Genetic generation of fuzzy knowledge bases: new perspectives

Marcos Evandro Cintra
Genetic generation of fuzzy knowledge bases: new perspectives*

Marcos Evandro Cintra

Advisor: Prof. Maria Carolina Monard
Co-Advisor: Prof. Heloisa de Arruda Camargo

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP, in partial fulfilment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. FINAL VERSION.

USP - São Carlos
June, 2012

*Work financially supported by the São Paulo State Research Agency - FAPESP, Proc. No. 2007/05390-0.
“Now, here, you see, it takes all the running you can do to keep the same place. If you want to get somewhere else, you must run at least twice as fast as that!”

Lewis Carroll, 1832-98: Through the Looking-Glass (1872).
Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisors, professors Maria Carolina Monard and Heloisa de Arruda Camargo, who have supported me throughout my thesis with much patience and knowledge whilst allowing me the room to work in my own way. Carolina, thank you for your support. I have surely learned much with you, both about researching as well as about life. You are a tender and sweet person, always willing to help. Thank you also for the many t-shirts and lots of chocolate and cookies :)

Thank you also Heloisa, for being such an exemplary person. I admire your elegant posture towards life. I deeply appreciate your confidence in me, as well as your continuous support, which so positively affected my work.

Professor Martin, thank you for your support while I was in Bristol, for the ideas which inspired my work, and for the nice chats about books and movies.

I have been truly blessed for sharing a lab and part of my life with such great people as you, Andrei, Everton, Fahdlina, Merley, Tati, and Ricardo. You inspired me with your examples and you were always there, sharing the burdens and blessings of life with me.

I also thank all my friends and colleagues from LABIC (Celso, Diego, Fabiano, Jean, Jorge, Ígor, Nathalie, Newton, Rafael Rossi, Ricardinho, Ronaldo Prati, Pedro, Vinicius, and Victor), from CIG (Dino, Pri, Fábio, Cris, and Mariana), and from the Intelligent Systems Lab (Anice, Daniel, Fiz, Sebastian, Viorel, who even taught me how to play poker, but skipped the lesson on how to lose with elegance :), Olga, Tang, and Zheng).

Warlene, my dear friend, I will always remember all the encouraging words I heard from you. How can we explain the fact that one day we met in a place so distant from where we call home? Was it just coincidence or were you bound to introduce me to bitter cyder and boss me around? Thank you for all the great moments. May we be blessed with many more!

My dear friends, Gustavo, Izilda, Lenira, Vagner, and Zé, thank you for your support, love, and friendship.

A special thank you for all the constant support received from Ana Paula, Ana Carolina, Glauca, Laura, and Lhais, as well as from all the ICMC staff.

I also thank FAPESP and CAPES for the financial support that made it possible for me to fulfill this important step in my life.

Finally, I thank my parents, my brothers and sisters, and my grandma Adelina, the most important people in my life, for all the support and love you all give me.
Este trabalho foca na geração genética de sistemas fuzzy. Uma das principais contribuições deste trabalho é a proposta do método FCA-BASED, que gera o espaço de busca genético usando a teoria de análise de conceitos formais por meio da extração de regras dos dados. Os resultados da avaliação experimental do método FCA-BASED demonstram sua robustez. O método FCA-BASED também produz um bom “trade-off” entre acurácia e interpretabilidade dos modelos gerados. Além disso, o método FCA-BASED apresenta melhorias em relação ao método DOC-BASED, uma abordagem proposta anteriormente. Essas melhorias estão relacionadas à redução do custo computacional para a geração do espaço de busca genético. Para ser capaz de trabalhar com conjuntos de dados de alta dimensão, foi também proposto o método FUZZYDT, uma versão fuzzy da clássica árvore de decisão C4.5. FUZZYDT é um método altamente escalável que apresenta baixo custo computacional e acurácia competitiva. Devido a essas características, o FUZZYDT é usado nesse trabalho como um método “baseline” para a avaliação experimental e comparações de outros métodos de classificação, fuzzy e clássicos. Também está incluído nesse trabalho a aplicação do método FUZZYDT em um problema do mundo real, o alerta da doença da ferrugem cafeeira em plantações brasileiras. Além disso, esse trabalho investiga a tarefa de seleção de atributos como forma de atacar o problema da dimensionalidade de sistemas fuzzy. Para esse fim, foi proposto o método FUZZYWRAPPER, uma abordagem baseada em “wrapper” que seleciona atributos levando em consideração as informações relevantes sobre a fuzificação dos atributos durante o processo de seleção. Esse trabalho também investiga a construção automática de bases de dados fuzzy, incluindo a proposta do método FUZZYDBD, que estima o número de conjuntos fuzzy que define todos os atributos de um conjunto de dados e distribui os conjuntos fuzzy proporcionalmente nos domínios dos atributos. Uma versão modificada do método FUZZYDBD, o método FUZZYDBD-II, também é proposta nesse trabalho. O método FUZZYDBD-II define números independentes de conjuntos fuzzy para cada atributo de um conjunto de dados por meio de funções de estimação.
Abstract

This work focuses on the genetic generation of fuzzy systems. One of the main contributions of this work is the proposal of the FCA-BASED method, which generates the genetic search space using the formal concept analysis theory by extracting rules from data. The experimental evaluation results of the FCA-BASED method show its robustness, producing a good trade-off between the accuracy and the interpretability of the generated models. Moreover, the FCA-BASED method presents improvements to the Doc-BASED method, a previously proposed approach, related to the reduction of the computational cost for the generation of the genetic search space. In order to tackle high dimensional datasets, we also propose the FUZZYDT method, a fuzzy version of the classic C4.5 decision tree, a highly scalable method that presents low computational cost and competitive accuracy. Due to these characteristics, FUZZYDT is used in this work as a baseline method for the experimental evaluation and comparisons of other classic and fuzzy classification methods. We also include in this work the use of the FUZZYDT method to a real world problem, the warning of the coffee rust disease in Brazilian crops. Furthermore, this work investigates the task of feature subset selection to address the dimensionality issue of fuzzy systems. To this end, we propose the FUZZYWRAPPER method, a wrapper-based approach that selects features taking the relevant information regarding the fuzzyfication of the attributes into account, in the feature selection process. This work also investigates the automatic design of fuzzy data bases, proposing the FUZZYDBD method, which estimates the number of fuzzy sets defining all the attributes of a dataset and evenly distributing the fuzzy sets in the domains of the attributes. A modified version of the FUZZYDBD method, FUZZYDBD-II, which defines independent numbers of fuzzy sets for each attribute of a dataset, by means of estimation functions, is also proposed in this work.
Contents

Acknowledgements ........................................................... vii
Resumo ........................................................................... ix
Abstract .......................................................................... xi
Table of Contents ............................................................. xvi
List of Figures ..................................................................... xviii
List of Tables ....................................................................... xx
List of Algorithms ............................................................ xxi
List of Abbreviations ......................................................... xxiv

1 Introduction ................................................................. 1
  1.1 Motivation ................................................................. 2
  1.2 Objectives ................................................................. 3
  1.3 Contributions of this Work .......................................... 4
  1.4 Outline ...................................................................... 7

2 Rule-Based Fuzzy Systems ............................................... 9
  2.1 Fuzzy Sets ................................................................. 10
  2.2 Computing with Fuzzy Rules ....................................... 13
    2.2.1 Linguistic Variables ............................................. 13
    2.2.2 Syntax of Fuzzy Rules ......................................... 14
    2.2.3 Semantic of Fuzzy Rules ...................................... 16
    2.2.4 Inference with Fuzzy Rules ................................. 18
  2.3 Rule-Based Fuzzy Systems .......................................... 19
    2.3.1 Fuzzy Classification Systems .............................. 20
    2.3.2 The Classic Fuzzy Reasoning Method ................. 20
    2.3.3 The General Fuzzy Reasoning Method ............... 21
  2.4 Final Remarks ......................................................... 22

3 Methods for the Automatic Definition of Fuzzy Data Bases .... 23
  3.1 Related Work ............................................................. 25
3.2 The FUZZYDBD Method ........................................... 27
  3.2.1 Definition of the Number of Fuzzy Sets for the FUZZYDBD
       Method ......................................................... 28
  3.2.2 The Distribution of the Fuzzy Sets ............................ 29
    3.2.2.1 The Equalized Universe Method ....................... 30
    3.2.2.2 The Same Frequency Method ......................... 30
    3.2.2.3 The 1-R Supervised Method ......................... 31
  3.2.3 Experimental Evaluation of the FUZZYDBD Method ........ 34
3.3 The FUZZYDBD-II Method ..................................... 37
  3.3.1 Estimation Functions .................................... 38
    3.3.1.1 The Information Gain Measure as Estimation Func-
            tion for FUZZYDBD-II .............................. 39
    3.3.1.2 Using ReliefF as Estimation Function for FUZZYDBD-
            II ...................................................... 40
    3.3.1.3 The Wang & Mendel Method as Estimation Func-
            tion for FUZZYDBD-II .............................. 40
  3.3.2 Experimental Evaluation of the FUZZYDBD-II Method ...... 41
3.4 Final Remarks .................................................. 44

4 Feature Subset Selection ......................................... 45
  4.1 Approaches for Feature Subset Selection ...................... 45
  4.2 Related Work .................................................. 47
  4.3 The FUZZYWRAPPER Method .................................. 49
    4.3.1 Description of the FUZZYWRAPPER Method .............. 49
    4.3.2 Experimental Evaluation of the FUZZYWRAPPER Method .. 50
    4.3.3 Expanded Experimental Evaluation — FUZZYWRAPPER .... 54
  4.4 Feature Subset Selection Using FUZZYDT ..................... 57
  4.5 Final Remarks ................................................ 60

5 Fuzzy Decision Trees ............................................ 63
  5.1 Decision Trees and the FUZZYDT Algorithm ................... 64
    5.1.1 C4.5 ...................................................... 64
    5.1.2 The FUZZYDT algorithm ................................. 65
    5.1.3 Classic VERSUS Fuzzy Decision Trees .................. 67
  5.2 The Use of FUZZYDT on the Prediction and Control of the Coffee
       Rust Disease ............................................... 70
    5.2.1 Contextualization and Related Work .................... 70
    5.2.2 Data Collection ......................................... 71
    5.2.3 Experimental Evaluation - Coffee Rust Prediction and Con-
            trol ...................................................... 72
    5.2.4 Results and Discussion ................................ 73
5.3 Final Remarks .......................................................... 79

6 Genetic Fuzzy System ............................................. 81
6.1 Genetic Fuzzy Systems ............................................. 81
   6.1.1 Challenges for the Automatic Generation of Fuzzy Systems 83
      6.1.1.1 The Dimensionality Issue .............................. 83
      6.1.1.2 The Issue of the Accuracy and Interpretability Trade- off .................. 84
6.2 Related Work ......................................................... 85
   6.2.1 Genetic Fuzzy Systems Proposals ............................ 85
   6.2.2 Genetic Fuzzy Systems - Applications to Real-World Problems ........................... 87
   6.2.3 Genetic Fuzzy Systems: New Trends .......................... 88
6.3 The DOC-BASED Method ........................................ 89
   6.3.1 Description of the DOC-BASED Method ...................... 89
   6.3.2 Experimental Evaluation ..................................... 93
6.4 Formal Concept Analysis ........................................ 97
6.5 The FCA-BASED Method ......................................... 103
   6.5.1 Description of the FCA-BASED Method ...................... 104
      6.5.1.1 Chromosome Codification ................................ 104
      6.5.1.2 Chromosome Initialization ............................... 106
      6.5.1.3 Fitness Calculation ...................................... 106
      6.5.1.4 Population Diversity Maintenance ..................... 107
      6.5.1.5 Selection of rules ....................................... 108
   6.5.2 Experimental Evaluation - FCA-BASED Method ............. 109
6.6 Final Remarks ....................................................... 115

7 Classic versus Fuzzy Classification Methods ................... 117
7.1 Initial Comparisons of Classification Methods .................. 117
   7.1.1 Description of the Involved Methods ....................... 118
   7.1.2 Experimental Evaluation ..................................... 119
7.2 Comparing FuzzyDT, C4.5, and PART ............................ 121
   7.2.1 PART .......................................................... 121
   7.2.2 Experimental Evaluation ..................................... 122
7.3 Comparing FuzzyDT, SLAVE, CHI, WF-C, Wang & Mendel, C4.5, and PART .................... 126
   7.3.1 Description of the Methods Selected for Comparison ....... 126
      7.3.1.1 CHI .................................................... 126
      7.3.1.2 WF-C .................................................. 127
      7.3.1.3 SLAVE ............................................... 127
   7.3.2 Experimental Evaluation ..................................... 128
List of Figures

1.1 Contributions of this work. .................................................. 5
2.1 Triangular membership function. .......................................... 11
2.2 Trapezoidal membership function. ...................................... 11
2.3 S function. ........................................................................ 12
2.4 Gaussian membership function. .......................................... 12
2.5 Unitary function. ................................................................ 13
2.6 A fine and a coarse partition defining the linguistic variable Temperature. ................................................................. 14
2.7 Classic Fuzzy Reasoning Method. ........................................ 21
2.8 General Fuzzy Reasoning Method. ........................................ 22
3.1 General illustration of the FUZZYDBD method. ...................... 28
3.2 Attribute defined by 5 fuzzy sets — Equalized Universe Method. 30
3.3 Attribute defined by 5 fuzzy sets — Same frequency method. .... 31
3.4 Attribute defined by 5 fuzzy sets distributed using the 1-R adapted method. ................................................................. 33
3.5 Accuracy rates - classic fuzzy reasoning method. .................... 36
3.6 Frequency of the fuzzy sets - FUZZYDBD. ............................. 36
3.7 The FUZZYDBD method. .................................................... 37
3.8 FUZZYDBD-II- Estimation of the number of fuzzy sets. .......... 38
4.1 FUZZYWRAPPER- a toy example. ............................................ 51
4.2 Evaluation model for FSS algorithms. ................................... 53
5.1 The FUZZYDT algorithm - a toy example. ......................... 65
5.2 A classic and a fuzzy decision tree for the Iris dataset. .......... 66
5.3 Fuzzy sets defining attributes Petal Length and Petal Width. ... 66
5.4 Decision tree generated by J48 for M3G10. .......................... 77
5.5 Decision tree generated by FUZZYDT for M3G10. ............... 78
6.1 Classification of Genetic Fuzzy Systems. .......................... 82
6.2 Genetic search space and binary chromosome codification. ... 92
6.3 Genetic search space and chromosome population. ............... 92
6.4 Conceptual lattice. ................................................. 99

7.1 Credit - normalized error rates [1.48, 15.64] and syntactic complexity [22.50, 222.00]. ............................................. 131
7.2 Dermatology - normalized error rates [4.66, 8.10] and syntactic complexity [26.10, 172.50]. ............................................. 131
7.3 Diabetes - normalized error rates [25.52, 30.32] and syntactic complexity [11.20, 112.00]. ............................................. 132
7.4 Heart - normalized error rates [8.52, 22.67] and syntactic complexity [16.80, 88.20]. ............................................. 132
7.5 Credit - normalized error rate [9.8, 37.3] and syntactic complexity [11.2, 95.4]. ............................................. 137
7.6 Cylinder - normalized error rate [25.8, 34.2] and syntactic complexity [1.0, 248.5]. ............................................. 138
7.7 Diabetes - normalized error rate [22.9, 31.5] and syntactic complexity [3.7, 150.2]. ............................................. 138
7.8 Glass - normalized error rate [27.0, 49.0] and syntactic complexity [10.1, 137.8]. ............................................. 139
List of Tables

3.1 Characteristics of the datasets - FuzzyDBD. .............................. 34
3.2 Error rates - FuzzyDBD. .......................................................... 35
3.3 Estimated number of fuzzy sets - FuzzyDBD. .............................. 36
3.4 Information gain rates for the Diabetes attributes. ......................... 40
3.5 ReliefF rates for the attributes of the Diabetes dataset. ................. 40
3.6 Error rates of the Wang & Mendel method for each attribute of the Diabetes dataset, considering each attribute individually. 41
3.7 Characteristics of the datasets - FuzzyDBD-II. ............................ 42
3.8 Number of fuzzy sets: Credit and Heart - FuzzyDBD-II. ................. 42
3.9 Error rates obtained by FuzzyDT using the FuzzyDBD-II method. 43
3.10 Number of generated rules - FuzzyDT. ..................................... 43

4.1 Characteristics of the datasets - FuzzyWrapper. .......................... 51
4.2 Original and selected number of features - FuzzyWrapper. ............. 52
4.3 Error rates - FuzzyWrapper. ..................................................... 52
4.4 Framework evaluation - FuzzyWrapper. ...................................... 54
4.5 Characteristics of the datasets - FuzzyWrapper expanded. ............... 54
4.6 Original and selected number of features - FuzzyWrapper expanded. 55
4.7 Error rates - FuzzyWrapper expanded. ....................................... 56
4.8 Dataset characteristics - FuzzyDT. ........................................... 58
4.9 Original and selected number of features - FuzzyDT. ..................... 58
4.10 Error rates - FuzzyDT. ............................................................ 59

5.1 Characteristics of the datasets and number of fuzzy sets - Coffee rust prediction and control. ................................................. 73
5.2 Description of the variables - Coffee rust prediction and control. .... 74
5.3 Error rates - Coffee rust prediction and control. .......................... 74
5.4 Number of rules, conditions, and features - Coffee rust prediction and control. ............................................................. 75
6.1 Characteristics of the datasets - DOC-BASED. .......................... 94
6.2 Total number of attributes, classes and fuzzy sets - DOC-BASED. 95
6.3 Number of selected attributes - DOC-BASED. .............................. 95
6.4 Total number of possible rules and number of extracted rules - DOC-BASED. ............................................. 96
6.5 Number of generated rules - DOC-BASED. ................................ 96
6.6 Error rates - DOC-BASED. ...................................................... 97
6.7 Attribute × value table - toy example. ........................................ 99
6.8 Formal Context based on Table 6.7. ........................................... 99
6.9 Fitness Penalization Rates - FCA-BASED. .................................... 107
6.10 SAGA Parameters. ................................................................. 108
6.11 Characteristics of the datasets - FCA-BASED. .............................. 109
6.12 Original and selected number of features using FUZZYWRAPPER. 110
6.13 Average number of fuzzy sets - FCA-BASED. ............................... 111
6.14 Average number of extracted rules using FCA and total number of rules. ................................................................. 111
6.15 Error rates - FCA-BASED. ...................................................... 112
6.16 Number of generated rules - FCA-BASED. .................................. 113
6.17 Number of rules and syntactic complexity - FCA-BASED. ............. 114

7.1 Characteristics of the datasets - Initial comparisons. ................. 119
7.2 Error rates - Initial comparisons. .............................................. 120
7.3 Number of generated rules - Initial comparisons. ......................... 120
7.4 Ranking for the Friedman test - Initial comparisons. .................... 121
7.5 Characteristics of the datasets - DT-based methods. .................... 122
7.6 Original and selected number of features - DT-based methods. ....... 123
7.7 Error rates - DT-based methods. .............................................. 124
7.8 Number of generated rules - DT-based methods. ........................ 124
7.9 Average number of conjunctions - DT-based methods. ................ 125
7.10 Number of Rules × Average Number of Conjunctions - DT-based methods. ......................................................... 125
7.11 Error rates - further comparisons. ........................................... 129
7.12 Number of rules and syntactic complexity - further comparisons. 129
7.13 Execution time - further comparisons. ..................................... 130
7.14 Error rates - DT and GA based methods. .................................. 136
7.15 Number of rules and syntactic complexity - DT and GA based methods. ......................................................... 137

8.1 Methods proposed in this work. .............................................. 142
List of Algorithms

1. The 1-R adapted algorithm for fuzzy set distribution. .......................... 32
2. The FuzzyWRAPPER algorithm. .................................................. 50
3. The FuzzyDT algorithm. ............................................................... 65
4. The DoC-based Method. ................................................................. 90
5. Extraction of Fuzzy Classification Rules using FCA. ......................... 102
6. The FCA-BASED Method. .............................................................. 105
7. Algorithm for the adjustment of the crossover and mutation rates. 108
8. Post pruning of rules for the FCA-BASED method. ....................... 109
| Abbreviation | Description                        |
|--------------|------------------------------------|
| CCR          | Correct Classification Rate        |
| CFRM         | Classic Fuzzy Reasoning Method     |
| CF           | Confidence Level                   |
| DoC          | Degree of Coverage                 |
| DT           | Decision Tree                      |
| EUM          | Equalized Universe Method          |
| EMC          | Error of the Majority Class        |
| EFS          | Error without Feature Selection    |
| FCA          | Formal Concept Analysis            |
| FDB          | Fuzzy Data Base                    |
| FKB          | Fuzzy Knowledge Base               |
| FRB          | Fuzzy Rule Base                    |
| FS           | Fuzzy System                       |
| FSS          | Feature Subset Selection           |
| GA           | Genetic Algorithm                  |
| GD           | Genetic Diversity                  |
| GFRM         | General Fuzzy Reasoning Method     |
| GFS          | Genetic Fuzzy System               |
| IG           | Information Gain                   |
IM  Inference Mechanism
KB  Knowledge Base
ME  Majority Error
ML  Machine Learning
MLP  MultiLayer Perceptron
MPLCS  Memetic Pittsburgh Learning Classifier System
NR  Number of Rules
PSO  Particle Swarm Optimization
RBFS  Rule Based Fuzzy System
RF  ReliefF
SFM  Same Frequency Method
SAGA  Self-Adaptive Genetic Algorithm
SC  Syntactic Complexity
SD  Standard Deviation
WM  Wang & Mendel
Machine learning is concerned with the development of methods for the extraction of patterns from data in order to make intelligent decisions based on these patterns. Classification, a supervised learning task, is an important field of research for machine learning, with relevant challenges and open issues. Soft computing methods, including the fuzzy logic, neural networks, and evolutionary systems, are based on the idea of mimicking the human intelligence in order to create tools provided with some human-like capabilities, such as learning, reasoning, and decision making with the specific goal of handling uncertainty and imprecision in its various forms. In this sense, fuzzy systems are part of both areas, machine learning and soft computing. Fuzzy classification systems, in particular, are able to provide highly accurate and interpretable models, naturally handling imprecise and uncertain information. Due to such characteristics of fuzzy systems, allied with the increasing amount of available data, the automatic generation of fuzzy systems is a very active area of research, with its pertinent open questions and challenges.

The focus of this work is on the proposal of automatic methods for the generation of fuzzy classification systems, as well as on methods that favour the scaling of the proposed approaches for large datasets. In this work, we propose methods for the generation of different parts of rule-based fuzzy systems, rules and fuzzy sets. Furthermore, we also tackle the problem of dimensionality reduction by means of fuzzy feature subset selection methods, and the trade-off between accuracy and interpretability. In order to enrich the investigations with additional viewpoints concerning aspects such as feature selection, interpretability and ease of model generation, we approached the automatic generation of rule-based fuzzy systems by means of decision tree
models, in addition to the genetic methods, is also emphasized in this work, which investigates the automatic generation of rule-based fuzzy systems by means of the genetic and decision tree approaches.

This chapter presents the motivation, objectives, and main contributions generated from this work, as well as an outline of the following chapters.

1.1 Motivation

Fuzzy systems are based on the fuzzy sets and fuzzy logic theories (Zadeh, 1965). Fuzzy systems provide automatic mechanisms to represent and process linguistic information, as well as to tackle imprecision and uncertainty, which exist in real-world information. Fuzzy systems are able to translate the knowledge of a domain into linguistic sentences that are easily interpreted. Moreover, the robust reasoning methods applied by fuzzy systems are able to deal with the information represented by these linguistic sentences.

The fuzzy systems of interest for this work are the classification rule-based fuzzy systems, which are formed by a knowledge base and an inference mechanism. The knowledge base contains a fuzzy rule base and a fuzzy data base. The fuzzy data base includes the definition of the attributes, also called features, in terms of fuzzy sets. The inference mechanism uses the information of the fuzzy data base and the fuzzy rule base to classify new examples.

The generation of fuzzy systems is a relevant issue, which is the focus of intense research activity. There are several approaches for the generation of fuzzy systems. Moreover, the increasing amount of available data generates a demand for the proposal of automatic methods that are able to learn from data. Some relevant information regarding the issue of the increasing amount of available data include the following:

- The volume of digital data in the planet will grow 48% in 2012, reaching 2.7 zettabytes. For 2015, the forecast is 8 zettabytes\textsuperscript{1}.

- By the year 2015, more than 85% of the organizations of the Fortune 500 rank will not be able to explore their generated data efficiently in order to obtain competitive advantage\textsuperscript{2}.

Regarding the automatic generation of fuzzy systems, it requires the definition of the knowledge base. In this sense, many approaches have been applied for the automatic generation of fuzzy systems from data. Neural networks (Aliev et al., 2011), genetic algorithms (Herrera, 2008), and, more

\textsuperscript{1}International Data Corporation - Predictions for 2012, \url{http://www.idc.com/research/Predictions12/Main/index.jsp}

\textsuperscript{2}Gartner Inc, \url{http://www.gartner.com/technology/research/predicts/}
recently, other bio-inspired approaches, such as particle swarm optimization (Al-Aawar et al., 2011; Pousinho et al., 2011), ant colony optimization (Ahmadizar and Soltanpanah, 2011), and artificial immunological systems (Prakash and Deshmukh, 2011) are among the most successful techniques for the automatic generation of fuzzy systems.

This work involves the proposal of methods for the automatic generation of fuzzy systems based on the genetic and decision tree paradigms. Genetic fuzzy systems are one of the most researched approaches in the area due to their flexibility in providing means for users to explore more than one objective at the same time while performing a powerful global search. Decision trees, due to their low computational cost, can be considered as suitable alternatives to the genetic fuzzy systems. Furthermore, this work also includes the proposal of automatic methods for the design of fuzzy data bases, as well as the proposal of feature subset selection methods for the generation of fuzzy systems.

The data dimensionality issue and the trade-off between accuracy and interpretability are among the most challenging topics investigated in the context of this work. In fact, interpretability is considered one of the main advantages of fuzzy systems. The concept of interpretability is related to being able to understand the behaviour of the fuzzy system by inspecting its fuzzy knowledge base (Mikut et al., 2005). The dimensionality issue, often addressed as the dimensionality curse, is related to the fact that the increase of attributes or fuzzy sets generates an exponential increase in the number of possible rules for a fuzzy system. In this sense, the generation of fuzzy rule bases with large sets of rules compromises the interpretability of the models. This fact takes us to the second issue, the trade-off between accuracy and interpretability, since these are competitive objectives.

1.2 Objectives

The objectives of this work are related to the investigation and proposal of methods for the design of fuzzy systems. In this sense, we aim at proposing methods that are able to cope with the issues of dimensionality and the trade-off between accuracy and interpretability, previously mentioned. This way, we worked on the proposal of methods to define fuzzy data bases tuned for the generation of fuzzy rule bases using the genetic and the decision tree-based approaches. Our goals also included the proposal of methods for feature subset selection adjusted for the generation of fuzzy systems, and, thus, able to take advantage of the information regarding the definition of the attributes in terms of fuzzy sets. Moreover, we aimed at reducing the number of rules.
in the fuzzy rule bases generated using the genetic approach by adding a measure of interpretability in the fitness calculation.

More specifically, in this work we tackled the issue of the *dimensionality curse* with:

- The proposal of feature subset selection methods focused on the characteristics of fuzzy systems;
- The proposal of methods to individually estimate the number of fuzzy sets describing each feature;
- The proposal of a method for the extraction of rules from data to generate the search space of a genetic algorithm based on the formal concept analysis theory, which presents considerable scalability to large datasets;
- The pruning of rules in the fuzzy rule bases generated by a genetic process.

Regarding the trade-off between accuracy and interpretability, in this work this issue is tackled in two ways:

1. The definition of a mechanism to calculate the fitness of chromosomes which is able to consider the number of rules and the accuracy of the rule bases encoded in the chromosomes;
2. The proposal of a fuzzy decision tree, which can be applied to datasets of high dimensionality, producing competitive accuracy and interpretability levels.

The main contributions of this work are detailed next.

### 1.3 Contributions of this Work

The main contributions of this work are related to the following areas:

1. Feature subset selection methods.
2. The automatic definition of fuzzy data bases;
3. The genetic generation of fuzzy rule bases;
4. Fuzzy decision trees;

Figure 1.1 presents the methods proposed in this work (named in the light blue rectangles) considering the well-defined steps for the automatic definition of fuzzy systems: i) an optional and previous selection of features from a
dataset; ii) the definition of the fuzzy data base; iii) and the generation of the fuzzy rule base. Notice that fuzzy decision trees and the genetic generation of fuzzy rule bases, although belonging to distinct approaches, are related to the same step, i.e., the generation of fuzzy rule bases.

Figure 1.1: Contributions of this work.

It follows a thorough description of our contributions according to the previously mentioned tasks.

**The Genetic Generation of Fuzzy Systems** — Our main contribution to this area, which is also the main contribution of this work, is the proposal of the FCA-BASED method in (Cintra et al., 2012c). The FCA-BASED method extracts the rules forming the genetic search space from data, based on the theory of formal concept analysis, which was investigated and proposed in (Majidian et al., 2011; Cintra et al., 2011d,a). We experimentally evaluated and compared the FCA-BASED method to similar approaches in (Cintra et al., 2012a,b). The FCA-BASED method presents an innovative and improved alternative for the generation of the genetic search space when compared to the DoC-BASED method (Cintra and Camargo, 2007a), previously proposed. The DoC-BASED method uses a process to extract rules to form its genetic search space based on the degree of coverage of the rules. This rule extraction process poses an exponential complexity problem. As another contribution, in this work we optimized the DoC-BASED method itself in (Cintra et al., 2008c, 2010a) (Chapter 6).

**The Automatic Definition of Fuzzy Data Bases** — The first contribution of this work is a method for the automatic definition of fuzzy data bases,
namely FuzzyDBD (Cintra et al., 2009a). An initial study on this topic was published in (Cintra et al., 2008b). FuzzyDBD uses the Wang & Mendel method to estimate the number of fuzzy sets for all the attributes of a dataset. Regarding the distribution of the fuzzy sets in the domains of the attributes, the equalized universe method was selected as default for FuzzyDBD (among three methods empirically evaluated). Triangular shaped fuzzy sets are adopted. A modified version of this method, named FuzzyDBD-II, was proposed in (Cintra et al., 2011e), which is able to define different numbers of fuzzy sets for each attribute of a dataset by means of estimation functions. The information gain measure, ReliefF filter, and Wang & Mendel method were experimentally evaluated as estimation functions for the FuzzyDBD-II method (Chapter 3).

**Fuzzy Decision Trees** — Our contribution to this area is the proposal of a fuzzy version of the classic C4.5 decision tree, namely FuzzyDT, in (Cintra and Camargo, 2010). Due to its low computational cost and competitive accuracy, FuzzyDT was evaluated and compared to several methods in (Cintra et al., 2010b, 2011c, 2012a), and it was used as estimation function for the FuzzyDBD method in (Cintra et al., 2011e). FuzzyDT was also applied to a real-world problem, the prediction and control of the coffee rust disease in Brazilian crops, in a collaborative work with researchers from The Brazilian Agricultural Research Corporation - EMBRAPA and from the State University of Campinas - UNICAMP (Cintra et al., 2011b) (Chapter 5).

**Feature Subset Selection Methods** — The task of feature subset selection is tackled in this work with the wrapper and embedded approaches. Firstly, we proposed the FuzzyWrapper method in (Cintra et al., 2008a), which was further evaluated in (Cintra et al., 2009b). The FuzzyWrapper method uses a wrapper approach, with the Wang & Mendel method as its base algorithm, in order to select optimized subsets of features. The advantages of the FuzzyWrapper approach over traditional methods for feature subset selection include the fact that, by using a fuzzy method as its base algorithm, it selects features taking the relevant information regarding the fuzzyfication of the attributes into account in the process. The FuzzyWrapper method presents low computational cost since it uses a best-first search strategy. This way, the FuzzyWrapper showed to be suitable for use in conjunction with the genetic generation of fuzzy systems (Cintra et al., 2012b). Secondly, we applied FuzzyDT for the task of feature subset selection, as an embedded method, in (Cintra and Camargo, 2010). FuzzyDT, a feature subset selection method, also includes the advantage of FuzzyWrapper which considers the informa-
tion regarding the fuzzy definition of the attributes in the feature subset selection process. Moreover, FUZZYDT presents low computational cost and, thus, high scalability (Chapter 4).

1.4 Outline

This work is organized as follows.

Chapter 2 — In this chapter, an introduction to the fuzzy sets and fuzzy logic theory is presented. We introduce the topic of computing with fuzzy rules, the rule-based fuzzy systems, as well as the fuzzy reasoning methods adopted in this work.

Chapter 3 — In this chapter, a literature review on methods for the automatic definition of fuzzy data bases is presented. Our proposed methods, FUZZYDBD and FUZZYDBD-II are also presented and detailed, including the experimental evaluation and discussion of the obtained results.

Chapter 4 — The topic of feature subset selection is discussed in this chapter. We present the main approaches for feature subset selection, filters, wrappers, and embedded methods. The FUZZYWRAPPER method, as well as the use of the FUZZYDT method for feature subset selection, are detailed. This chapter also includes the experimental evaluation and discussion of the results obtained with the FUZZYWRAPPER and FUZZYDT methods.

Chapter 5 — Chapter 5 is dedicated to decision trees. The proposed FUZZYDT method, together with its experimental evaluation, is detailed in this chapter. It follows a thorough comparison of fuzzy and classic decision tree models considering their inference methods, induction, handling of features, as well as the reuse of features in the induced models. Last, we present a real-world application of the FUZZYDT method on the prediction and control of the coffee rust disease in Brazilian crops, detailing the data collection process, the definition of the datasets, as well as the experimental results and comparisons.

Chapter 6 — This chapter is dedicated to the genetic fuzzy systems. A discussion of the challenges for the automatic generation of fuzzy systems begins the chapter, followed by a literature review on the genetic fuzzy systems. It follows the description of the DOc-BASED method, along with its experimental evaluation and results. In the sequel, we present an introduction on formal concept analysis, as well as on fuzzy formal
concepts. We also describe our proposal for the extraction of fuzzy classification rules from data using the formal concept analysis theory. The FCA-BASED method is then detailed. We present the description of the main features of the FCA-BASED method, including chromosome codification and initialization, fitness calculation, and the adopted algorithm for biodiversity maintenance. Finally, chapter 6 presents the experimental evaluation, comparisons and results of the FCA-BASED method.

Chapter 7 — In this chapter, several comparative evaluations of classic and fuzzy methods for classification are presented. The chapter presents a description of the methods used in the experimental comparisons. Special attention is given to comparisons between fuzzy and classic methods based on similar approaches, such as decision trees, rule inducing methods, as well as genetic systems. The selected methods are compared in terms of accuracy as well as in terms of interpretability.

Chapter 8 — The last chapter presents the conclusions of this work along with topics for future work based on the results obtained by the methods proposed in this work and motivating challenges faced during this work.
Rule-Based Fuzzy Systems

Fuzzy Systems (FS) can be defined as systems that use the fuzzy set theory proposed by prof. Lofti A. Zadeh (Zadeh, 1965) to represent at least one of its variables. The fuzzy set theory allows the computational representation and processing of imprecise and uncertain information, which are abundant in the real world. In fact, most of the available computer approaches cannot directly process information with imprecision and uncertainty, making fuzzy systems a valuable alternative to work with domains presenting such characteristics.

Rule-based fuzzy systems, a particular type of fuzzy systems, use a reasoning mechanism based on approximate reasoning that has the ability to express the ambiguity and subjectivity present in the human reasoning (Pedrycz and Gomide, 1998). The rule bases of fuzzy systems store knowledge represented by means of rules, such as:

\[ \text{If } \rightarrow \text{ Then} \]

The antecedent part of the rules contains the input attributes of the system, while their consequents contain the output attributes of the system. The antecedents and consequents are combined by means of logic operators. The output of the system is obtained by an inference mechanism, based on the membership degrees of the input data.

The principles of the fuzzy logic and fuzzy systems are presented in the next sections. The complete definitions regarding the fuzzy set theory included in this chapter can be found in (Klir and Yuan, 1995) and in (Pedrycz, 1998).
2.1 Fuzzy Sets

The idea of a set is quite simple and common in our daily lives and arises frequently when we try to organize or summarize knowledge about objects. Let X be a set of objects named universe of discourse, and x an element of X. A set $A$, $A \subseteq X$, is defined as a collection of objects of $x \in X$.

In the classic theory of sets, known as crisp sets, an element belongs or not to a given set. This fact can be expressed by the function defined by Equation 2.1, named characteristic function, which attributes a value 1 (one) to an element if it is a member of set $A$, and 0 (zero), otherwise.

$$F_A(X) = \begin{cases} 
1 & \text{if } x \in A \\
0 & \text{if } x \notin A 
\end{cases} \tag{2.1}$$

Notwithstanding, many real-world situations cannot be properly expressed by the classic set theory, since it does not present flexibility and it forces an abrupt membership transition, i.e., an element either belongs to a set (full membership) or does not belong to it (null membership). Instead, the fuzzy sets, which can be considered as a generalization of the classic sets, allow a gradual transition between the extreme states, thus, an element can partially belong to a given set.

A fuzzy set $A$ is characterized by a membership function that associates the elements of the universe of discourse $X$ to a value defined in an interval, usually $[0, 1]$. The expression $A_x : A \rightarrow [0, 1]$ is one of the most frequently used representations to denote the membership degree of an element $x$ in a given fuzzy set $A$ as $A(x)$.

The membership degree 0 (zero) represents total exclusion of an element from a set, while 1 (one) represents the total membership of the element in the set. For discrete sets, it is also possible to represent a set $A$ as a set of ordered pairs in the form $A(x)/x$, as:

$$A = \{(A(x)/x) | x \in X\}$$

where $A(x)$ is the membership degree of element $x$. For instance, consider $X = \{15, 18, 22, 28, 30, 33, 35, 39\}$, a collection of values for temperature. The fuzzy set defined by $A = \{(0.8/15), (0.9/18), (1/22), (0.7/28), (0.5/30), (0.3/33), (0.1/35), (0/39)\}$, can be understood as the temperature set for the “medium” category.

The core of the fuzzy set theory is the concept of membership. Basically, any function in the form $A_x : A \rightarrow [0, 1]$ describes a membership function associated to a fuzzy set. However, the most widely used membership functions are usually represented in the triangular, trapezoidal, $S$, or Gaussian forms, due to the fact that they have a simple and easy representation with a quite direct
usage. The unitary set, also named *singleton*, is another widely used membership function for practical applications. These most well known functions, as well as their graphic representations, are presented next.

- **Triangular function** – defined by parameters $a, m,$ and $b$, where $a \leq m \leq b$ (Equation 2.2 and Figure 2.1).

$$A(x) = \begin{cases} 
0 & \text{if } x \leq a \\
\frac{x-a}{m-a} & \text{if } x \in (a, m) \\
1 & \text{if } x = m \\
\frac{b-x}{b-m} & \text{if } x \in (m, b) \\
0 & \text{if } x \geq b 
\end{cases}$$

(Equation 2.2)

![Figure 2.1: Triangular membership function.]

- **Trapezoidal function** — defined by parameters $a, m, n,$ and $b$, where:
  $a \leq m < n \leq b$ (Equation 2.3 and Figure 2.2).

$$A(x) = \begin{cases} 
0 & \text{if } x \leq a \\
\frac{x-a}{m-a} & \text{if } x \in (a, m) \\
1 & \text{if } x \in [m, n] \\
\frac{b-x}{b-n} & \text{if } x \in (n, b) \\
0 & \text{if } x \geq b 
\end{cases}$$

(Equation 2.3)

![Figure 2.2: Trapezoidal membership function.]

11
• *S* function – defined by parameters *a*, *m*, and *b*. \( m = (a + b)/2 \) is known as the crossing point of function *S* (Equation 2.4 and Figure 2.3).

\[
A(x) = \begin{cases} 
0 & \text{if } x \leq a \\
2 \left( \frac{x - a}{b - a} \right)^2 & \text{if } x \in (a, m] \\
1 - 2 \left( \frac{x - b}{b - m} \right)^2 & \text{if } x \in (m, b] \\
1 & \text{if } x > b 
\end{cases} \tag{2.4}
\]

![Figure 2.3: S function.](image)

• Gaussian function – defined by parameters *m* and *k*, where *k > 0* (Equation 2.5 and Figure 2.4).

\[
A(x) = e^{-k(x-m)^2} \tag{2.5}
\]

![Figure 2.4: Gaussian membership function.](image)

• Unitary set (singleton) – defined by parameters \((m, h)\), where *h ≤ 1* (Equation 2.6 and Figure 2.5).

\[
A(x) = \begin{cases} 
h & \text{if } x = m \\
0 & \text{if } x \neq m 
\end{cases} \tag{2.6}
\]
2.2 Computing with Fuzzy Rules

Rules are useful to represent the knowledge related to many domains. For example, the task of modelling sentences in natural or artificial language is relatively simple by using fuzzy rules (Jang et al., 1997). Moreover, fuzzy rules allow the representation of imprecise knowledge, simplifying the modelling of real-world problems. Rules, fuzzy or not, can be used by computation systems in their reasoning mechanisms to infer conclusions based on known facts.

In this section, the concepts of fuzzy linguistic variable and fuzzy rules are described, as well as the principles of the inference mechanisms based on fuzzy rules.

2.2.1 Linguistic Variables

A large part of the human experience and knowledge can be well represented in a suitable linguistic way, which is more general and imprecise. Nevertheless, this representation makes it difficult to use such information by computers, which process precise information. The linguistic variables allow the proper representation of imprecise and complex information by means of linguistic terms, simplifying the processing of such information by computers.

Informally, a linguistic variable can be defined as a variable whose values are words or sentences in natural language instead of numbers (Zimmermann, 1991). These variables are defined on a base set, which is granulated by linguistic terms (or values), defined by fuzzy sets. This granulation process of a given domain in terms of fuzzy sets define a fuzzy partition for the given variable. This way, the larger the number of linguistic terms, the larger the partition granulation (fine partition), while a smaller number of linguistic terms produces a coarse partition.

Figure 2.6 presents a variable named Temperature defined by partitions with different granularities. The top figure presents a fine partition, while the bottom figure presents a coarse partition. For the fine partition, five linguistic
values are used to define the linguistic variable *temperature*: very low, low, medium, high, very high. These linguistic values are defined by the membership functions representing each linguistic value, which in this example are triangular. This way, the linguistic variable *Temperature* is defined by the set of linguistic terms $D_{Temperature} = \{\text{very low, low, medium, high, very high}\}$. For the coarse partition, the linguistic values *low* and *high* are used to define the same linguistic variable.

![Temperature - Fine Partition](image1)

![Temperature - Coarse Partition](image2)

Figure 2.6: A fine and a coarse partition defining the linguistic variable *Temperature*.

### 2.2.2 Syntax of Fuzzy Rules

Rules are among the first techniques used to represent knowledge. In fact, rules are still widely used due to the fact that they make it possible to clearly express directives and strategies, as well as capturing the knowledge from human experts. Rules also have the advantage of their linguistic format, which is easily understandable.

Fuzzy rules are an easy way to express imprecise knowledge, thus, giving fuzzy systems transparency and comprehensibility (Klir and Yuan, 1995). In general, fuzzy rules have the following form:

$$\textbf{IF} \ (\text{antecedent}) \ \textbf{THEN} \ (\text{consequent})$$
Fuzzy rules aim at establishing relations among the variables of their antecedents, also called condition or premise, and the variables of their consequents, also called conclusion or action.

The antecedent part of fuzzy rules are always a fuzzy proposition, or a conjunction/disjunction of them. These propositions can be atomic or compound. Atomic propositions specify a linguistic value \( A \) for a linguistic variable \( V \) and it has the form: \( V \) is \( A \). Compound propositions can be formed by conjunctions or disjunctions of atomic propositions. For instance, consider the following propositions:

\[
X_1 \text{ is } A \text{ AND } X_2 \text{ is } B
\]

and

\[
X_1 \text{ is } C \text{ OR } X_2 \text{ is } D
\]

These propositions can represent knowledge of the type:

- The temperature is low AND the humidity is high;
- The water is good for human consumption OR inflation is high.

A fuzzy rule assumes the format of a conditional proposition, which, in the simplest case, can be expressed as:

\[
\text{IF } X \text{ is } A \text{ THEN } Y \text{ is } B
\]

where \( X \) and \( Y \) are linguistic variables of the universes \( \mathbf{X} \) and \( \mathbf{Y} \), respectively, and \( A \) and \( B \) are linguistic terms that label fuzzy sets defined on universes \( \mathbf{X} \) and \( \mathbf{Y} \), respectively.

In a fuzzy rule, both the antecedent and the consequent can be formed by compound propositions. This fact originates, for example, rules in the form:

\[
\text{IF } X_1 \text{ is } A_1 \text{ AND } X_2 \text{ is } A_2 \text{ AND } X_3 \text{ is } A_3 \text{ THEN } Y_1 \text{ is } B_1 \text{ AND } Y_2 \text{ is } B_2
\]

In the most general case, both the antecedent and the consequent of the rule can be formed by a finite number of atomic propositions related by conjunctions or disjunctions. Thus, assuming that \( p \) and \( q \) denote fuzzy propositions, a fuzzy rule can be expressed in the form:

\[
\text{IF } p \text{ THEN } q,
\]

and proposition \( p \) can be expressed in the form:

\[
X_1 \text{ is } A_1 \text{ AND } X_2 \text{ is } A_2 \text{ AND } \ldots \text{ AND } X_n \text{ is } A_n, n \geq 1
\]
or

\[ X_1 \text{ is } A_1 \text{ OR } X_2 \text{ is } A_2 \text{ OR } \ldots \text{ OR } X_n \text{ is } A_n, \quad n \geq 1 \]

while proposition \( q \) can be expressed in the form:

\[ Y_1 \text{ is } B_1 \text{ AND } Y_2 \text{ is } B_2 \text{ AND } \ldots \text{ AND } Y_m \text{ is } B_m, \quad m \geq 1 \]

or

\[ Y_1 \text{ is } B_1 \text{ OR } Y_2 \text{ is } B_2 \text{ OR } \ldots \text{ OR } Y_m \text{ is } B_m, \quad m \geq 1 \]

where \( A_1, A_2, \ldots, A_n \) and \( B_1, B_2, \ldots, B_m \) are fuzzy sets in the universes \( X_1, X_2, \ldots, X_n \), and \( Y_1, Y_2, \ldots, Y_m \), respectively, while \( X_1, X_2, \ldots, X_n \) and \( Y_1, Y_2, \ldots, Y_m \) are variables of the domains \( X_1, X_2, \ldots, X_n \), and \( Y_1, Y_2, \ldots, Y_m \), respectively.

2.2.3 Semantic of Fuzzy Rules

Considering a fuzzy rule in its simplest form:

\[
\text{IF } X \text{ is } A \text{ THEN } Y \text{ is } B
\]

this rule can be understood as expressing a relation between variables \( X \) and \( Y \), which can be described as a fuzzy relation \( R \). The membership function of \( R \) describes the degree with which the pair \((x, y)\) is compatible with the relation between variables \( X \) and \( Y \).

Consider \( A \) and \( B \) as fuzzy sets on \( X \) and \( Y \), respectively. Relation \( R \) can be determined as:

\[
R(x, y) = f(A(x), B(y)), \quad \forall (x, y) \in X \times Y
\]

in which \( f \) is a function in the form:

\[
f : [0, 1]^2 \rightarrow [0, 1]
\]

The selection of function \( f \) is essential for fuzzy systems, since it determines the semantic of the rules in various processing aspects of the system. In general, the relations induced by fuzzy rules are derived from three main function classes: fuzzy conjunctions, fuzzy disjunctions, or fuzzy implications (Pedrycz and Gomide, 1998), described next.

- **Fuzzy Conjunction** — A fuzzy conjunction is a function

\[
f_t : [0, 1]^2 \rightarrow [0, 1]
\]
defined by:
\[ f_t(A(x), B(y)) = A(x)tB(y) \]
in which \( t \) denotes a \( t \)-norm. The most common definitions are the Mamdani (Mamdani and Assilian, 1975; Mamdani, 1977) and Larsen conjunctions (Larsen, 1980), which use the minimum (\( \land \)) and the algebraic product (\( \bullet \)), respectively, as defined next.

- **Mamdani**:
  \[ f_m(A(x), B(y)) = A(x) \land B(y), \forall x, y \in X \]

- **Larsen**:
  \[ f_p(A(x), B(y)) = A(x) \bullet B(y), \forall x, y \in X \]

**Fuzzy Disjunction** — A fuzzy disjunction is a function

\[ f_t : [0, 1]^2 \rightarrow [0, 1] \]

defined by:

\[ f_p(A(x), B(y)) = A(x)sB(y) \]
in which \( s \) denotes an \( s \)-norm. The most common \( s \)-norms, also called \( t \)-conorms, are:

- **Maximum \( t \)-conorm**:
  \[ f_{\text{max}}(A(x), B(y)) = \text{max}(A(x), B(y)), \forall x, y \in X \]

- **Probabilistic sum**:
  \[ f_p(A(x), B(y)) = A(x) + B(y) - A(x) \bullet B(y), \forall x, y \in X \]

**Fuzzy Implication** — A fuzzy implication is a function

\[ f_t : [0, 1]^2 \rightarrow [0, 1] \]
derived by a set of properties and classified in categories characterized by the formalism of the classic logic and intuitionist logic. The formal definition of fuzzy implication is not presented here since it is not fundamental for this work. More details can be found in (Pedrycz and Gomide, 1998).

One of the most well-known examples of fuzzy implications is the Lukasiewicz implication (Lukasiewicz, 1970):

\[ f_l(A(x), B(y)) = \min[1, 1 - A(x) + B(y)] \]
If a given rule has multiple variables in its antecedent or consequent, relation \( R \) is defined similarly. For rule

\[
\text{IF } X_1 \text{ is } A_1 \text{ AND } X_2 \text{ is } A_2 \text{ AND } \ldots \text{ AND } X_n \text{ is } A_n \\
\text{THEN } Y_1 \text{ is } B_1 \text{ AND } Y_2 \text{ is } B_2 \text{ AND } \ldots \text{ AND } Y_m \text{ is } B_m
\]

relation \( R \) is defined by assuming that the compound preposition \( X_1 \text{ is } A_1 \text{ AND } X_2 \text{ is } A_2 \text{ AND } \ldots \text{ AND } X_n \text{ is } A_n \) induces an \( n \)-ary relation defined by:

\[
P_a(x_1, x_2, \ldots, x_n) = A_1(x_1) \, t \, A_2(x_2) \, t \, \ldots \, t \, A_n(x_n)
\]

and the compound proposition \( Y_1 \text{ is } B_1 \text{ AND } Y_2 \text{ is } B_2 \text{ AND } \ldots \text{ AND } Y_m \text{ is } B_m \) induces an \( m \)-ary relation defined by:

\[
P_c(y_1, y_2, \ldots, y_m) = B_1(y_1) \, t \, B_2(y_2) \, t \, \ldots \, t \, B_m(y_m)
\]

The rule then induces an \( n + m \) relation defined by:

\[
R(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m) = f(P_a(x_1, x_2, \ldots, x_n), P_c(y_1, y_2, \ldots, y_m))
\]

If the antecedent or the consequent of the rule is formed by the connection \textbf{OR} instead of \textbf{AND}, we substitute the \( t \)-norm by an \( s \)-norm for the calculation for the induced relations \( P_a \) or \( P_c \).

2.2.4 Inference with Fuzzy Rules

The computation with fuzzy rules involves the usage of a reasoning method based on inference rules. The compositional inference rule is the basic inference rule for the reasoning processes of fuzzy systems. The compositional inference rule, also named composition rule, is presented for the basic case, \textit{i.e.}, when the rule has only one variable in the antecedent and one variable in the consequent.

The inference process of interest in this context is applied for a rule, which establishes a generic knowledge on a given domain and on a given fact, stipulating a specific information on the situation from which a conclusion must be inferred.

Let \( A \) be a fuzzy set on \( X \), and \( B \) a fuzzy set on \( Y \). According to what was stated previously about the semantic of the rules, the rule:

\[
\textbf{IF } X \text{ is } A \textbf{ THEN } Y \text{ is } B
\]

induces a fuzzy relation \( R(x, y) \) on \( X \times Y \). For this rule to be used in an inference, it is necessary to know some fact that establishes a specific value for the antecedent variable \( X \).

Given the fact that \( X \) is \( A' \), it is possible to infer the conclusion \( Y \) is \( B' \), where \( B' \) is defined by (Pedrycz and Gomide, 1998):
\[ B'(y) = \sup_{x \in X} [A'(x) \ t \ R(x, y)] \ \forall y \in Y \]

in which \( \sup \) denotes the supreme operator (the lowest upper bound) and \( t \) is a \textit{t-norm}.

The use of the compositional inference rule can be easily extended to rules with multiple variables, as well as to situations where the knowledge of the problem is described by a set of rules. The general process of computation using fuzzy rules is based on the compositional inference rule.

As an alternative to it, it is possible to apply the scaled inference method, which involves a small number of numeric calculations in relation to the general method. These two general computation processes with rules are described in (Pedrycz and Gomide, 1998). The general inference process and the scaled inference are not presented here since they are not used in this work. The presented discussion focus on reasoning methods specific for classification problems, which is the focus of this work.

### 2.3 Rule-Based Fuzzy Systems

As stated previously, a fuzzy system is any system that has some mechanism derived from the fuzzy logic and which includes, at least, one variable assuming linguistic values defined by fuzzy sets (Klir and Yuan, 1995; Zadeh, 1973).

The type of fuzzy system of interest for this work are the Rule-Based Fuzzy Systems (RBFS), or simply Fuzzy Rule Systems. The RBFS have been successfully applied in several areas, such as control, classification and pattern recognition (Ichiba et al., 2006; Navale and Nelson, 2010; Cintra et al., 2011b; Niknam et al., 2011). RBFS have two main components: a Knowledge Base (KB) and an Inference Mechanism (IM).

The KB is formed by the Fuzzy Data Base (FDB), which contains the definitions of the fuzzy sets related to the linguistic terms used in the fuzzy rules. The Fuzzy Rule Base (FRB) stores a set of rules that models a given problem. The IM is responsible for the processing of the rules, which is carried out by a reasoning method. The reasoning method is an inference process that derives conclusions from rules and known facts.

The definition of the several parameters involved in the RBFS generates the so called \textit{fuzzy system model}. Some of these models have received extra attention due to the success obtained in practical applications. In this sense, two main models for fuzzy systems can be highlighted: the Mamdani model (Mamdani and Assilian, 1975; Mamdani, 1977), which has a fuzzy proposition in the consequent part of the rules, and the Takagi-Sugeno model (Takagi and
Sugeno, 1983, 1985), which has a function applied to the input values in its consequent part.

The fuzzy classification systems, which are the type of fuzzy systems of interest for this work, are described next.

2.3.1 Fuzzy Classification Systems

Classification is an important task from the machine learning area. The classification task can be roughly described as: given a set of objects \( E = \{e_1, e_2, ..., e_n\} \), also named examples, cases, or instances, which are described by \( m \) features (or attributes), assign a class \( c_i \) from a set of classes \( C = \{c_1, c_2, ..., c_j\} \) to an object \( e_p \), described by its attribute values \( e_p = (a_{p1}, a_{p2}, ..., a_{pm}) \).

Fuzzy rules contain linguistic variables defining the features in their antecedent, and a class in their consequent part. A typical fuzzy classification rule can be expressed by:

\[
R_k : \text{IF } X_1 \text{ is } A_{1l_1} \text{ AND } X_2 \text{ is } A_{2l_2} \text{ AND } ... \text{ AND } X_m \text{ is } A_{ml_m} \text{ THEN Class } = c_i
\]

where \( R_k \) is the rule identifier, \( X_1, ..., X_m \) are the features of the set of examples considered in the problem (represented by linguistic variables), \( A_{1l_1}, ..., A_{ml_m} \) are the linguistic values used to represent the feature values, and \( c_i \in C \) is the class. Notice that fuzzy classification rules usually present a variable number of features in their antecedents.

The inference mechanism applies the set of fuzzy rules to an input example in order to determine the class it belongs to. The classic and general fuzzy reasoning methods are widely used in the literature. These methods, proposed in (Cordon et al., 1999), are described next.

2.3.2 The Classic Fuzzy Reasoning Method

Several fuzzy classification systems use the Classic Fuzzy Reasoning Method (CFRM), which classifies an example using the rule that has the highest compatibility degree with the example. Let \( e_p = (a_{p1}, a_{p2}, ..., a_{pm}) \) be an example to be classified, and \( \{R_1, R_2, ..., R_s\} \) the set of \( s \) rules of a classification system, each with up to \( m \) antecedents. Let \( A_{ili}(a_{pi}), \ i = 1, ..., m, \) be the membership degree of feature value \( a_{pi} \) to the \( i \)-th fuzzy set of fuzzy rule \( R_k \). The CFRM applies the following steps to classify example \( e_p \):

1. Calculate the compatibility degree between example \( e_p \) and each rule \( R_k \), for \( k = 1, ..., s \), and a \textit{t-norm} \( \textbf{t} \), given by:

\[
\text{Compat}(R_k, e_p) = \textbf{t}(A_{1l_1}(a_{p1}), A_{2l_2}(a_{p2}), ..., A_{ml_m}(a_{pm}))
\]
2. Find rule \( R_{k_{\text{max}}} \) as the rule with the highest compatibility degree with the example, i.e.,

\[
\text{Compat}(R_{k_{\text{max}}}, e_p) = \max\{\text{Compat}(R_k, e_p)\}, \quad k = 1, 2, ..., s
\]

3. Assign the class \( c_j \) to the example \( e_p \), where \( c_j \) is the class predicted by the rule \( R_{k_{\text{max}}} \) found in the previous step.

Figure 2.7 presents graphically the CFRM. The compatibility of the input example is calculated in relation to all \( s \) rules, and class \( c_j \) from rule \( R_k \) is assigned to the input example, which is the class of the rule with the highest compatibility with it.

![Figure 2.7: Classic Fuzzy Reasoning Method.](image)

### 2.3.3 The General Fuzzy Reasoning Method

The General Fuzzy Reasoning Method (GFRM) applies the following steps to classify a given example \( e_p \):

1. Similarly to CFRM, it calculates the compatibility degree between example \( e_p \) and each rule \( R_k \), for \( k = 1, ..., s \).

2. Calculate a classification value \( \text{Class}_{c_i} \), for each class. \( \text{Class}_{c_i} \) is defined as the aggregation of the compatibility degree, calculated in the previous step, of all rules with class \( c_i \), and represents the compatibility degree of the example with all the rules whose predicted class is \( c_i \), given by:

\[
\text{Class}_{c_i} = f\{\text{Compat}(R_k, e_p) | c_i \text{ is the class of } R_k\}
\]

where \( f \) is an aggregation operator.
3. The class with the highest classification degree is assigned to example \( e_p \).

Figure 2.8 presents graphically the GFRM. The compatibility of the input example is calculated in relation to all \( s \) rules, and class \( c_j \) is assigned to the input example, such that \( c_j \), is the class that obtained the highest classification degree with the input example, among all classes, \( C = \{c_1, c_2, \ldots, c_j\} \).

![Figure 2.8: General Fuzzy Reasoning Method.](image)

In this work we use an extension of the general fuzzy reasoning method, which calculates the sum of the compatibility degree instead of the aggregation operator in step 2.

### 2.4 Final Remarks

The fuzzy logic and fuzzy set theories introduced by prof. Zadeh provide a means for the representation and processing of imprecise and uncertain information, abundant in the real world.

In this chapter we presented the basic concepts of fuzzy sets, fuzzy logic, computation with fuzzy rules, and the rule-based fuzzy systems.

Chapter 3 discusses the automatic definition of fuzzy data bases.
CHAPTER

3

Methods for the Automatic Definition of Fuzzy Data Bases

The definition of fuzzy data bases, i.e., the definition of the fuzzy sets that model the linguistic variables of a given domain, is a fundamental task due to the possible variance on the results obtained by fuzzy systems. The methods for the automatic definition of fuzzy data bases can be roughly classified as follows:

• Methods that generate the fuzzy data base prior to the generation of the fuzzy rule base;

• Methods that generate both, the fuzzy data base and the fuzzy rule base, at the same time;

• Methods that tune the generated fuzzy data base after the generation of the fuzzy rule base, by adjusting the fuzzy sets to the generated rules.

The focus of this work is on the generation of the fuzzy data base prior to the generation of the fuzzy rule base. In other words, we focus on generating the fuzzy rule base with a predefined and fixed fuzzy data base. The reason for choosing to work with predefined and fixed data bases are related to the approach adopted for the generation of the fuzzy rule bases using a genetic algorithm to select rules. The definition of the rules and, thus, the definition of the fuzzy sets, is required before the use of the genetic algorithm, i.e., the rules must be known a priori in order to apply the genetic algorithm for their selection. Another reason, which is independent from the rule selection approach, is the fact that, although it is possible to generate both, the fuzzy rule
base and the fuzzy data base at the same time, since it is common to find
datasets described by many features for classification problems, the compu-
tational cost of generating both, the fuzzy rule base and the fuzzy data base,
becomes an issue.

The tasks involved in the process of automatically defining fuzzy data bases
include the following:

1. The definition of the shape of the membership function that defines the
   fuzzy sets (triangular, trapezoidal, S-function, among others);

2. The definition of the number of fuzzy sets that describe each attribute of
   the domain;

3. The distribution of the fuzzy sets describing each attribute in the at-
   tribute domain.

An expert in a given domain can assist in the process of defining fuzzy
data bases. In the absence of an expert, an automatic method can be used to
define fuzzy data bases. Nevertheless, in spite of the numerous approaches
found in the literature for this task, there are no general rules or guidelines
for the selection of a suitable method. In practice, many methods may have to
be evaluated in order to find the appropriate definitions for a given application
and domain.

Considering the definition of fuzzy data bases as the combination of the
three tasks previously described, next we discuss the proposals found in the
literature. For the first task (definition of the fuzzy set shape), the Gaussian,
trapezoidal and triangular shapes are the most used in the literature, probably
because they produce comparable results and are easily interpreted (Kaya and
Alhajj, 2003). Regarding the second task (the definition of the number of fuzzy
sets), most papers in the literature use from 2 to 10 fuzzy sets to define each
attribute, sometimes repeating experiments using a different number of fuzzy
sets for all attributes of each domain. It should be observed that the number
of fuzzy sets for each attribute must be reasonably small in order to produce
interpretable rule bases. For the last task (the distribution of the fuzzy sets
in their domains), most papers adopt their equal distribution in the domains
of the attributes. This approach is named the Equalized Universe Method
(EUM) (Chen and Wang, 1999).

Another aspect to be considered when defining fuzzy sets that make this
task non trivial is context dependent interpretations. For instance, let us con-
sider a linguistic variable Temperature; the interpretation of the linguistic term
“low temperature” is closely related to the region where one lives (consider a
continental country such as Brazil, for instance). Thus, personal interpreta-
tions may generate strong variations for the same concept: since each person
may have different and personal interpretations on the meaning of a concept, it is natural that different membership functions may be created to define the same concept. This flexibility on the subjective definition of membership functions and, as previously stated, in the distribution of the fuzzy sets in the partitions of the attributes, show the robustness of the fuzzy logic, which is closely related to the inherent characteristics of the fuzzy sets (Zhang and Zhang, 2005).

Although this flexibility for the definition of membership functions exists, the selection of a method depends on the particular application and domain (Chen and Wang, 1999). While the shape of the fuzzy sets may cause less expressive differences in the results obtained by fuzzy systems, the number of fuzzy sets and their distribution are relevant parameters that affect the system in terms of performance and interpretability.

3.1 Related Work

The approaches found in the literature for the definition of fuzzy data bases include the use of heuristic methods, bio-inspired algorithms (such as genetic algorithms, clone selection, and artificial neural networks), clustering algorithms, the use of indexes, histograms (Medasani et al., 1998), the adaptation of classic machine learning methods, such as the K-NN algorithm (Keller et al., 1985), among others. Next, some of these methods are presented in more detail.

The use of artificial neural networks is explored in (Aliev et al., 2011) for the generation of fuzzy systems. The proposed method includes the optimization of the fuzzy data base by means of neural networks. In (Abraham, 2001), the author presents an overview of several neuro-fuzzy methods for the definition of fuzzy sets. These methods, although some of them present good results, have the additional problem of tuning parameters for the neural networks, and, in some cases, might need further experimental evaluation tests.

Fuzzy clustering algorithms are explored by Liao et al. (2003), who propose the use of a Fuzzy C-Means variant for the generation of fuzzy term sets with $1/2$ overlap. This method uses the mean squared error criterion to determine the number of fuzzy sets and the optimal shape of the membership functions associated with each term.

Genetic algorithms are used in (Pulkkinen and Koivisto, 2010) to learn the granularities of fuzzy partitions, tune the membership functions, and learn fuzzy rules as well. Genetic algorithms are also used in (Alcalá et al., 2011) for an embedded genetic learning of fuzzy data bases. Also, in (Hong et al., 2006), genetic algorithms are used to find membership functions suitable for mining
problems and then uses the final best set of membership functions to mine fuzzy association rules. A genetic approach for the genetic fuzzy modelling of user perception of three-dimensional shapes is proposed in (Achiche and Ahmed-Kristensenæ, 2011). The genetic approach proposed in (Achiche and Ahmed-Kristensenæ, 2011) is based on a previous proposal (Achiche et al., 2003), which uses a combination of real coded GAs for the fuzzy data base and a binary coded GA for the rule base. The proposal obtained promising results.

Several methods explore the use of special parameters for the distribution of fuzzy sets. In (Alcalá et al., 2007), for example, the authors use the concept of 2-tuple to define linguistic terms. The 2-tuple strategy is used to define both, the centre of a fuzzy set (support), and a value for a symbolic translation, which enables the fuzzy sets to be moved sideways, always keeping the same base width. This symbolic translation of a linguistic term is a number within the interval $[-0.5, 0.5)$. A genetic algorithm is then used to define the best position for the fuzzy sets. The experiments presented in (Alcalá et al., 2007) used a fixed number of triangular shaped sets.

There are also proposals focusing on the use of special indexes to generate membership functions, such as the concept of entropy (Ishibuchi et al., 2001). Nieradka and Butkiewicz (2007) propose a method for the definition of fuzzy sets based on two indexes, the fuzzy entropy and the fuzziness index, adopting the $S$ shaped fuzzy set (Figure 2.3, page 12). These papers focus on image recognition applications. The Kappa measure was also used. Dou et al. (2007) use the Kappa measure to calculate the fuzzy Kappa index in order to define a suitable segmentation of partitions.

Chen et al. (2009) propose the divide-and-conquer genetic-fuzzy mining algorithm for association rules, which is able to find minimum supports, membership functions, and fuzzy association rules.

Sanz et al. (2010) propose a method for the enhancement of fuzzy systems for classification. The method uses a post-processing genetic tuning step that adjusts the amplitude of the upper bound of interval-valued fuzzy sets to contextualize the fuzzy partitions and maximize accuracy.

Acilar and Arslan (2011) use a clonal selection algorithm to determine the membership functions in a fuzzy system. The clonal selection algorithm is used to explain the basic features of an adaptive immune response to an antigenic stimulus. Their proposal determines and adjusts the shape of membership functions.

Although such a variety of methods exist, it is important to emphasize the fact that many studies adopt the triangular, Gaussian or trapezoidal shapes for the fuzzy sets, defining the number of fuzzy sets empirically, and employing
the equalized universe method (Chen and Wang, 1999) for an even distribution of the fuzzy sets in the partitions of the attributes. Regarding the number of fuzzy sets, most researchers define this variable empirically using a range usually varying from 2 to 10 fuzzy sets per attribute. The main reasons why researchers decide to define such important variables without the aid of any formal method may include:

- The complexity of the available methods, which may require more time and effort to be implemented than the actual application in focus;

- The flexibility of the fuzzy logic for the definition of fuzzy variables and fuzzy terms, which allows the users to define their own fuzzy data base. This flexibility exists because since the fuzzy data base is used for the generation of the fuzzy rules, these rules, in turn, are adjusted to provide a suitable performance for the fuzzy system;

- The lack of consensus and/or guidelines for the selection of suitable methods for a given application or domain.

The drawbacks of the presented approaches for the automatic definition of fuzzy data bases (related to the need for tuning parameters, tests with few datasets, high computational cost, among others) motivated us to propose a fast, feasible, and comprehensive method for this task (Cintra et al., 2009a). The proposed method, named FUZZYDBD, is described next.

3.2 The FUZZYDBD Method

FUZZYDBD is a simple and fast approach that defines a fuzzy data base using examples from a given domain. In fact, these characteristics of the FUZZYDBD method favours its use against costly approaches for the generation of fuzzy rule bases, such as genetic algorithms. Considering the three tasks involved in defining a fuzzy data base (definition of the shape of the fuzzy sets, number of fuzzy sets, and their distribution), FUZZYDBD uses the following approaches:

- Fuzzy set shape: the triangular shape is adopted for all attributes assuming the fact that the triangular, Gaussian and trapezoidal shapes usually produce similar results;

- Definition of the number of fuzzy sets: this parameter is estimated by the Wang & Mendel method, described in Section 3.2.1, page 28, and used for all attributes of the dataset;
• Distribution of the fuzzy sets in the domains of the attributes: FUZZYDBD considered three distinct methods for the distribution of the fuzzy sets:

- The same width distribution;
- The same frequency distribution;
- The distribution using a method that takes the classes into consideration in the process.

Figure 3.1 illustrates the FUZZYDBD method considering the three aforementioned approaches for the distribution of the fuzzy sets.

Figure 3.1: General illustration of the FUZZYDBD method.

In Figure 3.1, after the number of fuzzy sets defining the attributes of the input dataset is estimated using the Wang & Mendel method, one of the three approaches for their distribution can be applied to generate the final fuzzy data base. Notice that $A_{t_{ij}}$ represents the $jth$ fuzzy set defining attribute $i$, while $a_{ij}$, $b_{ij}$, and $c_{ij}$ contain the values that define points $a$, $b$, and $c$ of the triangular membership function of the $jth$ fuzzy set for attribute $i$.

The definition of the number of fuzzy sets and the methods investigated for the distribution of the fuzzy sets are described next.

### 3.2.1 Definition of the Number of Fuzzy Sets for the FUZZYDBD Method

The definition of the number of fuzzy sets is an open area of research. Moreover, some approaches proposed to generate fuzzy rules demand a considerable computational effort, such as genetic algorithms that can take from minutes to days to converge to good solutions. This way, when the approach used to generate the fuzzy rule base is costly, it is preferable to combine it with a fast method to define the fuzzy data base. Thus, as already mentioned, it is common to find researchers using a few different numbers of fuzzy sets for each attribute in order to empirically validate their methods, usually ranging from 2 to 9, repeating the complete round of experiments several times, according to the different numbers of fuzzy sets used.
The FUZZYDBD method, on the other hand, uses a heuristic strategy to define the number of fuzzy sets for all the attributes of a dataset using the Wang & Mendel method (Wang, 2003), in a fast and efficient way. The Wang & Mendel method has been used as a benchmark approach for comparison purposes due to its low complexity ($O(m \times n)$, $m$ attributes, $n$ examples) and the fact that it produces rule bases with good classification rates and no conflicting or redundant rules. The Wang & Mendel method basically generates a rule for each example in the training set by replacing the continuous values for the attributes of an input example with the fuzzy set with highest membership degree with the example. In the sequence, redundant and conflicting rules are removed in order to form the final fuzzy rule base.

However, nowadays it is possible to generate more precise fuzzy rule bases with lower number of rules using other approaches, such as fuzzy decision trees or genetic fuzzy systems, instead of the Wang & Mendel method. The heuristic used by the FUZZYDBD method is based on the assumption that the Wang & Mendel method can be used as an indicator of the quality of the results that can be achieved by other methods, when using the same data base, i.e., the same number, distribution, and shape of fuzzy sets defining the attributes of the dataset.

The FUZZYDBD method was evaluated using from 2 to 10 triangular shaped fuzzy sets to define each attribute of the domains. The idea is to execute the Wang & Mendel method several times, using from 2 to 10 fuzzy sets, in order to select the number of fuzzy sets that produces the rule base with the best accuracy rates, or, in case of ties, the smaller number of rules for interpretability sake. Notice that the number of fuzzy sets is defined for all attributes of a given dataset. The FUZZYDBD method also uses a tolerance interval to allow the user to select the smaller number of fuzzy sets when the difference in accuracy is not significant, while the difference in the number of fuzzy sets is.

### 3.2.2 The Distribution of the Fuzzy Sets

For a proper distribution of the fuzzy sets in the domains of the attributes in order to obtain transparent fuzzy data bases, it is important to consider the following general guidelines (Pulkkinen and Koivisto, 2010):

- **Coverage**: the domains of the attributes must be covered by at least one fuzzy set;
- **Overlap**: adjacent fuzzy sets should be moderately overlapped. The overlap cannot be too large or too small. Generally, 0.5 is a good choice;
- **Normality**: the fuzzy sets must have full membership in at least one point.
The three methods tested for the distribution of the fuzzy sets in the domains of the attributes fulfil the general guidelines previously described, guaranteeing the quality of the generated fuzzy data bases. These methods are:

1. The Equalized Universe Method (EUM);
2. The Same Frequency Method (SFM);
3. An adaptation of the 1-R method for attribute discretization.

3.2.2.1 The Equalized Universe Method

This method uses the same width for each fuzzy set (Chen and Wang, 1999), producing strong fuzzy partition. Figure 3.2 presents an attribute described by five triangular fuzzy sets using the equalized universe method. Notice that, usually, the fuzzy sets at the extremities of the domain have half the width of the fuzzy sets in the middle of the domain.

![Figure 3.2: Attribute defined by 5 fuzzy sets — Equalized Universe Method.](image)

This method is widely used in the literature. Some of the reasons for that may include the fact that it is a simple and easy method to use, generating isosceles triangles with $\frac{1}{2}$ overlap, guaranteeing the semantic meaning of the fuzzy sets and guaranteeing that no area of the domain has a membership degree inferior to 0.5. Another important advantage of this method is that the fuzzy data bases it generates are quite easy to interpret.

3.2.2.2 The Same Frequency Method

This method divides the domains of the attributes according to the frequency distribution of the examples. In our implementation, we adopted percentiles (Bai et al., 1989) to find the suitable distributions of the fuzzy sets.

To illustrate, consider an attribute described by 5 triangular fuzzy sets: the maximum membership degree will be on the 16.7th percentile for the first fuzzy set, on the 33.3th percentile for the second fuzzy set, on the 50.0th percentile for the third one, on the 66.7th percentile for the fourth one, and
on the 83.3th percentile for the last fuzzy set. This way, the fuzzy sets do not have the same base width, on the contrary, depending on the distribution of the examples, they are likely to be different. Notice that in case of many examples with identical values, two or more fuzzy sets might be defined with exactly the same middle point. In order to avoid this situation, if the same example is chosen by two or more different percentile values, the neighbouring examples of the left and right are chosen to replace the original value.

Figure 3.3 illustrates the use of percentiles for the distribution of fuzzy sets for an attribute represented by 5 triangular fuzzy sets. The distribution of the fuzzy sets for this attribute was done considering 20 examples in the interval \([0,10]\) \((0, 0, 0, 1, 1, 2, 2, 2, 3, 5, 5, 6, 7, 7, 7, 9, 9, 9, 9, 9)\). For this example, the 16.7th percentile (of the 20 examples), is rounded to 4, the 33.3th percentile is 7, the 50th percentile is 10, the 66.7th percentile is 14, while the 83.3th percentile is 17. This way, the 4th, 7th, 10th, 14th and 17th examples are selected to define the centre of the fuzzy sets.

![Figure 3.3: Attribute defined by 5 fuzzy sets — Same frequency method.](image)

Notice that the fuzzy sets of the laterals are defined beyond the actual domain limits. This strategy is used to avoid leaving any regions of the domain with membership degrees smaller than 0.5. The space between the actual beginning and end of the domain and the vertices of the triangles is calculated as the same space between the middle vertex of the triangle and the adjacent extremity of the domain.

### 3.2.2.3 The 1-R Supervised Method

The 1-R method is used for discretization of attributes. It basically divides the domain of an attribute into \(k\) predefined intervals, in relation to the frequency of the examples, by calculating the optimum (majority) class for each interval, thus, trying to find the best splitting point using the class. The original algorithm is proposed in (Holte, 1993). Algorithm 1 presents the adapted
version of the 1-R algorithm used in this work.

**Algorithm 1:** The 1-R adapted algorithm for fuzzy set distribution.

```
Input: \( n \) = number of examples;
\( m \) = number of attributes;
\( i \) = number of fuzzy sets describing the attribute:

1. for \( a = 1 \) to \( m \) do
   2. Sort the examples by attribute \( a \);
   3. Divide the examples in \( i \) intervals;
   4. for \( b = 1 \) to \( i - 1 \) do
      5. Calculate the optimal (majority) class for the interval;
      6. for \( c = 1 \) to interval \( (i - 2) \) do
         7. if first example of interval \( (c + 1) \) belongs to optimal class of interval \( c \) then
            Adjust intervals so that the first example of interval \( (c + 1) \) is transferred to interval \( c \);
         else \( d = \) index of last example of interval \( (c + 1) \);
         8. if \( d = \) index of last example of interval \( (c + 1) \) then split the examples of interval \( c \) between intervals \( c \) and \( (c + 1) \);
      end
   end
   9. For each interval, create a fuzzy set with core in the middle of the interval and laterals in the cores of the neighbouring intervals;
end
```

To avoid leaving any regions of the domain with membership degrees smaller than 0.5, the same strategy used for the same frequency method was used similarly for the 1-R adapted method.

To illustrate the 1-R adapted method for the distribution of fuzzy sets, consider the following 20 examples (attribute value, class) in the interval \([0, 10]\): \((0, 3), (0, 3), (0, 1), (1, 3), (1, 3), (2, 3), (2, 2), (2, 2), (3, 2), (5, 2), (5, 1), (6, 1), (7, 1), (7, 3), (7, 3), (9, 2), (9, 3), (9, 1), (9, 2), (9, 3)\). Notice that the examples are ordered according to the attribute value. These 20 examples are first equally divided into the following 5 sets:

1. \([0, 3), (0, 3), (0, 1), (1, 3)]\);
2. \([1, 3), (2, 3), (2, 2), (2, 2)]\);
3. \([3, 2), (5, 2), (5, 1), (6, 1)]\);
4. \([7, 1), (7, 3), (7, 3), (9, 2)]\);
5. \([9, 3), (9, 1), (9, 2), (9, 3)]\).

Since 1-R tries to split the subsets examples in order to preserve a majority class, 1-R checks if the first example of the following subset also belongs to
the majority class of the subset at hand, and, if so, it moves the first example to the current subset. This process is repeated until an example that does not belong to the majority class of the current subset is found, which is when the next subsets are checked similarly. This way, by considering the majority class of each set, the final distribution of the examples using the 1-R adapted method is:

1. \{(0, 3), (0, 3), (0, 1), (1, 3), (1, 3), (2, 3)\};
2. \{(2, 2), (2, 2), (3, 2), (5, 2)\};
3. \{(5, 1), (6, 1), (7, 1)\};
4. \{(7, 3), (7, 3), (9, 2), (9, 3)\};
5. \{(9, 1), (9, 2), (9, 3)\}.

Based on the examples of each set and the interval of the domain \([0, 10]\), the cores of the fuzzy sets are defined as follows.

1. 0 to 2: 1;
2. 2 to 5: 3.5;
3. 5 to 7: 6;
4. 7 to 9: 8;
5. 9 to 10: 9.5.

Figure 3.4 shows the final distribution of the 5 fuzzy sets, according to the 1-R adapted method, based on the 20 examples previously presented.

Figure 3.4: Attribute defined by 5 fuzzy sets distributed using the 1-R adapted method.
3.2.3 Experimental Evaluation of the FuzzyDBD Method

All experiments presented in this work were performed using a 10-fold cross validation strategy. For the FuzzyDBD method, we used the 10 datasets from UCI - Machine Learning Repository (Frank and Asuncion, 2010), presented in Table 3.1, which shows a summary of their characteristics including the number of examples in each dataset, number of features (including the number of continuous and discrete features in brackets), number of classes, and the majority error (ME) — representing the error obtained by the most naive classifier, i.e., a classifier which classifies all examples using the majority class of the dataset.

Table 3.1: Characteristics of the datasets - FuzzyDBD.

| Dataset    | Examples | Features | Classes | ME   |
|------------|----------|----------|---------|------|
| Breast     | 683      | 9 [9, 0] | 2       | 34.60|
| Bupa       | 345      | 4 [4, 0] | 2       | 42.03|
| Credit     | 653      | 15 [6, 9]| 2       | 45.93|
| Diabetes   | 769      | 8 [8, 0] | 2       | 34.90|
| Glass      | 220      | 9 [9, 0] | 7       | 65.46|
| Heart      | 270      | 13 [13, 0]| 2      | 44.44|
| Iris       | 150      | 4 [4, 0] | 3       | 66.67|
| Segmentation| 210    | 19 [19, 0]| 7      | 85.72|
| Vehicle    | 846      | 18 [18, 0]| 4      | 74.33|
| Wine       | 178      | 13 [13, 0]| 3      | 99.74|

The classic and the general fuzzy reasoning methods were used to evaluate the rule bases. Nevertheless, in this work we only show the results with the classic fuzzy reasoning method, due to the poor quality of the results with the general method.

Table 3.2 presents the error rates obtained by Wang & Mendel using the classic fuzzy reasoning method (Section 2.3.2, page 20), and the three methods for the distribution of the fuzzy sets. The first row (Sets) describes the number of fuzzy sets defining each attribute, column EUM represents the equalized universe method, column SFM represents the same frequency method, while column 1-R represents the 1-R adapted method. The best results for each dataset and for each distribution method, among the results using from 2 to 10 fuzzy sets, are dark-gray shaded. Notice that a tolerance interval of 0.02 was used in order to select the smallest subset of features within this interval.

One of the reasons for the good performance of the equalized universe method may include the fact that it generates fuzzy sets with the same distance amongst them, promoting the creation of rules that cover equal areas of the partitions of the attributes. The other methods, instead, promote the creation of fuzzy sets in areas with a concentration of examples but with less differences regarding the classes, which could have a deeper effect on the classification rates. The 1-R method, since it uses the classes to adjust the fuzzy sets defining each attribute, was expected to perform better than the
To test whether there is a significant difference among the methods, the Friedman test was used with the null-hypothesis that the performance of the three methods for the distribution of fuzzy sets, assessed in terms of the error rate, are comparable (Demsar, 2006). As the null-hypothesis was rejected with a 95% confidence level, the Bonferroni-Dunn post-hoc test (Demsar, 2006) (to detect whether the differences among the methods are significant) was used. Results showed that the equalized universe method is significantly better than the 1-R and frequency methods, with a 95% confidence level. Results also showed that the 1-R method performed significantly better than the frequency method with a 95% confidence level.

Figure 3.5 presents the accuracy rates (vertical axis) obtained with the experiments, graphically. Notice that, in Figure 3.5, \( W \) represents the equalized universe method (width method), \( F \) the frequency method, and \( 1R \) represents the 1-R adapted method.

Table 3.3 shows the best estimated number of fuzzy sets, using the equalized universe method (EUM), same frequency method (SFM) and the 1-R adapted method (1-R). The best classification rates used to define these values are light-gray shaded in Tables 3.2.

Figure 3.6 presents the frequency that the numbers of fuzzy sets were used to define the attributes of the dataset, according to Table 3.3.

Two and three fuzzy sets were chosen to define the attributes of most of the datasets (23 times, in total). In fact, two fuzzy sets were selected 13 times, while three fuzzy sets 10 times. This way, the estimated number of fuzzy sets defined by the three distribution methods can be considered reasonably small.

### Table 3.2: Error rates - FuzzyDBD.

| Sets | Datasets | EUM | SFM | 1-R | EUM | SFM | 1-R | EUM | SFM | 1-R | EUM | SFM | 1-R | EUM | SFM | 1-R |
|------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|      | Breast   | 0.04| 0.47| 0.36| 0.04| 0.41| 0.11| 0.05| 0.29| 0.16| 0.08| 0.31| 0.23| 0.11| 0.34| 0.29|
|      | Bupa     | 0.51| 0.54| 0.57| 0.48| 0.49| 0.43| 0.45| 0.47| 0.43| 0.44| 0.48| 0.46| 0.45| 0.62| 0.48|
|      | Diabetes | 0.46| 0.42| 0.40| 0.46| 0.42| 0.27| 0.47| 0.45| 0.37| 0.47| 0.45| 0.42| 0.47| 0.45| 0.44|
|      | Class    | 0.59| 0.42| 0.52| 0.51| 0.45| 0.33| 0.45| 0.40| 0.35| 0.39| 0.44| 0.39| 0.39| 0.47| 0.43|
|      | Heart    | 0.02| 0.02| 0.07| 0.00| 0.00| 0.03| 0.00| 0.01| 0.02| 0.00| 0.02| 0.00| 0.00| 0.03| 0.00|
|      | Iris     | 0.33| 0.37| 0.43| 0.35| 0.09| 0.35| 0.08| 0.07| 0.09| 0.09| 0.10| 0.09| 0.13| 0.17| 0.13|
|      | Segmentation | 0.53| 0.86| 0.74| 0.26| 0.85| 0.46| 0.29| 0.85| 0.50| 0.33| 0.85| 0.61| 0.34| 0.86| 0.70|
|      | Vehicle  | 0.01| 0.05| 0.00| 0.00| 0.02| 0.00| 0.00| 0.00| 0.00| 0.02| 0.00| 0.00| 0.02| 0.00| 0.02|
|      | Wine     | 0.06| 0.01| 0.01| 0.01| 0.02| 0.03| 0.00| 0.00| 0.02| 0.00| 0.01| 0.01| 0.01| 0.02| 0.01| 0.01|

### Table 3.3: Estimated number of fuzzy sets.

| Sets | Datasets | EUM | SFM | 1-R | EUM | SFM | 1-R | EUM | SFM | 1-R | EUM | SFM | 1-R | EUM | SFM | 1-R |
|------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|      | Breast   | 0.09| 0.35| 0.30| 0.10| 0.37| 0.31| 0.26| 0.37| 0.31| 0.20| 0.39| 0.34| 0.20| 0.39| 0.34|
|      | Bupa     | 0.45| 0.54| 0.52| 0.39| 0.55| 0.58| 0.42| 0.57| 0.58| 0.45| 0.57| 0.55| 0.45| 0.57| 0.55|
|      | Credit   | 0.46| 0.45| 0.44| 0.46| 0.45| 0.45| 0.40| 0.45| 0.45| 0.43| 0.45| 0.45| 0.43| 0.45| 0.45|
|      | Diabetes | 0.01| 0.00| 0.03| 0.03| 0.01| 0.05| 0.07| 0.07| 0.11| 0.12| 0.09| 0.11|
|      | Glass    | 0.37| 0.48| 0.45| 0.42| 0.49| 0.46| 0.40| 0.50| 0.49| 0.39| 0.50| 0.50| 0.39| 0.50| 0.50|
|      | Heart    | 0.07| 0.05| 0.01| 0.08| 0.02| 0.01| 0.09| 0.11| 0.02| 0.09| 0.13| 0.03|
|      | Iris     | 0.07| 0.20| 0.13| 0.11| 0.29| 0.24| 0.11| 0.37| 0.28| 0.11| 0.41| 0.32|
|      | Segmentation | 0.38| 0.85| 0.77| 0.46| 0.86| 0.79| 0.47| 0.85| 0.82| 0.55| 0.86| 0.84|
|      | Vehicle  | 0.03| 0.00| 0.01| 0.05| 0.05| 0.01| 0.06| 0.03| 0.03| 0.07| 0.05| 0.02|
|      | Wine     | 0.01| 0.07| 0.00| 0.07| 0.12| 0.05| 0.19| 0.15| 0.07| 0.27| 0.19| 0.09|
for most of the datasets, with two exceptions: the Bupa and Credit datasets using the equalized universe method.

Due to the fact that FuzzyDBD had better results using the equalized universe method for the distribution of the fuzzy sets, this method was chosen as the default method for the distribution of the fuzzy sets for the FuzzyDBD
method. Figure 3.7 illustrates the FuzzyDBD method.

In Figure 3.7, a dataset is the input for the first step of the FuzzyDBD method, which estimates the number of triangular shaped fuzzy sets defining each attribute of the dataset using the Wang & Mendel method. The equalized universe method is then applied to distribute the fuzzy sets in the domains of the attributes and the final fuzzy rule base is defined.

In conclusion, by using the FuzzyDBD method, reasonably small numbers of fuzzy sets were defined for almost all datasets, contributing for the interpretability of the generated fuzzy rule bases. Although the FuzzyDBD method is efficient, fast and consistent, FuzzyDBD defines all the attributes of a given dataset with the same number of fuzzy sets. This fact can be seen as negative, especially when there is information, from an expert, for instance, indicating that the attributes might have different numbers of fuzzy sets.

In order to overcome this problem, we proposed a second version of the FuzzyDBD method in (Cintra et al., 2011e), presented next.

### 3.3 The FuzzyDBD-II Method

The new version of the FuzzyDBD method, named FuzzyDBD-II, uses three distinct estimation functions to individually estimate the number of fuzzy sets for each attribute. In order to propose a new and improved method, keeping the simplicity, easiness of usage, and speed of the FuzzyDBD method, we adopted two reasonable simplifications for the process, reducing the computational cost of the whole process. These two simplifications are related to:

1. The largest number of fuzzy sets considered suitable for each attribute;
2. The assumption of independence among attributes of a given dataset.

The first simplification allows us to search for the best estimation of the number of fuzzy sets within a feasible range. In our experiments this range
comprised from 2 to 9 fuzzy sets per attribute. The upper limit was defined considering that a larger number of fuzzy sets for any given attribute can highly impact the interpretability of the induced models. Regarding the second simplification, it was proposed in order to avoid the exponential complexity of a full search when considering all possible attribute combinations of larger datasets.

Having set these two basic restrictions, the idea of the FUZZYDBD-II method is to evaluate each attribute independently according to different estimation functions, in order to individually define the number of fuzzy sets for each attribute. This idea is described in Figure 3.8: for a given set described by \( m \) attributes, an estimation function is applied to each attribute, individually, considering 2 to 9 fuzzy sets. An optimized number of fuzzy sets is estimated for each attribute, namely the number that produced the best value for the estimation function.

To individually estimate the best number of fuzzy sets for each attribute, we applied three distinct approaches:

1. The information gain measure;
2. The ReliefF ranking filter;
3. The Wang & Mendel method to generate fuzzy rule bases.

The estimation functions used by FUZZYDBD-II are described next.

3.3.1 Estimation Functions

The estimation functions were used to define the number of fuzzy sets for each attribute. For this purpose, each attribute is evaluated considering the
class attribute only. In other words, the set of each attribute and the class attribute is individually evaluated in order to estimate the number of fuzzy sets for each attribute.

In this work, three methods were tested as estimation functions in order to evaluate our proposal for the estimation of the number of fuzzy sets defining each attribute of a dataset. The methods used as estimation functions (information gain, ReliefF, and Wang & Mendel), were assessed in terms of the error rates obtained by a fuzzy decision tree, namely the FuzzyDT. Information gain and ReliefF are described next. FuzzyDT, described in Section 5.1, page 64, is a fuzzy decision tree algorithm based on the classic C4.5 algorithm proposed by Quiln (Quinlan, 1993) that uses the same measures of entropy and information gain used by C4.5 to recursively induce the model.

For the FuzzyDBD-II method, in order to find a direct connection between the estimated number of fuzzy sets and the fuzzy rule base generated using this estimated number, the first step of the process consists of replacing the values of the continuous attributes of the set of examples with fuzzy linguistic terms, represented by fuzzy sets. This discretization process using fuzzy sets is usually referred to as “fuzzyfication”. This way, once the training set is “fuzzyfied” by this first step, replacing the values of the continuous attributes with the label of the fuzzy set with the highest membership degree with this value, the estimation value of each continuous attribute is calculated and stored. In case of ties, a fuzzy set is randomly chosen. The process is repeated 8 times, using from 2 to 9 fuzzy sets for each attribute. The final fuzzy data base is defined using the number of fuzzy sets that produced the highest estimation value for each attribute.

3.3.1.1 The Information Gain Measure as Estimation Function for FuzzyDBD-II

The Information Gain (IG) measure is commonly used in decision tree algorithms to select an attribute defining a test in a given branch of the tree. The IG of a given attribute $X$ with respect to the class attribute $Y$ is the reduction in uncertainty about the value of $Y$ when the value of $X$ is known, noted as $I(Y; X)$.

To illustrate, Table 3.4 presents the IG rates obtained for each attribute ($A_{t1}$ to $A_{t8}$) of the Diabetes dataset from the UCI - Machine Learning Repository (Frank and Asuncion, 2010), defined with 2 to 9 fuzzy sets (first column). The last row presents the chosen number of sets for each attribute. The best results are dark-gray shaded.

Notice that a 0.005 tolerance, empirically defined, was set in order to give preference to smaller numbers of fuzzy sets which always produce a smaller
number of possible rule combinations. For attribute $At_6$, for instance, instead of choosing 9 fuzzy sets, the algorithm chose 7, due to this tolerance. Similarly, for attribute $At_3$, the algorithm chose 4, instead of 8 or 9 sets.

### 3.3.1.2 Using ReliefF as Estimation Function for FuzzyDBD-II

The ReliefF measure (Kononenko, 1994) evaluates the quality of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and of a different class. In other words, ReliefF measures the usefulness of attributes based on their ability to distinguish between similar examples belonging to different classes. It produces an average merit ranking of the features and it is able to operate on both discrete and continuous data.

Table 3.5 presents the ReliefF rates obtained for each attribute ($At_1$ to $At_8$) of the Diabetes dataset defined with 2 to 9 fuzzy sets (first column). The last row presents the chosen number of sets for each attribute. As before, a 0.005 tolerance was also used for choosing the optimal number of fuzzy sets. The best results are dark-gray shaded.

### 3.3.1.3 The Wang & Mendel Method as Estimation Function for FuzzyDBD-II

Due to the simplicity and low computational cost of the Wang & Mendel method (Wang, 2003), in (Cintra et al., 2009a), it was the base for the first version of FuzzyDBD. This first proposal consisted on defining FRBs based on the partitioning of the input attributes into evenly distributed triangular
shaped fuzzy sets. The attributes were all defined by a fixed number of fuzzy sets ranging from 2 to 10. For the FuzzyDBD-II method, instead of using the Wang & Mendel method to estimate a fixed number of fuzzy sets for all attributes of a given dataset, we use it to estimate flexible, and possibly different, numbers of fuzzy sets for each attribute. For this estimation process, each attribute is defined using from 2 to 9 fuzzy sets, and then used individually to generate a Wang & Mendel classifier. The error rates of these classifiers are then used to select the number of fuzzy sets for this attribute, (i.e., the number of fuzzy sets that generated the smallest error rate). A 0.005 tolerance was also used, similarly to the estimation process using IG and ReliefF.

Table 3.6 presents the error rates obtained by the Wang & Mendel method for each attribute (\(At_1\) to \(At_8\)) of the Diabetes dataset defined with 2 to 9 fuzzy sets (first column). The last row presents the chosen number of sets for each attribute. The best results are dark-gray shaded.

Table 3.6: Error rates of the Wang & Mendel method for each attribute of the Diabetes dataset, considering each attribute individually.

| Attribute | \(At_1\) | \(At_2\) | \(At_3\) | \(At_4\) | \(At_5\) | \(At_6\) | \(At_7\) | \(At_8\) |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|
| 2         | 43.70  | 46.80  | 53.70  | 56.30  | 43.70  | 43.70  | 46.80  | 50.00  |
| 3         | 50.00  | 56.30  | 50.00  | 43.70  | 56.30  | 46.80  | 56.30  | 56.30  |
| 4         | 59.50  | 53.20  | 50.00  | 50.00  | 50.00  | 43.70  | 50.00  | 34.20  |
| 5         | 50.00  | 56.30  | 43.70  | 56.30  | 53.20  | 50.00  | 50.00  | 37.40  |
| 6         | 56.30  | 46.80  | 46.80  | 43.70  | 56.30  | 53.20  | 46.80  | 34.20  |
| 7         | 46.80  | 53.20  | 56.30  | 46.80  | 59.50  | 46.80  | 53.20  | 34.20  |
| 8         | 53.20  | 46.80  | 56.30  | 46.80  | 43.70  | 53.20  | 50.00  | 34.20  |
| Best      | 2      | 2      | 5      | 3      | 2      | 2      | 2      | 4      |

Next, we present the experiments and results.

### 3.3.2 Experimental Evaluation of the FuzzyDBD-II Method

The FuzzyDBD-II method was tested using FuzzyDT to induce fuzzy decision trees. The comparisons of the generated models were carried out considering their error rates and the total number of rules. We used 11 datasets from UCI (Frank and Asuncion, 2010) in our experiments. Table 3.7 summarizes the dataset characteristics, presenting the total number of examples, total number of features, including the number of continuous and discrete features in brackets, the number of classes, and the majority error (ME).

To allow further comparisons, we also defined three fuzzy data bases using 3, 5, and 7 evenly distributed fuzzy sets for each attribute of the datasets, representing the most commonly found approach in the literature (i.e., the use of the same number of fuzzy sets for all attributes). Triangular shaped fuzzy sets, evenly distributed in their partitions were used in all experiments.

To illustrate the results obtained in the process of defining the number of fuzzy sets for each attribute, Table 3.8 presents the number of fuzzy sets de-
Table 3.7: Characteristics of the datasets - FuzzyDBD-II.

| Dataset | Examples | Features | Classes | ME |
|---------|----------|----------|---------|----|
| Credit  | 653      | 15       | 6, 9    | 2  |
| Cylinder| 277      | 32       | 19, 13  | 2  |
| Diabetes| 769      | 8        | 8, 0    | 2  |
| Gamma   | 19,020   | 10       | 10, 0   | 2  |
| Glass   | 220      | 9        | 9, 0    | 7  |
| Heart   | 270      | 13       | 13, 0   | 2  |
| Ionosphere| 351     | 34       | 34, 0   | 2  |
| Iris    | 150      | 4        | 4, 0    | 3  |
| Segmentation| 210   | 19       | 19, 0   | 7  |
| Vehicle | 946      | 18       | 18, 0   | 4  |
| Wine    | 178      | 13       | 13, 0   | 3  |

scribing each attribute defined for the Credit and Heart datasets, using the information gain, ReliefF, and Wang & Mendel approaches. Notice that the discrete attributes from the Credit dataset, requiring no estimation, are represented by the symbol ⊗.

Table 3.8: Number of fuzzy sets: Credit and Heart - FuzzyDBD-II.

**Credit**

| Attribute     | At1 | At2 | At3 | At4 | At5 | At6 | At7 | At8 | At9 | At10 | At11 | At12 | At13 | At14 | At15 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|
| Information gain| ⊗  | 5   | 6   | ⊗   | 9   | ⊗   | 9   | ⊗   | 9   | ⊗    | 9    | ⊗    | 9    |
| ReliefF       | ⊗  | 8   | 4   | ⊗   | ⊗   | ⊗   | 9   | ⊗   | 7   | ⊗    | 9    | ⊗    | 9    |
| Wang & Mendel | ⊗  | 3   | 2   | ⊗   | ⊗   | ⊗   | 2   | ⊗   | 2   | ⊗    | 2    | ⊗    | 2    |

**Heart**

| Attribute     | At1 | At2 | At3 | At4 | At5 | At6 | At7 | At8 | At9 | At10 | At11 | At12 | At13 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|
| Information gain| 8   | 2   | 3   | 5   | 6   | 2   | 7   | 2   | 5   | 3    | 3    | 3    |
| ReliefF       | 5   | 4   | 5   | 4   | 9   | 3   | 3   | 9   | 5   | 6    | 5    | 3    | 4    |
| Wang & Mendel | 3   | 2   | 4   | 3   | 3   | 2   | 3   | 2   | 3   | 2    | 3    | 3    | 3    |

It is possible to notice strong variations for the same attributes of the Credit dataset. It is also possible to notice that Wang & Mendel tends to estimate a lower number of fuzzy sets when compared to the other two methods. For the Credit dataset, no agreement is found for any of the attributes regarding all three methods. For the Heart dataset, with no discrete attributes, the differences in the number of estimated fuzzy sets are smaller than for the Credit dataset. The sixth, seventh, and thirteenth attributes present very similar estimations, while the twelfth one has identical estimation for all methods.

Table 3.9 presents the error rates and standard deviation, in brackets, obtained by FuzzyDT, executed with default parameters, using the Information Gain, ReliefF, and Wang & Mendel methods to estimate the number of fuzzy sets for all attributes. Table 3.9 also presents the error rates obtained by FuzzyDT using 3, 5, and 7 fixed sets for all attributes of each dataset. The dark-shaded cells indicate the smallest error rates for each dataset. The average ranking of each method is presented in the last row.

FuzzyDT obtained the smallest error rates for four datasets with the fuzzy data base defined using the information gain measure, with a tie with the fuzzy data base defined using ReliefF, which produced the smallest error rates for other 4 datasets and a triple tie among ReliefF, Wang & Mendel, and the
Table 3.9: Error rates obtained by FUZZYDT using the FUZZYDBD-II method.

| Dataset      | Information Gain | ReliefF | Wang & Mendel | 3  | 5  | 7  |
|--------------|------------------|---------|---------------|----|----|----|
| Credit       | 11.25 (5.36)     | 11.25 (5.80) | 12.81 (6.40) | 12.81 (6.40) | 12.81 (6.40) | 12.19 (5.96) |
| Cylinder     | 16.15 (7.05)     | 11.92 (5.56) | 13.46 (6.93) | 12.31 (5.65) | 10.77 (3.98) | 11.54 (5.16) |
| Diabetes     | 20.53 (4.09)     | 19.87 (4.76) | 22.24 (3.70) | 25.53 (4.53) | 22.24 (3.70) | 23.03 (5.14) |
| Gamma        | 16.16 (6.63)     | 15.80 (6.88) | 16.76 (6.73) | 21.13 (6.70) | 16.76 (6.73) | 16.63 (6.92) |
| Glass        | 24.30 (4.03)     | 36.96 (6.52) | 42.61 (6.68) | 40.43 (5.16) | 42.61 (6.68) | 55.85 (12.27) |
| Heart        | 13.33 (6.02)     | 12.22 (4.40) | 11.48 (4.81) | 14.44 (5.60) | 11.48 (4.81) | 12.59 (6.24) |
| Ionosphere   | 5.00 (3.49)      | 4.41 (3.54)  | 7.65 (3.99)  | 7.35 (5.92)  | 7.65 (3.99)  | 6.76 (5.10)  |
| Iris         | 4.00 (4.42)      | 5.33 (5.81)  | 5.33 (5.81)  | 5.00 (2.67)  | 5.33 (5.81)  | 5.33 (4.00)  |
| Segmentation | 5.71 (3.56)      | 6.67 (4.86)  | 10.48 (6.32) | 12.38 (2.33) | 9.05 (5.81)  | 8.10 (5.24)  |
| Vehicle      | 16.10 (5.85)     | 15.49 (2.38) | 18.17 (2.34) | 27.20 (3.45) | 18.54 (3.57) | 16.83 (2.60) |
| Wine         | 7.14 (5.53)      | 3.57 (3.57)  | 3.57 (3.57)  | 3.57 (3.57)  | 3.57 (3.57)  | 3.57 (3.57)  |
| **Avg.Ranking** | **2.59**        | **2.00**     | **4.09**     | **5.18**     | **3.73**     | **3.41**     |

partition defined with 5 fuzzy sets for all attributes. Wang & Mendel shows two ties with the static definition of 5 fuzzy sets, which, in turn, shows the best error rates for one dataset. Discarding the ties, the static definition of 5 fuzzy sets produced the smallest error rates for FUZZYDT only for the Cylinder dataset. All remaining datasets obtained the smallest error rates for FUZZYDT with the fuzzy data bases defined using the information gain, ReliefF, or Wang & Mendel method.

As previously, in order to check for statistically significant differences among the approaches, we used the Friedman test (Demsar, 2006), which found no statistically significant differences among the results with a 95% confidence level.

Table 3.10 presents the number of rules generated by FUZZYDT using the Information Gain, ReliefF, and Wang & Mendel methods to estimate the number of fuzzy sets of each attribute. It also presents the number of rules generated by FUZZYDT using 3, 5, and 7 fixed fuzzy sets for all attributes of each dataset. Dark-shaded cells highlight the smaller number of rules for each dataset. The average rank of each method is presented in the last row.

Table 3.10: Number of generated rules - FUZZYDT.

| Dataset      | Information Gain | ReliefF | Wang & Mendel | 3  | 5  | 7  |
|--------------|------------------|---------|---------------|----|----|----|
| Credit       | 34               | 29      | 5  | 5  | 5  | 11 |
| Cylinder     | 54               | 78      | 59 | 76 | 69 | 58 |
| Diabetes     | 51               | 54      | 29 | 29 | 29 | 25 |
| Gamma        | 369              | 636     | 229| 229| 229| 679|
| Glass        | 67               | 45      | 25 | 25 | 25 | 49 |
| Heart        | 14               | 31      | 41 | 41 | 41 | 35 |
| Ionosphere   | 33               | 44      | 17 | 21 | 17 | 37 |
| Iris         | 6                | 5       | 5  | 5  | 5  | 13 |
| Segmentation | 66               | 35      | 45 | 45 | 45 | 49 |
| Vehicle      | 287              | 258     | 201| 193| 295|    |
| Wine         | 10               | 18      | 21 | 17 | 21 | 19 |
| **Average Rank** | **4.00**        | **4.23** | **3.09** | **2.05** | **3.00** | **4.64** |

Regarding the number of rules, the Wang & Mendel method statically defining only three fuzzy sets for each attribute produced the smallest sets of rules. Nevertheless, notice that there is a direct connection between the set of possible rules formed by the combination of the fuzzy sets and the total number
of fuzzy sets itself. However, Wang & Mendel shows the lowest error rates (Table 3.9). Due to the large number of examples in the Gamma dataset (Table 3.7), large sets of rules were generated for all induced trees. The Iris and Wine datasets show a comparable number of rules for all approaches.

It should be observed that the FuzzyDBD-II method was proposed in order to overcome the problem of defining only one single number of fuzzy sets for all attributes of a given dataset, as the FuzzyDBD method does.

Although FuzzyDBD-II is able to individually estimate the number of fuzzy sets defining each attribute, depending on the application, it might be necessary to reduce the range of possible fuzzy sets. This reduction in the largest number of possible fuzzy sets for each attribute highly decreases the resulting amount of possible rules for a given domain. This issue is discussed in depth in Chapter 6.

### 3.4 Final Remarks

The fuzzy data base, an important component of fuzzy systems, contains the definitions of the domain attributes in terms of fuzzy sets. The automatic definition of fuzzy data bases is an important task with several methods proposed in the literature. In this chapter, we presented a brief bibliographic review on methods for the automatic definition of fuzzy data bases. We also presented the FuzzyDBD method, which estimates a number of fuzzy sets to define all the attributes of a dataset, as well as the FuzzyDBD-II method, which estimates individual numbers of fuzzy sets for each attribute of a dataset.

In the next chapter, the topic of feature subset selection methods is introduced, as well as our contributions to this topic, which include a wrapper-based method (FuzzyWrapper) and an embedded method (FuzzyDT) for the task of feature subset selection.
The task of Feature Subset Selection (FSS) aims at finding small subsets of features that describe the dataset at hand as well as, or even better than the original set of features. The importance of FSS lies in the fact that, in practice, the performance of most learning algorithms is affected by the presence of irrelevant and/or redundant features.

When automatically generating fuzzy rule bases, a particular challenge to be tackled is the dimensionality problem, usually called dimensionality curse, since for a linear increase in the number of input/output attributes there is an exponential increase in the total number of possible rules. Other advantages associated with feature selection are related to reducing the potential hypothesis space by improving data quality, thus increasing the efficiency of the learning algorithm, while enhancing the comprehensibility of the induced classifier (Yu and Liu, 2004).

This chapter presents the main approaches for the feature subset selection task, filters, wrappers, and embedded methods, followed by a literature review focused on feature subset selection methods for fuzzy systems. This chapter also presents the FUZZYWRAPPER and the FUZZYDT methods proposed in this work, in the context of feature subset selection.

4.1 Approaches for Feature Subset Selection

FSS methods can be classified into three main categories, according to the dependence regarding the classifier:

1. Filters;
2. Wrappers;

3. Embedded methods.

These three categories are discussed next.

**Filters** — Filters have the characteristic of selecting features before the induction of the classifier. This way, filters perform a separate process that does not interact with and is independent from the learning algorithm itself. The basic idea of filters is to use general characteristics of the features to select the relevant ones, according to these characteristics, before the induction of the classifier takes place.

CFS (Hall, 2000), ReliefF (Kira and Rendell, 1992), and Consistency (Liu and Setiono, 1996) are examples of filters commonly used for feature subset selection. CFS stands for Correlation-based Feature Selection and uses the correlation between features for their selection. ReliefF, as previously explained, ranks the features by their usefulness on distinguishing between very similar examples belonging to different classes. The average merit rank of the features produced by ReliefF is used for their selection. Consistency, on the other hand, evaluates the subset of features using the level of consistency between feature values and class values.

These general characteristics used by filters (correlation, consistency, among others) might be appropriate for different domains and tasks. Another advantage of filters is the fact that they are fast and simple to use. Moreover, filters are implemented in most machine learning frameworks, such as KEEL (Bull et al., 2008) and WEKA (Witten et al., 2011).

**Wrappers** — The wrapper approach uses the induction algorithm itself as a black box to evaluate candidate subsets of features, repeating the process on each feature subset until a stopping criterion is met. This way, wrapper methods take into consideration all the important characteristics of the learning algorithm in the final decision of the feature subset selection process. On the other hand, since wrapper-based approaches basically repeatedly run the same learning algorithm with different subsets of features, their computational cost is much higher than filters. To avoid the exponential complexity of the search process, wrappers usually use a forward search strategy (Das, 2001). However, the computational cost of wrapper methods is still prohibitive on large datasets. Moreover, differently from the filter approach, the quality of the set of features selected by wrappers is only guaranteed for the specific learning algorithm used in the process, or to very similar approaches.
In (Cintra et al., 2008a), we proposed a wrapper-based feature selection method, described in Section 4.3.

**Embedded Methods For Feature Subset Selection** — Similarly to wrappers, the feature selection performed by embedded methods is linked with the induction algorithm itself. However, in this case this link is stronger than in wrappers, since the feature selection process is included in the classifier construction. Embedded methods for feature subset selection induce a classifier that usually do not include all possible features, thus, making it possible to define the set of features included in the classifier as the most relevant ones. Moreover, embedded methods select features as a secondary process, i.e., feature subset selection is not the main goal of the method. A typical example of embedded methods for feature subset selection is decision trees (Quinlan, 1993).

Decision trees are widely used in machine learning and various algorithms have been proposed for their generation, such as ID3 (Quinlan, 1986), CART (Breiman et al., 1984), and C4.5 (Quinlan, 1996). These algorithms generate a tree structure through recursively partitioning the feature space until the whole decision space is completely partitioned into a set of non-overlapping subspaces.

C4.5 is a well-known decision tree algorithm proposed and implemented by Quinlan (Quinlan, 1993), which uses the information gain and entropy measures when deciding on the importance of the features, making it possible to select features with the features of the generated branches.

Fuzzy approaches for the induction of decision trees have also been proposed in the literature (Olaru and Wehenkel, 2003), including a fuzzy C4.5 algorithm (Kazunor et al., 2001).

### 4.2 Related Work

Although some fuzzy feature subset selection methods have been proposed in the literature, they are not abundant, and many of the existing proposals are focused on specific applications. The reason for this lack of fuzzy feature subset selection methods lies in the fact that classic feature subset selection methods are often sufficiently suitable to be used with fuzzy systems. Some selected proposals found in the literature of fuzzy feature subset selection methods are presented next.

Hu et al. (2010) propose a filter-based approach for feature subset selection. The proposal uses a weighted distance learning algorithm for FSS by maximizing fuzzy dependencies. After the maximization of the fuzzy depen-
dency between features, a weight vector is generated. The quality of the fea-
tures is evaluated using this weight vector: features deriving great weights
are considered to be useful for supervised learning. Thus, the FSS can be
performed.

In (Sánchez et al., 2008), a feature subset selection method was proposed
using an extended definition of the mutual information measure between two
fuzzyfied continuous variables.

A fuzzy filter is proposed in (Lee et al., 2001), based on the fuzzy entropy
measure. This filter is used to select features for the Iris and Wisconsin
datasets from UCI - Machine Learning Repository (Frank and Asuncion, 2010).

A proposal based on the fitness evaluation of feature subsets is presented
in (Azzeh et al., 2008). The fitness, in this method, uses the minimum simi-
larity between fuzzy clusters which take information about data structure into
account. This method is applied to software effort estimation.

A feature selection method for clustering algorithms, i.e. non-supervised
learning, is proposed in (Wu et al., 2007). Its strategy is based on measuring
the feature importance factor to describe the contribution of each feature to
clustering. The feature importance factor of an attribute is determined based
on the position (or positions) of this attribute in a decision tree structure,
such as C4.5. The idea here is based on the fact that the root attribute in a
decision tree is the most important attribute, and the other ones found in the
subsequent branches have a smaller weight (importance). If a feature appears
more than once in the tree structure, its feature importance factor will be the
sum of its weights in each branch. The method proposed in (Wu et al., 2007)
uses a vector named “clustering feature importance factor” that reflects the
contribution that each feature has to clustering. Notice that this method can
only be considered a fuzzy approach for feature subset selection if the decision
tree used for the calculation of the feature importance factor values is a fuzzy
tree.

It is also possible to find feature subset selection approaches designed to be
used by fuzzy systems, which are, basically, traditional feature subset selec-
tion methods. For instance, in (Alonso and Magdalena, 2009), feature subset
selection is performed using the C4.5 algorithm (Quinlan, 1993). The ap-
proach simply selects the attributes included in the generated decision tree as
the subset of relevant attributes. Moreover, the splitting points for continuous
attributes defined in the decision tree are used to decide on the number of
fuzzy sets for each attribute. Although this method is not fuzzy-based, it is
relevant for this literature review. Moreover, for the process of feature sub-
set selection, the method proposed in (Alonso and Magdalena, 2009) can be
implemented using a fuzzy decision tree instead of a crisp one.
Next, two feature subset selection methods proposed in this work are presented. The first FuzzyWrapper (Cintra et al., 2008a, 2009b), has the Wang & Mendel method as base algorithm for a wrapper-based strategy. The second one, FuzzyDT (Cintra and Camargo, 2010), uses a fuzzy decision tree for the task of FSS.

4.3 The FuzzyWrapper Method

The FuzzyWrapper method preselects features using the Wang & Mendel method (Wang, 2003). Wang & Mendel generates Fuzzy Rule Bases (FRBs) from a dataset and is the base algorithm for FuzzyWrapper. The search technique adopted by FuzzyWrapper is the Best-First Heuristic Search, in order to avoid the exponential complexity of the search process.

The purpose of the FuzzyWrapper method is two-fold. First, to facilitate the use of genetic algorithms to automatically design fuzzy systems by providing a feature subset selection method (the topic of genetic generated fuzzy systems is discussed in details in Chapter 6). Second, FuzzyWrapper is proposed as a method that takes the information regarding the definition of the attributes in terms of fuzzy sets into consideration for the FSS process. Such information include the number and type of fuzzy sets, the membership functions defining each fuzzy set and their distribution, as well as the reasoning method.

Notice that, although wrapper-based approaches generally use the learning algorithm to generate several models, selecting the one with the best performance and the smallest subset of features, our proposal with the FuzzyWrapper method aims at using FuzzyWrapper only to estimate an optimized subset of features. The generation of the FRB is then carried out by a more robust algorithm, since Wang & Mendel cannot be considered competitive with more recent approaches.

4.3.1 Description of the FuzzyWrapper Method

This method uses a wrapper to sequentially generate FRBs using the Wang & Mendel method, described in Section 3.2.1, page 28, by combining the original features of the dataset, as previously stated. FuzzyWrapper is detailed
in Algorithm 2 and illustrated in Figure 4.1.

**Algorithm 2: The FUZZYWRAPPER algorithm.**

1. **Input:** \( m \) = number of attributes;
2. **for** \( a = m \) **to** 1 **do**
   1. Generate all FRBs, using the Wang & Mendel method, combining all possible \( a \) features;
   2. Remove the feature that does not appear in the FRB with the best (lowest) error rate, from the dataset;
3. **end**
4. **Rank** the generated FRBs by their error rates and number of features;
5. **Select** the smallest subset of features that produced the best FRB, i.e., the FRB with lowest error rate.

As the maximum number of fuzzy rules generated by the Wang & Mendel method cannot exceed the number of examples, the complexity of the FUZZYWRAPPER method is \( O(n \times m^2) \). The relevance of this approach for the task of feature subset selection relies on the fact that, by using the Wang & Mendel method as the base algorithm for the wrapper, it is able to take into consideration all important characteristics of the fuzzy logic in the process of feature subset selection. The sharing of the fuzzy definitions of the attributes in the feature subset selection process is an advantage of FUZZYWRAPPER over general feature subset selection methods.

Figure 4.1 illustrates the process used by the FUZZYWRAPPER method to find the best subset of features for a dataset described by 4 attributes (\( At_1 \), \( At_2 \), \( At_3 \), and \( At_4 \)), and the best-first search strategy. The error rate for the Wang & Mendel method using all four attributes is 0.27. The red (lighter) arrows indicate the sequence of the selected subsets. The subset with three attributes with smallest error is the one without attribute \( At_1 \), thus the process continues testing the set of attributes \( \{ At_2, At_3, At_4 \} \). The removal of attribute \( At_4 \) from this set produces the smallest error rate, thus, attributes \( At_2 \) and \( At_3 \) are tested. The best error rate is obtained with the set of attributes \( \{ At_2, At_3, At_4 \} \), which is dark-gray shaded.

### 4.3.2 Experimental Evaluation of the FUZZYWRAPPER Method

The FUZZYWRAPPER method was evaluated using both, the classic (CFRM) and the general (GFRM) fuzzy reasoning methods. All experiments based on the Wang & Mendel method were performed using a 10-fold cross validation strategy, with paired folds (i.e., all experiments used identical subsets of examples). Four datasets from the UCI Machine Learning repository (Frank and Asuncion, 2010), described by numerical-valued features, were used. Since the FUZZYWRAPPER method was proposed before the FUZZYDBD method, in this work a partition with three triangular fuzzy sets for each input feature was empirically defined for each domain. Table 4.1 summarizes the dataset.
characteristics giving the total number of examples, total number of features (including the number of continuous and discrete features in brackets), the majority error (ME), and the number of classes.

To allow further comparisons with the FuzzyWrapper method, three filters available in the WEKA framework (Witten et al., 2011) — CFS (Hall, 2000), ReliefF (Kira and Rendell, 1992), and Consistency (Liu and Setiono, 1996) — were used, as well as the original version of C4.5 (Quinlan, 1993) to select features. Since ReliefF ranks the features by their usefulness on distinguishing between very similar examples belonging to different classes, presenting an average merit ranking for each feature, this value was used to select the features presenting an average merit greater than 0.009. The experiments were carried out using the default settings of all algorithms.

Table 4.2 presents the original number of features of each dataset, the number of features selected by the FuzzyWrapper method using the classic and general fuzzy reasoning methods (FuzzyWrapper-C, and FuzzyWrapper-G, respectively), as well as the number of features selected using the CFS, ReliefF, and Consistency filters, as well as the C4.5 decision tree algorithm, as an embedded approach. The smallest subsets of selected features are dark-gray shaded.

It should be observed that for the Diabetes dataset, FuzzyWrapper using the classic and general fuzzy reasoning methods selected the same set of 6 attributes. Results show that for the Heart dataset, FuzzyWrapper-C selected the smallest subset of features, while for Vehicle, the smallest subset
Table 4.2: Original and selected number of features - FuzzyWrapper.

| Datasets | Features |FUZZYWRAPPER-C |FUZZYWRAPPER-G |CFS |ReliefF |C4.5 |Consistency |
|----------|----------|---------------|---------------|----|--------|-----|------------|
| Diabetes | 8 | 6 | 6 | 4 | 6 | 5 | 8 |
| Heart    | 13 | 3 | 6 | 8 | 12 | 12 | 10 |
| Vehicle  | 18 | 10 | 10 | 11 | 18 | 18 | 18 |
| Wine     | 13 | 7 | 7 | 10 | 3 | 13 | 5 |

of features was selected by FuzzyWrapper-C and FuzzyWrapper-G. For the Diabetes and Wine datasets, the smallest subset of features were selected by CFS and C4.5, respectively.

Since the Wang & Mendel method is the base algorithm of FuzzyWrapper, it was used to generate FRBs and calculate their error rates with the features selected by the filters and by C4.5. Table 4.3 presents the error rates and standard deviation, in brackets, for the fuzzy rule base generated by the proposed method (FuzzyWrapper) and for the FRBs generated with the features selected by CFS, C4.5, ReliefF and Consistency, using the CFRM and the GFRM reasoning methods.

Table 4.3: Error rates - FuzzyWrapper.

| Dataset   | Error Rates - Classic Fuzzy Reasoning Method | Error Rates - General Fuzzy Reasoning Method |
|-----------|---------------------------------------------|---------------------------------------------|
| Diabetes  | FuzzyWrapper | CFS | ReliefF | C4.5 | Consistency | FuzzyWrapper | CFS | ReliefF | C4.5 | Consistency |
| Diabetes  | 0.32(0.09) | 0.34(0.13) | 0.65(0.63) | 0.37(0.09) | 0.28(0.05) | 0.29(0.09) | 0.32(0.09) | 0.65(0.63) | 0.37(0.09) | 0.28(0.05) |
| Heart     | 0.19(0.12) | 0.24(0.08) | 0.32(0.09) | 0.31(0.08) | 0.29(0.12) | 0.16(0.08) | 0.22(0.07) | 0.29(0.09) | 0.30(0.09) | 0.26(0.11) |
| Vehicle   | 0.36(0.05) | 0.32(0.05) | 0.43(0.04) | 0.43(0.04) | 0.43(0.04) | 0.40(0.07) | 0.65(0.07) | 0.41(0.07) | 0.41(0.07) | 0.41(0.07) |
| Wine      | 0.03(0.05) | 0.05(0.08) | 0.11(0.08) | 0.13(0.08) | 0.14(0.10) | 0.04(0.06) | 0.04(0.06) | 0.08(0.08) | 0.12(0.07) | 0.05(0.08) |

For both, the classic and general fuzzy reasoning methods, FuzzyWrapper produced the smallest error rates for Heart, Vehicle, and Wine datasets, while Consistency produced the smallest error for Diabetes. Nevertheless, it should be observed that Consistency was not able to reduce any features. Moreover, for the Diabetes dataset, the error rates using the features selected by CFS, ReliefF, and C4.5 are greater than the majority error, for both reasoning methods. Also, for the Wine dataset, using the general fuzzy reasoning method, there is a tie between the results of the FuzzyWrapper method and the CFS filter, although the former uses less features (7, against 10 of CFS).

As previously done, the Friedman test (Demsar, 2006) was used to assess the methods in terms of the error rate. Results showed that the FuzzyWrapper method is significantly better than ReliefF and C4.5, with a 95% confidence level.

Since only the number of selected features or only the smallest error rates are not the best individual parameters to evaluate the obtained results, we used the multi criteria model proposed in (Lee et al., 2006) for the evaluation.
of the performance of the feature subset selection algorithms. This approach considers two aspects simultaneously: the reduction in the number of features versus the quality of the induced classifier using the subset of selected features.

In this framework, shown in Figure 4.2, where EFS stands for the classifier error without FSS and EMC represents the majority error (whenever this error is less than 50%, otherwise it is set to 50%), the performance of the feature subset selection algorithms is placed into five categories:

1. Excellent (⋆⋆⋆);
2. Very good (⋆⋆);
3. Good (⋆);
4. Poor (⊕);
5. Very poor (●).

Using this model, both versions of the proposed algorithm (using the CFRM and the GFRM), were empirically evaluated and compared with the filters and the embedded methods for feature subset selection.

Notice that, since this model is crisp, some rough transitions might occur. The most unnatural transition occurs when a method has a small reduction in the number of features and it obtains error rates close to the error rate obtained with no feature selection: for a small difference in the error rate, a method can be classified as “excellent”, “good”, or “very poor”.

For the classic fuzzy reasoning method, FUZZYWRAPPER obtained “excellent” for three datasets, while CFS for two datasets. ReliefF, C4.5 and Consistency obtained “excellent” for one dataset each. For the general fuzzy reasoning method, ReliefF obtained excellent results for one dataset, while the other methods obtained excellent results for two datasets.
### 4.3.3 Expanded Experimental Evaluation — FUZZYWRAPPER

Due to the promising results of the FUZZYWRAPPER method just described, using four datasets, we expanded the experiments using eight datasets and an automatic fuzzy data base definition method, the FUZZYDBD method (Cintra et al., 2009a), described in the previous chapter (Section 3.2, page 27), instead of empirically defining the parameters of the fuzzy data base, as done in the previous work (Cintra et al., 2008a). FUZZYDBD basically compares the performance of fuzzy rule bases generated by the Wang & Mendel method, in terms of accuracy, testing from 2 to 10 triangular fuzzy sets defining the attributes of a given dataset. The distribution of the fuzzy sets, using FUZZYDBD, is done according to the equalized universe method (Chen and Wang, 1999), which uniformly distributes the fuzzy sets in the partitions. All experiments based on the Wang & Mendel method were performed using a 10-fold cross validation strategy, with paired folds.

Table 4.5 summarizes the dataset characteristics giving the total number of examples, total number of features (including the number of continuous and discrete features in brackets), number of classes, the majority error (ME), and the number of fuzzy sets defining each of the attributes (FS).

| Dataset   | Examples | Features | Classes | ME     | FS |
|-----------|----------|----------|---------|--------|----|
| Credit    | 653      | 15 (6, 9)| 2       | 45.33  | 2  |
| Diabetes  | 769      | 8 (8, 0 )| 2       | 34.90  | 2  |
| Glass     | 220      | 9 (9, 0 )| 7       | 65.46  | 7  |
| Heart     | 270      | 13 (13, 0)| 2     | 44.44  | 2  |
| Ionosphere| 351      | 34 (34, 0)| 2     | 35.90  | 3  |
| Segmentation| 210   | 19 (19, 0)| 7     | 85.71  | 3  |
| Vehicle   | 846      | 18 (18, 0)| 4     | 74.23  | 2  |
| Wine      | 178      | 13 (13, 0)| 3     | 59.74  | 3  |

For the expanded experiments, we also compared FUZZYWRAPPER with the three filters used in the previous work, CFS (Hall, 2000), ReliefF (Kira and Rendell, 1992), and Consistency (Liu and Setiono, 1996), as well as the original version of the C4.5 (Quinlan, 1993), both in term of the number of selected features and in terms of the correct classification rates. Another difference from the previous work is the fact that we adopted a tolerance interval in order to select the smallest subset of features when the error rates are comparable, similarly to what was done with the FUZZYDBD method. The tolerance interval adopted is 0.05.
Table 4.6 presents the original number of features of each dataset, the number of features selected by the FUZZYWRAPPER method using the classic (FUZZYWRAPPER-C) and general (FUZZYWRAPPER-G) reasoning methods, as well as the number of features selected by CFS, C4.5, ReliefF, and Consistency. It should be observed that the features selected using filters were obtained considering all examples in the dataset, thus, the results can be considered optimistic when compared to the features selected using the FUZZYWRAPPER method, which were obtained using a cross validation strategy. The smallest sets of features are dark-gray shaded.

Table 4.6: Original and selected number of features - FUZZYWRAPPER expanded.

| Dataset   | Features | FUZZYWRAPPER-C | FUZZYWRAPPER-G | CFS | C4.5 | ReliefF | Consistency |
|-----------|----------|----------------|----------------|-----|------|---------|-------------|
| Credit    | 15       | 12             | 12             | 7   | 9    | 12      | 13          |
| Diabetes  | 8        | 2              | 7              | 4   | 6    | 5       | 8           |
| Glass     | 9        | 6              | 6              | 7   | 9    | 8       | 7           |
| Heart     | 13       | 3              | 6              | 8   | 12   | 12      | 10          |
| Ionosphere| 34       | 7              | 7              | 14  | 33   | 7       |             |
| Segmentation | 19  | 8              | 12             | 7   | 12   | 15      | 9           |
| Vehicle   | 18       | 10             | 10             | 11  | 18   | 18      | 18          |
| Wine      | 13       | 7              | 7              | 10  | 3    | 13      | 5           |

Notice that, due to the adoption of a tolerance interval, the results for the Diabetes dataset obtained in this work are different from the results obtained in the previous work. This difference reflects the fact that the subsets of features selected for the Diabetes dataset in the previous work are different from the ones selected in this work.

Results show that the FUZZYWRAPPER-C method was able to select the smallest number of features for 6 out of the 8 datasets, with one tie with Consistency. However, it is important to notice that for the Credit and Wine datasets, although CFS and C4.5, respectively, were able to select a smaller subset of features, the smallest error rates for these datasets were obtained using the subset of features selected by the FUZZYWRAPPER method (see Table 4.7).

Since the base algorithm for FUZZYWRAPPER is the Wang & Mendel method, FRBs were generated by Wang & Mendel using the features selected by the filters and by C4.5, in order to allow fair comparisons. Table 4.7 presents the error rates and standard deviation (in brackets) for FUZZYWRAPPER, and for the FRBs generated with the features selected by CFS, ReliefF, C4.5, and Consistency, using the classic and the general fuzzy reasoning methods. The best results are dark-gray shaded.

For both reasoning methods, FUZZYWRAPPER obtained the best (smallest) error rates for all datasets, except for a tie for the Diabetes dataset using Consistency and the classic fuzzy reasoning method. However, for Diabetes, results were poor, since the error rate of FUZZYWRAPPER-C is similar to the
### Table 4.7: Error rates - FUSWYWRAPPER expanded.

| Dataset   | All | FUSWYWRAPPER-C | CFS | ReliefF | C4.5 | Consistency |
|-----------|-----|----------------|-----|---------|------|-------------|
| Credit    | 0.21 (0.09) | 0.20 (0.06) | 0.43 (0.08) | 0.23 (0.03) | 0.43 (0.02) | 0.28 (0.04) |
| Diabetes  | 0.40 (0.07) | 0.40 (0.04) | 0.51 (0.08) | 0.47 (0.06) | 0.51 (0.06) | 0.40 (0.05) |
| Glass     | 0.42 (0.02) | 0.37 (0.09) | 0.54 (0.09) | 0.42 (0.07) | 0.50 (0.07) | 0.48 (0.06) |
| Heart     | 0.47 (0.03) | 0.19 (0.03) | 0.24 (0.03) | 0.32 (0.02) | 0.31 (0.06) | 0.29 (0.08) |
| Ionosphere| 0.34 (0.05) | 0.24 (0.07) | 0.41 (0.08) | 0.36 (0.09) | 0.44 (0.04) | 0.39 (0.10) |
| Segmentation | 0.25 (0.02) | 0.21 (0.01) | 0.27 (0.03) | 0.26 (0.03) | 0.26 (0.07) | 0.34 (0.01) |
| Vehicle   | 0.76 (0.02) | 0.36 (0.03) | 0.52 (0.09) | 0.43 (0.10) | 0.43 (0.09) | 0.43 (0.03) |
| Wine      | 0.10 (0.08) | 0.03 (0.04) | 0.05 (0.05) | 0.10 (0.01) | 0.13 (0.04) | 0.14 (0.10) |

| Dataset   | All | FUSWYWRAPPER-G | CFS | ReliefF | C4.5 | Consistency |
|-----------|-----|----------------|-----|---------|------|-------------|
| Credit    | 0.21 (0.09) | 0.17 (0.06) | 0.35 (0.08) | 0.21 (0.05) | 0.29 (0.04) | 0.22 (0.05) |
| Diabetes  | 0.28 (0.09) | 0.34 (0.13) | 0.48 (0.63) | 0.45 (0.41) | 0.44 (0.25) | 0.56 (0.32) |
| Glass     | 0.39 (0.03) | 0.39 (0.04) | 0.50 (0.04) | 0.39 (0.04) | 0.49 (0.03) | 0.50 (0.04) |
| Heart     | 0.56 (0.08) | 0.16 (0.08) | 0.22 (0.07) | 0.29 (0.01) | 0.30 (0.09) | 0.26 (0.11) |
| Ionosphere| 0.32 (0.03) | 0.23 (0.05) | 0.34 (0.04) | 0.32 (0.04) | 0.33 (0.05) | 0.36 (0.06) |
| Segmentation | 0.27 (0.03) | 0.20 (0.06) | 0.33 (0.05) | 0.29 (0.03) | 0.28 (0.03) | 0.34 (0.02) |
| Vehicle   | 0.41 (0.07) | 0.40 (0.07) | 0.65 (0.07) | 0.41 (0.07) | 0.41 (0.07) | 0.41 (0.07) |
| Wine      | 0.12 (0.07) | 0.04 (0.06) | 0.04 (0.06) | 0.08 (0.08) | 0.12 (0.07) | 0.05 (0.08) |

ME. Nevertheless, for Diabetes, FUSWYWRAPPER-G is the only one having an error smaller than the ME. The same happens with the Ionosphere dataset using FUSWYWRAPPER-C.

The Friedman test was also used to check for statistically significant differences in the performance of each algorithm. Since the Friedman test found differences, the Bonferroni-Dunn post-hoc test (Demsar, 2006) was used to detect if the differences were significant. Results of the Bonferroni-Dunn tests shows that the FUSWYWRAPPER method is significantly better than all the other four algorithms with a 95% confidence level.

In conclusion, the results show that the FUSWYWRAPPER method is able to achieve a good reduction in the number of features as well as keeping error rates comparable or inferior to the ones of the fuzzy rule bases generated with the features selected by filters and by the C4.5 algorithm, as an embedded method. In fact, FUSWYWRAPPER obtained better error rates than using all features, except for the Diabetes dataset using the classic fuzzy reasoning method.

It is important to emphasize the fact that FUSWYWRAPPER uses the Wang & Mendel method aiming only at selecting features. The Wang & Mendel method was also selected to be the base algorithm for accuracy comparisons due to the fact that it is used in the core of FUSWYWRAPPER. This way, although the Wang & Mendel method is not competitive as a fuzzy rule base algorithm, its results can be considered suitable to provide evidence of the effectiveness of FUSWYWRAPPER as a feature subset selection method.

The second FSS method proposed in this work is described next.
4.4 Feature Subset Selection Using FUZZYDT

The FUZZYDT method, described in detail in the next chapter, is a fuzzy version of the classic C4.5 decision tree algorithm (Quinlan, 1993) proposed in (Cintra and Camargo, 2010). As previously stated, decision trees generate models that usually include only a subset of the original features used for the induction of the model. Thus, since this subset contains only the relevant attributes for the decision tree algorithm, it is possible to use decision trees for the task of feature subset selection.

The FUZZYDT method uses the same measures of C4.5 (entropy and information gain) to decide on the importance of the features while inducing the tree. The main difference between them is the fact that FUZZYDT performs a discretization of continuous attributes using the fuzzy sets defining them. In other words, for FUZZYDT, the continuous attributes of the training set are fuzzyfied before the induction of the tree. Thus, the process can be seen as the induction of a tree using only discrete features, since the continuous features are defined in terms of fuzzy sets and the fuzzyfied training set is used for the decision tree induction.

After the induction of the tree, FUZZYDT, similarly to C4.5, performs a post pruning process with a default confidence level of 25%.

Decision trees can be seen as a set of disjunct rules in which only one rule is fired for a given input example. Regarding fuzzy decision trees, they can be seen as a set of rules that can be fired simultaneously, each one presenting a class and a degree of compatibility for a given input pattern to be classified. This degree of compatibility is calculated using the membership degrees of the linguistic values of the rules and the values of the attributes. This characteristic of the fuzzy decision trees allows the use of the classic and general fuzzy reasoning methods, i.e., since more than one rule derived from the decision tree can be fired, an input instance can be classified with the class of the rule with highest compatibility with the input pattern (classic fuzzy reasoning method), or with the class with the highest sum from the set of rules with that given class. Next, we present the experiments and results obtained by FUZZYDT as a feature subset selection method.

In order to evaluate the ability of FUZZYDT to select features, we compared its results with the FUZZYWRAPPER method, the classic C4.5 decision tree, and the three baseline filters (available in the WEKA framework (Witten et al., 2011)) used in the previous work: CFS, ReliefF and Consistency (described in Section 4.1). The datasets used in this work include the 8 datasets of the previous work, and two new ones, all available at UCI (Frank and Asuncion, 2010). Only the classic reasoning method was adopted since it performed bet-
ter than the general reasoning method in previous experiments (Cintra et al., 2009b).

Table 4.8 summarizes the dataset characteristics giving the total number of examples (Examples), total number of features (Features), including the number of continuous and discrete features (in brackets), number of classes (Classes), the majority error (ME), and the number of fuzzy sets defining the attributes (FS). Regarding the number of fuzzy sets describing each feature, they were defined by the FuzzyDBD method described in Section 3.2.

Table 4.8: Dataset characteristics - FuzzyDT.

| Dataset      | Examples | Features | Classes | ME   | FS |
|--------------|----------|----------|---------|------|----|
| Credit       | 653      | 15 (6, 9)| 2       | 45.33| 2  |
| Cylinder     | 277      | 32 (19, 13)| 2      | 35.74| 2  |
| Dermatology  | 358      | 34 (33, 1)| 6      | 68.99| 2  |
| Diabetes     | 769      | 8 (8, 0) | 2       | 34.90| 2  |
| Glass        | 220      | 9 (9, 0) | 7       | 65.46| 7  |
| Heart        | 270      | 13 (13, 0)| 2      | 44.44| 2  |
| Ionosphere   | 351      | 34 (34, 0)| 2      | 35.90| 3  |
| Segmentation | 210      | 19 (19, 0)| 7      | 85.71| 3  |
| Vehicle      | 846      | 18 (18, 0)| 4      | 74.23| 2  |
| Wine         | 178      | 13 (13, 0)| 3      | 59.74| 3  |

Table 4.9 presents the original number of features of each dataset, the number of features selected by FuzzyWRAPPER, as well as the ones selected by FuzzyDT, C4.5, CFS, ReliefF, and Consistency. The average number of features is also presented. The best results are dark-gray shaded. Notice that the fuzzy methods for feature subset selection are separated from the classic ones by a double bar.

Table 4.9: Original and selected number of features - FuzzyDT.

| Dataset      | Features | Fuzzy Methods | Classic Methods |
|--------------|----------|---------------|-----------------|
|              |          | FuzzyWRAPPER | FuzzyDT | C4.5 | CFS | ReliefF | Consistency |
| Credit       | 15       | 12            | 10       | 9    | 7   | 12      | 13          |
| Cylinder     | 32       | 9             | 24       | 24   | 12  | 14      | 16          |
| Dermatology  | 34       | 17            | 17       | 17   | 16  | 16      | 8           |
| Diabetes     | 8        | 2             | 5        | 6    | 4   | 5       | 8           |
| Glass        | 9        | 6             | 6        | 9    | 7   | 8       | 7           |
| Heart        | 13       | 3             | 5        | 12   | 8   | 12      | 10          |
| Ionosphere   | 34       | 7             | 10       | 14   | 14  | 33      | 7           |
| Segmentation | 19       | 8             | 10       | 12   | 7   | 15      | 9           |
| Vehicle      | 18       | 10            | 11       | 18   | 11  | 18      | 18          |
| Wine         | 13       | 7             | 8        | 13   | 10  | 13      | 5           |
| Average      | 19.5     | 8.1           | 10.6     | 10.8 | 9   | 14.8    | 9.4         |

Results show that the FuzzyWRAPPER method was able to select the smallest number of features for 5 out of the 10 datasets, with one tie with FuzzyDT. The FuzzyWRAPPER also shows the best (smallest) average number of selected features. ReliefF, on the other hand, obtained the worst average of selected features, reducing one of the original number of features for the Glass, Heart and Ionosphere datasets, and none for the Wine and Vehicle datasets.

As before, in order to evaluate the quality of the selected sets of features, we used the Wang & Mendel method to generate fuzzy rule bases and compare
their accuracy. The classification power of these sets of selected features can be seen as an indication of their quality to describe the dataset. Moreover, notice that, as previously stated, the Wang & Mendel method is a quite simple approach to generate fuzzy rule bases and was used in these experiments due to its simplicity and easy implementation only for preliminary comparisons. The results with the Wang & Mendel method, which is not a competitive method in terms of the accuracy of its generated models, are suitable to give us information in order to evaluate the feature subset selection methods.

Table 4.10 presents the error rates and standard deviation (in brackets) obtained with the Wang & Mendel method. Both FuzzyWrapper and FuzzyDT were used to generate rule bases with the features selected by FuzzyWrapper (FW/WM), as well as with C4.5 (C4.5/WM), CFS (CFS/WM), ReliefF (RF/WM), and Consistency (Cons/WM). Notice that column FDT presents the error rates obtained by the FuzzyDT model itself, while FDT/WM presents the error rates of the base generated by Wang & Mendel with the features selected by FuzzyDT. Column All presents the Wang & Mendel error obtained using all attributes. The double column division in the middle of the table separates the first three methods (FuzzyWrapper, FuzzyDT, and FuzzyDT- WM) that perform the selection of features using the fuzzyfied data, from the other methods that use the original data. The best results are dark-gray shaded. The last row presents the ranking of the methods according to their error rates.

Notice that FuzzyDT (FDT) is used as an embedded method, since it induces a fuzzy classifier and selects features at the same time. On the other hand, for the FDT/WM and the C4.5/WM, the decision tree (fuzzy or not) is used only to select the features which appear in the generated branches. These features are then used by the Wang & Mendel method to generate the fuzzy rule base.

For this setup, FuzzyDT was able to obtain the smallest error rates for 7 out of the 10 datasets, with one tie with the FuzzyWrapper method, which obtained the smallest error rates for the remaining datasets. FuzzyDT also performed reasonably well in terms of feature selection. Disregarding the re-
results obtained by FUZZYDT, FUZZYWRAPPER obtained the smallest error rates for all datasets. FUZZYWRAPPER was also able to select the smallest sets of features. The features selected by the ReliefF filter produced the third best results, but it is important to bear in mind that ReliefF was the worst approach in terms of selected features. Notice that Wang & Mendel did not yield good results for any approach, but for FUZZYWRAPPER. In fact, in many cases the results are worse than the majority error.

Although FUZZYWRAPPER shows a competitive performance, it is important to notice that the use of wrappers might not be a reasonable alternative to select features of large datasets due to its computational cost. In this sense, FUZZYDT can be considered a good alternative for the task of feature subset selection for fuzzy systems. Also, its excellent results, both in terms of precision and number of rules (discussed next), motivated us to perform more experiments to compare it with other more robust fuzzy rule-based methods, such as genetic approaches. Part of these experiments were submitted for publication (Cintra et al., 2012a) and are detailed in Section 7.4.

As before, the Friedman test was used to check for statistically significant differences in the results of the methods. The last row of Table 4.10 shows the Friedman ranking. Since significant differences were found, the Bonferroni-Dunn test was executed to compare each algorithm against the others (Demšar, 2006). Results show that the FUZZYWRAPPER method is better than the CFS method and significantly better than C4.5 (comparing the rule base generated by the Wang & Mendel method). The Bonferroni-Dunn test also shows that FUZZYDT using the rule base of the induced trees is significantly better than FUZZYDT and ReliefF, both using the Wang & Mendel fuzzy rule base. Finally, results show that FUZZYDT, using the decision tree rule base, is extremely better than C4.5 and CFS, both using the Wang & Mendel fuzzy rule base.

In conclusion, FUZZYDT is able to select subsets of features with high classification power, being a good alternative to wrapper approaches for datasets with many features. Furthermore, FUZZYDT is a quite simple and fast algorithm, producing comparable results, and having high scalability.

4.5 Final Remarks

This chapter briefly described the feature subset selection task, as well as its approaches, which are usually classified as filters, wrappers and embedded methods.

We proposed the FUZZYWRAPPER method in (Cintra et al., 2008a), a wrapper-based feature subset selection method that uses the Wang & Mendel method
as its core algorithm. The FuzzyWrapper method was further investigated in (Cintra et al., 2009b). The main advantage of the FuzzyWrapper method over traditional FSS methods is the fact that since it uses a fuzzy algorithm (Wang & Mendel) as its core algorithm, the fuzzy data base used by FuzzyWrapper to perform feature subset selection is the same used by the fuzzy method that generates the fuzzy rule bases. Thus, the FuzzyWrapper takes all the information regarding the definition of the attributes in terms of fuzzy sets into consideration when performing feature selection.

In (Cintra and Camargo, 2010), we proposed a decision tree-based method used for feature selection, namely the FuzzyDT method, which has the same advantage of the FuzzyWrapper method (it takes the information of the fuzzy data base into consideration for its feature selection process), but which is faster and requires less computation than FuzzyWrapper. FuzzyDT is also well scalable to datasets with many attributes, while FuzzyWrapper is not.

The FuzzyWrapper and FuzzyDT methods were presented, together with related experiments, comparisons and conclusions.

Next chapter discusses the topic of fuzzy decision trees and presents a real-world application of FuzzyDT.
Machine learning is concerned with the development of methods for the extraction of patterns from data in order to make intelligent decisions, based on these patterns. A relevant concern related to machine learning methods is the issue of interpretability, which is highly desirable for final users. In this sense, Decision Trees (DT) (Quinlan, 1993) are powerful and popular models for machine learning since they are easily understandable, quite intuitive, and produce models that can be expressed as rules. The induction process of decision trees is usually fast and the induced models are usually competitive with the ones generated by other interpretable machine learning methods. Another quality of decision trees is related to their induction process, which also performs a feature selection, as seen in Chapter 4.

ID3 (Quinlan, 1986), CART (Breiman et al., 1984), and C4.5 (Quinlan, 1996) are among the most relevant decision tree based methods. Regarding fuzzy decision trees, several proposals can be found in the literature (Olaru and Wehenkel, 2003), including a fuzzy version of the classic C4.5 algorithm in (Kazunor et al., 2001). Notice that the paper in (Kazunor et al., 2001) is written in Japanese. In this work, we propose our fuzzy version of C4.5, named FUZZYDT (Cintra and Camargo, 2010).

FUZZYDT was applied for the induction of classifiers, as well as for feature subset selection. As previously mentioned, FUZZYDT was also applied to a real-world problem, the prediction and control of the coffee rust disease in Brazilian crops.

This chapter discusses decision trees and presents the FUZZYDT algorithm.
5.1 Decision Trees and the FuzzyDT Algorithm

Decision trees are widely used in machine learning because of their simple induction and powerful representation of knowledge. As previously mentioned, fuzzy decision trees have also been proposed in the literature, including FuzzyDT, our fuzzy version of C4.5, was proposed in (Cintra and Camargo, 2010). Next, the C4.5 and the FuzzyDT algorithms are detailed and compared.

5.1.1 C4.5

C4.5 uses the information gain and entropy measures when deciding on the importance of the features, recursively creating branches corresponding to the values of the selected features, until a class is assigned as a terminal node. Each branch of the tree can be seen as a rule, whose conditions are formed by the attributes and their respective tests. As usual in decision tree induction algorithms, C4.5 includes a pruning process by means of a post-pruning strategy, i.e., the pruning takes place after the tree is completely induced. The pruning process basically assesses the error rates of the tree and its components directly on the set of training examples (Quinlan, 1993).

To understand the process of pruning, assume \( N \) training cases are covered by a leaf, \( E \) of them incorrectly. This way, the error rate for this leaf is defined by \( E/N \). If we take this set of \( N \) training cases as a sample, it can tell us something about the probability of error over the entire population of examples covered by this leaf. This probability cannot be precisely determined, but it has a probability distribution that is usually summarized by a pair of confidence limits. For a given confidence level \( CF \), the upper limit of this probability can be found from the confidence limits for the binomial distribution; this upper limit is here written as \( U_{CF}(E, N) \). As the upper and lower binomial distribution limits are symmetrical, the probability that the real error rate exceeds \( U_{CF}(E, N) \) is \( CF/2 \). As pointed out by Quinlan, although one might argue that this heuristic is questionable, it frequently yields acceptable results (Quinlan, 1993).

The default confidence limits used by C4.5 is 25%. However, it is important to notice that the smaller the confidence limit, the higher the chances of pruning, and the higher the confidence limit, the smaller the chances of pruning. Thus, if the confidence limit is set to 100%, we are stating that the predicted error, obtained with the examples at hand, is equal to the real error, and no pruning is performed. This idea conflicts with the natural intuition that a 25% confidence limit will produce less pruning than an 80% confidence limit, for instance. This way, one should not associate the default 25% confidence limits of C4.5 with actually pruning 25% of the generated tree.
5.1.2 The FuzzyDT algorithm

FuzzyDT\(^1\), proposed by us in (Cintra and Camargo, 2010), uses the same measures of the classic C4.5 algorithm (entropy and information gain) to decide on the importance of the features. However, for FuzzyDT, continuous features are defined in terms of fuzzy sets before the induction of the tree. This way, the process of inducing a tree using FuzzyDT takes a set of “discretized” features, since the continuous features are defined in terms of fuzzy sets and the training set is fuzzyfied before the decision tree induction takes place. Algorithm 3 describes FuzzyDT.

**Algorithm 3:** The FuzzyDT algorithm.

1. Define the fuzzy data base, i.e., the fuzzy granulation for the domains of the continuous features;
2. Replace the continuous attributes of the training set using the linguistic labels of the fuzzy sets with highest compatibility to the input values;
3. Calculate the entropy and information gain of each feature to split the training set and define the test nodes of the tree until all features are used or all training examples are classified;
4. Apply a post-pruning process, similarly to C4.5, using 25% confidence limits as default.

As the fuzzyfication of the training data is done before the induction of the tree, the third step of the FuzzyDT algorithm corresponds to the same step of the classic decision tree algorithm. Figure 5.1 illustrates the process of data fuzzyfication and tree induction for a toy dataset with \(n\) examples, 3 attributes (\(At_1\), \(At_2\), and \(At_3\)), and 3 classes (\(C_a\), \(C_b\), and \(C_c\)).

![Original dataset](#) ![Fuzzyfied dataset](#) ![Induced Tree](#)

**Figure 5.1:** The FuzzyDT algorithm - a toy example.

As previously stated, a special issue regarding classic decision trees is the fact that they can be seen as a set of disjunct rules in which only one rule is fired to classify a new example. However, although a fuzzy decision tree can be seen as a set of rules, differently from the classic model, many of these rules are fired simultaneously, each one with a degree of compatibility with

\(^1\)A first version of FuzzyDT is available for download at [http://dl.dropbox.com/u/16102646/FuzzyDT.zip](http://dl.dropbox.com/u/16102646/FuzzyDT.zip)
an input example, in order to classify this new example. Figure 5.2 illustrates this special characteristic of fuzzy decision trees with a toy example based on the Iris dataset from UCI (Frank and Asuncion, 2010). This example uses just two attributes to classify new inputs into three types of iris plants: Setosa, Versicolor, and Virginica. The left decision tree in Figure 5.2 is the classic one, while its similar fuzzy version is the right one in Figure 5.2. In order to be able to interpret the fuzzy linguistic variables and linguistic terms of the fuzzy decision tree, Figure 5.3 presents the fuzzy sets defining attributes Petal Length and Petal Width used in Figure 5.2. For simplification, just two triangular shaped sets (small and large) evenly distributed in the partitions were used. Figure 5.3 shows the membership functions for Petal Length and Petal Width.

![Classic Decision Tree](image1.png)  ![Fuzzy Decision Tree](image2.png)

**Figure 5.2:** A classic and a fuzzy decision tree for the Iris dataset.

![Fuzzy sets](image3.png)

**Figure 5.3:** Fuzzy sets defining attributes Petal Length and Petal Width.

The inference process of the classic decision tree is quite straightforward: if Petal Width is ≤ 0.6, the example belongs to the Virginica class, otherwise, the Petal Length attribute is tested; if it is ≤ 4.9, the example is classified as Versicolor, otherwise it is classified as Setosa. This way, considering a new input example to be classified having Petal Length = 4.9 and Petal Width = 0.6,
both values on the borderline of the crisp discretization of the classic decision tree, only the first rule is fired:

**IF Petal Width is ≤ 0.6 THEN Class is Virginica.**

However, for the fuzzy decision tree, the membership degrees of the input example, shown in Figure 5.3 (Petal Length Small = 0.66; Petal Length Large = 0.34; Petal Width Small = 0.79; Petal Width Large = 0.21) are used to calculate the compatibility degree of the input example with each rule. For this particular example, using minimum as t-norm, the compatibility degrees of the fuzzy rules are (in brackets):

1. **IF Petal Width is Small THEN Class is Virginica (0.79)**
2. **IF Petal Width is Large AND Petal Length is Small THEN Class is Versicolor (0.21)**
3. **IF Petal Width is Large AND Petal Length is Large THEN Class is Setosa (0.21)**

For this example, using the classic fuzzy reasoning method (best rule), the class of the first rule is used to classify the example as Virginica, which is the same class defined by the classic decision tree.

Now, let us assume that the Petal Width of the input example is 0.61, while the Petal Length continues the same. Notice that the difference in the Petal Width between this example and the last one is not significant (0.01). This way, we are likely to believe, intuitively, that the class of such similar examples should be the same. Nevertheless, the classic decision tree classifies this new example as belonging to the Versicolor class. The fuzzy decision tree, on the other hand, since it uses the compatibility degrees of the input values with the fuzzy sets defining the tree tests, still classifies this new example with the same class, Virginica. Notice that the same situation can occur for Petal Length, as well as with any continuous attribute with a crisp discretization. This robustness of fuzzy decision trees is highly desirable.

It follows a more detailed comparison between classic and fuzzy decision trees.

### 5.1.3 Classic VERSUS Fuzzy Decision Trees

Classic and fuzzy decision trees, although sharing the same basic idea of building a tree like structure by partitioning the feature spaces, also present some relevant differences. Next, we discuss these similarities and differences, using the classic C4.5 and FuzzyDT algorithms for the comparisons.

**Evaluation of features** — For the partitioning process, both versions use the same measures, entropy and information gain, in order to select the features to be used in the test nodes of the tree;
**Induction process** — Both versions use the same approach: repeated subdivision of the feature space using the most informative features until a leaf node is reached or no features or examples remain;

**Handling continuous features** — The classic version splits the domain into crisp intervals according to the examples at hand by minimizing entropy and maximizing information gain. This process might cause unnatural divisions that reflect on a lower interpretability of the rules. As a practical illustration, let us consider the Vehicle dataset, from UCI, which has *Compactness* as its first test attribute of a decision tree induced by C4.5. *Compactness* is a continuous attribute with real values ranging from 73 to 119. For the tree induced by C4.5, it is possible to find the following tests using *Compactness*:

1. **IF** *Compactness* is $\leq$ 95 **AND** ... **AND** *Compactness* is $\leq$ 89
2. **IF** *Compactness* is $\leq$ 95 **AND** ... **AND** *Compactness* is $>$ 89
3. **IF** *Compactness* is $>$ 95
4. **IF** *Compactness* is $\leq$ 102
5. **IF** *Compactness* is $>$ 102
6. **IF** *Compactness* is $\leq$ 109 **AND** ... **AND** *Compactness* is $\leq$ 106
7. **IF** *Compactness* is $\leq$ 109 **AND** ... **AND** *Compactness* is $>$ 106
8. **IF** *Compactness* is $>$ 109
9. **IF** *Compactness* is $\leq$ 82 **AND** ... **AND** *Compactness* is $\leq$ 81
10. **IF** *Compactness* is $\leq$ 82 **AND** ... **AND** *Compactness* is $>$ 81
11. **IF** *Compactness* is $>$ 82 **AND** ... **AND** *Compactness* is $\leq$ 84
12. **IF** *Compactness* is $>$ 82 **AND** ... **AND** *Compactness* is $>$ 84

These 12 tests make the understanding of the rules difficult since, for a whole understanding of the model, the user has to keep in mind the subspaces defined by each rule. A particular problem happens when the subdivisions are relatively close, strongly restraining the domain of the features, such as in rule 10 ($81 < \text{Compactness} \leq 82$).

Another issue regarding the handling of continuous features by C4.5 is related to the fact that the number of divisions used to split continuous attributes is not handled directly by the algorithm, even if this parameter is previously known or defined. In fact, for the algorithm to use a previously defined number of divisions for any attribute, the attribute in question has to be discretized before the induction of the decision tree, since the number of divisions splitting continuous attributes is dynamically determined during the tree induction process. This way, the number determined by the decision tree might be different from the number of divisions used by an expert, for example. Notice that in the example provided, the decision tree uses 8 different splitting points for the same
attribute (81, 82, 84, 89, 95, 102, 106, and 109), some of them very close to each other.

FuzzyDT, on the other hand, is able to use the partitions (in terms of fuzzy sets) defined by an expert. Furthermore, even if this information is not available, automatic methods for the generation of fuzzy partitions can be used, most of them controlling and preventing the creation of unnatural splitting points.

**Reuse of features** — for C4.5, the same continuous feature can be included several times in one single rule (for example, feature *Compactness* in the previous example). This repetition of the same feature and subdivision of the domain degrades the interpretation of the rule.

On the other hand, the induction process of FuzzyDT can be seen as inducing a decision tree with fuzzyfied (discretized) attributes, thus a feature is never used more than once in the same rule. This fact favours the interpretability of the generated rules.

**Inference** — the inference process for C4.5 is quite straightforward: it first tests the root attribute and then sequentially tests the following triggered nodes of the tree until a class node is reached. This way, in case the user requires an explanation regarding the classification of a new example, the explanation is the corresponding branch of the tree. The process is intuitive and clear. However, for continuous features, whenever the input values are located in the decision frontiers, misclassifications might occur, such as the ones exemplified in Figure 5.2.

FuzzyDT, on the other hand, determines the degree of membership for an input values in each fuzzy set defining the given attribute. This way, a confidence degree is calculated for each rule. This confidence degree is used by the classification process and gives credibility to the final classification. Notice that, for a well defined fuzzy data base, each input value presents membership degrees greater than zero in at most two fuzzy sets, i.e., the two corresponding branches of the test node are fired simultaneously. On the other hand, if the decision tree is large, the inference process might require considerable computational effort when compared to the classic C4.5, since more branches are fired. This issue can be softened by defining a minimum threshold of membership degree in order to continue testing the rules or not.

However, for FuzzyDT, in case the user requires an explanation regarding the classification of a new example, the explanation should include all the branches of the tree which have participated of the example classification.
In conclusion, decision trees are important algorithms for machine learning since they have a powerful representation of knowledge, easy usage, can handle continuous and discrete attributes, as well as being highly scalable for large datasets. Moreover, the fuzzy version of the classic C4.5 algorithm, FuzzyDT, proposed in this work, shows the robustness of fuzzy decision trees, including several advantages over the classic C4.5 algorithm.

5.2 The Use of FuzzyDT on the Prediction and Control of the Coffee Rust Disease

FuzzyDT was used in a real-world application for the coffee rust disease prediction and control (Cintra et al., 2011b), in a collaboration with researchers from EMBRAPA, Brazilian Agricultural Research Corporation and from the State University of Campinas - UNICAMP. For the induction of the models, a condensed dataset with data collected during 8 years was used. The fuzzy decision tree models were compared with models induced by a classic decision tree algorithm, taking into account their accuracy and syntactic complexity, as well as the quality of the models according to the opinion of an expert. The fuzzy models showed better accuracy power and interpretability than the classic one.

5.2.1 Contextualization and Related Work

Coffee rust (Hemileia vastatrix Berk. & Br.) is the main disease in coffee crops (Coffea arabica L.) in the world. In regions of Brazil, where climate conditions favour the disease, losses can reach about 35%, and sometimes even more than 50% (Zambolim et al., 1997). In addition to its economical importance, the coffee rust disease meets other requirements that justify the development of forecast and warning models, according to its variations among crop seasons and availability of economically viable control measures.

Several empirical models to predict some features of coffee rust have been proposed. Adjusting the observed data to regression equations is the most common modelling technique (Zambolim et al., 2002). In (Kushalappa et al., 1983), a model to explain the biological action course of the H. vastatrix is proposed. The values obtained with this model from data observed in the field were used in the development of regression equations to predict the progress rate of the disease (Kushalappa et al., 1984). In (Pinto et al., 2002), the potential of neural networks to tackle the coffee rust epidemic is evaluated. In

---

2 http://www.embrapa.br/english
3 http://www.unicamp.br/unicamp/en
(Garcon et al., 2004), a prediction model using a calculated matrix based on
the severity values of the coffee rust is empirically evaluated.

The induction of Decision Trees (DTs) is an alternative modelling technique. DTs
are easy to interpret. Furthermore, the visual approach of DTs is par-
ticularly helpful in comprehending sequential decisions and outcome depen-
dencies. Differently from regression techniques, the multicollinearity among
independent variables does not affect the performance of the DTs (Butt and
Royle, 1990). Also, several variables, numeric and categoric, can be analysed
at the same time, since the DT induction algorithm itself controls the selection
of the most relevant variables.

Decision trees were used in (Paul and Munkvold, 2004) to model and pre-
dict severity categories of gray leaf spot of maize, as well as in (Baker et al.,
1993) to predict the risk (high or low) of the mortality of pine trees due to the
Annosus root disease. Specifically related to coffee rust, epidemics of the dis-
ease were analysed with the aid of DTs in (Meira et al., 2008). DTs have also
been used to predict the infection rate of coffee rust in (Meira et al., 2009).

Next, the data collection, experiments and comparisons using FUZZYDT for
the prediction and control of coffee rust are presented.

5.2.2 Data Collection

The data used in (Cintra et al., 2011b) were collected by Japiassu et al.
(2007) and refer to the monthly following up of the coffee rust incidence
at the Experimental Farm of the Procafé Foundation, in Varginha, Minas
Gerais, Brazil (21°34’0”S, 45°24’22”W), during 8 years (October, 1998 – Octo-
ber, 2006). For each year, four plots with large spacing (about 4,000 plants/ha)
and four dense plots (about 8,000 plants/ha) were selected. For each spacing,
two plots with large fruit load (above 30 bags/ha) were selected. The sam-
pling method adopted was the one recommended by Chalfoun (1997). The
final condensed dataset includes 182 examples from the total of 192 available
examples (representing the monthly averages recorded during 8 years, for two
different plantation spacings). The remaining 10 samples were discarded due
to problems in the collection process. No disease control was done for the
agricultural year in the selected plots.

Meteorological data, such as air temperature (average, maximum, and min-
umum), pluvial precipitation, and relative humidity of the air, were recorded
every 30 minutes by an automatic meteorological station (Davis company,
Groweather Industrial model) installed close to the places of the coffee rust
incidence evaluation. The coffee rust progress between one evaluation and
the following, i.e., the infection rate, was defined as the dependent variable
(class attribute). Monthly infection rates were calculated by subtracting the
disease incidence of the current month from the incidence of the previous one. The numerical values of the infection rates were mapped into two categories (or classes) (Meira et al., 2009).

Two infection rates were used to define the class attribute. The first option of the binary infection rate was defined by creating variable INF_RATE_G5, with value 1 for infection rates equal or greater than 5 percentage points (pp), and 0 otherwise. The second option, INF_RATE_G10, was defined to allow further comparisons, with value 1 for infection rates equal or greater than 10 pp, and 0 otherwise. The decision threshold set in 5 pp was determined based on the 5% incidence limit of the coffee rust recommended by Zambolim et al. (1997) for the disease control. The decision threshold set to 10 pp was determined based in (Kushalappa et al., 1984), which proposed the risk limit of 10% of incidence to recommend the use of fungicide. It is important to notice that this value (10%) is close to the upper limit of 12% of sick leaves recommended for the use of systemic fungicides (Zambolim et al., 1997).

The meteorological predictive attributes were derived for probable periods of infection of the *H. vastatrix* determined by estimates of the incubation period of the fungus (calculated with a regression equation). Some of these attributes aim at contemplating known results and aspects of the disease epidemics found in the literature. Further details on the data preparation steps can be found in (Meira, 2008).

5.2.3 Experimental Evaluation - Coffee Rust Prediction and Control

Experiments were carried out with J48, a C4.5 implementation available in the WEKA (Witten et al., 2011) framework, as well as with our implementation of FuzzyDT. Both algorithms were executed using default parameters, except for the minimum number of examples in terminal nodes, which was set to 5. The following aspects were considered for the comparisons of the results obtained by C4.5 and FuzzyDT:

- Performance, in terms of error rates;
- Interpretability of the generated models, in terms of their syntactic complexity, taking into consideration the number of rules generated and the number of conditions of these rules, *i.e.*, the rule length.

To allow a more detailed analysis of the experiments, the original dataset was used to generate six different datasets. Table 5.1 describes the general characteristics of these datasets, presenting their number of features, including the number of continuous and discrete features, respectively (in brackets),
number of classes, majority error (ME), and number of fuzzy sets defining the attributes (FS). The number of fuzzy sets was empirically defined. Triangular fuzzy sets evenly distributed in the partitions were used. The experiments with FuzzyDT used the classic fuzzy reasoning method, which classifies a new example using the class with the highest compatibility with it.

Table 5.1: Characteristics of the datasets and number of fuzzy sets - Coffee rust prediction and control.

| Dataset | Features | Classes | ME   | FS |
|---------|----------|---------|------|----|
| M1G5    | 23 (22, 1) | 2       | 46.15 | 3  |
| M1G10   | 23 (22, 1) | 2       | 29.12 | 3  |
| M2G5    | 14 (13, 1) | 2       | 46.15 | 3  |
| M2G10   | 14 (13, 1) | 2       | 29.12 | 3  |
| M3G5    | 11 (10, 1) | 2       | 46.15 | 3  |
| M3G10   | 11 (10, 1) | 2       | 29.12 | 3  |

The names of the datasets are intended to present information regarding the set of variables used and the infection rate for the class. Three predictive attribute selections were chosen. The first option, identified by the prefix “M1”, included all 23 predictive attributes. The second selection, prefix “M2”, included attributes derived from meteorological variables measured by sensors available in most meteorological stations. This way, we aim at widening the use of the models. The third subset of original features, prefix “M3”, excluded from M2 the attributes whose preparation depended on hour recordings. This is due to the fact that, in general, public data available from meteorological stations of other institutions are often daily summarized. Again, the motivation here was to widen the use of the induced models. Both, the attributes and the three dataset options (M1, M2 and M3) are listed in Table 5.2. G5 and G10 refer to the infection rates, i.e., the binary class variable, separating examples according to the infection rate thresholds of 5 pp and 10 pp, respectively.

Table 5.2 describes the variable names, where the two first ones are class related, type (T: binary (B) or numeric (N)), unit (U) (mm: millimetres; h: hour; mm/h: millimetres per hour; km/h: kilometres per hour; °C: degrees Celsius; %: percentage), and descriptions of all attributes. Columns named 1, 2, and 3 present a check mark for the variables included in the corresponding datasets M1, M2, and M3. The contractions for average (avg.) and temperature (temp.) are used.

5.2.4 Results and Discussion

The focus of the coffee rust control must be placed on large fruit load years, when the disease progress is faster and the attack more severe. The warning models presented in this section predict the infection of a following month based on 5 pp and 10 pp increase rates. These models can be used to support
Table 5.2: Description of the variables - Coffee rust prediction and control.

| Variable Name | T | U | Description |
|---------------|---|---|-------------|
| INF_RATE_G5   | B | - | Infection rate: 1 if \(\geq 5\) pp, 0 otherwise |
| INF_RATE_G10  | B | - | Infection rate: 1 if \(\geq 10\) pp, 0 otherwise |
| 01 - RAINY_DAYS | N | days | Number of rainy days (precipitation \(\geq 1\) mm) |
| 02 - SPACING  | B | - | Spacing: dense or wide crops |
| 03 - RAIN_PREC_AVG | N | mm | Average of the daily rain precipitation |
| 04 - AVG_MAX_RAIN | N | mm/h | Maximum daily rain precipitation average |
| 05 - NH_RH_95 | N | h | Night hours with air relative humidity \(\geq 95\)% avg. |
| 06 - DH_RH_95 | N | h | Hours with air relative humidity \(\geq 95\)% avg. |
| 07 - AC_RAIN_PREC | N | mm | Rain precipitation accumulated |
| 08 - SUM_NH_RH_95 | N | h | Sum of NH_RH_95 |
| 09 - SUM_DH_RH_95 | N | h | Sum of DH_RH_95 |
| 10 - WIND_SUM_AVG | N | km/h | Wind speed sum average |
| 11 - D_TEMP_AVG_95 | N | °C | Daily temp. average - air relative humidity \(\geq 95\)% |
| 12 - AVG_MAX_T_IP | N | °C | Maximum daily avg. temp. for incubation period |
| 13 - AVG_TEMP | N | °C | Average of the temperatures |
| 14 - AVG_T_IP | N | °C | Daily temperature average for incubation period |
| 15 - AVG_MIN_T_IP | N | °C | Minimum daily avg. temp. for incubation period |
| 16 - REL_HUM | N | % | Daily air relative humidity |
| 17 - WIND_SPEED | N | km/h | Average of the daily wind speed |
| 18 - UID | N | days | Unfavourable days for infection |
| 19 - VFID | N | days | Favourable and very favourable days for infection |

The decision of the moment to act, as well as what type of measures must be taken, for the disease control.

Both methods presented similar execution time. Regarding the accuracy of the induced models, Table 5.3 presents the majority error of each dataset (ME) and the error rates (Error) and standard deviation (SD) obtained by the FUZZYDT and J48 algorithms. The error rates were calculated using a 10-fold cross-validation strategy. The smaller error rate for each model is light-grey shaded.

Table 5.3: Error rates - Coffee rust prediction and control.

| Dataset | ME | FUZZYDT | J4.8 |
|---------|----|---------|------|
| M1G5    | 46.15 | 15.29 (9.56) | 18.72 (8.59) |
| M1G10   | 29.12 | 17.06 (8.90) | 21.90 (8.23) |
| M2G5    | 46.15 | 16.47 (12.85) | 17.82 (7.91) |
| M2G10   | 29.12 | 17.06 (6.65) | 23.53 (10.08) |
| M3G5    | 46.15 | 16.47 (13.62) | 18.07 (8.71) |
| M3G10   | 29.12 | 16.47 (9.77) | 21.36 (9.59) |

It can be observed that FUZZYDT obtained the smallest error rates for all datasets. However, for both algorithms, J48 and FUZZYDT, the error rate standard deviations were considerably high.

Aiming to verify if the high error rate standard deviations were due to the kind of model induced (DTs), we investigated the results obtained by other models. To this end, the following learning algorithms, all of them available in the WEKA framework, were executed with default parameters:

- Nearest Neighbour using 3 neighbours – which classifies an input with the class of its closest examples;
• Naive Bayes – a probabilistic model;

• Random Forests – a DT-based approach that induces rules based on the divide and conquer strategy and partial decision trees;

• Multi-layer perceptron – a neural network approach.

Nevertheless, all these learning algorithms presented higher error rates than those obtained by FuzzyDT, also with high standard deviation rates. With these results, we can conclude that the high standard deviation rates are not related to the kind of model induced. For the DT models, the high error rate standard deviations indicate that there are some rules with high misclassification rates. This problem can be tackled by focusing on these rules in order to search for possible exceptions, as proposed in (Prati et al., 2004). Moreover, for FuzzyDT, the definition of the class as a fuzzy attribute may also help to alleviate the problem. The idea is to define the infection rate into three fuzzy sets, according to the rates used for prevention and intervention.

To test whether there was a statistically significant difference among the algorithms, the Wilcoxon (Demsar, 2006) matched pair test and the Mann-Whitney (Wilcoxon, 1945) test were performed. Both tests concluded that FuzzyDT and J48 differ significantly with 95% confidence.

Table 5.4 presents the number of rules (Rules), total number of conditions, and the number of features present in each model for FuzzyDT and J48, of the models generated with all the training examples. The smaller number of rules, conjunctions, and features for each model are dark-gray shaded.

Table 5.4: Number of rules, conditions, and features - Coffee rust prediction and control.

| Dataset | FuzzyDT | | | J48 | | |
|---------|---------|---------|---------|---------|---------|---------|
|         | Rules   | Conditions | Features | Rules   | Conditions | Features |
| M1G5    | 5       | 8        | 2       | 6       | 20        | 4       |
| M1G10   | 7       | 15       | 3       | 8       | 34        | 7       |
| M2G5    | 3       | 3        | 1       | 5       | 13        | 3       |
| M2G10   | 9       | 22       | 4       | 4       | 9         | 3       |
| M3G5    | 7       | 15       | 3       | 5       | 12        | 4       |
| M3G10   | 7       | 15       | 3       | 4       | 9         | 3       |

Regarding the number of rules and conditions, FuzzyDT had smaller values for the first three datasets, while J48 for the remaining ones. It is interesting to notice that for the first three datasets, although the difference in the number of rules is not expressive, J48 presented a considerable larger number of conditions when compared to FuzzyDT. In other words, J48 induced more specialized rules. On the other hand, FuzzyDT presented a larger number of conditions for the M2G10 dataset.

Another important issue regarding the interpretability of the models is the number of different attributes they include. FuzzyDT presented a smaller
number of attributes for four datasets and a tie for one dataset (M3G10), while J48 presented less attributes for only one dataset (M2G10). Notice that although FuzzyDT produced larger numbers of rules and conditions for the last three datasets, the number of attributes included in the models is not larger than the ones included in the J48 induced models. In fact, as previously discussed, in general, for a fuzzy decision tree, two branches of a node will be fired for a continuous input value, while for a classic decision tree, only one branch is fired. Consequently, a fuzzy decision trees is bound to include more conditions in its model than a classic decision tree.

For the M1G5 dataset, the model induced by J48 presented an issue related to classic decision trees: the repeated use of the same numerical attribute in a single branch/rule. This is shown in the following rules/branches. Notice that these two rules represent only part of the induced model. This characteristic makes it harder to understand the whole model.

1. IF $UID \leq 25$ AND $AVG\_MAX\_T\_IP \leq 28.6$ AND $D\_TEMP\_AVG\_95 \leq 17.6$ AND $WIND\_SUM\_AVG \leq 2.9$ AND $D\_TEMP\_AVG\_95 \leq 16.3$ THEN Class is 0

2. IF $UID \leq 25$ AND $AVG\_MAX\_T\_IP \leq 28.6$ AND $D\_TEMP\_AVG\_95 \leq 17.6$ AND $WIND\_SUM\_AVG \leq 2.9$ AND $D\_TEMP\_AVG\_95 > 16.3$ THEN Class is 1

In order to allow further analysis, next we present and discuss the model induced by J48 for the M3G10 dataset — Figure 5.4. This model uses three attributes with binary outcome tests. According to this model, the effect of the daily air relative humidity ($REL\_HUM$) has prevailed if compared to the temperature influence. Feature $REL\_HUM$ is included with higher importance in the faster evolution periods of the coffee rust. Higher average temperatures in the incubation period ($AVG\_MAX\_T\_IP$) and lower daily average temperatures ($AVG\_TEMP$) during the infection period had a negative effect on the infection rates, confirming epidemiological studies of the disease (Meira et al., 2008).

The fuzzy model contains the following set of rules:

• IF $AVG\_MIN\_TEMP$ is Low THEN Class is 0

• IF $AVG\_MIN\_TEMP$ is Medium THEN Class is 0

• IF $AVG\_MIN\_TEMP$ is High AND $REL\_HUM$ is Low THEN Class is 1

• IF $AVG\_MIN\_TEMP$ is High AND $REL\_HUM$ is Medium THEN Class is 0

• IF $AVG\_MIN\_TEMP$ is High AND $REL\_HUM$ is High AND $AVG\_MAX\_T\_IP$ is Low THEN Class is 1

• IF $AVG\_MIN\_TEMP$ is High AND $REL\_HUM$ is High AND $AVG\_MAX\_T\_IP$ is Medium THEN Class is 1

• IF $AVG\_MIN\_TEMP$ is High AND $REL\_HUM$ is High AND $AVG\_MAX\_T\_IP$ is High THEN Class is 0

The tree representation of the fuzzy model is presented in Figure 5.5. Although the model also uses three attributes, one of them differs from the J48 model (for the FuzzyDT model, the average minimum temperature is
used, while J48 uses the average temperature). From the interpretability point of view, it is more pleasant to read the fuzzy model. Although the linguistic values must be representative for the user, the fuzzy model does not have unnatural separations of attributes as in the J48 model, which shadows the understanding of the model and might force different classifications for very similar examples.

The fuzzy model shows the importance of the daily minimum temperatures average attribute (AVG_MIN_TEMP). It should be observed that the leaf wetness necessary for the germination of the spores of the *H. vastatrix* generally happens during the night period, when the minimum daily temperature also occurs. The other two variables are the same as the J48 model, with compatible behaviours in both models, with exception to the following rule:

**IF** AVG_MIN_TEMP is **High** AND REL_HUM is **Low** **THEN** Class is 1.

From the expert’s point of view, specifically for this rule, class 0 would be expected.

For large fruit load years, late atomizations are not recommended after the confirmation that the incidence levels of the coffee rust is higher than 5% (*Zambolim et al.*, 2002). This way, the MxG5 models are more indicated for the decision support of the disease control. MxG10 models, on the other hand,
might be used as additional instruments informing and alerting about the fact that measures taken should be more urgent and/or effective, based on the fact that conditions are favourable for an even more accelerated development of the disease.

In conclusion, FUZZYDT presented competitive error rates and models. FUZZYDT also has the advantage of producing better interpretable models with no unnatural discretization of the attributes. It is important to highlight the fact that, although the fuzzy models showed better performance for all datasets, the improvement, when compared to the J48 models, was more expressive for the MxG10 datasets.

Regarding the selection of features for the definition of the three models (M1, M2, and M3), models M3 (with the smallest subset of attributes) had competitive performance, having the advantage that meteorological hourly records are not included.
5.3 Final Remarks

In this Chapter, we discussed decision trees and fuzzy decision trees. The advantages of decision trees are related to their powerful representation of knowledge in the form of rules, their easy usage, high scalability, and the fact that they can handle discrete and numeric variables.

The FuzzyDT algorithm was presented. FuzzyDT is a fuzzy decision tree based on the classic C4.5 algorithm that incorporates the characteristics of fuzzy systems related to their ability to handle imprecision and uncertainty. FuzzyDT has been used to induce fuzzy decision trees, as well as for feature subset selection, presenting competitive results.

FuzzyDT was also applied to a real-world application, related to the control of coffee rust, the most important existing coffee disease. Experiments and comparisons with J48, an implementation of C4.5, showed that FuzzyDT produced better accuracy than J48 in the WEKA framework, with comparable models, in terms of interpretability.

Next chapter discusses the genetic fuzzy systems, including a taxonomy of existing approaches. We present a literature review as well as the DoC-BASED and FCA-BASED methods proposed in this work.
Genetic Fuzzy Systems (GFS) combine the power of the global search of Genetic Algorithms (GA) with fuzzy systems, providing accurate and interpretable rule-based systems. There has been a considerable effort on the research of GFSs, with several proposals published in the literature. Specifically for our research work on GFSs, we focus on the genetic generation of fuzzy rule bases using a predefined and fixed fuzzy data base. Our main contribution to this area is the proposal of the FCA-BASED method, which uses the theory of formal concept analysis (Cintra et al., 2011a) to extract rules from data in order to generate the genetic search space, followed by a genetic process to generate the final fuzzy rule bases. The FCA-BASED method presents an alternative with reduced computational cost and better scalability for the definition of the genetic search space when compared to a previous proposal, the DoC-BASED method.

In this chapter, first we introduce the topic of the GFSs, presenting a literature review of the most up-to-date methods for the genetic generation of fuzzy rule bases. In the sequel, we present our contribution to the area, first the DoC-BASED method and then the FCA-BASED method, as well as the experimental evaluations and results. In order to make this work self-contained, we also present an introduction including the most relevant aspects of formal concept analysis as well as illustrative examples.

6.1 Genetic Fuzzy Systems

A large number of different GFS approaches has been proposed in the literature. In 2004, on the occasion of the 10th anniversary of the first proposal of
a genetic fuzzy system, a special paper, which can be considered a milestone in the area of genetic fuzzy systems, was published (Cordon et al., 2004). Its importance is related to the classification and overall description of the most relevant approaches proposed until then, the indications of future work and open problems, as well as a critical evaluation of the contributions of the genetic fuzzy systems for the area.

More recently, in (Herrera, 2008), prof. Francisco Herrera proposed a new classification of the GFS according to a more updated and comprehensive taxonomy, presenting examples for each group. This classification is shown in Figure 6.1.

![Figure 6.1: Classification of Genetic Fuzzy Systems (Herrera, 2008).](image)

The classification of GFSs presented in Figure 6.1-(a) separates the genetic approaches into two main groups:

1. Methods that perform genetic tuning;
2. Methods that perform genetic learning.
The genetic tuning group includes methods for the tuning of the FDB, the FRB, or both, *i.e.*, the FKB, as well as tuning methods related to inference engine. The genetic learning of components, on the other hand, includes the methods for the learning of the FKB, or its components, as well as methods related to the learning of the fuzzy knowledge base and the inference engine parameters.

Figure 6.1-(b) details the approaches applied to the genetic learning of the fuzzy rule base, which are classified into:

1. Genetic rule learning;
2. Genetic rule selection;
3. Genetic learning of fuzzy data bases;
4. Genetic learning of the whole fuzzy knowledge base, *i.e.*, the FRB and the FDB.

Our work focuses on the genetic learning of FRBs using a predefined FDB and explicitly generating the search space of the genetic process. Such approaches are shown in Figure 6.1-(b): "genetic rule selection with a priori rule extraction".

6.1.1 Challenges for the Automatic Generation of Fuzzy Systems

Independently of the adopted approach, some issues are frequently singled out in the literature as the most challenging for the automatic design of fuzzy systems, such as:

- The data and, consequently, rule dimensionality issue;
- The trade-off between accuracy and interpretability.

These issues are discussed briefly in the sequel, due to their importance for this work.

6.1.1.1 The Dimensionality Issue

The well-known expression "dimensionality curse" characterizes the increasing complexity of processes searching for a "good" expression to represent a concept. A concept is represented by the fuzzy rule base, in the context of this work. A general approach to tackle this problem, suiting classic and fuzzy methods, is the use of a feature subset selection method.

Feature subset selection methods aim at selecting the most important attributes characterizing a concept. In this work, we investigated two approaches
for feature subset selection: wrapper based and embedded approaches. To this end, the FuzzyWrapper method (Cintra et al., 2008a, 2009b) (see Section 4.3, page 49), a wrapper-based feature subset selection method, as well as an embedded approach based on the FuzzyDT decision tree algorithm (Cintra and Camargo, 2010) (see Section 5.1.2, page 65), were proposed.

The second alternative to tackle the dimensionality problem, specifically in the context of the genetic generation of fuzzy rules by means of the genetic rule selection approach with a priori rule extraction (Figure 6.1-(b)), takes into consideration previous knowledge about the data domain in order to reduce the search space prior to the rule generation process. This approach was proposed in (Ishibuchi et al., 1992) and adopted in (Ishibuchi and Yamamoto, 2004), where a set of candidate rules is generated from numerical data using rule evaluation measures (support and confidence). Following this generation step, a genetic process is used to select the rule set that defines the final fuzzy rule base. A similar approach, named Doc-Based, was proposed in (Cintra and Camargo, 2007a), where the pre-selection of candidate rules is based on the degree of coverage of the rules. Furthermore, in order to reduce the computational cost of the extraction process of the candidate rules, as well as for scalability sake, a method that extracts the candidate rules from data, using the formal concept analysis theory, the FCA-Based method (Cintra et al., 2012c) was proposed. Both the Doc-Based and the FCA-Based methods are described in this chapter.

6.1.1.2 The Issue of the Accuracy and Interpretability Trade-off

Accuracy and interpretability are requirements for a rule-based fuzzy system. Accuracy, or precision, is related to the ability of properly representing a system, while interpretability is related to the ability of expressing the behaviour of the system in an understandable manner. As previously explained, in practice, these goals are conflicting, and one of them tends to prevail. Due to the importance of both goals, several recent fuzzy systems present some mechanism to balance accuracy and interpretability.

Notice that both issues (the dimensionality issue and the trade-off between accuracy and interpretability) are connected: large domains tend to require large rule sets in order to obtain suitable accuracy, leading to low interpretability.

In order to tackle the accuracy versus interpretability issue, it is possible to search for rule bases that are both, accurate and interpretable, simultaneously. This approach is adopted by the FCA-Based method, proposed in this work, as well as by the previously proposed Doc-Based method.

Another possible approach to tackle the ‘dimensionality curse”, in order to
obtain reasonable levels of accuracy and interpretability, and which is widely used in the literature, is the use of dedicate processes for the reduction of rule bases, by discarding redundant and conflicting rules. This rule selection process allows the definition of an optimized rule set, based on a previous rule base. Such approach is adopted by the FCA-BASED method.

It follows a bibliographic review of genetic and bio-inspired proposals for the automatic generation of fuzzy systems, in which we discuss the adopted approaches, when one is used, to tackle the “dimensionality curse” and the trade-off between interpretability and accuracy.

6.2 Related Work

An updated and comprehensive bibliographic compilation on genetic fuzzy rule-based fuzzy systems including publications from 2007 on, can be found in http://sci2s.ugr.es/gfs/biblio.php. This compilation is maintained by one of the leading research groups in fuzzy systems, the “Soft Computing and Intelligent Information Systems” research group., with the Granada University, Spain, lead by Dr. Francisco Herrera. The following discussed proposals were selected from this bibliographic compilation, according to their relevance for this work.

Recently, there has been several GFS proposals for real-world applications. In fact, real-world applications of genetic fuzzy systems is listed as a trend in (Cordón et al., 2007). Although the several proposals on GFSs tackle relevant problems, most of them present adaptations of previously proposed methods, focusing on the applications at hand.

6.2.1 Genetic Fuzzy Systems Proposals

Some of the most relevant proposals in the area of GFSs might include the following:

The problem of rule explosion for high-dimensional regression problems is tackled in (Alcalá et al., 2011). The proposal is based on the embedded genetic data base learning, allowing the fast learning of simple and accurate linguistic models. The authors propose some mechanisms to ensure a fast, but not premature, convergence in problems with a high number of variables. Also, in order to scale the method for datasets with large numbers of instances, a general mechanism for the estimation of the model error when using evolutionary algorithms, by only considering a reduced subset of the examples, is proposed. This way, a fast post processing stage for further refining of the learned models is applied. The results show the effectiveness of the proposed method in terms of scalability as well as the simplicity of the obtained models.
Alcalá et al. (2011) perform a statistical study on multiobjective genetic fuzzy rule selection and extend it by combining the single granularity-based approach with a lateral tuning of the membership functions.

Another interesting proposal involves the discovery of association rules, which is tackled in (Jesus et al., 2011) by means of evolutionary algorithms.

An evolutionary optimization algorithm is presented in (Ducange et al., 2010). The proposal uses the well-known NSGA-II multi-objective optimization algorithm to generate an approximation of a Pareto front composed of fuzzy classifiers with different trade-offs among sensitivity, specificity, and interpretability. Sensitivity and specificity are evaluated in a ROC plane, where each classifier is described as a point whose coordinates represent the true positive rate and the false positive rate, respectively. After projecting the overall Pareto front onto the ROC plane (Prati et al., 2011), the potentially optimal classifiers are determined by using the ROC convex hull method.

In (Sánchez and Couso, 2010), the authors formulate the learning of a fuzzy rule-based classifier as a problem of statistical inference, proposing the learning of rules by maximizing the likelihood of the classifier. The authors also extend the proposed methodology to interval censored data, which can be roughly described as data that is either known through a pair of bounds, or an upper or lower bound of the true value.

In (Evsukoff et al., 2009), the authors present a method for the generation of fuzzy rule-based systems that performs data modelling according to the symbolic relations expressed by the rules. The proposal focus on the interpretability of the rules and the accuracy of the model. The number of rules of the model is defined using the eigenstructure analysis of the similarity matrix, computed from data. Once the number of rules is defined, a rule induction algorithm based on a clustering algorithm using the number of rules of the model as its number of centres, is applied to the data to define the rules of the model. The proposal produces weighted rules. The model is then simplified by a genetic algorithm that simultaneously selects the most representative subset of variables and number of fuzzy sets in the fuzzy partition of the selected variables.

Another important proposal on GFSs is found in (Ishibuchi and Yamamoto, 2004), in which the authors propose a rule selection process with three objectives: maximization of accuracy, minimization of the rule set, and minimization of the length of the rules. The proposal in (Ishibuchi and Yamamoto, 2004) includes the extraction of a set of candidate rules by evaluation measures and the genetic selection of rules.
6.2.2 Genetic Fuzzy Systems - Applications to Real-World Problems

The following proposals are related to the application of genetic fuzzy systems on real-world problems.

Abadeh et al. (2011) propose the use of GFSs to deal with the intrusion detection problem. The authors propose and compare three types of GFSs based on the Michigan, Pittsburgh and iterative rule learning approaches. In the Michigan approach each chromosome represents a rule, and the rule base is formed by the set of chromosomes. The Pittsburgh approach, instead, codifies a whole rule base in each chromosome, while the iterative approach codifies a single rule in each chromosome and selects a rule at each execution of the GA. Since intrusion detection is a high-dimensional classification problem, the authors propose the separation of the rules in the genetic population according to their class labels, thus, the GFS focus on learning each class independently. The drawback of this proposal is the need to repeat the genetic algorithm for each of the classes in the classification problem. In order to generate the rules to form the genetic search space of the Pittsburgh version, this method uses the training set to define the antecedent part of a relatively small number of fuzzy if-then rules (e.g., 100 rules), and then calculates the compatibility of each antecedent with each of the classes of the problem, based on the proposal in (Ishibuchi et al., 1992). This way, the generated rules have considerable classification power. The same procedure is used to determine the rules of the Michigan version of the proposal at each generation. Results demonstrated the efficiency of all three versions of the proposed approach for intrusion detection. Furthermore, the Pittsburgh version of this proposal obtained better results than the Michigan and iterative versions.

A new method for the synthesis of planar antenna arrays for wireless communication using GFSs is proposed in (Bousahla et al., 2010). The problem is tackled by adjusting the control parameters of a standard GA based on the best individual fitness and population diversity measurements.

Genetic algorithms are used in (Navale and Nelson, 2010) for the proposal of an adaptive approach to control a water valve for a cooling coil. The authors use genetic algorithms to improve the fuzzy rule matrix as well as the fuzzy membership functions of its internal systems. Three different methods are used and experimentally tested in real time, starting with a fuzzy rule matrix based on the human intuition and then adapting these rules in real time, obtaining good results.

A GFS for the genetic fuzzy modelling of user perception of three-dimensional shapes is proposed in (Achiche and Ahmed-Kristensen, 2011). The GFS proposed in this method is based on a previous proposal (Achiche et al.,
which uses a combination of real coded GAs for the fuzzy data base and a binary coded GA for the rule base. The proposal obtained promising results.

6.2.3 Genetic Fuzzy Systems: New Trends

Regarding the new trends in the research of genetic fuzzy systems, Herrera (2008) points out the following directions (discussed next):

- Interpretability quality;
- New data mining tasks: frequent and interesting pattern mining, mining data streams, among others;
- Dealing with high dimensional datasets.

Regarding the interpretability of the genetic fuzzy systems, a possible approach is the inclusion of novel interpretability measures in the fitness function of GFS models, in order to improve the generated fuzzy rule bases.

Data mining includes several tasks which can be explored by genetic fuzzy systems, such as frequent pattern mining open questions, data streams, sequential and time series data, adversary data mining, anomaly detection, non-static, imbalanced data, among others.

The problem of high dimensional datasets is tackled in (Alcalá et al., 2011), as previously stated, by means of a scalable multiobjective genetic fuzzy system.

Recently, several bio-inspired approaches for the generation of fuzzy systems have been proposed in the literature. While little innovation can be found in the recent proposals of GFSs, many novel proposals using bio-inspired approaches have been published lately, using a diversity of hybrid methods. Some of these approaches for the definition of fuzzy systems include:

- Fuzzy neural networks (Aliev et al., 2011; Pousinho et al., 2011; Arotaritei, 2011);
- Species based approaches (Lee et al., 2011);
- Particle swarm optimization (Marinaki et al., 2011; Al-Aawar et al., 2011; Niknam et al., 2011);
- Ant colony based approaches (Ahmadizar and Soltanpanah, 2011; Niknam et al., 2011);
- Artificial immune systems (Prakash and Deshmukh, 2011);
- Clonal selection (Acilar and Arslan, 2011).

Next, we present the DOG-BASED method.
6.3 The DoC-BASED Method

The DoC-BASED method, proposed in (Cintra and Camargo, 2007a) and optimized in (Cintra et al., 2010a), generates fuzzy rule bases using a predefined and fixed fuzzy data base. The GA is applied to select a set of rules to form the fuzzy rule base of a fuzzy classification system. This set of rules is selected from a pool of candidate rules that form the search space of the GA. The process to preselect rules for the genetic search space basically forms all possible rules by combining the labels of the fuzzy sets of each attribute, and then calculates their Degree of Coverage (DoC) with the training examples in order to consider their usefulness for the genetic search space. A minimum threshold for the DoC is set to decide weather to discard a rule or not.

The process of generating the search space used by the DoC-BASED method poses an exponential problem. In order to minimize the complexity of the process, a feature subset selection approach can be used.

Next, we describe the most relevant characteristics of the DoC-BASED method, as well as experiments carried out with it, using the FUZZYDBD method to define the fuzzy data bases (see Section 3.2, page 27) and the FuzzyWrapper method to select attributes (see Section 4.3, page 49).

6.3.1 Description of the DoC-BASED Method

For the automatic generation of FRBs using GAs based on the DoC-BASED method, the search space is formed by a pool of rules that potentially includes all rules with degree of coverage higher than a previously fixed threshold. This way, all rules resulting from the combination of the linguistic terms that define the variables in the rule premise are considered. Since fuzzy systems suffer with the dimensionality curse, as previously stated, as the number of variables increases, the set of possible rules increases exponentially, interfering in the result of the learning process, and, in some situations, making it unfeasible.

The DoC-BASED method (Cintra and Camargo, 2007a,b), described in Algorithm 4, uses a criteria based on heuristic knowledge for the pre-selection of candidate rules to be considered by the GA. This heuristic allows the reduction of the dimension of the search space and the simplification of the codification of the chromosomes, also called individuals or solutions. As previously stated, this heuristic is associated to the DoC of the rules. This way, once the fuzzy partitions of the domains of the attributes are defined, the DoC values of all possible rules are calculated and only the ones with a DoC higher than a predefined threshold are selected to be used by the GA.

Notice that for a dataset described by \( m \) attributes, the maximum number of possible rules in step 3 is defined as the maximum number of possible
Algorithm 4: The DoC-based Method.

| Input: A given dataset |
|------------------------|
| 1 Define the fuzzy data base (see Chapter 3); |
| 2 Define $DoC_{\text{min}}$ as the minimum DoC threshold for rule selection; |
| for $a = 1$ to the maximum number of possible rules do |
| 4 Form $\text{Rule}_a$ by combining the fuzzy sets defining the attributes of the domain; |
| 5 Calculate the DoC value of $\text{Rule}_a$, $DoC_a$; |
| 6 if $(DoC_a \geq DoC_{\text{min}})$ then store $\text{Rule}_a$; |
| end |
| 8 Apply the GA to find a set of rules, from the rules stored previously, defining the final fuzzy rule base |

combinations considering all $m$ attributes. Rules formed by the combination of less than $m$ attributes are not considered by the algorithm in order to reduce the computational cost of the process.

The last step of the algorithm, the use of the GA, involves the following tasks:

- The definition of an initial population of chromosomes;
- An iterative process that executes until a previously defined number of generations is reached;
- The application of genetic operators to evolve the population;
- The evaluation of the generated solutions.

It follows the description of the most relevant processes and definitions used by the DoC-BASED method.

Degree of Coverage Calculation — The heuristic used to define the selection criteria is based on the idea, derived from empirical studies, that the DoC value of the rules can be a good parameter to preselect candidate rules. This way, the process excludes the rules that have a very low chance to be in the final fuzzy rule base from the GA process, namely the rules with null or very low DoC values. This selection step enhances the entire process by avoiding useless search effort in the GA learning process, without any significant loss for the final FRB performance.

The exhaustive calculation of the DoC values for all possible rules also increases exponentially with the increase in the number of features and fuzzy sets. This additional cost in calculation does not harm the benefits of the method for small datasets, since the DoC of the rules is calculated only once before the search process begins. Nevertheless, the process is prohibitive for large datasets and the use of a feature subset selection might be necessary.
Criteria for the Extraction of Candidate Rules — Based on heuristic knowledge, two criteria were experimentally evaluated:

1. Extraction of the rules with DoC higher than the smallest DoC found in the set of rules generated by the Wang & Mendel method;

2. Extraction of all rules with a non-null DoC value.

Experimental results show that the use of the more restrictive criterion, the first one, besides producing genetic search spaces with less rules, also produces fuzzy rule bases with better accuracy. The experiments presented here were conducted using the first criterion.

Chromosome Codification — The DoC-BASED method was implemented and experimentally evaluated using the binary and integer chromosome codifications based on the Pittsburgh approach, in which each chromosome represents a complete rule base. In both codification versions the chromosomes represent and entire rule base. In the binary codification, each gene represents the position of a rule in the list of ordered rules extracted using the DoC. Each chromosome has the same size of the pool of preselected rules. A gene value equals to 1 represents an active rule in the rule base codified by the chromosome (the corresponding rule is actually in the rule base) and 0 represents an inactive rule (the corresponding rule is not in the rule base). For the adoption of this codification, the number of rules in the genetic search space must be reasonably small. Otherwise, long chromosomes are generated, requiring considerable amounts of memory as well as computational effort for the application of genetic operators. The binary codification is shown in Figure 6.2.

Figure 6.2 shows a population of \( n \) chromosomes, each with \( m \) genes, for the representation of the \( m \) rules of the genetic search space. Taking the genes displayed in Figure 6.2, the first chromosome has rules 1, 2, and \( m \) active, and rules 0, 3 and 4 inactive.

For the integer codification, the size of the chromosome is equal to the maximum number of rules expected in the final rule base. Therefore, the size of the chromosomes is not directly related to the size of the list with the preselected rules, i.e., the size of the genetic search space. In fact, each gene simply contains an integer representing the index of a rule in the list of preselected rules. A “−1” value represents the absence of rules in the rule base. Thus, a gene may represent an inactive rule, permitting a reduction in the number of active rules initially defined for the chromosomes. This codification requires less memory and computational effort than the binary codification. However, it has the disadvantage
of requiring the previous definition of the maximum number of rules, in order to define the number of genes of the chromosomes. Figure 6.3 illustrates the search space and population of chromosomes using the integer codification.

In Figure 6.3, the search space is formed by $m$ rules, while the population of $n$ chromosomes contains 9 genes. Thus, each chromosome can contain a maximum of 9 rules. The first chromosome of the population represents a fuzzy rule base containing rules 15, 92, 28, 6, 99, -1, 45, and 38, each.
one described in the genetic search space. Two of the genes of the first chromosome contain no rules (the 5th and 7th genes contain “-1”).

For both codifications, the chromosomes of the initial populations are randomly defined. The existence of conflicting and redundant rules is verified in the initialization of the population and during the use of genetic operators. Notice that since the rules are automatically formed by the combination of the fuzzy set labels of the attributes, the process generates rules with the same antecedent for each class, i.e, conflicting rules. This way, the verification process guarantees that all chromosomes always represent a consistent and valid rule base.

**Fitness Calculation** — For the DoC-BASED method, we proposed an approach that is able to reduce the number of rules in the final FRB, by considering the accuracy and the number of rules of each chromosome in the calculation of the fitness of the chromosomes. This process is used by the FCA-BASED method, but with different adjusts for the parameters. Since the FCA-BASED method is one of the main contributions of this work, and it was able to obtain better results than the DoC-BASED method, our focus has been on the FCA-BASED method. This way, the details for the fitness calculation, including the definition of its parameters are described for the FCA-BASED method in Section 6.5.1.3, page 106.

### 6.3.2 Experimental Evaluation

The optimized version of the DoC-BASED method, which stores the rules that are positively evaluated to form the genetic search space saving memory and processing time, was evaluated and the results published in (Cintra et al., 2010a). The experiments were carried out using 10 datasets from the UCI - Machine Learning Repository (Frank and Asuncion, 2010), and a 10-fold stratified cross-validation strategy.

Table 6.1 presents a summary of the characteristics of the 10 datasets, including the number of examples, features, with the number of continuous and discrete features in brackets, and the number of classes, as well as the majority error.

The experiments carried out using each dataset to generate the fuzzy rule bases involved the following steps:

1. The definition of the fuzzy data bases using the FuzzyDBD method (notice that the FuzzyDBD-II method was proposed after this evaluation, thus, it was not considered in these experiments);
Table 6.1: Characteristics of the datasets - DoC-BASED.

| Dataset   | Examples | Features | Classes | ME   |
|-----------|----------|----------|---------|------|
| Breast    | 683      | 9        | 2       | 34.60|
| Credit    | 653      | 15       | 2       | 45.33|
| Glass     | 220      | 9        | 7       | 64.46|
| Heart     | 270      | 13       | 2       | 44.44|
| Hepatitis | 130      | 16       | 2       | 79.36|
| Ionosphere| 351      | 34       | 2       | 35.90|
| Segmentation | 210     | 19       | 7       | 85.72|
| Sonar     | 208      | 60       | 2       | 46.63|
| Vehicle   | 846      | 18       | 4       | 74.23|
| Wine      | 176      | 13       | 3       | 59.74|

2. The selection of attributes using:
   
   (a) The CFS filter available in the Weka framework (Witten et al., 2011), using the whole data set (notice that this step is independent from the previous one, since it does not involve the definition of the fuzzy data base);
   
   (b) FuzzyWrapper, using 10-fold stratified cross-validation (notice that this step depends on the definition of the fuzzy data base).

3. The generation of the fuzzy rule bases using the Wang & Mendel method with the attributes selected by CFS as well as with the attributes selected by the FuzzyWrapper method;

4. The calculation of the DoC of the rules generated by the Wang & Mendel method in the previous step in order to define $\text{DoC}_{\text{min}}$, the minimum DoC threshold for the rule extraction process (see step 2 of Algorithm 4, page 90);

5. The extraction and storage of rules with $\text{DoC} \geq \text{DoC}_{\text{min}}$. Notice that $\text{DoC}_{\text{min}}$ is defined in step 4 (see steps 3 to 6 of Algorithm 4);

6. The generation of the fuzzy rule bases using the DoC-BASED method, considering the attributes selected by:
   
   (a) The CFS filter;
   
   (b) The FuzzyWrapper method.

7. The analysis of the results.

As previously stated, the FuzzyDBD method was used to define the number of fuzzy sets describing each attribute of the datasets used in the experiments, as well as their distribution. Table 6.2 presents the original number of attributes (Total), the number of classes (Classes), and the number of fuzzy sets (Fuzzy sets) used to describe each dataset attribute.

As previously mentioned, the FuzzyWrapper method and the CFS filter were adopted to evaluate the results of the DoC-BASED method. Other feature
subset selection methods were not included since the focus of the experiments is no the evaluation of feature subset selection methods. Nevertheless, four different feature subset selection methods were tested in order to select the most suitable one: C4.5, CFS, Consistency, and ReliefF. Table 6.3 presents the original number of attributes (Total) for each dataset and the number of attributes selected by the FUZZYWRAPPER, CFS, C4.5, and Consistency methods. CFS was chosen for our experiments since it was able to select the most reasonable and feasible subset of features for all datasets, i.e., the selected feature subset selection method was the one that was able to select suitable numbers of features for all datasets. For instance, that ReliefF selected 33 out of the 34 original attributes, making the use of the DOC-BASED method impracticable. Regarding the C4.5 and Consistency methods, they performed reasonably well. However, they selected 18 features for the Vehicle dataset, whereas CFS selected only 11 features and FUZZYWRAPPER selected 10.

Table 6.3: Number of selected attributes - DOC-BASED.

| Dataset   | Total | FUZZYWRAPPER | CFS | C4.5 | ReliefF | Consistency |
|-----------|-------|--------------|-----|------|---------|-------------|
| Breast    | 10    | 6            | 7   | 7    | 10      | 7           |
| Credit    | 15    | 12           | 7   | 9    | 12      | 13          |
| Glass     | 9     | 6            | 7   | 9    | 8       | 7           |
| Heart     | 13    | 10           | 8   | 12   | 12      | 10          |
| Hepatitis | 16    | 6            | 6   | 8    | 18      | 12          |
| Ionosphere| 34    | 7            | 14  | 14   | 33      | 7           |
| Segmentation| 19  | 8            | 7   | 12   | 15      | 9           |
| Sonar     | 60    | 6            | 14  | 12   | 14      | 58          |
| Vehicle   | 18    | 10           | 11  | 18   | 18      | 18          |
| Wine      | 13    | 7            | 10  | 3    | 13      | 5           |

Table 6.4 presents the total number of possible rules according to the databases generated using the attributes selected by the FUZZYWRAPPER (FW-Total) and CFS (CFS - Total) methods, as well as the number of extracted rules for the FUZZYWRAPPER (FW-extracted) and CFS (CFS-extracted) methods. For each dataset, the rules extracted were the ones with a DoC higher than the smallest one found in the rules generated by the Wang & Mendel method for all datasets, except for the Breast dataset, due to the fact that the number of possible rules was quite small. For this specific dataset, all possible rules were used for the attributes selected by the FUZZYWRAPPER method, and all
rules with a non null DoC were used for the attributes selected by the CFS filter.

Table 6.4: Total number of possible rules and number of extracted rules - DoC-BASED.

| Dataset  | FW-Total | FW-extracted | CFS-Total | CFS-extracted |
|----------|-----------|--------------|-----------|---------------|
| Breast   | 128       | 128          | 256       | 182           |
| Credit   | 8,192     | 752          | 256       | 141           |
| Glass    | 823,543   | 663          | 5,764,801 | 602           |
| Heart    | 2,048     | 461          | 512       | 223           |
| Hepatitis| 1,458     | 140          | 1,458     | 110           |
| Ionosphere| 4,374    | 1,468        | 9,565,938 | 852           |
| Segmentation| 45,927  | 2,420        | 15,309    | 602           |
| Sonar    | 1,458     | 402          | 9,565,938 | 105,633       |
| Vehicle  | 4,096     | 3,647        | 8,192     | 6,437         |
| Wine     | 6,561     | 679          | 177,147   | 6,465         |

It is important to notice that the number of possible rules for the Glass, Ionosphere, and Sonar datasets using the attributes selected by CFS is quite large and requires considerable computational effort.

Table 6.5 shows the average number of rules present in the fuzzy rule bases generated by the DoC-BASED method using the sets of selected attributes (the numbers of selected attributes are present in columns FUZZYWRAPPER and CFS), as well as the corresponding standard deviation in brackets. The best results, i.e., the smallest number of rules, are dark-gray shaded.

Table 6.5: Number of generated rules - DoC-BASED.

| Datasets  | FUZZYWRAPPER Rules - FUZZYWRAPPER | CFS Rules - CFS |
|-----------|----------------------------------|-----------------|
| Breast    | 6 | 9.43 (1.73) | 7 | 30.73 (13.78) |
| Credit    | 12 | 14.14 (3.28) | 7 | 4.81 (0.90) |
| Glass     | 6 | 30.05 (12.40) | 7 | 30.66 (8.75) |
| Heart     | 10 | 26.57 (5.20) | 8 | 17.88 (2.48) |
| Hepatitis | 6 | 6.94 (1.21) | 6 | 4.23 (1.48) |
| Ionosphere| 7 | 14.48 (6.65) | 14 | 156.23 (11.92) |
| Segmentation| 8 | 42.12 (9.13) | 7 | 29.28 (8.87) |
| Sonar     | 6 | 10.52 (2.06) | 14 | 24.85 (5.59) |
| Vehicle   | 10 | 12.53 (3.32) | 11 | 9.95 (3.54) |
| Wine      | 7 | 20.98 (8.13) | 10 | 27.34 (6.91) |

Notice that, as expected, the fuzzy rule bases generated using less attributes present less rules. Since only the number of rules in the generated fuzzy rule bases do not allow a thorough comparison of the results, Table 6.6 presents the error rates obtained by the fuzzy rule bases generated by the DoC-BASED method using the attributes selected by FUZZYWRAPPER and CFS, as well as the corresponding standard deviation in brackets. The dark-gray shaded cells highlight the smallest error rates.

The databases generated with the attributes selected by the FUZZYWRAPPER method produced the smallest error rates for 8 datasets, and show comparable results for the remaining 2 datasets.

The Wilcoxon signed-rank test (Demšar, 2006) was used to compare the results in terms of error rates, showing that, with a 95% confidence, the
Table 6.6: Error rates - DoC-BASED.

| Datasets   | FUZZYWRAPPER | CFS     |
|------------|--------------|---------|
| Breast     | 5.00 (5.00)  | 36.60 (16.00) |
| Credit     | 15.50 (9.00) | 15.80 (9.00)   |
| Glass      | 37.70 (16.0) | 47.10 (11.00)  |
| Heart      | 22.60 (6.00) | 23.70 (7.00)   |
| Hepatitis  | 29.80 (13.00)| 33.60 (13.00)  |
| Ionosphere | 37.40 (2.00) | 35.80 (2.00)   |
| Segmentation | 22.40 (7.00)  | 27.60 (5.00)   |
| Sonar      | 37.00 (13.00)| 32.40 (20.00)  |
| Vehicle    | 50.50 (5.00) | 59.20 (6.00)   |
| Wine       | 6.80 (6.00)  | 12.10 (8.00)   |

DoC-BASED method obtained better results with the attributes selected by the FUZZYWRAPPER method, when compared to the CFS filter. However, notice that the error rates using FUZZYWRAPPER were higher than the majority error for the Ionosphere dataset, while the error rates using CFS were higher than the majority error for Breast and Ionosphere.

In conclusion, the DoC-BASED method is able to automatically generate fuzzy rule bases from data. The DoC-BASED method includes a step for the definition of the genetic search space that extracts rules, selecting the relevant ones according to their degree of compatibility with the set of examples. This pre-selection of rules greatly reduces the search space, simplifying the codification of the chromosomes, as well as reducing the computational cost of the genetic search. Nevertheless, the generation of the genetic search space poses an exponential problem. In this sense, a feature subset selection method can reduce the cost of the genetic search space generation. In this work, two different FSS methods, CFS and FUZZYWRAPPER, were used, showing better results with the FUZZYWRAPPER method. One of the reasons for the good performance of the method using the attributes selected by the FUZZYWRAPPER method, when compared to using the ones selected by CFS, is the fact that FUZZYWRAPPER uses the same fuzzy data base used by the DoC-BASED method to generate the fuzzy rule bases, thus, taking into account the fuzzy definition of the attributes.

In order to tackle the exponential complexity of the genetic search space generation process used by the DoC-BASED method, we proposed the FCA-BASED method, which uses the theory of Formal Concept Analysis (FCA) to form the genetic search space. It follows a description of the basic concepts of FCA used in this work, as well as a full description of the FCA-BASED method.

### 6.4 Formal Concept Analysis

FCA (Wille, 1982), is a mathematical technique for extracting concepts and structures from data. It was introduced in the 1980s and is becoming increasingly popular. The basic data structure in FCA is the formal context, which is
normally represented in a table form where the columns represent the attributes and the rows represent the objects (see Table 6.8). In other words, a formal context is a representation of the relation between objects and their attributes. Notice that attributes in a formal context must be binary. The table representing the formal context contains 1 (true) in cell \((i, j)\) if object \(i\) has attribute \(j\), and 0 (false) otherwise. It is possible to generate a conceptual lattice from the formal context. A conceptual lattice is basically a graph whose vertices correspond to the formal concepts represented by sets of examples or attributes.

Formally, a context is a triple \(k = (G, M, I)\), where \(G\) is a set of objects, \(M\) is a set of attributes, and \(I\) is a binary relation \(I \subseteq G \times M\). Given a set of objects \(A \subseteq G\), the shared image of \(A\) in \(M\) is defined as:

\[
A^\uparrow := \{m \in M|(g, m) \in I \forall g \in A\}
\]  

(6.1)

While the shared image of \(B\) in \(G\), given a set of attributes \(B \subseteq M\), is:

\[
B^\downarrow := \{g \in G|(g, m) \in I \forall m \in B\}
\]  

(6.2)

The pair \((A, B)\) constitutes a formal concept of \((G, M, I)\) if and only if \(A \subseteq G, B \subseteq M\) and \(A = B^\downarrow, B = A^\uparrow\). \(A\) is called the extent of the concept and \(B\) is called the intent of the concept (Wille, 1982). In other words, Equation 6.1 defines the collection of all attributes shared by all objects in \(A\), while Equation 6.2 defines the collection of all objects sharing all the attributes in \(B\).

In traditional FCA, the relation is binary, although multi-valued contexts are much more common than binary-valued ones. For attributes that can take a range of values, the idea of "conceptual scaling", which transforms a many-valued attribute (e.g. a number) into a symbolic attribute, can be used. For example, the attribute "height in centimetres", whose domain ranges from 0 and 200, can generate the following binary attributes "height-less-than-50", "height-from-50-to-100", and "height-more-than-100". Since these derived attributes have true/false values, they can be treated within the FCA framework.

Using the formal context, it is possible to generate a conceptual lattice that presents the information in a nice visual way. It follows a toy example based on the attribute \(\times\) value table shown in Table 6.7, which shows the name, Age, Sex, and Hair Colour of six people (Cintra et al., 2011a).

The first step to define the formal context is the creation of binary attributes to represent the Age, Sex, and Hair Colour attributes of Table 6.7, as shown in Table 6.8.

In Table 6.8, the continuous attribute age was discretized, generating three
Table 6.7: Attribute × value table - toy example.

| Name | Age | Sex | Hair Colour |
|------|-----|-----|-------------|
| Andy | 48  | M   | Black       |
| Lina | 29  | F   | Black       |
| Mark | 23  | M   | Brown       |
| Martina | 46 | F   | Blond       |
| Mike | 18  | M   | Brown       |
| Suzy | 17  | F   | Blond       |

Table 6.8: Formal Context based on Table 6.7.

| Name | Age ≤20 | Age >20 | Age >30 | Sex M | Sex F | Hair Colour Blond | Hair Colour Brown | Hair Colour Black |
|------|---------|---------|---------|-------|-------|------------------|------------------|------------------|
| Andy | 0       | 0       | 1       | 1     | 0     | 0                | 0                | 1                |
| Lina | 0       | 1       | 0       | 0     | 1     | 0                | 0                | 1                |
| Mark | 0       | 1       | 0       | 1     | 0     | 1                | 0                | 1                |
| Martina | 0 | 0 | 1 | 0 | 1 | 1 | 0 | |
| Mike | 1       | 0       | 0       | 0     | 1     | 0                | 0                | 0                |
| Suzy | 1       | 0       | 0       | 0     | 1     | 0                | 1                | 0                |

binary attributes. The multi-valued attributes sex and hair colour generated a binary attribute for each of its values. Notice that FCA works with datasets that have a class attribute and with datasets that do not have a class, as in the case of this toy example. Figure 6.4 shows the generated conceptual lattice based on the formal context shown in Table 6.8.

![Conceptual lattice](image)

Figure 6.4: Conceptual lattice.

In the lattice structure, formal concepts are represented by the nodes. Attributes are noted slightly above the nodes while objects are noted slightly under the nodes. The positioning of the nodes can be arranged in a variety of ways. In the lattice illustrated in Figure 6.4, the nodes were arranged to minimize the number of intersections, thus, attributes are not displayed in the order they are shown in the formal context. To retrieve extensions, one must simply trace all paths leading down from the node of the attribute of interest. Similarly, it is possible to retrieve intentions by tracing all paths leading up from the node representing the object of interest. For instance, the intention of the formal concept represented by the node named *Mark* is the
set of attributes \(\{> 20 \& \leq 30, M, Brown\}\). The extension of the formal concept represented by the node named *Blond* is the set of objects \(\{\text{Martina, Suzy}\}\).

As previously stated, the transformation of continuous attributes into binary ones is commonly called *scaling*. The fuzzy theory has also contributed to this task. An introduction to the use of the fuzzy theory for the task of FCA scaling is presented in (Wolff, 2002). Some of the advantages of using the fuzzy theory for FCA scaling include:

- The partitioning process uses linguistic values, easily interpretable by humans, such as *young, old, tall, short*, conveying this important interpretability feature;
- Unnatural divisions are easily avoided by the fuzzy logic with the use of the membership degrees that allow one object to belong to different categories with different degrees of membership;
- Fuzzy scaling is a natural choice when the interest in FCA is for the extraction of fuzzy rules, since these rules reflect the same FDB used by the inference mechanism of the induced classifier.

As long as the formal context is defined, the formal concepts can be extracted automatically. Some of the algorithms available in the literature for the extraction of formal concepts include the NextClosure algorithm, which works by finding neighbouring concepts (Ganter, 2002), and the algorithm proposed in (Krajca et al., 2010), which features a parallel search process. Since the final set of formal concepts extracted from a formal context is independent from the extraction algorithm, any algorithm will produce the same set of formal concepts.

One of the applications of formal concept analysis is related to the extraction of association rules, as well as classification rules. For the extraction of classification rules from a formal context, the basic idea is to extract formal concepts and use their intentions to form association rules. In the sequel, the process selects the association rules with a class to form the set of classification rules.

In this work, we propose an approach based on the use of FCA to extract fuzzy classification rules from data in order to form the search space of a genetic rule generation process. The first step of our proposal consists in transforming an attribute \(\times\) value table, such as the one in Table 6.7, into a formal context, such as the one in Table 6.8. Once the formal context is generated, the formal concepts are extracted and the classification rules are then generated from this set of formal concepts.

To illustrate the process of extracting classification rules from formal concepts, let us consider a formal context representing a dataset with three binary
attributes, $At_1$, $At_2$, and $At_3$, two classes, $Class_A$ and $Class_B$, and $n$ examples (called objects in the FCA context), $Obj_1$ to $Obj_n$. Let us assume that the following formal concepts were extracted from this formal context:

1. $\{At_1, At_2, Class_A\}, \{Obj_1, Obj_2\}$; 
2. $\{At_2, At_3, Class_B\}, \{Obj_3, Obj_4\}$.

By discarding the extensions (sets of objects) of these formal concepts, we obtain the following item-sets:

1. $\{At_1, At_2, Class_A\}$; 
2. $\{At_2, At_3, Class_B\}$.

Since these item sets contain a class, they can generate the following classification rules:

1. IF $At_1$ is true AND $At_2$ is true THEN $Class$ is $A$; 
2. IF $At_2$ is true AND $At_3$ is true THEN $Class$ is $B$.

Algorithm 5 describes the proposed approach of extracting classification rules from data using a formal context for a dataset with $m$ attributes and $n$ examples.

Notice that our proposal contains a fuzzy scaling process. To guide the scaling of real values, a predefined threshold $A_{min}$ for the membership degrees is used. Whenever the membership degree of a certain input value for a fuzzy set is equal or higher than $A_{min}$, the corresponding attribute is set to true in the formal context, otherwise, the attribute is set to false. As previously stated, since the extraction of the formal concepts from a formal context is independent from the extraction algorithm, i.e., different algorithms will extract exactly the same sets of formal concepts for a given formal context, step 32 can be performed by any extraction algorithm. In our experiments, we used the NextClosure algorithm (Ganter, 2002).

An important issue regarding the extraction of the formal concepts when using the binary fuzzyfication of the formal context is related to the increase in the number of attributes. The total number of attributes after the scaling of continuous and multi-valued attributes is equal to the sum of all fuzzy sets describing each original attribute, plus all the values of each discrete attribute, and the number of classes. This way, an increase in the number of fuzzy sets defining the attributes leads to an exponential increase in the number of possible formal concepts. The total number of formal concepts in a formal context can be estimated using the Metropolis-Hastings algorithm for sampling formal
Algorithm 5: Extraction of Fuzzy Classification Rules using FCA.

**Input:** A given dataset described by \( m \) attributes and \( n \) examples

1. Define the FDB, i.e., the definition of the continuous attributes in terms of fuzzy sets;
2. for \( a = 1 \) to \( m \) do
   3. if Attribute \( \text{Att}_a \) is continuous then create a binary attribute for each fuzzy set defining \( \text{Att}_a \);
   4. else create a binary attribute for each discrete value of \( \text{Att}_a \);
3. end
4. Define \( A_{min} \) as the minimum membership degree to guide the fuzzy scaling process;
5. for \( e = 1 \) to \( n \) do
   6. for \( a = 1 \) to \( m \) do
      7. if \( \text{Att}_a \) is a continuous attribute then
         8. for \( x = 1 \) to total number of binary attributes generated from attribute \( \text{Att}_a \) do
            9. calculate \( A_{\text{Att}_a x} \), as the membership degree of the input value of attribute \( \text{Att}_a \), example \( e \), in the fuzzy set defining the binary attribute \( x \);
            10. if \( A_{\text{Att}_a x} \geq A_{min} \) then
                11. set binary attribute \( \text{Att}_a x \) as true;
                12. end
            13. else
                14. set binary attribute \( \text{Att}_a x \) as false;
                15. end
         16. end
      7. else \( \text{Att}_a \) is a discrete attribute
         8. for \( x = 1 \) to total number of label values of attribute \( \text{Att}_a \) do
            9. if \( x \) = the input value of attribute \( \text{Att}_a \), example \( e \) then
               10. set binary attribute \( \text{Att}_a x \) as true;
               11. end
            12. else
               13. set binary attribute \( \text{Att}_a x \) as false;
               14. end
         15. end
   9. end
5. end
6. Extract all formal concepts from the formal context;
7. Discard the extension of the extracted formal concepts;
8. Store the formal concepts with a class, i.e., the ones that originate classification rules.

concepts, described in (Boley et al., 2010). Nevertheless, it is important to notice that for our purpose, the total number of formal concepts is much larger than the number of formal concepts we are interested in extracting; since we want to extract classification rules, we are only interested in extracting formal concepts that have a class in their intention.

In order to reduce the number of possible formal concepts, the fuzzy sets defining each continuous attribute can be evenly distributed in the partition, so the maximum possible membership degree in the intersections is 0.5. This way, if \( A_{min} \) is set to 0.5, for each original fuzzyfied attribute, only one of its binary attributes will be activated. Notice that ties must also be handled: in our implementation we used a random variable to select one of two tied fuzzy sets.

Another possibility for the reduction of the computational cost of extracting
all formal concepts, as well as for the reduction of the resulting number of formal concepts in a formal context, is to reduce the number of examples in the training set. A simple alternative for this reduction in the number of examples is to remove similar examples considering the fuzzy definition of the attributes. The idea here is to separate the examples into sets with similar membership degrees for all the values of the attributes of the example, and then eliminating the most similar ones. For instance, consider the following two examples after the fuzzyfication step:

1. **Attribute 1** is **Low**, **Attribute 2** is **Low**, **Class** is **A**;

2. **Attribute 1** is **Low**, **Attribute 2** is **Medium**, **Class** is **A**;

These two examples, after the fuzzyfication step, differ only in the second attribute (**Low** for the first example and **Medium** for the second one). Moreover, these fuzzy sets defining both examples are adjacent (**Low** and **Medium**). Suppose that the membership degrees of example 1 is 0.52 for **Low** and 0.48 for **Medium**, and for example 2, 0.30 for **Low** and 0.70 for **Medium**. Following this strategy, in this case, the first example could be considered to be discarded since its second attribute covers fuzzy sets **Low** and **Medium** with similar membership degrees. We intend to investigate this strategy in future work.

### 6.5 The FCA-BASED Method

The FCA-BASED method uses the FCA theory to extract the rules forming the genetic search space. Once the genetic search space is formed, FCA-BASED uses the powerful search of genetic algorithms to generate fuzzy rule bases. As previously explained, the main motivation for the proposal of the FCA-BASED method was to tackle the exponential complexity involved in generating the genetic search space by the DoC-BASED method, while still using a codification of the chromosomes that does not require modifications in the generated rules. Notice that the rules of the genetic search space are predefined and fixed, *i.e.*, the genetic process do not alter the existing rules, it simply selects rules from the predefined set of rules forming the genetic search space. This way, the genetic search can concentrate its efforts to find suitable combinations of rules without generating new rules, which could insert useless and even redundant rules in the genetic search space.

Experimental results show that FCA-BASED is able to extract a suitable number of classification rules with the following desirable characteristics:

- The rules are extracted directly from data, avoiding the random extraction of a potentially large number of unnecessary/useless rules;
• Unlike the exhaustive generation of rules, which has exponential complexity, by using FCA the complexity of the process is typically polynomial (Lindig, 1995);

• The rules extracted using the proposed approach present variable number of conditions, improving the interpretability of the generated FRBs.

The FCA-BASED method was experimentally evaluated and compared to the DOC-BASED method, as well as to a baseline genetic fuzzy algorithm, namely SLAVE, describe in 7.3.1.3, page 127.

Since the DOC-BASED method differs from the FCA-BASED method only in the strategy to form the GA search space, the DOC-BASED method is a natural choice for comparisons and evaluation of FCA-BASED. The comparisons of both methods took into account the accuracy and syntactic complexity of the generated models, and the cost of generating the search space of the GA, as well. As done previously, in this work the syntactic complexity of a rule base is calculated as the number of conditions in the rules of the rule base. The FUZZYDBD method which adopts the equalized universe method for the distribution of the fuzzy sets in the partitions of the attributes was used to predefine the FDBs.

The FCA-BASED method is detailed next.

6.5.1 Description of the FCA-BASED Method

The FCA-BASED method fits the same category of the DOC-BASED method, “the genetic rule selection with a priori rule extraction”, in the classification shown in Figure 6.1-b, page 82.

Algorithm 6 presents the FCA-BASED method, detailed in the sequel.

As previously stated, notice that the extraction of the classification rules that form the genetic search space is independent from the extraction algorithm itself.

6.5.1.1 Chromosome Codification

The FCA-BASED method uses an integer chromosome codification, based on the Pittsburgh approach, i.e., each chromosome represents a rule base. The size of the chromosome, i.e., its number of genes, is defined as the number of rules found in the rule base produced by the Wang & Mendel method applied to a given dataset. By using this reference value for the size of the chromosomes, the FCA-BASED method is able to work with reasonably small chromosomes. An alternative to the use of the number of rules in the rule base generated by Wang & Mendel as the number of genes of the chromosomes include the definition of large chromosomes i.e., each chromosome would be
Algorithm 6: The FCA-BASED Method.

input: A given dataset
1 Define the fuzzy data base (see Section 3.2);
2 Generate and extract the classification rules from the formal context (see Algorithm 5);
3 Using the WM method, define the values for the initial number of rules (Best_NR) and correct classification rate (Best_CCR) to be used by the GA;
4 Create the initial population of the GA formed by k chromosomes, each with Best_NR genes;
5 for a = 1 to the maximum number of iterations do
6   for c = 1 to k do
7     Calculate $f_c$ as the fitness value of chromosome c;
8   end
9   for c = 1 to k do
10      if $f_c > \text{Best}\_\text{CCR}$ then
11         Best_CCR = $f_c$;
12         Best_NR = number of rules in chromosome c;
13      end
14      if $f_c = \text{Best}\_\text{CCR}$ then
15         if number of rules in chromosome c < Best_NR then
16            Best_NR = number of rules in chromosome c;
17         end
18      end
19   end
20   for c = 1 to k do
21      if $f_c < \text{Best}\_\text{CCR}$ then
22         Apply a penalization to $f_c$ according to Table 6.9;
23      end
24   end
25   Apply the elitism, mutation, and crossover genetic operators;
26 end
27 Perform a rule pruning process (see Algorithm 8, page 109).

formed by a large set of genes (rules). Nevertheless, this alternative tends to generate large rule bases. This way, a post process of rule selection to remove useless rules might be necessary. Our heuristic defines that the number of rules in the rule bases produced by the Wang & Mendel method is the largest number of rules considered acceptable for any rule base. In fact, the Wang & Mendel method tends to generate a large number of rules, since its first step creates a rule for each example of the rule set.

In the integer codification, each gene contains an index to a rule in the search space. To allow the generation of rule bases with less rules than the number of rules in the Wang & Mendel rule base, a “−1” value is used to indicate that a gene represents an inactive rule — See Figure 6.3, page 92.

The chromosomes of the initial populations are randomly generated. The existence of conflicting or redundant rules is verified in the initialization of the process as well as when genetic operators are applied. Although this verification process increases the computational cost of the method, it is necessary to guarantee that each chromosome always represents a consistent rule base.
6.5.1.2 Chromosome Initialization

Our heuristic to define the maximum number of rules in each chromosome is based on the fact that we believe the rule bases generated by Wang & Mendel can be reduced, while keeping or improving the original accuracy of the rule base. This way, we adopted a strategy to induce the reduction of the number of active rules in the chromosomes based on the initialization of chromosomes with different percentages of active rules. 25% of the chromosomes is initialized with all rules activated, i.e., each gene contains the index for a valid rule in the search space, 25% of the chromosomes are initialized with 80% of active rules, 25% with 60% of active rules, and the remaining 25% with 40% of active rules. By initializing 75% of the population with less rules than the number defined by the Wang & Mendel method, we allow the generation of reduced rule bases by means of the genetic operators. The percentages of active rules in the chromosomes were defined empirically.

6.5.1.3 Fitness Calculation

The fitness calculation process adopted by FCA-BASED aims at reducing the number of rules of the chromosomes during the search process while maintaining the best classification rates found. This way, the fitness value is calculated using the Correct Classification Rate (CCR) and the Number of Rules (NR) in the FRB represented by each chromosome. This evaluation process uses a self-adaptive algorithm that updates referential values of ideal CCR (Best_CCR) and NR (Best_NR). The initial values of Best_CCR and Best_NR are set with the corresponding values found in the FRB generated by the WM method for the same dataset before the use of the GA, and using the same FDB. These values are updated after each generation if a better CCR is obtained by a chromosome in the population. The NR is used in a penalization mechanism that decreases the fitness value of a chromosome when its NR is larger than the current reference number (Best_NR). The general evaluation process applied in each generation is described in the following steps:

1. Calculate the CCR for each of the chromosomes and set it as their initial fitness value;
2. If a CCR > Best_CCR is obtained, update the reference value for Best_CCR using this new best CCR;
3. Whenever Best_CCR is updated, update the reference value for Best_NR using the NR of the best chromosome;
4. Calculate the final fitness value of each chromosome applying the penalization mechanism.
Notice that this mechanism does not guarantee that the \textit{Best}_{NR} found in any generation of the GA is always \( \geq \) the \textit{Best}_{NR} of the previous generations. In fact, this mechanism was defined focusing on the improvement of the accuracy.

The fitness calculation process of the DoC-BASED method only updated \textit{Best\_CCR} and \textit{Best\_NR} when a chromosome with better accuracy and same or smaller number of rules was found, or when a chromosome with the same accuracy and less rules was generated. Empirical tests showed that this strategy produced lower accuracy rates for the FCA-BASED method. One of the reasons for this is related to the coverage power of the rules used by both methods. The rules extracted by the DoC-BASED method are more generalized, including all attributes in their antecedents, while the rules extracted by FCA-BASED have, on average, less attributes in their antecedents, thus, they are more specialized.

Table 6.9 presents the penalization rates applied in the calculation of the chromosomes fitness. These values were defined empirically.

| Number of rules | Penalization rate |
|-----------------|-------------------|
| \( \leq \text{Best\_NR} \) | no penalization |
| \( \leq (\text{Best\_NR} \times 1.5) \) | 10% |
| \( \leq (\text{Best\_NR} \times 2) \) | 30% |
| \( \leq (\text{Best\_NR} \times 3) \) | 40% |
| \( > (\text{Best\_NR} \times 3) \) | 50% |

6.5.1.4 Population Diversity Maintenance

Aiming at avoiding a premature convergence of the GA, FCA-BASED uses a Self-Adaptive Genetic Algorithm (SAGA) (Castro and Camargo, 2004) that dynamically adjusts the crossover and mutation rates at each new generation. The SAGA algorithm keeps the Genetic Diversity (GD) at reasonable levels.

In this work, the GD of the population is defined as the average fitness of the population divided by the maximum fitness value found in each generation, as shown in Equation 6.3.

\[
GD = \frac{\text{Average\_Fitness}}{\text{Maximum\_Fitness}} \tag{6.3}
\]

Thus, the GD represents a value in the interval \([0,1]\).

When the GD gets close to 1, we have an indication that the individuals of the population are becoming quite similar. This way, in order to avoid a premature convergence, the crossover rate must be reduced while the mutation rate must be increased. Similarly, when the GD gets close to 0, the individuals of the population are quite different, and the search process can become random, with no direction. Thus, it is necessary to reduce the GD.
of the population by increasing the crossover rate and reducing the mutation rate.

By keeping the genetic diversity at reasonable levels, it is possible to better explore the fittest individuals of each generation. The dynamic adjust process of the crossover and mutation rates is detailed in Algorithm 7.

**Algorithm 7:** Algorithm for the adjustment of the crossover and mutation rates.

1. if $GD \geq V_{\text{max}}$ then
2.   $M_{\text{Rate}} = \min(M_{\text{Rate}} \times v, M_{\text{upp}})$;
3.   $C_{\text{Rate}} = \max(C_{\text{Rate}}/v, C_{\text{low}})$;
4. end
5. else
6.   if $GD \leq V_{\text{min}}$ then
7.     $M_{\text{Rate}} = \max(M_{\text{Rate}}/v, M_{\text{low}})$;
8.     $C_{\text{Rate}} = \min(C_{\text{Rate}} \times v, C_{\text{upp}})$.
9. end
10. end

The auxiliary constant $v$ is responsible for the modification rate of the mutation and crossover operators. Table 6.10 shows the defined values for the parameters in order to control the genetic diversity of the GA population.

| Variable      | Value  |
|---------------|--------|
| Crossover rate| 70%    |
| Mutation rate | 5%     |
| Maximum GD    | 0.8    |
| Minimum GD    | 0.3    |
| $v$           | 1.3    |
| Maximum Crossover | 90% |
| Minimum Crossover | 40% |
| Maximum mutation | 25%  |
| Minimum mutation | 0.05% |

6.5.1.5 Selection of rules

In order to improve the interpretability of the final rule bases it generates, FCA-BASED performs a simple post selection process. This rule selection process, described in Algorithm 8, checks the ability of each individual rule to improve the classification power of the rule set.
Algorithm 8: Post pruning of rules for the FCA-BASED method.

1. Order the rule set by class, from the rule with largest number of conditions to the one with the least number of conditions;
2. Calculate the error rate of the rule set with all rules ($ER_1$);
3. for class 1 to class n do
   4. for i = first rule to the last rule with the current class do
      5. Remove rule i from the rule set;
      6. Calculate the new error rate for the whole rule set ($ER_2$);
      7. if $ER_2 \leq ER_1$ then
         8. Remove permanently the current rule from the rule set;
      else
         9. Reinsert the rule into the rule set;
   end
end

The process of rule selection first orders, for each class, the rules in the rule set according to their number of conditions. Afterwards, if a rule which is more specific than another rule is found, the more specific rule is removed from the rule set if the error rate of the classifier without this rule does not increase. This process is repeated until all rules are tested. The heuristic of this algorithm is based on the fact that the more specialized (specific) a rule is, the less impact it has on the whole set of examples. Notice that this heuristic is only valid for the classic fuzzy reasoning method (best rule). The procedure aims at removing as many rules as possible keeping (or improving) the accuracy of the whole rule base.

6.5.2 Experimental Evaluation - FCA-BASED Method

The FCA-BASED method was tested using 10 datasets from UCI - Machine Learning Repository (Frank and Asuncion, 2010) with a 10-fold cross-validation strategy (Cintra et al., 2012c). Table 6.11 summarizes the dataset characteristics, presenting the total number of examples, total number of features, including the number of continuous and discrete features (in brackets), the number of classes, and the majority error (ME).

Table 6.11: Characteristics of the datasets - FCA-BASED.

| Dataset   | Examples | Features | Classes | ME    |
|-----------|----------|----------|---------|-------|
| Credit    | 653      | 15 (6, 9)| 2       | 45.33 |
| Cylinder  | 277      | 32 (19, 13) | 2     | 35.74 |
| Diabetes  | 769      | 8 (8, 0) | 2       | 34.90 |
| Glass     | 220      | 9 (9, 0) | 7       | 65.46 |
| Heart     | 270      | 13 (13, 0)| 2     | 44.44 |
| Ionosphere| 351      | 34 (34, 0)| 2     | 35.90 |
| Iris      | 150      | 4 (4, 0) | 3       | 66.66 |
| Segmentation | 210    | 19 (19, 0)| 7     | 85.71 |
| Vehicle   | 846      | 18 (18, 0)| 4     | 74.23 |
| Wine      | 178      | 13 (13, 0)| 3     | 39.74 |

Notice that two of the datasets detailed in Table 6.1, page 94, are not used.
in the evaluation of the FCA-BASED method: the Hepatitis dataset was discarded since most of its 16 attributes are discrete, thus, these discrete attributes cannot be defined in terms of fuzzy sets, while the Sonar dataset was replaced due to the fact that it presents a number of attributes much higher than the average of the other datasets.

As previously stated, the FUZZYDBD algorithm was used to define the FDBs using information gain, ReliefF and Wang & Mendel as estimation functions. For SLAVE, we used the implementation available in the KEEL framework (Bull et al., 2008). Since this implementation of SLAVE does not allow the definition of different numbers of fuzzy sets for the attributes of a dataset, we defined the partitions used by SLAVE with 3 fuzzy sets and also tested the FCA-BASED method using 3 fixed fuzzy sets evenly distributed in the partitions to define each attribute, allowing a fair comparison of the methods.

The FUZZYWRAPPER method was used to select features for the FCA-BASED and Doc-BASED methods. Table 6.12 presents the original number of features of each dataset and the number of features selected by FUZZYWRAPPER. Notice that SLAVE also performs a feature selection, although its procedure is embedded in its GA and works at a rule level, not at a dataset level, i.e., each rule generated by SLAVE uses its own set of attributes.

| Dataset    | Features | Selected Features |
|------------|----------|------------------|
| Credit     | 15       | 5                |
| Cylinder   | 32       | 5                |
| Diabetes   | 8        | 7                |
| Glass      | 9        | 7                |
| Heart      | 13       | 7                |
| Ionosphere | 34       | 8                |
| Iris       | 4        | 2                |
| Segmentation | 19     | 7                |
| Vehicle    | 18       | 7                |
| Wine       | 13       | 5                |

Notice that, for each of the 10 folds, a FDB was defined by the FUZZYDBD method. This way, Table 6.13 shows the average number of fuzzy sets and standard deviation, in brackets, estimated by the FUZZYDBD method using information gain (Info Gain), ReliefF, and the Wang & Mendel method as estimation functions. The smallest number of fuzzy sets obtained are dark-gray shaded.

Using Wang & Mendel as estimation function led to a smaller number of fuzzy sets, on average, for 8 of the 10 datasets. ReliefF presented smaller number of fuzzy sets for the remaining two datasets. Consequently, information gain produced larger numbers of fuzzy sets than the other two estimation functions for all datasets.

Once the FDB is defined, it is possible to apply the procedure, based on FCA, to extract the classification rules for the given dataset. Experimental
results showed that our method, which uses FCA to extract fuzzy classification rules (see Algorithm 5), is able to extract a suitable number of rules in reasonable time to form the search space of a GA.

Table 6.14 shows the average number of extracted rules and the standard deviation (in brackets) for each dataset used in our experiments, using a 10-fold cross-validation strategy and the 4 different FDBs. The last column presents the total number of rules that could be generated for preselection by the DOC-BASED method using three fuzzy sets to define each attribute (considering only rules that include all attributes).

Following the extraction of the classification rules using FCA, a GA is used to generate the final rule base by means of selection.

For the GA to have a good performance, the search space must be formed by a reasonable number of rules, so the population can be randomly initialized with an appropriate genetic diversity. Thus, a small number of rules can lead to a poor performance of the GA, while a large number of rules increases its search time. Nevertheless, when a large number of rules is available, it is always possible to form the search space using only part of the available rules, in order to reduce the computational cost of the GA. This approach was used for the Cylinder and Ionosphere datasets, whose genetic search spaces were formed by 50% of the total rules extracted using FCA, randomly chosen.

Although the number of extracted rules for the Cylinder and Ionosphere are quite large and only 50% of the available rules extracted using FCA form the search space, it is interesting to compare the number of rules that would

---

### Table 6.13: Average number of fuzzy sets - FCA-BASED.

| Dataset  | Info Gain | ReliefF | Wang & Mendel | DOC-BASED  |
|----------|-----------|---------|---------------|------------|
| Credit   | 4.22 (2.97) | 3.82 (2.80) | 3.99 (3.23) | 79 billion |
| Cylinder | 5.41 (2.74) | 4.50 (2.47) | 3.88 (1.64) | 402 billion |
| Diabetes | 6.99 (1.86) | 6.20 (1.89) | 4.09 (0.39) | 13,122.0 |
| Glass    | 8.44 (0.70) | 6.27 (2.07) | 3.79 (0.51) | 137,781.0 |
| Heart    | 8.68 (1.79) | 4.76 (1.48) | 3.84 (0.72) | 3.2 billion |
| Ionosphere | 7.75 (1.43) | 6.55 (1.47) | 4.16 (0.76) | 33 quatrillion |
| Iris     | 8.38 (0.24) | 6.03 (1.81) | 4.20 (1.43) | 243.0 |
| Segmentation | 7.48 (2.12) | 5.02 (2.42) | 3.04 (0.79) | 1.5 billion |
| Vehicle  | 7.82 (1.45) | 6.71 (1.92) | 3.33 (0.57) | 4.8 billion |
| Wine     | 8.16 (0.56) | 3.85 (0.99) | 4.54 (0.74) | 4.8 billion |

---

### Table 6.14: Average number of extracted rules using FCA and total number of rules.

| Dataset  | Info Gain | ReliefF | Wang & Mendel | FCA-BASED-3 sets | DOC-BASED |
|----------|-----------|---------|---------------|-----------------|-----------|
| Credit   | 68,766.8 (26.2) | 59,455.5 (21.4) | 28,689.1 (18.3) | 36,723.4 (14.4) | 79 billion |
| Cylinder | 311,358.3 (267.1) | 482,216.1 (811.4) | 864,579.2 (223.5) | 862,890.9 (223.5) | 402 billion |
| Diabetes | 10,544.8 (7.0) | 9,347.5 (4.9) | 3,470.9 (9.8) | 11,255.4 (1.0) | 13,122.0 |
| Glass    | 1,371.3 (1.1) | 1,254.9 (1.9) | 558.7 (0.9) | 717.8 (0.5) | 137,781.0 |
| Heart    | 27,813.3 (27.8) | 31,678.8 (20.2) | 30,022.5 (38.0) | 32,997.2 (8.3) | 3.2 billion |
| Ionosphere | 405,408.3 (338.4) | 583,823.8 (397.4) | 1,125,316.8 (3,083.4) | 394,036.1 (276.2) | 33 quatrillion |
| Iris     | 601.1 (0.2) | 223.8 (0.3) | 103.8 (0.4) | 104.7 (0.1) | 243.0 |
| Segmentation | 2,322.5 (1.6) | 2,549.5 (3.0) | 767.5 (1.7) | 744.3 (0.4) | 8 billion |
| Vehicle  | 78,249.3 (61.8) | 121,420.7 (46.7) | 41,291.3 (141.8) | 77,693.2 (8.1) | 1.5 billion |
| Wine     | 5,281.7 (3.0) | 8,086.3 (12.3) | 6,868.8 (14.5) | 6,302.5 (1.7) | 4.8 billion |
be evaluated by the DoC-BASED method, shown in the last column of Table 6.14. The task of generating the rules and having their DoC calculated can be considered unfeasible for most of the datasets. In order to use the DoC-BASED method to generate the search spaces, a feature selection process is mandatory.

A possible alternative to use the DoC-BASED method for large datasets is to combine the feature selection process with the random generation of a predefined number of rules, representing a small set of the total possible set of rules. Nevertheless, even if a FSS method is used, the generation of all possible rules for a dataset with \( m \) features (i.e., the generation of all possible rules that include 1, 2, ..., \( m \) feature combination) might still require a considerable computational effort. As stated before, by using only rules containing all \( m \) attributes of a dataset, the interpretability of the model is compromised.

Experiments with the FCA-BASED method were carried out using 200 chromosomes, 400 iterations, 10% elitism, 5% mutation, and 70% crossover. Table 6.15 shows the error rates (Error) and standard deviation, in brackets, obtained by the FCA-BASED, DoC-BASED and SLAVE methods. Notice that the two groups, the methods using fixed FRBs defined by 3 triangular fuzzy sets for each attribute (FCA-BASED (FCA - 3), DoC-BASED and SLAVE), and FCA-BASED (FCA-IG, FCA-RF, and FCA-WM) using partitions with independent numbers of fuzzy sets for each attribute are separated by a double column. The smallest error rates obtained are dark-grey shaded. The last row of the table presents the average ranking of each approach.

Table 6.15: Error rates - FCA-BASED.

| Datasets | ME      | SLAVE   | DoC-BASED | FCA - 3 | FCA - IG | FCA - RF | FCA - WM |
|----------|---------|---------|-----------|---------|---------|---------|---------|
| Credit   | 45.33   | 37.36   | 39.20     | 39.82   | 24.21   | 20.95   | 21.26   |
| Cylinder | 35.74   | 35.74   | 38.08     | 25.77   | 37.68   | 36.93   | 34.23   |
| Diabetes | 34.90   | 25.66   | 20.10     | 23.09   | 49.16   | 33.95   | 26.46   |
| Glass    | 65.46   | 38.25   | 36.51     | 37.56   | 43.04   | 30.87   | 39.99   |
| Heart    | 44.44   | 20.74   | 27.75     | 19.72   | 26.27   | 22.20   | 22.94   |
| Ionosphere | 35.90 | 12.27   | 17.06     | 11.76   | 20.88   | 11.85   | 15.83   |
| Iris     | 66.67   | 4.67    | 10.67     | 4.68    | 6.01    | 17.35   | 2.01    |
| Segmentation | 55.71 | 13.43   | 29.98     | 24.75   | 36.67   | 33.34   | 25.58   |
| Vehicle  | 74.23   | 39.59   | 50.01     | 44.15   | 48.79   | 58.72   | 33.43   |
| Wine     | 59.74   | 9.54    | 12.84     | 4.27    | 15.00   | 6.40    | 4.40    |
| Avg. Ranking | 2     | 6       | 1         | 5       | 4       | 3       |

Table 6.15 shows that the FCA-BASED method using 3 fuzzy sets for each attribute (FCA - 3) obtained the smallest error rates for 8 of the 10 datasets with one tie with FCA - RF and another one with SLAVE. SLAVE had the smallest error rates for three datasets, with one tie. Regarding the ranking of each approach, it is interesting to notice that while the FCA-BASED method with 3 fuzzy sets defining each attribute had, on average, the best results, the DoC-BASED method is in the last position. Thus, the results show the superiority of the FCA-BASED method over the DoC-BASED method in terms
of accuracy. The definition of the FDBs using the FUZZYDBD method did not produce rule bases with competitive error rates.

To test whether there is a statistically significant difference among the approaches used to generate the FRBS, regarding the obtained error rates, we used the Friedman test (Demsar, 2006). The Friedman test found that there is no statistically significant difference among the results with a 95% confidence level.

In order to analyse the models produced by each approach, Table 6.16 presents the number of rules and standard deviation, in brackets, for each approach. The methods using three fuzzy sets defining each attribute of the datasets are separated from the methods that define independent numbers of fuzzy sets for each attribute of a dataset by a double bar.

### Table 6.16: Number of generated rules - FCA-BASED.

| Dataset | SLAVE | Doc - 3 | FCA - 3 | Flexible number of fuzzy sets |
|---------|-------|---------|---------|-----------------------------|
|         |       | Three fuzzy sets | FCA - IG | FCA - RF | FCA - WM |
| Credit  | 5.10 (0.70) | 229.8 (44.99) | 16.50 (3.53) | 106.00 (24.42) | 144.70 (57.65) | 127.40 (82.12) |
| Cylinder| 1.00 (0.00) | 126.00 (16.73) | 15.80 (6.27) | 78.30 (39.13) | 62.90 (26.51) | 68.10 (26.33) |
| Diabetes| 4.20 (0.75) | 125.10 (38.76) | 9.10 (3.01) | 50.60 (47.18) | 114.00 (24.93) | 56.10 (46.35) |
| Glass   | 11.90 (2.17) | 81.00 (17.92) | 6.80 (2.36) | 31.40 (18.19) | 25.90 (6.21) | 12.60 (4.95) |
| Heart   | 8.30 (1.68) | 235.20 (38.31) | 14.10 (6.20) | 45.70 (11.06) | 46.20 (14.29) | 53.30 (22.70) |
| Iono    | 15.10 (2.90) | 74.90 (28.78) | 19.90 (6.62) | 111.40 (50.62) | 93.00 (57.90) | 78.00 (24.05) |
| Iris    | 3.20 (0.40) | 5.40 (1.07) | 4.50 (0.81) | 18.70 (4.79) | 11.30 (2.21) | 8.40 (4.62) |
| Segment| 3.50 (0.50) | 92.80 (23.61) | 11.50 (2.54) | 38.50 (10.20) | 29.90 (4.92) | 27.20 (11.39) |
| Vehicle | 21.70 (4.58) | 148.00 (32.71) | 30.20 (7.31) | 146.90 (38.00) | 149.60 (21.77) | 131.10 (42.62) |
| Wine    | 4.50 (0.50) | 47.40 (10.37) | 4.90 (1.04) | 28.70 (3.86) | 26.90 (6.67) | 27.80 (7.67) |
| Ranking | 1 | 6 | 2 | 5 | 4 | 3 |

SLAVE produced the rule bases with smallest number of rules for 9 of the 10 datasets. The remaining dataset had the smallest number of rules with FCA-BASED using three uniformly distributed triangular fuzzy sets. For the DOC-BASED method and FCA-BASED using estimation functions to define the FDB (FCA-IG, FCA-RF, and FCA-WM), the number of rules is quite large, compromising the interpretability of the rule bases.

Regarding the ranking of the methods on the number of generated rules, SLAVE is in first place, followed by FCA-BASED using a FDB defined by 3 fuzzy sets. Nevertheless, it is important to notice that the SLAVE method used only the majority class as a single rule for the Cylinder dataset, i.e., for this dataset, SLAVE does not present any learning. The FCA-BASED approach using FDBs defined by the FUZZYDBD method produced rule bases with poor interpretability due to their large amount of rules.

Similarly to what happened with the error rates of the models, the FCA-BASED and the DOC-BASED methods are in opposite positions: while the FCA-BASED method produced rule bases with reasonable number of rules, the DOC-BASED method produced rule bases with large numbers of rules. This shows the superiority of the FCA-BASED over the DOC-BASED method in terms of interpretability, as well.
Although the number of rules is a good parameter to evaluate the interpretability of a rule base, it is also important to analyse the interpretability of the rule base at a rule level. The model produced by SLAVE for the Cylinder dataset is a good example of the type of contradictions that might be found: although it contains only one rule, which could be seen as a highly interpretable model, its error rate is equal to the error of the majority class, thus, there is no learning. In this sense, Table 6.17 presents the number of rules and syntactic complexity (Complexity) for SLAVE and FCA-BASED using 3 fuzzy sets. The syntactic complexity is defined here as the average number of rules × the average number of conjunctions of the rules. The results for FCA using estimation functions to define the FDB are not presented due to the poor quality of the generated rule bases.

| Datasets   | SLAVE Rules | SLAVE Complexity | FCA-BASED Rules | FCA-BASED Complexity |
|------------|-------------|------------------|-----------------|----------------------|
| Credit     | 5.10        | 18.60            | 10.50           | 32.80                |
| Cylinder   | 1.00        | 1.00             | 15.80           | 70.10                |
| Diabetes   | 4.20        | 9.10             | 33.10           | 9.10                 |
| Glass      | 11.90       | 51.10            | 6.80            | 39.40                |
| Heart      | 8.30        | 43.90            | 14.10           | 58.50                |
| Ionosphere | 15.10       | 75.00            | 19.90           | 77.50                |
| Iris       | 3.20        | 10.40            | 4.50            | 12.80                |
| Segment    | 3.50        | 15.90            | 11.30           | 49.20                |
| Vehicle    | 21.70       | 151.90           | 30.20           | 172.90               |
| Wine       | 4.50        | 30.20            | 4.90            | 13.30                |

Discarding the syntactic complexity for the cylinder dataset, since in this case SLAVE was not able to learn, SLAVE produced rules with less conjunctions for 7 datasets against two for FCA-BASED. Nevertheless, it is important to consider the fact that SLAVE produces rules formed by conjunctions with sets of disjunctions, while FCA-BASED produces standard rules, formed only by conjunctions in their antecedent. Thus, since the variation in the SC of the models generated by both methods for Heart, Ionosphere, Iris, and Vehicle is relatively small, it is difficult to state that the models generated by SLAVE are more interpretable than the ones generated by FCA-BASED without the support of experts in the domains.

Concluding, the experimental results show that the FCA-BASED method produces rule bases with lower error rates and less rules than DoC-BASED. The FCA-BASED method also showed better accuracy than SLAVE for most of the datasets used in our experiments. Regarding the interpretability of the compared methods, DoC-BASED produced rule bases with many rules, and, thus, with poor interpretability. Due to the different types of rules produced by FCA-BASED and SLAVE, although they produced reasonably interpretable fuzzy rule bases, the comparisons are inconclusive.
6.6 Final Remarks

The successful application of fuzzy systems in several areas has created the need for robust methods to automatically generate fuzzy systems. The genetic approach has been widely researched and several proposals can be found in the literature.

The main challenges with the generation of fuzzy systems, independently of the adopted approach, include the “dimensionality curse” and the search for a suitable trade-off between accuracy and interpretability of the generated models. In this work we present the DoC-BASED method, a genetic approach for the automatic definition of fuzzy systems. The DoC-BASED method generates and evaluates a large number of rules according to their degree of coverage, in order to decide whether to include them or not in the genetic search space. Since this process of defining the search space poses an exponential complexity, the use of a feature subset selection might be essential in order for the DoC-BASED method to produce suitable results for datasets with large numbers of attributes.

In order to overcome the exponential complexity of the process of rule extraction used by the DoC-BASED method, in this work we propose the FCA-BASED method, which uses the theory of formal concept analysis to produce the rules forming the genetic search space. It also performs a rule selection process after the definition of the fuzzy rule bases. Experimental results showed that FCA-BASED produces highly accurate and interpretable models.

In the next chapter, some selected methods from the classic and fuzzy communities that are based on similar approaches, as well as experimental results and comparisons between these classification methods are presented.
Chapter 7

Classic versus Fuzzy Classification Methods

The task of classification is widely researched by the machine learning and fuzzy communities. Moreover, these communities have proposed classification methods based on similar approaches, such as rule learning, decision trees, and genetic algorithms, among others. In spite of the similarities, each method presents its own characteristics. Aiming at analysing the quality and drawbacks of similar algorithms from both communities, this work experimentally evaluates and compares several algorithms. Special emphasis is given to the decision tree-based methods, due to the fact that they usually produce highly competitive and interpretable models with low computational cost. As a result of this work, four papers with distinct comparisons of classic and fuzzy classification models were published.

In this chapter, the most relevant results, according to the sequence of their publication, are presented.

7.1 Initial Comparisons of Classification Methods

Our first comparison of classic and fuzzy classification methods included five classic methods (C4.5, Multilayer Perceptron, NaiveBayes, and OneRule), and two fuzzy methods (DoC-BASED and Wang & Mendel). The comparisons were carried out using four datasets from UCI (Frank and Asuncion, 2010), and taking only the error rates of the models into account.

In the sequel we presented the methods used in the comparisons.
7.1.1 Description of the Involved Methods

C4.5 is described in Section 5.1.1, page 64. The DoC-BASED method is described in Section 6.3, page 89, while Wang & Mendel is described in Section 3.2.1, page 28. The remaining selected methods are presented next.

**NaiveBayes** — This classifier (Domingos and Pazzani, 1997) is a simple probabilistic classifier based on applying Bayes’ theorem with strong or naive independence assumptions. These naive assumptions come from the fact that a naive Bayes classifier assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature. Since the NaiveBayes assumes independence of the variables, only the probability of each variable for each class needs to be determined and not the entire covariance matrix.

**OneRule** — This algorithm, also known as OneR, is a simple classification algorithm that generates a one-level decision tree (Holte, 1993). The OneR algorithm creates one rule for each attribute in the training data by determining the most frequent class for each attribute value, then it selects the rule with the smallest error rate as its *one rule*. Since OneR selects the rule with the lowest error rate, if two or more rules have the same error rates, the rule is chosen at random. The OneR implementation of WEKA, on the other hand, selects the rule which obtains the highest number of correct instances, not necessarily the one with the lowest error rate (Witten et al., 2011).

**Multilayer Perceptron** — An artificial neural network, usually called *neural network*, is a computational model that tries to simulate the structure and/or functional aspects of biological neural networks (Mitchell, 1997). It consists of an interconnected group of artificial neurons and processes using a connectionist approach to computation. In most cases a neural network is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Neural networks are non-linear statistical data modelling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data, thus, they can be used for classification problems.

A Multilayer Perceptron (MLP) (Haykin, 1998) is a feed-forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. It is a modification of the standard linear perceptron. It uses three or more layers of neurons (nodes) with non-linear activation functions in order to be able to distinguish data that is not linearly separable,
or separable by a hyperplane.

One of the problems generally attributed to neural networks is the fact that they do not generate an interpretable structure, such as rules, for instance, although there are several proposals for the extraction of rules from neural networks (Milaré et al., 2004; Zhou, 2004; Hayashi et al., 2000).

Regarding the fuzzy methods evaluated in this first work, Wang & Mendel is described in Section 3.2.1, page 28, while DoC-BASED is described in Section 6.3, page 89.

Next, we present the initial experiments and comparisons of the machine learning and fuzzy methods described previously.

7.1.2 Experimental Evaluation

The experiments were conducted using 5-fold cross validation on 4 datasets from UCI (Frank and Asuncion, 2010). Table 7.1 shows a summary of the characteristics of the datasets used in the experiments presenting the number of examples in each dataset, number of features, including the number of continuous and discrete features, in brackets, number of classes, and the majority class error (ME). Notice that all the attributes for the 4 datasets are continuous.

Table 7.1: Characteristics of the datasets - Initial comparisons.

| Dataset  | Examples | Features | Classes | ME    |
|----------|----------|----------|---------|-------|
| AutoMPG  | 209      | 4 (4, 0) | 3       | 53.57 |
| Diabetes | 769      | 8 (8, 0) | 2       | 34.90 |
| Iris     | 150      | 4 (4, 0) | 3       | 66.67 |
| Machine  | 392      | 7 (7, 0) | 3       | 64.37 |

For the Machine dataset, two attributes were discarded: vendor name and model name, since they were unique for most examples. Moreover, the class for Machine, which is continuous, was discretized using 3 triangular fuzzy sets evenly distributed in the domain. AutoMPG had the class fuzzyfied in a similar way. The fuzzy data bases consisted of 3 triangular shaped fuzzy sets for each attribute, evenly distributed in the partitions.

The experiments with the J4.8, NaiveBayes, OneRule, and MLP were carried out using the Experimenter tool in the Weka framework (Witten et al., 2011), with default parameters. Although the default parameters of these algorithms might not produce their best results, they were adopted due to the fact that the fine tuning of these parameters depend on the knowledge of experts on these methods, which could also be divergent. Moreover, the best settings for each method can only be defined considering each dataset individually. This way, it is important to bear in mind that by fine tuning the
parameters of these algorithms, better results could be obtained. Notice that for the fuzzy methods, the default parameters were also adopted.

Table 7.2 presents the error rates and the standard deviation (in parenthesis). The dark-gray shaded cells highlight the smallest error rates.

Table 7.2: Error rates - Initial comparisons.

| Datasets  | Fuzzy Approaches     | Classic Approaches |
|-----------|----------------------|--------------------|
|           | Wang & Mendel        | J4.8               | NaiveBayes   | OneRule | MLP          |
| AutoMPG   | 13.68 (2.12)         | 27.73 (0.72)       | 16.58 (3.82) | 18.39 (2.89) | 23.21 (2.42) | 16.20 (3.27) |
| Diabetes  | 0.03 (0.05)          | 28.52 (0.22)       | 28.78 (3.31) | 24.57 (3.76) | 28.25 (2.87) | 25.38 (3.01) |
| Iris      | 0.29 (0.40)          | 8.67 (0.32)        | 4.00 (2.79)  | 4.53 (3.55)  | 6.00 (1.49)  | 3.60 (2.92)  |
| Machine   | 4.90 (0.40)          | 45.79 (0.93)       | 4.80 (4.48)  | 14.84 (5.11) | 12.43 (1.91) | 7.51 (4.01)  |

Table 7.3 presents the number of rules and the standard deviation rates obtained by the rule-based methods. The dark-grey shaded cells highlight the smallest rates. Since the NaiveBayes and the MLP methods do not produce rules, they cannot be assessed in terms of the number of induced rules. Moreover, once OneRule uses a single rule to classify any input, it was also discarded from the comparisons.

Table 7.3: Number of generated rules - Initial comparisons.

| Datasets  | DoC-BASED       | Wang & Mendel | J4.8       |
|-----------|----------------|---------------|------------|
| AutoMPG   | 13.00 (2.12)   | 20.00 (1.41)  | 7.20 (4.39) |
| Diabetes  | 14.20 (0.44)   | 24.00 (2.34)  | 4.60 (2.41) |
| Iris      | 7.60 (1.34)    | 15.00 (1.92)  | 4.40 (0.55) |
| Machine   | 6.20 (1.30)    | 13.8 (1.78)   | 10.60 (1.14) |

J48 obtained the best (smallest) average number of rules for all datasets, but for the Machine dataset, which obtained the best average number of rules with the DoC-BASED method.

It is important to notice that for the Machine dataset, the DoC-Based method shows the smallest number of rules and error rate comparable to the best one, obtained by J4.8, while for the remaining datasets, although J4.8 shows the smallest number of rules, its performance in terms of error rate is not comparable to the DoC-Based method. In fact, we can say that the reduction of the models generated by J4.8 does not pay for their poor performance in terms of error rates, when compared with the DoC-BASED method.

The Friedman test (Demsar, 2006) was used to find statically significant differences among the methods, according to their error rates (Demsar, 2006). Results showed that the DoC-Based method is significantly better than the NaiveBayes, and OneRule methods with a 95% confidence level.

Table 7.4 shows the general and average rankings of each method, for the classic and general fuzzy reasoning methods. DoC-Based is the first one, followed by Wang & Mendel, MLP, J4.8, NaiveBayes, and OneRule.

Summing up, it is possible to observe that although the DoC-Based method did not perform as well as the J4.8 method in terms of the number of rules...
Table 7.4: Ranking for the Friedman test - Initial comparisons.

| Algorithm | Fuzzy Approaches | Classic Approaches |
|-----------|------------------|--------------------|
|           | DoC-Based | Wang & Mendel | J4.8 | NaiveBayes | OneRule | MLP |
| General Ranking | 1 | 4 | 3 | 5 | 6 | 2 |
| Average Ranking | 1.14 | 4.13 | 3.37 | 4.14 | 5.00 | 3.00 |

of the generated models, its performance in terms of error rates is satisfactory. The Wang & Mendel method had reasonable accuracy, while OneRule, although very simple, lacks accuracy power.

The proposal of the FuzzyDT method motivated us to include it in our comparisons. The experimental evaluation and comparisons including the FuzzyDT method are presented next.

7.2 Comparing FuzzyDT, C4.5, and PART

In order to evaluate the performance of the FuzzyDT method, we compared it with two classic decision tree-based algorithms, C4.5 and PART, on ten datasets from UCI. The comparisons considered the error rates and the syntactic complexity of the models generated by these algorithms. Since these algorithms have an embedded feature subset selection method, we also compared them in terms of the number of features present in their models.

The FuzzyDT method is described in Section 5.1.2, page 65, while C4.5 is described in Section 5.1.1, page 64. It follows the description of PART.

7.2.1 PART

The name PART is related to the fact that the algorithm is based on partial decision trees (Frank and Witten, 1998). PART is a rule-induction procedure that adopts the separate-and-conquer strategy. In essence, it builds a rule, removes the instances it covers, and continues creating rules recursively for the remaining instances until none is left. In order to generate this rule, PART generates a decision tree and prunes all but one leaf (specifically the leaf with the largest coverage) and makes this leaf into a rule, discarding the rest of the tree. According to its authors, using a pruned tree to obtain a rule, instead of building it incrementally by adding conjunctions one at a time, avoids the over-pruning problem of the basic separate-and-conquer rule learner. In fact, using the separate-and-conquer methodology in conjunction with decision trees adds flexibility and speed to the process.

Since it is wasteful to build a full decision tree just to obtain a single rule, the process described is accelerated significantly without sacrificing the above advantages by building a ‘partial’ decision tree instead of a complete one. A partial decision tree is an ordinary decision tree that contains branches to
undefined subtrees. An integration of the construction and pruning operations is used in order to find a 'stable' subtree that can be simplified no further. This way, once this subtree has been found, the tree induction ceases and a single rule is selected.

### 7.2.2 Experimental Evaluation

The comparison of these three decision tree-based algorithms (FUZZYDT, C4.5, and PART) aimed at further evaluating FUZZYDT. For this purpose, we analysed and compared their error rates and syntactic complexity rates. Furthermore, the PART algorithm was also executed using the features present in the models produced by C4.5, as well as the features selected by three filter approaches (CFS, Consistency, and ReliefF). The J4.8 implementation of C4.5, available at WEKA was used, as well as WEKA's implementation of PART, CFS, Consistency, and ReliefF. Similarly to previous work, an average merit threshold of 0.009 was used to selected features with ReliefF. The experiments were carried out using the default settings of the algorithms. The classic fuzzy reasoning method was adopted for the fuzzy approaches.

Table 7.5 summarizes the dataset characteristics giving the total number of examples, total number of features, including the number of continuous and discrete features in brackets, number of classes, the majority error (ME), and the number of fuzzy sets for each of the attributes (FS) used by FUZZYDT. Examples with missing values were removed from the datasets.

| Dataset     | Examples | Features | Classes | ME     | FS |
|-------------|----------|----------|---------|--------|----|
| Credit      | 653      | 15 (6, 9)| 2       | 45.33  | 2  |
| Cylinder    | 277      | 32 (19, 13)| 2   | 35.74  | 2  |
| Dermatology | 358      | 34 (33, 1)| 6     | 68.99  | 2  |
| Diabetes    | 769      | 8 (8, 0) | 2       | 34.90  | 2  |
| Glass       | 220      | 9 (9, 0)| 7       | 65.46  | 7  |
| Heart       | 270      | 13 (13, 0)| 2     | 44.44  | 2  |
| Ionosphere  | 351      | 34 (34, 0)| 2     | 35.90  | 3  |
| Segmentation| 210      | 19 (19, 0)| 7     | 85.71  | 3  |
| Vehicle     | 846      | 18 (18, 0)| 4     | 74.23  | 2  |
| Wine        | 178      | 13 (13, 0)| 3     | 59.74  | 3  |

Notice that the AutoMPG and Machine datasets used in the previous work, were not included in this work due to the fact that they have continuous classes, which would require a fuzzyfication. For this work, only datasets with discrete classes were included.

Regarding the number of fuzzy sets describing each feature, they were defined by the FUZZYDBD method (see Section 3.2, page 27) with triangular shaped fuzzy sets evenly distributed in the features domains.

Table 7.6 presents the original number of features of each dataset (Features), the number of features used by the models induced by FUZZYDT, PART,
and C4.5 algorithms, as well as the features selected by the three filters: CFS, Consistency, and ReliefF. The two groups of FSS methods (embedded methods and filters) are separated by a double bar. The average number of features is also presented. The number of features is compared for each group and the smallest values in each group are dark-gray shaded.

Table 7.6: Original and selected number of features - DT-based methods.

| Dataset    | Features | Embedded Methods | Filters |
|------------|----------|------------------|---------|
|            |          | FUZZYDT | PART | C4.5 | CFS | Consistency | ReliefF |
| Credit     | 15       | 15     | 10   | 9    | 7   | 13         | 12      |
| Cylinder   | 32       | 26     | 24   | 18   | 6   | 9          | 16      |
| Dermatology| 34       | 17     | 11   | 7    | 16  | 8          | 16      |
| Diabetes   | 8        | 5      | 8    | 6    | 4   | 8          | 5       |
| Glass      | 9        | 6      | 9    | 9    | 7   | 7          | 8       |
| Heart      | 13       | 5      | 13   | 12   | 8   | 10         | 12      |
| Ionosphere | 34       | 10     | 14   | 14   | 7   | 9          | 33      |
| Segmentation| 19       | 11     | 11   | 12   | 7   | 9          | 15      |
| Vehicle    | 18       | 11     | 19   | 15   | 11  | 18         | 18      |
| Wine       | 13       | 8      | 4    | 3    | 10  | 5          | 13      |
| Average    | 19.5     | 10.6   | 13.0 | 10.8 | 9.0 | 9.4        | 14.8    |

Regarding the embedded methods, it is possible to observe that the models induced by the FUZZYDT and C4.5 algorithms present fewer features. On average, they only use about 50% of the original features, while PART always includes more features in its model. Considering now the filters, CFS and Consistency selected a smaller number of features than ReliefF.

The error rates of the models induced by the three algorithms are shown in Table 7.7. To give a chance for the PART rule induction algorithm to improve its performance, PART was executed with all the features, similarly to the FUZZYDT and C4.5 algorithms, as well as having access only to the features selected by C4.5 and the ones selected by the other three FSS methods.

Table 7.7 presents the majority error (ME) and error rates and standard deviation (in brackets) for the FUZZYDT, C4.5, and PART algorithms, as well as the error rates and standard deviation for the models induced by PART using only the features selected by C4.5 (PART-C4.5), CFS (PART-CFS), Consistency (PART-Consistency), and ReliefF (PART-ReliefF).

FUZZYDT was able to obtain the best (smallest) error rates for 7 out of the 10 datasets, and shows the smallest average error, as well as the best ranking. In general, PART was able to have an improved performance using the features selected by C4.5, CFS and Consistency, although not significantly.

The good results obtained by FUZZYDT are related to its fuzzyfication stage. Although we used the FUZZYDBD method for all datasets, it was observed that using three fuzzy sets instead of two, as defined by FUZZYDBD — Table 7.5 — for the Dermatology and Heart datasets, the error rates drop from 6.06 to 5.67 for Dermatology, and from 18.52 to 14.80 for Heart.

As done previously, the Friedman test (Demsar, 2006) was applied using the obtained error rates. However, the Friedman test found that there is no
Table 7.7: Error rates - DT-based methods.

| Datasets | ME | FUZZYDT | C4.5 | PART | PART-C45 |
|----------|----|---------|------|------|----------|
| Credit   | 45.53 | 14.03 | 15.64 | 15.77 |
| Cylinder | 35.74 | 32.34 | 31.72 | 27.88 |
| Dermatology | 68.99 | 4.66 | 5.33 | 3.31 |
| Diabetes | 34.90 | 25.51 | 26.55 | 26.37 |
| Glass    | 65.46 | 32.55 | 32.95 | 32.95 |
| Heart    | 44.44 | 21.85 | 22.67 | 22.93 |
| Ionosphere | 35.90 | 10.26 | 9.17 | 9.12 |
| Segmentation | 85.71 | 27.79 | 27.79 | 27.79 |
| Vehicle  | 74.23 | 8.79 | 7.95 | 7.95 |
| Wine     | 59.74 | 3.54 | 3.54 | 3.54 |
| Average  | 55.04 | 19.04 | 19.13 | 19.03 |

Table 7.8 presents the number of rules of the models induced by FUZZYDT, C4.5, and PART, as well as the ones induced by PART using the features selected by C4.5 (PART-C4.5), CFS (PART-CFS), Consistency (PART-Consistency), and ReliefF (PART-ReliefF). The average is also presented in the last row of the table. The results of the algorithms executed without feature selection are separated from the results of PART with feature selection by a double bar. The best (smallest) results are dark-grey shaded for each of the groups.

Table 7.8: Number of generated rules - DT-based methods.

| Datasets | Embedded Methods | PART with feature selection |
|----------|------------------|-----------------------------|
|          | FUZZYDT | C4.5 | PART | PART-C45 | PART-CFS | PART-Consistency | PART-ReliefF |
| Credit   | 9 | 29 | 20 | 23 | 20 | 23 | 40 |
| Cylinder | 38 | 22 | 25 | 20 | 23 | 20 | 27 |
| Dermatology | 25 | 9 | 7 | 9 | 7 | 9 | 9 |
| Diabetes | 55 | 16 | 17 | 16 | 17 | 17 | 16 |
| Glass    | 6 | 13 | 20 | 13 | 20 | 16 | 22 |
| Heart    | 19 | 10 | 8 | 10 | 8 | 8 | 10 |
| Ionosphere | 28 | 11 | 10 | 11 | 11 | 11 | 11 |
| Segmentation | 38 | 29 | 24 | 29 | 29 | 29 | 29 |
| Vehicle  | 17 | 4 | 4 | 4 | 4 | 4 | 4 |
| Average  | 24.00 | 16.50 | 13.60 | 13.90 | 14.30 | 17.70 |

PART, using the attributes selected by C4.5 shows the smallest average of rules. When comparing the algorithms with no previous feature selection, PART produced the smallest sets of rules for 6 datasets, FUZZYDT for 3, and C4.5 for one dataset.
Table 7.9 presents the average number of conjunctions of the induced models. The smallest values for each dataset in both groups are dark-grey shaded.

Table 7.9: Average number of conjunctions - DT-based methods.

| Embedded Methods | PART with feature selection |
|------------------|-----------------------------|
|                  | PART-C45 | PART-CFS | PART-Consistency | PART-ReliefF |
| Datasets         | FuzzyDT | C4.5    | PART |                  |              |
| Credit           | 4.2     | 7.4     | 4.4  | 2.9              | 2.4          | 3.4          | 4.1          |
| Cylinder         | 10.3    | 7.9     | 3.8  | 3.5              | 2.2          | 5.2          | 3.0          |
| Dermatology      | 7.5     | 4.4     | 2.9  | 2.0              | 2.0          | 2.0          | 2.9          |
| Diabetes         | 3.3     | 5.6     | 2.9  | 2.0              | 2.2          | 2.9          | 2.4          |
| Glass            | 9.4     | 4.4     | 2.9  | 2.7              | 3.1          | 3.1          | 2.8          |
| Heart            | 2.8     | 4.9     | 2.8  | 2.8              | 2.8          | 2.7          | 2.8          |
| Iono.            | 5.1     | 6.7     | 2.6  | 3.1              | 2.6          | 2.5          | 2.6          |
| Segmen.          | 5.5     | 5.0     | 1.8  | 1.9              | 1.7          | 1.7          | 1.8          |
| Vehicle          | 6.0     | 9.4     | 3.8  | 5.5              | 4.0          | 3.8          | 3.8          |
| Wine             | 3.3     | 2.4     | 1.3  | 1.3              | 1.3          | 1.3          | 1.3          |
| Average          | 5.1     | 6.0     | 2.9  | 2.8              | 2.4          | 2.9          | 2.8          |

The smallest average number of conjunctions was obtained by PART using the features selected by CFS. Clearly, PART (with and without previous feature selection) produced the set of rules with the smallest number of conjunctions, having in average from 47% (PART-CFS × FuzzyDT) to 57% (PART or PART-Consistency × FuzzyDT) less conjunctions in the set of rules than the other two methods (C4.5 and FuzzyDT).

However, to analyse the syntactic complexity of the models, both results should be taken into account, i.e., the product of the average number of rules and the average number of conjunctions — Table 7.10. The average and ranking are also presented in the last rows of the table. The models having the lowest syntactic complexity in both groups are dark-gray shaded.

Table 7.10: Number of Rules × Average Number of Conjunctions - DT-based methods.

| Embedded Methods | PART with feature selection |
|------------------|-----------------------------|
|                  | PART-C45 | PART-CFS | PART-Consistency | PART-ReliefF |
| Datasets         | FuzzyDT | C4.5    | PART |                  |              |
| Credit           | 37.8    | 222.0   | 127.6| 58.0             | 31.2         | 78.2         | 164.0        |
| Cylinder         | 291.4   | 199.5   | 83.6 | 87.5             | 23.0         | 104.0        | 81.0         |
| Dermatology      | 172.5   | 33.2    | 26.1 | 14.0             | 16.0         | 12.0         | 26.1         |
| Diabetes         | 23.1    | 112.0   | 37.7 | 14.0             | 11.0         | 37.7         | 21.6         |
| Glass            | 176.0   | 179.2   | 43.5 | 43.2             | 52.7         | 52.7         | 44.8         |
| Heart            | 16.8    | 88.2    | 64.4 | 44.8             | 56.0         | 43.2         | 61.6         |
| Iono.            | 96.9    | 120.6   | 26.0 | 21.7             | 20.8         | 10.0         | 26.0         |
| Segmen.          | 154.0   | 65.0    | 19.8 | 19.0             | 18.7         | 18.7         | 19.8         |
| Vehicle          | 228.0   | 91.1    | 110.2| 132.0            | 172.0        | 110.2        | 110.2        |
| Wine             | 56.1    | 12.0    | 5.2  | 5.2              | 5.2          | 5.2          | 5.2          |
| Average          | 135.3   | 193.6   | 54.4 | 43.9             | 40.6         | 47.2         | 56.0         |
| Rank             | 6       | 7       | 4    | 2                | 1            | 3            | 5            |

When taking into consideration the number of conjunctions and the number of rules, PART also shows the best results, producing the smallest rule sets.

In summary, FuzzyDT shows the smallest error rates, while PART with the features preselected by CFS induced the smallest rule sets. However, the error rates of most of the models induced by PART are not competitive with
the ones induced by FuzzyDT. Recall that we are interested in models with low syntactic complexity, as they are simple to be interpreted, but not at the cost of a high increment in the error rate.

Another aspect to be considered regarding the interpretability of the induced models is related to the splitting of continuous attributes performed by C4.5 and FuzzyDT. As discussed in detail in Section 5.1.3, page 67, one of the advantages of FuzzyDT over C4.5 is the fact that FuzzyDT granulates continues attributes using fuzzy sets described by linguistic terms which are highly understandable and handle imprecision and uncertainty. This way, no crisp or rough separations are generated when splitting the attributes during the process of the tree induction.

In order to perform a more thorough comparison between classic and fuzzy rule-based methods with similar approaches, we included five fuzzy and two classic methods in the experiments described next.

7.3 Comparing FuzzyDT, SLAVE, CHI, WF-C, Wang & Mendel, C4.5, and PART

The fuzzy methods included in these experiments are FuzzyDT, SLAVE, a genetic fuzzy approach, and CHI (Cordon et al., 1999), WF-C (Cordon et al., 1999), and Wang & Mendel, which generate fuzzy rule bases. For the classic approaches, we included the C4.5 and PART algorithms. FuzzyDT was selected for comparisons as a decision tree-based method. SLAVE was included as a baseline method for our experiments since it is a well-known genetic fuzzy approach, a widely researched area in the fuzzy community. The other fuzzy methods, CHI, WF-C, and Wang & Mendel were included due to their simplicity in generating fuzzy rule bases from data. The choice of the classic methods was based on the fact that both C4.5 and PART had already been included in previous comparisons, with competitive results.

7.3.1 Description of the Methods Selected for Comparison

The Wang & Mendel method is described in Section 3.2.1, page 28, the FuzzyDT method is described in Section 5.1.2, page 65, C4.5 is described in Section 5.1.1, page 64, and PART is described in Section 7.2.1, page 121. Next, we describe the remaining methods included in these experiments.

7.3.1.1 CHI

The CHI method (Cordon et al., 1999) generates a set of fuzzy rules from a dataset and determines a mapping between the feature space and the class
set. This method can be summarized by the following two steps:

1. The fuzzyfication of the feature space;
2. The generation of the fuzzy rules from data, similarly to the process employed by Wang & Mendel.

CHI implements the use of weights for the rules in the fuzzy rule base in order to improve the classification process.

7.3.1.2 WF-C

WF-C (Cordon et al., 1999) stands for Weighted Fuzzy Classifier. This method builds a classifier based on if-then fuzzy rules that allows the incorporation of weighted training patterns in order to adjust the sensitivity of the classification with respect to certain classes. The antecedent of the rules are defined by partitioning each attribute into fuzzy sets. The consequent of these rules and their degree of certainty are defined using the compatibility and weights of the training patterns. By giving different weights to different training patterns, the authors tackle an important issue regarding the misclassification of input patterns with high cost. An example of this situation is the diagnosis of diseases, such as cancer: diagnosing a person who has cancer as healthy can have stronger consequences than the opposite.

7.3.1.3 SLAVE

SLAVE (Gonzalez and Perez, 2001) is a genetic learning algorithm that uses the iterative approach to generate a FRB. In the iterative approach, chromosomes represent individual rules, and a single rule is selected at each iteration of the GA. The set of selected rules form the rule base of the model. SLAVE includes an embedded feature selection process. The pre-selection of attributes minimizes the problems caused by large search spaces, such as excessive execution time, while improving the interpretability of the generated models.

The main idea of SLAVE is to reduce the original problem of obtaining a complete set of rules to a simpler problem which consists in obtaining only one rule at a time. In this approach, each chromosome of the population represents a single rule, but only the best individual in each iteration is considered, the remaining chromosomes being discarded. In fact, in the iterative model, one execution of the GA provides a partial solution (a rule) to the learning problem.

Regarding the feature selection process adopted by SLAVE, it dynamically explores the set of possible variables in order to find the most useful rule and the most relevant variables for this rule. Thus, this feature selection process
is implemented for each single rule, not for the whole set of rules. The basic schema of this process consists of modifying the rule representation in the search mechanism of SLAVE in order to allow the learning algorithm to search not only for the best rule, but also the best set of variables for each rule. Similarly to CHI and WF-C, SLAVE also produces rules with different weights, which are used by the inference mechanism to improve the classification performance. SLAVE usually produces reasonably small rule sets.

### 7.3.2 Experimental Evaluation

The reason to compare such different approaches for the automatic generation of classifiers is to provide information and insight when selecting a classification method. This way, we considered the following aspects for the comparisons of the methods:

- Performance in terms of error rate;
- Interpretability of the generated models in terms of their syntactic complexity, taking into consideration the average number of rules generated and average number of conjunctions of the rule sets;
- Computational cost of these methods measured in terms of execution time.

As done previously, the experiments were carried out using 10 datasets from UCI (Frank and Asuncion, 2010) and a 10-fold cross-validation strategy, using the same datasets from the previous work, described in Table 7.5, page 122.

Notice that all fuzzy methods evaluated in this work used similar fuzzy partitions, i.e., the same number of fuzzy sets and their distribution, as well as the type of membership function.

Table 7.11 presents the error rates and standard deviation (in brackets) for each of the tested methods. The dark-gray shaded cells highlight the smallest error rates obtained for this setup. Notice that the results obtained by CHI, Wang & Mendel, and WF-C are not presented here due to the fact that they were quite poor and would interfere in the comparisons and analysis carried out. In fact, for several datasets CHI, Wang & Mendel and WF-C presented error rates comparable or even greater than the majority error.

FuzzyDT shows the smallest error rates for 6 out of 10 datasets followed by C4.5 which obtained the smallest error rates for 3 datasets with a technical tie with SLAVE for Diabetes. PART shows the smallest error rate for the remaining dataset, Segmentation.
Table 7.11: Error rates - further comparisons.

| Dataset    | ME       | FUZZYDT  | SLAVE    | C4.5     | PART     |
|------------|----------|----------|----------|----------|----------|
| Credit     | 45.33    | 11.48 (5.79) | 14.71 (2.75) | 14.03 (3.91) | 15.64 (4.31) |
| Cylinder   | 35.74    | 16.71 (4.01) | 28.52 (8.19) | 32.34 (8.41) | 31.72 (7.76) |
| Derma      | 68.99    | 6.06 (4.59) | 8.1 (4.08) | 4.66 (3.21) | 5.33 (3.48) |
| Diabetes   | 64.90    | 30.31 (4.95) | 25.66 (4.49) | 25.51 (5.37) | 26.55 (5.91) |
| Glass      | 65.46    | 23.02 (16.78) | 38.23 (10.74) | 32.35 (9.86) | 32.95 (8.42) |
| Heart      | 44.44    | 8.52 (6.83) | 20.74 (10.65) | 21.85 (7.42) | 22.67 (7.81) |
| Ionosphere | 35.90    | 7.8 (6.38) | 12.27 (5.07) | 10.26 (4.38) | 9.17 (4.66) |
| Segmentation | 85.71  | 12.38 (2.33) | 28.1 (6.53) | 12.67 (6.88) | 11.57 (7.3) |
| Vehicle    | 74.23    | 37 (4.56) | 39.59 (5.08) | 27.72 (4.32) | 27.79 (3.61) |
| Wine       | 59.74    | 4.64 (7.15) | 9.54 (7.43) | 8.79 (7.71) | 7.95 (7.44) |
| Average    | 55.04    | 16.99 (6.24) | 22.55 (6.5) | 19.04 (6.14) | 19.13 (6.03) |

To test whether there is a statistically significant difference among the error rates of the algorithms, the Friedman test (Demsar, 2006) was used, as previously. However, no statistically significant difference was found among the tested algorithms with a 95% confidence level.

In order to analyse the models generated by these methods, Table 7.12 presents their average number of rules (Rules) and syntactic complexity (SC). The dark-gray shaded cells highlight the smallest number of rules and smallest syntactic complexity values obtained for this setup. Notice that SLAVE produces rules that might include conjunctions of disjunctions, as detailed in Section 7.4.2. For this work, we only considered the number of conjunctions in the rules in order to calculate the syntactic complexity of the models produced by SLAVE. The smallest sets of rules are medium-gray shaded, while the smallest syntactic complexity are dark-gray shaded.

Table 7.12: Number of rules and syntactic complexity - further comparisons.

| Datasets     | FUZZYDT | SLAVE | C4.5 | PART |
|--------------|---------|-------|------|------|
|              | Rules   | SC    | Rules | SC    | Rules   | SC    | Rules | SC    |
| Credit       | 9.00    | 37.80 | 6.70  | 22.50 | 30.00   | 22.00 | 29.00 | 127.60 |
| Cylinder     | 38.00   | 391.40| 7.90  | 26.40 | 24.00   | 189.60| 22.00 | 83.60  |
| Derma        | 23.00   | 172.50| 12.20 | 30.50 | 8.00    | 35.20 | 9.00  | 26.10  |
| Diabetes     | 7.00    | 35.80 | 4.20  | 11.20 | 20.00   | 112.00| 13.00 | 37.70  |
| Glass        | 55.00   | 176.00| 11.90 | 34.80 | 28.00   | 179.20| 15.00 | 43.50  |
| Heart        | 6.00    | 19.80 | 8.30  | 26.80 | 18.00   | 88.20 | 23.00 | 64.40  |
| Ionosphere   | 19.00   | 96.90 | 15.10 | 51.10 | 18.00   | 120.60| 10.00 | 26.00  |
| Segmentation | 28.00   | 154.00| 11.00 | 35.80 | 13.00   | 65.00 | 11.00 | 19.80  |
| Vehicle      | 38.00   | 228.00| 21.70 | 101.60| 97.00   | 911.80| 29.00 | 110.20 |
| Wine         | 17.00   | 56.10 | 4.50  | 17.40 | 5.00    | 12.00 | 4.00  | 5.20   |
| Average      | 24.00   | 135.26| 10.35 | 23.40 | 26.10   | 193.56| 16.50 | 54.41  |

Regarding the number of generated rules, SLAVE presented the smallest values for 6 datasets with a tie with PART for Segmentation. PART presented the smallest number of rules for 3 datasets, while C4.5 and FUZZYDT presented the smallest number of rules for one dataset each.

Regarding the syntactic complexity of the generated models, SLAVE presented the smallest values for 5 datasets. PART presented the smallest values for 4 datasets, while FUZZYDT shows the smallest results for Heart. It is interesting to notice that, on average, PART presented the smallest number of
rules, but SLAVE, although with a higher average number of rules, presented the smallest syntactic complexity.

Table 7.13 presents the total execution time of FuzzyDT, SLAVE, C4.5, and PART, for all experiments with the 10 datasets and a 10-fold cross-validation setup. The computer used has an Intel® Core™2 Duo T7250 (2.00GHz, 2MB L2 Cache, 800MHz FSB).

Table 7.13: Execution time - further comparisons.

| Method   | Time       |
|----------|------------|
| FuzzyDT  | 00:00:58   |
| SLAVE    | 02:26:38   |
| C4.5     | 00:00:30   |
| PART     | 00:00:57   |

While FuzzyDT, C4.5, and PART took less than a minute to complete all executions, SLAVE took two hours and 26 minutes. The greater time taken by SLAVE is directly connected to the genetic search it uses. Nevertheless, the execution time of SLAVE can be considered reasonably low for a genetic approach.

In order to evaluate the performance of the methods in terms of error rates and syntactic complexity, the graphs in Figures 7.1, 7.2, 7.3, and 7.4 for the first four datasets of the experiments, using the normalized values of error rate (vertical axis) and syntactic complexity (horizontal axis) for each method and each dataset, were produced. These graphs exemplify the use of a visual alternative to evaluate the methods. Notice that the origins do not represent null error and null syntactic complexity. In fact, the origins are defined by the smallest error rate and syntactic complexity found among the results of the tested methods for each dataset. Similarly, (1,1) represents the maximum error rate and syntactic complexity found among the results obtained.

Using the normalized values, instead of the real ones, helps choosing the most appropriate algorithm for a specific dataset by focusing on the ones that are plotted closest to the origin of the graph. The idea is to discard the methods whose values are plotted farthest from the origin and just compare and analyse those closest to the origin in order to obtain the best compromise between error rate and syntactic complexity.

For the credit dataset, shown in Figure 7.1, FuzzyDT seems to be the natural choice since, although its syntactic complexity is slightly higher than SLAVE, it is reasonably low, while its error rate is the smallest (0% → 11.48), while SLAVE presents high syntactic complexity.

For the Dermatology dataset, shown in Figure 7.2, the best choice would include C4.5 and Part, which are close to the origin with low or minimum error rates and syntactic complexity.
Another interesting example is presented in Figure 7.3, which shows the results for the Diabetes dataset. SLAVE is isolated in the area near the origin of the graph.

Figure 7.4 shows the results for Heart. In this case, FuzzyDT is plotted exactly in the origin, making it the best option. It is interesting to notice that, differently from the other examples, for Heart, all the other algorithms are distant from FuzzyDT, making the choosing process visually quite easy.

The results show that, regarding the compared methods, FuzzyDT shows competitive error rates and execution time, presenting reasonably low syntactic complexity. SLAVE, on the other hand, produced models that can be considered simpler than the others, but its efficiency in classifying was limited and its execution time was high when compared to the other methods. C4.5 and PART both yielded good results in competitive time. The visual representation of the error rates and syntactic complexity is a useful tool in order to choose a compromise between accuracy and interpretability.
7.4 Comparing Genetic-based and Decision Tree-based Methods

With the proposal of the FCA-BASED method in (Cintra et al., 2012c), we included it in our comparisons between fuzzy and classic methods. Our experiments evaluated two groups of rule learning methods for classification:

- Methods based on decision trees;
- Methods based on genetic algorithms.

As already mentioned, genetic-based rule learning methods take advantage of the powerful search process of GAs, although the computational cost of this search is usually high. Decision trees, on the other hand, present low computational cost, producing highly interpretable models with competitive
accuracy. This way, methods based on Decision Trees (DTs), might be a good alternative to the genetic approaches.

For the decision tree-based methods, we included the ones that were considered in our previous comparisons: FUZZYDT, C4.5, and PART. For the genetic-based methods, we kept SLAVE and included FCA-BASED, and MPLCS (Bacardit and Krasnogor, 2009). MPLCS was selected due to the fact that it is a crisp genetic rule learning method.

Most of the methods included have been previously detailed (FUZZYDT: Section 5.1.2, page 65; C4.5: Section 5.1.1, page 64; PART: Section 7.2.1, page 121; FCA-BASED: Section 6.5.1, page 104; SLAVE: Section 7.3.1.3, page 127). The remaining method, MPLCS, is described next.

7.4.1 MPLCS

MPLCS (Bacardit and Krasnogor, 2009) stands for Memetic1 Pittsburgh Learning Classifier System (MPLCS). The MPLCS method has many variants according to the adopted local search mechanism. The version used in our experiments adopts the local search based on the rule set-wise operator, which has three main stages:

1. Evaluation of the candidate rules (the set of candidate rules is generated using the example set by sampling, with no replacement, instances from the training set and creating rules that are generalizations of those instances);

2. Selection of the rules that will form the offspring rule set;

3. Generation of the final individual.

In the first stage, all rules are evaluated with all the examples of the training set, producing a map of correct and incorrect classifications for each rule. The next stage uses this map to evaluate how much each candidate rule can contribute to improve the accuracy of the offspring rule set without re-evaluating the rule set.

It follows a detailed comparison of the models generated by all the included methods.

---

1Memetics is a theory of mental content based on an analogy with the Darwinian evolution. Memes are similar to genes in GA, but represent ideas, believes, patterns of behaviour, which can reproduce.
7.4.2 Comparing the Models Generated by the SLAVE, FCA-BASED, and MPLCS

As previously mentioned, an issue with the models generated by SLAVE is that they contain rules with sets of fuzzy label disjunctions in their antecedents. For example, one of the models generated for the Iris dataset, with three rules, is presented next.

1. IF \( X_2 \) is \([L_0]\) THEN Class is Iris-setosa (W 0.977)

2. IF \( X_0 \) is \([L_0 \text{ OR } L_1]\) AND \( X_2 \) is \([L_0 \text{ OR } L_1]\) AND \( X_3 \) is \([L_0 \text{ OR } L_1]\) THEN Class is Iris-versicolor (W 0.402)

3. IF \( X_0 \) is \([L_1 \text{ OR } L_2]\) AND \( X_2 \) is \([L_1 \text{ OR } L_2]\) AND \( X_3 \) is \([L_0 \text{ OR } L_2]\) THEN Class is Iris-virginica (W 0.719)

Although the model has only 3 rules, with a total of 7 conditions, 6 of these conditions contain disjunctions of fuzzy labels, impacting on the model interpretability. It is also possible to find the association of linguistic values that are not defined by neighbouring fuzzy sets (see rule 3, variable \( X_3 \), for instance, using the fuzzy sets \( L_0 \) and \( L_2 \), which are not neighbours) making the understanding of each individual rule difficult.

Similarly to SLAVE, the models generated by MPLCS, also contain conjunctions of disjunctions, and, for continuous attributes, the splitting points can be quite unnatural and difficult to be interpreted. As an example, the rule set for the Iris dataset, with 4 rules, is shown next.

1. IF \( \text{sepalLength} \) is \( > 6.243 \) AND \( \text{petalLength} \) is \( > 5.085 \) THEN Class is Iris-virginica

2. IF \( \text{sepalLength} \) is \( < 6.340 \) AND \( > 7.020 \) and \( \text{petalWidth} \) is \( > 1.627 \) THEN Class is Iris-virginica

3. IF \( \text{petalLength} \) is \( < 1.983 \) THEN Class is Iris-setosa

4. Default rule: Class is Iris-versicolor

The cutting points defined by the algorithm can be quite similar and close to each other. For instance, the splitting points for \( \text{sepalLength} \) in rules 1 and 2 discard values from a very close interval from 6.243 to 6.340, which makes the understanding of the whole model difficult.

Instead, the FCA-BASED models, contain only conjunctions of conditions in the antecedent of their rules, and, due to the fact that it is based on the fuzzy logic, the discretization of continuous attributes is done using highly interpretable linguistic valued fuzzy sets. The rule set generated by FCA-BASED for the iris dataset is presented next.

1. IF \( \text{sepalLength} \) is medium AND \( \text{petalLength} \) is medium AND \( \text{petalWidth} \) is medium THEN Class is Iris-virginica

2. IF \( \text{petalLength} \) is large AND \( \text{petalWidth} \) is medium THEN Class is Iris-versicolor

3. IF \( \text{petalLength} \) is small AND \( \text{petalWidth} \) is medium AND \( \text{sepalWidth} \) is small THEN Class is Iris-setosa

4. IF \( \text{petalLength} \) is medium AND \( \text{petalWidth} \) is small AND \( \text{sepalWidth} \) is large THEN Class is Iris-virginica
For experts in a domain, the fuzzy linguistic values *small*, *medium* and *large* are directly interpreted. Nevertheless, for those who are unfamiliar with the domain, it is necessary to check the FDB for the information regarding the number, type and distribution of the fuzzy sets defining each attribute, in order to interpret the rules and the rule set. This process, although quite straightforward, requires extra effort. To reduce the time to understand the FDB, the information can be presented graphically.

A detailed comparison of the models generated by C4.5, PART, and FuzzyDT can be found in Section 5.1.3, page 67.

### 7.4.3 Experimental Evaluation

As done previously, our experiments were carried out using the same 10 datasets (see Table 7.5, page 122) from UCI (Frank and Asuncion, 2010) used in the previous work, and a 10-fold cross-validation strategy. The KEEL framework (Bull et al., 2008) was used for the SLAVE, C4.5, PART, and MPLCS algorithms, all executed with default parameters, except for the number of fuzzy sets for SLAVE, which was set to 3. For FCA-BASED and FuzzyDT, we used our own implementations.

Similarly to previous comparisons, the performance of the models in terms of error rates and their interpretability, in terms of their syntactic complexity, which takes into consideration the average number of rules generated and the average number of conditions of these rules, were considered.

Notice that all fuzzy methods used the same (FCA-BASED and FuzzyDT) or similar (SLAVE) fuzzy partitions, *i.e.*, the same number of fuzzy sets and their distribution, as well as the type of membership function. Since the KEEL implementation of SLAVE does not allow the definition of distinct numbers of fuzzy sets for different attributes of a given dataset, the FuzzyDBD-II method was not adopted in order to perform the comparisons using similar fuzzy data bases. The FuzzyDBD method, on the other hand, was not used in our experiments since it defined, on average, large numbers of fuzzy sets for the datasets. As previously mentioned, fuzzy data bases defined by large numbers of fuzzy sets tend to produce a negative impact on the interpretability of the generated models, as well as increasing exponentially the computational cost of the experiments.

Table 7.14 presents the average error rates and standard deviation for the tested methods. The majority error (ME) is also presented (second column). The last rows present the average and final ranks for each approach. For the average and final ranks, when computing the error rank for each dataset, if two or more error measures are equal, the fractional strategy for assigning rankings was used, *i.e.*, they receive the same ranking numbers, which is
the mean of what they would have under ordinary rankings. The results for the DT-based approaches are presented in the first columns, and thus, the results for the GA-based approaches in the last columns. The smallest error rates obtained among all approaches are dark-gray shaded.

Table 7.14: Error rates - DT and GA based methods.

| Datasets   | ME          | FuzzyDT     | C4.5        | PART         | SLAVE       | FCA-BASED   | MPLCS       |
|------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|
| Credit     | 45.33       | 15.78 (6.68)| 12.09 (4.39)| 37.35 (13.54)| 37.38 (3.86)| 9.82 (3.90) | 11.02 (4.89) |
| Cylinder   | 35.74       | 34.20 (0.09)| 27.69 (10.95)| 31.32 (10.68)| 35.74 (0.96)| 25.77 (5.15)| 25.95 (10.85) |
| Diabetes   | 34.90       | 26.10 (0.05)| 22.89 (8.07) | 31.47 (10.49)| 25.66 (4.26)| 23.09 (2.30)| 23.67 (8.41)  |
| Glass      | 65.46       | 48.26 (8.12)| 27.63 (11.29)| 48.98 (17.50)| 38.25 (10.19)| 39.96 (5.21)| 29.39 (12.17) |
| Heart      | 44.44       | 21.85 (5.60)| 20.57 (8.32) | 38.15 (13.86)| 20.74 (10.10)| 19.72 (7.44)| 17.41 (8.61)  |
| Ion         | 35.90       | 13.53 (9.95)| 11.11 (4.51) | 20.52 (13.02)| 12.27 (4.81)| 11.76 (4.97)| 9.98 (4.66)   |
| Iris        | 66.67       | 8.00 (2.67) | 4.00 (5.33)  | 60.00 (20.00)| 4.67 (4.27)| 4.68 (6.33)| 2.67 (5.33)   |
| Segmentation| 85.71      | 20.48 (5.65)| 0.48 (0.45)  | 10.70 (4.88) | 13.43 (0.43)| 24.75 (7.39)| 0.69 (0.55)   |
| Vehicle    | 74.23       | 35.85 (4.05)| 24.33 (9.09) | 58.36 (19.96)| 39.59 (4.82)| 44.15 (4.73)| 27.04 (10.64) |
| Wine       | 50.74       | 12.86 (11.43)| 6.67 (6.94)  | 36.31 (14.51)| 9.54 (7.05)| 4.27 (4.59)| 5.00 (3.89)   |
| Average Rank| 4.9        | 2.3         | 5.6         | 3.3         | 3.0         | 1.9         |
| Final Rank  | 5           | 2           | 6           | 4           | 3           | 1           |

Considering only the DT-based approaches, C4.5 obtained the smallest error rates for all datasets. FuzzyDT presented smaller error rates than PART for 8 datasets. It should be observed that in most cases learning was very poor for PART, as well as for FuzzyDT using Cylinder, i.e., the error rate for these models is similar to the most naive learning algorithm that always predicts the most frequent class in the dataset (ME).

Considering only the GA-based approaches, MPLCS had the smallest error rates for 6 datasets while FCA-BASED for 4 datasets. Moreover, the error rate of the model generated by SLAVE for Cylinder is the ME, thus, there was no learning.

Comparing all methods, i.e., DT-based and GA-based approaches, C4.5 obtained the smallest error rates for 4 datasets, while FCA-BASED and MPLCS for three datasets each.

To test whether there was a statistically significant difference among the six algorithms we used the Friedman test (Demsar, 2006) with the null-hypothesis that the performance of all algorithms, assessed in terms of the error rates, was comparable. The Friedman test found there is no statistically significant difference among the tested algorithms with a 95% confidence level.

As discussed previously, some methods tend to present good error rates, but low interpretability, or vice-versa. This way, to analyse the interpretability of the generated models, Table 7.15 presents the average number of rules and the Syntactic Complexity (SC) of the models generated by the six analysed algorithms, as well as the average rank and final rank of their SC. The dark-gray shaded cells highlight the smallest syntactic complexity values obtained in both approaches. As before, notice that although the rules produced by SLAVE and MPLCS present conjunctions of disjunctions, Table 7.15 does not
consider the number of disjunctions in the rules of these models. The smallest syntactic complexity values found for each dataset are dark-gray shaded.

Table 7.15: Number of rules and syntactic complexity - DT and GA based methods.

| Datasets    | Approaches | DT-based approaches | GA-based approaches |
|-------------|------------|---------------------|---------------------|
|             |            | FuzzyDT | C4.5 | PART | SLAVE | FCA-BASED | MPLCS |
| Credit      | Rules      | SC      | Rules | SC   | Rules | Rules    | SC   |
| Cylinder    | 31.8       | 83.7    | 12.8  | 248.5| 1.3   | 2.5      | 11.9 |
| Diabetes    | 12.6       | 34.4    | 23.6  | 150.2| 1.0   | 11.9     | 5.1  |
| Glass       | 26.6       | 95.0    | 24.1  | 137.8| 1.4   | 10.1     | 18.6 |
| Heart       | 17.4       | 49.0    | 18.5  | 86.1 | 1.4   | 8.3      | 32.8 |
| Iono        | 20.2       | 54.4    | 13.9  | 72.4 | 2.4   | 15.1     | 10.5 |
| Heart       | 17.4       | 49.0    | 18.5  | 86.1 | 1.4   | 8.3      | 32.8 |
| Average Rank|            | 4.9     | 5.1   | 1.1  | 3.1   | 4.1      | 2.7  |
| Final Rank  |            | 5       | 6     | 1    | 3     | 4        | 2    |

As can be observed, PART produced the models with the smallest syntactic complexity for 9 of the datasets and SLAVE for the remaining one, which is the Cylinder dataset, having the ME as its error rate (Table 7.15). Thus, the model generated consists of only one rule assigning the majority class to any new instance. Moreover, PART was ranked last in accuracy (Table 7.14). On the other hand, although C4.5 is ranked last regarding the SC, it is ranked second for accuracy. Furthermore, MPLCS is ranked second regarding the SC, and it is ranked first for accuracy.

As before, in order to consider the performance of the methods both in terms of error rates and syntactic complexity, we used the normalized values of the error rate and SC to produce some graphs and visually analyse the results. To illustrate, Figures 7.5, 7.6, 7.7, and 7.8 present the results for Credit, Cylinder, Diabetes, and Glass, the first four datasets, respectively.

Figure 7.5: Credit - normalized error rate [9.8, 37.3] and syntactic complexity [11.2, 95.4].

Recall that, since the values are normalized, the origins of the graphs do
not represent null error and null syntactic complexity. Instead, the origins are defined by the smallest error rate and SC of the results of the tested methods for each dataset. Similarly, point (1, 1) represents the maximum error and syntactic complexity obtained on the dataset.

For the Credit dataset, shown in Figure 7.5, the FCA-BASED algorithm presents the smallest error rate and low syntactic complexity. The second best would be MPLCS. However, it is important to notice that the rules produced by MPLCS contain the disadvantage of being formed by conjunctions of disjunctions, while FCA-BASED produces more interpretable rules.

![Graph showing error rates and syntactic complexity for Credit dataset](image)

Figure 7.6: Cylinder - normalized error rate [25.8, 34.2] and syntactic complexity [1.0, 248.5].

For the Cylinder dataset, shown in Figure 7.6, FCA-BASED should be chosen. C4.5, although having low error rates, had the worst SC for these datasets. PART and SLAVE show poor error rates and good syntactic complexity.

![Graph showing error rates and syntactic complexity for Cylinder dataset](image)

Figure 7.7: Diabetes - normalized error rate [22.9, 31.5] and syntactic complexity [3.7, 150.2].

Similarly to Credit, for the Diabetes dataset, shown in Figure 7.7, the FCA-
BASED algorithm presents the smallest error rate and low syntactic complexity. MPLCS is the second best, but it produces less interpretable rules.

Figure 7.8: Glass - normalized error rate [27.0, 49.0] and syntactic complexity [10.1, 137.8].

For the Glass dataset, shown in Figure 7.8, MPLCS presents the best results. PART and SLAVE shows both good syntactic complexity, but poor error rates for this dataset.

7.5 Final Remarks

In this chapter we presented four sets of experiments, comparing methods from the classic and fuzzy machine learning communities. The methods selected for the comparisons, although proposed by different communities, share similar cores, such as decision trees, genetic algorithms, and rule-based methods.

The first set of experiments compared two fuzzy methods and 6 classic methods. After the execution of these experiments, we proposed FUZZYDT, we compared it to two classic methods in the second set of experiments. The comparisons with FUZZYDT were broadened to include a genetic fuzzy method in our third set of experiments. Finally, FCA-BASED, the last fuzzy classification proposal of this work, was included in a new set of experiments comparing decision tree-based methods and genetic-based methods for classification.

The comparisons included the accuracy of the models, in terms of error rates, their syntactic complexity, in terms of the number of conjunctions in the rule set, and, for one set of experiments, the execution time was also included. As future work, we intend to compare the FCA-BASED and FUZZYDT methods. We also intend to consider other fuzzy and classic methods in our comparisons, including the FURIA algorithm (Huhn and Hullermeier, 2009) that generates unordered fuzzy rules and other fuzzy decision tree algorithms.
Results show that FCA-BASED produces highly accurate models, with competitive syntactic complexity. C4.5 and FuzzyDT also obtained competitive results, both in terms of accuracy and syntactic complexity.
Conclusions and Future Research

In this work we investigated the automatic generation of fuzzy systems for classification problems and related tasks, such as feature subset selection, the definition of fuzzy data bases, and the definition of fuzzy rule bases using the genetic and the decision tree paradigms.

In this final chapter, we summarize the research contributions of this work, as well as discuss directions for future research.

8.1 Contributions

This work makes a number of contributions to the state of the art in the area of fuzzy systems for classification by proposing automatic methods to generate the knowledge base of fuzzy systems using different paradigms.

One of the main contributions of this work is the proposal of the FCA-BASED method for the genetic generation of fuzzy systems, which generates the genetic search space using the formal concept analysis theory to extract rules from data (Cintra et al., 2012c). The task of generating the genetic search space is investigated and published in (Majidian et al., 2011; Cintra et al., 2011d). Due to its robustness and the generation of highly interpretable rules, the FCA-BASED method was evaluated and compared to several methods (Cintra et al., 2012a,b).

The FCA-BASED method presents improvements to a method proposed in a previous work, the DoC-BASED method (Cintra et al., 2008c, 2010a). The improvements are related to the reduction of the computational cost for the generation of the genetic search space.

Moreover, we proposed the FUZZYDT method (Cintra and Camargo, 2010),
a fuzzy version of the classic C4.5 decision tree, presenting low computational cost and competitive accuracy. For this reason, FUZZYDT was selected as a baseline method for the evaluation of several other fuzzy classification methods (Cintra et al., 2010b, 2011c, 2012a), and as an estimation function for the FUZZYDBD-II method (Cintra et al., 2011e). FUZZYDT was also applied to a real-world problem, the prediction and control of the coffee rust disease in Brazilian crops (Cintra et al., 2011b).

Regarding the task of feature subset selection, we proposed the FUZZYWRAPPER method (Cintra et al., 2008a, 2009b), a wrapper-based approach able to select optimized subsets of features. Since FUZZYWRAPPER uses a fuzzy method as its base algorithm, it has the advantage, over non-fuzzy methods, of selecting features taking the relevant information regarding the fuzzyfication of the attributes into account in the process. Also, the exponential complexity involved in the search for subsets of features is tackled by a best-first search strategy, which makes the FUZZYWRAPPER suitable to be applied in the genetic generation of fuzzy systems (Cintra et al., 2012b).

We also proposed, in (Cintra and Camargo, 2010), the use of the FUZZYDT method as an embedded feature subset selection method, with promising results.

Furthermore, we presented a method for the automatic definition of fuzzy data bases, namely FUZZYDBD (Cintra et al., 2009a), based on related studies (Cintra et al., 2008b), which is able to automatically estimate the number of fuzzy sets defining the attributes of a dataset based on the Wang & Mendel method. FUZZYDBD uses the equalized universe method to distribute the triangular shaped fuzzy sets in the domains of the attributes. We also presented a modified version of this method, FUZZYDBD-II (Cintra et al., 2011e), which is able to define different numbers of fuzzy sets for each attribute of a dataset, by means of estimation functions.

Table 8.1 summarizes the methods proposed in this work, distributed according to the task they tackle.

| Task                                      | Proposed Methods                  |
|-------------------------------------------|-----------------------------------|
| The automatic definition of fuzzy data bases | FUZZYDBD and FUZZYDBD-II         |
| Fuzzy decision trees                      | FUZZYDT                           |
| Feature subset selection methods          | FUZZYWRAPPER and FUZZYDT          |
| The genetic generation of fuzzy rule bases | DOC-BASED and FCA-BASED           |

Table 8.1: Methods proposed in this work.

8.2 Future Research

Several research opportunities arise from this work, regarding improvements of the proposed approaches, as well as new ideas for their combination.
For the FuzzyDBD and FuzzyDBD-II methods that automatically generate fuzzy data bases, we intend to investigate the adjustment of the fuzzy sets defining each attribute by means of their lateral displacement, as well as changing their base, and even using different fuzzy set shapes. We understand that the task of tuning the fuzzy sets might improve the accuracy of the generated fuzzy rule bases. Specifically for the FuzzyDBD-II method, we intend to investigate the combination of the three numbers of fuzzy sets estimated by the three different estimation functions, instead of experimenting with the three numbers.

The use of the FuzzyDT method for the prediction and control of the coffee rust disease with crisp classes has motivated us to further investigate the problem with the definition of the class attribute as a linguistic variable. This way, it is possible to have three linguistic values for the class, including the three thresholds defined in the literature for the problem.

Furthermore, as an alternative to tackle the genetic generation of fuzzy systems for large datasets, we intend to investigate alternatives for the reduction of examples in the example sets, combined with the use of feature subset selection methods, in order to alleviate the process of fitness calculation.

Regarding the FCA-BASED method, we intend to investigate and propose a method for the reduction in the number of rules extracted using formal concept analysis. The strategy we intend to adopt involves the removal of similar examples from the training set, regarding the values of the attributes in the fuzzy sets defining each attribute. This reduction in the number of examples in the training set benefits the scaling of the method, since the number of rules extracted with it depends on both, the number of examples and the number of attributes in the training set.

For the reduction in the number of rules of the fuzzy rule bases generated by the FCA-BASED method, we intend to evaluate the use of a genetic algorithm to search for smaller subsets of rules using the fuzzy rule base produced by the FCA-BASED method as the genetic search space. We believe that the genetic algorithm might yield good results in reasonable time due to the reduced search space and the reduced size of the chromosomes. Our focus will be on the investigation of automatic methods to define the number of chromosomes and genes, maximum number of generations, as well as the rates for the genetic operators.

As previously stated, since the FuzzyDT method has low computational cost and is highly scalable, it was used in this work as baseline algorithm for a series of comparisons among classic and fuzzy classification methods. We also intend to compare the FuzzyDT method with the more recently proposed FCA-BASED method. Our goal is to analyse the gains in accuracy obtained by
the genetic search of FCA-BASED, in regards to its high computational cost, compared to the results of the FUZZYDT method.

Another important topic to be investigated in future work includes the more recent bio-inspired proposals for the automatic generation of fuzzy systems. We also intend to include some of these bio-inspired approaches in future comparisons with the FUZZYDT and FCA-BASED methods.
Abadeh, M. S., Hamid, M., and Habibi, J. (2011). Design and analysis of genetic fuzzy systems for intrusion detection in computer networks. *Expert Systems with Applications*, 38-6:7067–7075. (Cited on page 87)

Abraham, A. (2001). Neuro fuzzy systems: State-of-the-art modeling techniques. *Lecture Notes in Comp. Science*, 2084:269–276. (Cited on page 25)

Achiche, S. and Ahmed-Kristensena, S. (2011). Genetic fuzzy modeling of user perception of three-dimensional shapes. In *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, volume 25-1, pages 93–107. (Cited on pages 26 and 87)

Achiche, S., Baron, L., and Balazinski, M. (2003). Real/binary like coded genetic algorithm to automatically generate fuzzy knowledge bases. In *Proceedings of the 4th IEEE Int. Conf. on Control and Automation*, pages 799–803. (Cited on pages 26 and 87)

Acilar, A. M. and Arslan, A. (2011). Optimization of multiple input-output fuzzy membership functions using clonal selection algorithm. *Expert Systems with Applications*, 38:1374–1381. (Cited on pages 26 and 88)

Ahmadizar, F. and Soltanpanah, H. (2011). Reliability optimization of a series system with multiple-choice and budget constraints using an efficient ant colony approach. *Expert Systems with Applications*, 38-4:3640–3646. (Cited on pages 3 and 88)

Al-Aawar, N., Hijazí, T., and Arkadan, A. (2011). Particle swarm optimization of coupled electromechanical systems. *IEEE Transactions on Magnetics*, 47-5:1314–1317. (Cited on pages 3 and 88)

Alcalá, R., Gacto, M., and Herrera, F. (2011). A fast and scalable multi-objective genetic fuzzy system for linguistic fuzzy modeling in high-
dimensional regression problems. *Transactions on Fuzzy Systems Conference - FUZZ-IEEE*, 19-4:666–681. (Cited on pages 25, 85, and 88)

Alcalá, R., Alcalá-Fdez, J., Gacto, M., and Herrera, F. (2007). Genetic learning of membership functions for mining fuzzy association rules. In *IEEE International Conference on Fuzzy Systems*, pages 1538–1543. (2007). (Cited on page 26)

Alcalá, R., Nojima, Y., Herrera, F., and Ishibuchi, H. (2011). Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions. *Soft Computing*, 15(12):2303–2318. (Cited on page 85)

Aliev, R. A., Pedrycz, W., Guirimova, B. G., Aliev, R. R., Ilhan, U., Babagil, M., and Mammadlid, S. (2011). Type-2 fuzzy neural networks with fuzzy clustering and differential evolution optimization. *Information Sciences*, 181:1591–1608. (Cited on pages 2, 25, and 88)

Alonso, J. M. and Magdalena, L. (2009). An interpretability-guided modeling process for learning comprehensible fuzzy rule-based classifiers. *International Conference on Intelligent Systems Design and Applications*, pages 432–437. (Cited on page 48)

Arotaritei, D. (2011). Genetic algorithm for fuzzy neural networks using local crossover. *International Journal of Computers Communications & Control*, 6-1:8–20. (Cited on page 88)

Azzeh, M., Neagu, D., and Cowling, P. (2008). Improving analogy software effort estimation using fuzzy feature subset selection algorithm. In *Proceedings of the 4th international workshop on Predictor Models in Software Engineering*, pages 71–78. (Cited on page 48)

Bacardit, J. and Krasnogor, N. (2009). Performance and efficiency of memetic Pittsburgh learning classifier systems. *Evolutionary Computation*, 17:3:307–342. (Cited on page 133)

Bai, J., Jakeman, J., and Mcaleer, M. (1989). The effects of misspecification in estimating the percentiles of some two and three-parameter distributions. Technical report, Institute of Social and Economic Research, Osaka University. 14 pages. (Cited on page 30)

Baker, F. A., Verbyla, D. L., Hodges, C. S., and Ross, E. W. (1993). Classification and regression tree analysis for assessing hazard of pine mortality caused by *Heterobasidion annosum*. *Plant Disease*, 77:136–139. (Cited on page 71)
Boley, M., Grosskreutz, H., and Gärtner, T. (2010). Formal concept sampling for counting and threshold-free local pattern mining. In Proc. of the Int. Conf. on Data Mining (SDM 2010), pages 177–188. (Cited on page 102)

Bousahla, M., Kadri, B., and Bendimerad, F. T. (2010). Circular antenna array synthesis using fuzzy genetic algorithm. In International Review Of Electrical Engineering-IREE, pages 785–792. (Cited on page 87)

Breiman, L., Friedman, J. H., Olshen, R., and Stone, C. (1984). Classification and Regression Trees. Wadsworth and Brooks. (Cited on pages 47 and 63)

Bull, L., Bernadó-Mansilla, E., and Holmes, J. (2008). Learning Classifier Systems in Data Mining. Springer. (Cited on pages 46, 110, and 135)

Butt, D. J. and Royle, D. J. (1990). Multiple regression analysis in the epidemiology of plant diseases. In: Epidemics of plant diseases: mathematical analysis and modeling. Springer-Verlag. 2nd edition. (Cited on page 71)

Castro, P. A. D. and Camargo, H. A. (2004). Learning and optimization of fuzzy rule base by means of self adaptative genetic algorithms. IEEE International Conference on Fuzzy Systems, 2:1037–1042. (Cited on page 107)

Chalfoun, S. M. (1997). Coffee tree diseases: importance, identification and control methods. Federal University of Lavras, Brazil, FAEPE (Foundation for the Support of Teaching, Researching and Extension). (in Portuguese). (Cited on page 71)

Chen, C.-H., Hong, T.-P., and Tseng, V. S. (2009). An improved approach to find membership functions and multiple minimum supports in fuzzy data mining. Expert Sys. with Applications, 36:10016–10024. (Cited on page 26)

Chen, M. S. and Wang, S. W. (1999). Fuzzy clustering analysis for optimizing membership functions. Fuzzy Sets and Systems, 103:239–254. (Cited on pages 24, 25, 27, 30, and 54)

Cintra, M. E. and Camargo, H. A. (2007a). Fuzzy rules generation using genetic algorithms with self-adaptive selection. IEEE International Conference on Information Reuse and Integration - IRI, pages 261–266. (Cited on pages 5, 84, and 89)

Cintra, M. E. and Camargo, H. A. (2007b). Genetic generation of fuzzy rules with pre-selection of candidate rules. VI ENIA - Brazilian Meeting on Artificial Intelligence, 1:1341–1350. (in Portuguese). (Cited on page 89)
Cintra, M. E. and Camargo, H. A. (2010). Feature subset selection for fuzzy classification methods. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems - IPMU*, volume 80, pages 318–327. Springer. Communications in Computer and Information Science - Springer. *(Cited on pages 6, 49, 57, 61, 63, 64, 65, 84, 141, and 142)*

Cintra, M. E., Camargo, H. A., and Martin, T. P. (2009a). Optimising the fuzzy granulation of attribute domains. In *Int. Fuzzy Systems Association World Conference - IFSA 2009*, pages 742–747. *(Cited on pages 6, 27, 40, 54, and 142)*

Cintra, M. E., Camargo, H. A., and Monard, M. C. (2008a). Fuzzy feature subset selection using the Wang & Mendel method. In *Hybrid Intelligent Systems - HIS 2008*, volume 1, pages 590–595. *(Cited on pages 6, 47, 49, 54, 60, 84, and 142)*

Cintra, M. E., Camargo, H. A., and Monard, M. C. (2008b). A study on techniques for the automatic generation of membership functions for pattern recognition. In *C3N - Congress of the Trinational Academy of Sciences*, volume 1, pages 1–10. *(Cited on pages 6 and 142)*

Cintra, M. E., Camargo, H. A., R. Hruschka Jr, E., and Nicoletti, M. C. (2008c). Automatic construction of fuzzy rule bases: a further investigation into two alternative inductive approaches. *Journal of Universal Computer Science*, 14:2456–2470. *(Cited on pages 5 and 141)*

Cintra, M. E., Martin, T. P., Monard, M. C., and Camargo, H. A. (2009b). Feature subset selection using a fuzzy method. In *International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 214–217. *(Cited on pages 6, 49, 58, 61, 84, and 142)*

Cintra, M. E., Martin, T. P., Monard, M. C., and Camargo, H. A. (2011a). On the definition of an approach for the extraction of classification rules from fuzzy formal contexts. Technical report, Mathematics and Computer Science Institute - ICMC. *(Cited on pages 5, 81, and 98)*

Cintra, M. E., Meira, C. A. A., Monard, M. C., Camargo, H. A., and Rodrigues, L. H. A. (2011b). The use of fuzzy decision trees for coffee rust warning in Brazilian crop. In *11th International Conference on Intelligent Systems Design and Applications*, volume 1, pages 1347–1352. *(Cited on pages 6, 19, 70, 71, and 142)*

Cintra, M. E., Monard, M. C., and Camargo, H. A. (2010a). Data base definition and feature selection for the genetic generation of fuzzy rule bases. *Evolving Systems*, 1-4:241–252. *(Cited on pages 5, 89, 93, and 141)*
Cintra, M. E., Monard, M. C., and Camargo, H. A. (2010b). An evaluation of rule-based classification models induced by a fuzzy method and two classic learning algorithms. In *The Brazilian Symposium on Artificial Neural network (SBRN)*, volume 1, pages 188–193. (Cited on pages 6 and 142)

Cintra, M. E., Monard, M. C., and Camargo, H. A. (2011c). Fuzzy and classic rule learning methods: a comparative analysis. In *World Conference on Soft Computing, 2011*, volume 1, pages 182–190. (Cited on pages 6 and 142)

Cintra, M. E., Monard, M. C., and Camargo, H. A. (2012a). *On Rule Learning Methods: A Comparative Analysis of Classic and Fuzzy Approaches*. Soft Computing: State of the Art Theory and Novel Applications. (submitted for publication). (Cited on pages 5, 6, 60, 141, and 142)

Cintra, M. E., Monard, M. C., and Camargo, H. A. (2012b). On the automatic generation of genetic fuzzy systems. *Applied Soft Computing*. (to be submitted). (Cited on pages 5, 6, 141, and 142)

Cintra, M. E., Monard, M. C., Camargo, H. A., and Martin, T. P. (2012c). Using fuzzy formal concepts in the genetic generation of fuzzy systems. In *2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2012)*. (in print). (Cited on pages 5, 84, 109, 132, and 141)

Cintra, M. E., Monard, M. C., Camargo, H. A., Martin, T. P., and Majidian, A. (2011d). On rule generation approaches for genetic fuzzy systems. In *ENIA - Brazilian Meeting on Artificial Intelligence*, volume 1, pages 263–274. (Cited on pages 5 and 141)

Cintra, M. E., Monard, M. C., Chermen, E. A., and de Arruda Camargo, H. (2011e). On the estimation of the number of fuzzy sets for fuzzy rule-based classification systems. In *11th International Conference on Hybrid Intelligent Systems (HIS)*, volume 1, pages 211–216. (2011). (Cited on pages 6, 37, and 142)

Cordón, O., Alcalá, R., Alcalá-Fdez, J., and Rojas, I. (2007). Guest editorial genetic fuzzy systems: What’s next? an introduction to the special section. *IEEE T. Fuzzy Systems*, 15(4):533–535. (Cited on page 85)

Cordon, O., del Jesus, M. J., and Herrera, F. (1999). A proposal on reasoning methods in fuzzy rule-based classification systems. *International Journal of Approximate Reasoning*, 20(1):21–45. (Cited on pages 20, 126, and 127)

Cordon, O., Gomide, F. A. C., Herrera, F., Hoffmann, F., and Magdalena, L. (2004). Ten years of genetic fuzzy systems: Current framework and new trends. *Fuzzy Sets and Systems*, 141(1):5–31. (Cited on page 82)
Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. *Proceedings of the 18th Int. Conf. on Machine Learning*, pages 74–81. (Cited on page 46)

Demsar, J. (2006). Statistical comparison of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30. (Cited on pages 35, 43, 52, 56, 60, 75, 96, 113, 120, 123, 129, and 136)

Domingos, P. and Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130. (Cited on page 118)

Dou, W., Ren, Y., Wu, Q., Ruan, S., Chen, Y., Bloyet, D., and Constans, J. M. (2007). Fuzzy kappa for the agreement measure of fuzzy classifications. *Neurocomputing*, 70(4-6):726–734. (Cited on page 26)

Ducange, P., Lazzerini, B., and Marcelloni, F. (2010). Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets. *Soft Computing*, 14:7:713–728. (Cited on page 86)

Evsukoff, A. G., Galichet, S., de Lima, B. S., and Ebecken, N. F. F. (2009). Design of interpretable fuzzy rule-based classifiers using spectral analysis with structure and parameters optimization. *Fuzzy Sets and Systems*, 160:857–881. (Cited on page 86)

Frank, A. and Asuncion, A. (2010). UCI machine learning repository. (Cited on pages 34, 39, 41, 48, 50, 57, 66, 93, 109, 117, 119, 128, and 135)

Frank, E. and Witten, I. H. (1998). Generating accurate rule sets without global optimization. In ICML ’98: *Proceedings of the 15th Int. Conf. on Machine Learning*, pages 144–151. Morgan Kaufmann. (Cited on page 121)

Ganter, B. (2002). Formal concept analysis: Algorithmic aspects. Technical report, Technische Universitat Dresden. 52 pages. (Cited on pages 100 and 101)

Garcon, C. L. P., Zambolim, L., Mizubuti, E. S. G., Vale, F. X. R., and Costa, H. (2004). Coffee leaf rust control based on rust severity values. *Fitopatologia Brasileira*, 29:486–491. (in Portuguese). (Cited on page 71)

Gonzalez, A. and Perez, R. (2001). Selection of relevant features in a fuzzy genetic learning algorithm. *IEEE Transactions on Systems and Man and and Cybernetics Part B: Cybernetics*, 31(3):417–425. (Cited on page 127)
Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class. In Proceedings of the 17th International Conference on Machine Learning, pages 359–366. Morgan Kaufmann. (Cited on pages 46, 51, and 54)

Hayashi, Y., Setiono, R., and Yoshida, K. (2000). A comparison between two neural network rule extraction techniques for the diagnosis of hepatobiliary disorders. Artificial Intelligence in Medicine, 20:205–216. (Cited on page 119)

Haykin, S. (1998). Neural Networks: A Comprehensive Foundation. Macmillan College Publishing Company, 2 edition. (Cited on page 118)

Herrera, F. (2008). Genetic fuzzy systems: taxonomy, current research trends and prospects. Evolutionary Intelligence, 1(1):27–46. (Cited on pages 2, 82, and 88)

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. Machine Learning, 11(1):63–90. (Cited on pages 31 and 118)

Hong, T. P., Chen, C. H., Wu, Y. L., and Lee, Y. C. (2006). A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions. Soft Computing, 10(11):1091–1101. (Cited on page 25)

Hu, Q., Zhu, P., Liu, J., Yang, Y., and Yu, D. (2010). Feature selection via maximizing fuzzy dependency. Fundamental Informaticae, 98:167–181. (Cited on page 47)

Huhn, J. C. and Hullermeier, E. (2009). Furia: an algorithm for unordered fuzzy rule induction. Data Min. Knowl. Discov., 19(3):293–319. (Cited on page 139)

Ichiba, D., Hara, K., and Kanoh, H. (2006). Spatial interpolation of traffic data by genetic fuzzy system. International Symposium on Evolving Fuzzy Systems, 1:280–285. (Cited on page 19)

Ishibuchi, H., Nozaki, K., and Tanaka, H. (1992). Distributed representation of fuzzy rules and its application to pattern classification. Fuzzy Sets and Systems, 52-1:21–32. (Cited on pages 84 and 87)

Ishibuchi, H. and Yamamoto, T. (2004). Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. Fuzzy Sets and Systems, 141:59–88. (Cited on pages 84 and 86)
Ishibuchi, H., Yamamoto, T., and Nakashima, T. (2001). Fuzzy data mining: Effect of fuzzy discretization. In *IEEE Int. Conf. on Data Mining - ICDM 2001*, pages 241–248, USA. IEEE Computer Society.  (Cited on page 26)

Jang, S. R., Sun, C. T., and Mizutani, E. (1997). *Neuro-fuzzy and Soft computing*. Prentice Hall.  (Cited on page 13)

Japiassu, L. B., Garcia, A. W. R., Miguel, A. E., Carvalho, C. H. S., Ferreira, R. A., Padilha, L., and Matiello, J. B. (2007). Effect of crop load, tree density and weather conditions on the development of the coffee leaf rust. *Brazilian Coffee Research Symposium*, pages 1–5. 5th edition.  (Cited on page 71)

Jesus, M., Gámez, J., González, P., and Puerta, J. (2011). On the discovery of association rules by means of evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.  (Cited on page 86)

Kaya, M. and Alhajj, R. (2003). A clustering algorithm with genetically optimized membership functions for fuzzy association rules mining. *12th IEEE International Conference on Fuzzy Systems*, 2:881–886.  (Cited on page 24)

Kazunor, H., motohide, U., Hiroshi, S., and Yuushi, U. (2001). Fuzzy C4.5 for generating fuzzy decision trees and its improvement. *Faji Shisutemu Shinpojitumu Koen Ron bunshu*, 15:515–518. (in Japanese).  (Cited on pages 47 and 63)

Keller, J., Gray, M., and Jr., J. G. (1985). A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(4):580–585.  (Cited on page 25)

Kira, K. and Rendell, L. (1992). A practical approach to feature selection. *International Conference on Machine Learning*, 1:368–377.  (Cited on pages 46, 51, and 54)

Klir, G. J. and Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic - Theory and Applications*. Prentice-Hall.  (Cited on pages 9, 14, and 19)

Kononenko, I. (1994). Estimating attributes: Analysis and extension of RELIEF. *Proc. of European Conference on Machine Learning - Springer Verlag*, 784:171–182.  (Cited on page 40)

Krajca, P., Outrata, J., and Vychodil, V. (2010). Parallel algorithm for computing fixpoints of galois connections. *Annals of Mathematics and Artificial Intelligence*, 59:257–272.  (Cited on page 100)
Kushalappa, A. C., Akutsu, M., and Ludwig, A. (1983). Application of survival ratio for monocyclic process of *Hemileia vastatrix* in predicting coffee rust infection rates. *Brazilian Phytopathology*, 73:96–103. (Cited on page 70)

Kushalappa, A. C., Akutsu, M., Oseguera, S. H., Chaves, G. M., and Melles, C. (1984). Equations for predicting the rate of coffee rust development based on net survival ratio for monocyclic process of *Hemileia vastatrix*. *Brazilian Phytopathology*, 9:255–271. (Cited on pages 70 and 72)

Larsen, P. M. (1980). Industrial applications of fuzzy logic control. *International Journal of Man-Machine Studies*, 12:3–10. (Cited on page 17)

Lee, C.-H., Li, C.-T., and Chang, F.-Y. (2011). A species-based improved electromagnetism-like mechanism algorithm for tsk-type interval-valued neural fuzzy system optimization. *Fuzzy Sets and Systems*, 171-1:22–43. (Cited on page 88)

Lee, H., ming Chen, C., ming Chen, J., and lu Jou, Y. (2001). An efficient fuzzy classifier with feature selection based on fuzzy entropy. *IEEE Transactions on Systems, Man, and Cybernetics*, 31:426–432. (Cited on page 48)

Lee, H. D., Monard, M. C., Voltolini, R., Prati, R. C., and WU, F. C. (2006). A simple evaluation model for feature subset selection algorithms. *Artificial Intelligence*, 10:9–17. (Cited on page 52)

Liao, T. W., Celmins, A. K., and Hammell, R. J. (2003). A fuzzy c-means variant for the generation of fuzzy term sets. *Fuzzy Sets and Systems*, 135:241–257. (Cited on page 25)

Lindig, C. (1995). Concept-based component retrieval. In J. Kohler, F., Giunchiglia, C. G., and Walther, C., editors, *Working Notes of the IJCAI-95 Workshop: Formal Approaches to the Reuse of Plans, Proofs, and Programs*, pages 21–25. (1995). (Cited on page 104)

Liu, H. and Setiono, R. (1996). A probabilistic approach to feature selection - a filter solution. *Proceedings of the 13th International Conference on Machine Learning*, 1:319–327. (Cited on pages 46, 51, and 54)

Lukasiewicz, J. (1970). Philosophical remarks on many-valued systems of Propositional Logic. *In Studies in Logic and the Foundation of Mathematics*, pages 153–178. (Cited on page 17)

Majidjian, A., Martin, T., and Cintra, M. E. (2011). Fuzzy formal concept analysis and algorithm. In *Proceedings of the 11th Annual Workshop on Computational Intelligence - UKCI*, pages 1–7. (Cited on pages 5 and 141)
Mamdani, E. H. (1977). Application of fuzzy logic to approximate reasoning using linguistic systems. *Fuzzy Sets and Systems*, 26:1182–1191. *(Cited on pages 17 and 19)*

Mamdani, E. H. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):135–147. *(Cited on pages 17 and 19)*

Marinaki, M., Marinakis, Y., and Stavroulakis, G. E. (2011). Fuzzy control optimized by a multi-objective particle swarm optimization algorithm for vibration suppression of smart structures. *Structural and Multidisciplinary Optimization*, 43-1:29–42. *(Cited on page 88)*

Medasani, S., Kim, J., and Krishnapuram, R. (1998). An overview of membership function generation techniques for pattern recognition. *International Journal of Approximate Reasoning*, 19:391–417. *(Cited on page 25)*

Meira, C. A. A. (2008). *Knowledge discovery process in databases for analysis and warning of crop diseases and its application on coffee rust*. PhD thesis, University of Campinas, Brazil. *(in Portuguese). (Cited on page 72)*

Meira, C. A. A., Rodrigues, L. H. A., and Moraes, S. A. (2008). Analysis of coffee leaf rust epidemics with decision tree. *Tropical Plant Pathology*, 33:114–124. *(in Portuguese). (Cited on pages 71 and 76)*

Meira, C. A. A., Rodrigues, L. H. A., and Moraes, S. A. (2009). Warning models for coffee rust control in growing areas with large fruit load. *Brazilian Agricultural Research*, 44:233–242. *(in Portuguese). (Cited on pages 71 and 72)*

Mikut, R., Jakel, J., and Groll, L. (2005). Interpretability issues in data-based learning of fuzzy systems. *Fuzzy Sets and Systems*, 150-2:179–197. *(Cited on page 3)*

Milaré, C. R., Batista, G. E., Carvalho, A. C., and Monard, M. C. (2004). Applying genetic and symbolic learning algorithms to extract rules from artificial neural networks. In *Proceedings of the Mexican International Conference on Artificial Intelligence - MICAI 2004*, volume 2972, page 833–834. *(Cited on page 119)*

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill. *(Cited on page 118)*

Navale, R. L. and Nelson, R. M. (2010). Use of genetic algorithms to develop an adaptive fuzzy logic controller for a cooling coil. *Energy and Buildings*, 42-5:708–716. *(Cited on pages 19 and 87)*
Nieradka, G. and Butkiewicz, B. (2007). A method for automatic membership function estimation based on fuzzy measures. In *International Fuzzy Systems Association World Congress - IFSA 2007*, volume 4529, pages 451–460. (Cited on page 26)

Niknam, T., Mojarrad, H. D., and Meymand, H. Z. (2011). A novel hybrid particle swarm optimization for economic dispatch with valve-point loading effects. *Energy Conversion and Management*, 52-4:1800–1809. (Cited on pages 19 and 88)

Olaru, C. and Wehenkel, L. (2003). A complete fuzzy decision tree technique. *Fuzzy Sets and Systems*, 138(2):221–254. (Cited on pages 47 and 63)

Paul, P. A. and Munkvold, G. P. (2004). A model-based approach to preplanting risk assessment for gray leaf spot of maize. *Phytopathology*, 94:1350–1357. (Cited on page 71)

Pedrycz, W. (1998). *Computational Intelligence: An Introduction*. CRC Press. (Cited on page 9)

Pedrycz, W. and Gomide, F. (1998). *An Introduction to Fuzzy Sets*. MIT Press. (Cited on pages 9, 16, 17, 18, and 19)

Pinto, A. C. S., Pozza, E. A., Souza, P. E., Pozza, A. A. A., Talamini, V., Boldini, J. M., and Santos, F. S. (2002). Description of epidemics of coffee rust with neural network. *Brazilian Phytopathology*, 27:517–524. (in Portuguese). (Cited on page 70)

Pousinho, H., Mendes, V., and Catalão, J. (2011). A hybrid pso-anfis approach for short-term wind power prediction in portugal. *Energy Conversion and Management*, 52-1:397–402. (Cited on pages 3 and 88)

Prakash, A. and Deshmukh, S. (2011). A multi-criteria customer allocation problem in supply chain environment: An artificial immune system with fuzzy logic controller based approach. *Expert Systems with Applications*, 38-4:3199–3208. (Cited on pages 3 and 88)

Prati, R. C., Batista, G. E., and Monard, M. C. (2011). A survey on graphical methods for classification predictive performance evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 23(11):1601–1618. (Cited on page 86)

Prati, R. C., Monard, M. C., and Carvallho, A. C. P. L. F. (2004). Looking for exceptions on knowledge rules induced from HIV cleavage data set. *Genetics and Molecular Biology*, 27(4):637–643. (Cited on page 75)
Pulkkinen, P. and Koivisto, H. (2010). A dynamically constrained multiobjective genetic fuzzy system for regression problems. *IEEE Transactions on Fuzzy Systems*, 18:161–177. (Cited on pages 25 and 29)

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1:81–106. Reprinted in Shavlik and Dieterich (eds.) Readings in Machine Learning. (Cited on pages 47 and 63)

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, 1 edition. (Cited on pages 39, 47, 48, 51, 54, 57, 63, and 64)

Quinlan, J. R. (1996). Bagging, Boosting and C4.5. In *Proc. of the 13th Conf. on Artificial Intelligence*, pages 725–730. (Cited on pages 47 and 63)

Sánchez, L. and Couso, I. (2010). Obtaining fuzzy rules from interval censored data with genetic algorithms and a random sets-based semantic of the linguistic labels. *Soft Computing*, 15-10:1945–1957. (Cited on page 86)

Sanz, J. A., Fernández, A., Bustince, H., and Herrera, F. (2010). Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning. *Information Sciences*, 180:3674–3685. (Cited on page 26)

Sánchez, L., Suárez, M. R., Villar, J. R., and Couso, I. (2008). Mutual information-based feature selection and fuzzy discretization of vague data. *International Journal of Approximate Reasoning*, 49 (3):607–622. (Cited on page 48)

Takagi, T. and Sugeno, M. (1983). Derivation of fuzzy control rules from human operator’s control actions. In *Proceedings of the IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis*, pages 55–60. (Cited on page 19)

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132. (Cited on page 20)

Wang, L. (2003). The WM method completed: a flexible fuzzy system approach to data mining. *IEEE Transactions on Fuzzy Systems*, 11:768–782. (Cited on pages 29, 40, and 49)

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1:80–83. (Cited on page 75)
Wille, R. (1982). Restructuring lattice theory: an approach based on hierarchies of concepts. In Rival, I., editor, *Ordered Sets*, pages 445–470. Boston Publisher.  
(Cited on pages 97 and 98)

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition.  
(Cited on pages 46, 51, 57, 72, 94, 118, and 119)

Wolff, K. E. (2002). Concepts in fuzzy scaling theory: order and granularity. *Fuzzy Sets and Systems*, 132(1):63–75.  
(Cited on page 100)

Wu, N., Li, X., Yang, J., and Liu, P. (2007). Improved clustering approach based on fuzzy feature selection. In *2007 International Conference on Service Systems and Service Management*, pages 1–5.  
(Cited on page 48)

Yu, L. and Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224.  
(Cited on page 45)

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8:338–353.  
(Cited on pages 2 and 9)

Zadeh, L. (1973). Outline of a new approach to the analysis of complex systems and decision process. *IEEE Transactions on Systems, Man and Cybernetics*, 3:28–44.  
(Cited on page 19)

Zambolim, L., Vale, F. X. R., Costa, H., Pereira, A. A., and Chaves, G. M. (2002). *The state-of-the-art of the coffee production technologies. Epidemiology and integrated control of the coffee rust*. Suprema. (in Portuguese).  
(Cited on pages 70 and 77)

Zambolim, L., Vale, F. X. R., Pereira, A. A., and Chaves, G. M. (1997). *Coffee (Coffea arabica L.): diseases control - diseases caused by fungi, bacterium, viruses. In: Plant diseases control: large cultures*. volume 1. Federal University of Varginha - UFV. (in Portuguese).  
(Cited on pages 70 and 72)

Zhang, L. and Zhang, B. (2005). The structure analysis of fuzzy sets. *International Journal of Approximate Reasoning*, 40:92–108.  
(Cited on page 25)

Zhou, Z. (2004). Rule extraction: Using neural networks or for neural networks. *Journal of Computer Science and Technology*, 19:249–253.  
(Cited on page 119)

Zimmermann, H. J. (1991). *Fuzzy Set Theory and its Applications*. Kluwer Academic Publishers.  
(Cited on page 13)