Rough Cognitive Networks

Acknowledgements

The picture of the hermit mathematician isolated in the woods is increasingly obsolete and unrealistic. In my opinion, every professional achievement comprises a conjunction of efforts where supervisors, colleagues, friends and family play a paramount role. I was never alone during this three-year journey. I feel blissful to have had the kind support of many people and now it is time to thank them.

In the first place, I cannot find the proper words (even in Spanish) to express my sincere gratitude to my supervisors for their amazing guidance. Prof. dr. Koen Vanhoof, Prof. dr. Rafael Bello and Prof. dr. Bart Kuijpers are examples of straightness and flexibility, genius, diplomacy, professional ethics and open mind. They never had to mention how respected they are to receive my admiration.

About Prof. dr. Koen Vanhoof. The first time I came to Hasselt someone told me: “If you have a problem and Koen cannot solve it, then do not worry, the problem has no solution”. I found in each bureaucratic paperwork, visa application or research-related issue the opportunity to check the veracity of that sentence.

About Prof. dr. Bart Kuijpers. I am still impressed by his exhaustive comments toward improving the mathematical notation of this thesis!

About Prof. dr. Rafael Bello. I am in the academia because of Bello and his moral principles. Bello taught me that academia is not always fair and the importance of working hard with the ethics in research as a flag.

Equally important has been the support of my closest collaborators and personal friends: Isel, Maikel, Falcon and Frank. The secret behind our team is that there is no place for status; a senior member of IEEE contributes the same as a PhD student and we all have the same probability to be nicely criticized.

Likewise, I would like to thank to Prof. dr. Elpiniki Papageorgiou and Prof. dr. Wojciech Froelich for their support in the last years.

I would additionally like to thank to all my colleagues from our research group: Ahmad, Hamzah, Lia, Aziz, Farahnaz, Benoît, Mieke, Marleen, Gert, Toon, Mathijs, Jonas and Mernush (my lunch partner). A special thanks goes to Marijke, Niels and Hanne for their kind guidance in times of high stress. Ivett and Marilyn, thank you for your emotional support in this last period.

I also received a lot of support from the three women in my life: my mom Miriam, my grandma Juana and my almost-wife Isel (yes, the one who have been collaborating with me from the very beginning). They celebrated each accepted paper as the most important achievement of the entire world. But they also suffered the issues inherent to the academia. I have special words for my partner in life:

My love, you are the most patient, smart, funny, sweet and sensible person I know. You were by my side in every step of this journey providing me the composure I often need. I could not have done it without your support.

I received the same support from the rest of my family and friends; they trusted on me even when the overall evidence indicated otherwise :)

I cannot finish my acknowledgements without thanking to my family-in-law. Libia, Ricardo, Edilio, Magaly and Eddy welcomed me as another member of the family and supported me in every difficult decision I had to face.

Gonzalo Nápoles
Hasselt, May 2017

Contents

List of Tables

List of Figures

Preface

It is undeniable the impact of *Artificial Intelligence* (AI) in our modern, highly technological society. In practice, AI is not about conscious, omnipotent or even reasoning machines, but about computational models that produce near-optimal solutions for very complex problems. Computer vision, natural language processing, anomaly detection, autonomous systems, digital image analysis, affective computing and other fields have made immense advances in the past few years.

Pattern classification is perhaps the most popular field within AI as a result of its link with real-world problems. In short, it may be defined as the process of identifying the right category (among those in a predefined set) to which an observation belongs. The ease with which we recognize our black cat from hundreds similar to it, read handwritten characters or decide whether a banana is ripe only by its smell belies the astoundingly complex processes that underlie these scenarios. That is why researchers have developed a wide range of classification algorithms called *classifiers* with the goal of facing these situations with the best possible accuracy.

Regrettably, most accurate classification models do not provide any mechanism to explain how they arrived at each conclusion and behave like *black-boxes*. This means that their reasoning mechanism is not transparent, therefore negatively affecting their practical usability in scenarios where understanding the decision process is required. As an example, in the context of Decision Support Systems, the experts are assisted by a computer program that should elucidate its reasoning mechanism; otherwise the experts have to trust on results that they cannot understand.

The purpose of this thesis is to develop a transparent classifier as an alternative to black-box models. This suggests that our algorithm should be capable of computing high-quality prediction rates when compared to traditional classifiers, and providing an introspection mechanism into its decision process.

In order to accomplish our research goal, this thesis introduces the notion of rough cognitive mapping in the context of pattern classification. The proposed methodology comprises three well-defined steps that are materialized through the *Rough Cognitive Networks*. In the first step, we discover information granules on the available information using the *Rough Set Theory*. This mathematical theory computes three disjoint regions for each decision class with a precise meaning for the classification problem. In the second step, we build a *Fuzzy Cognitive Map* where input neurons represent the previously discovered granules, while output ones denote the decision classes to be considered. The last step focuses on performing the neural reasoning process using intelligible inclusion equations and causal relations.

Moreover, we introduce two extensions to deal with the parametric requirement of building information granules in presence of numerical variables: the *Rough Cognitive Ensembles* and the *Fuzzy-Rough Cognitive Networks*. The first model uses an ensemble of Rough Cognitive Networks, each performing at a different granularity degree. The latter replaces the crisp information granules with fuzzy ones, thus providing further flexibility. While both models perform similarly in terms of prediction rates, the latter is preferred since it is simpler and more transparent. The numerical simulations using a wide variety of synthetic datasets have shown that our model performs comparably with regards to most successful classification algorithms.

The main advantage of our classifiers (referred to as Rough Cognitive Networks in general) relies on their *transparency* as they allow understanding the decision process at a granular level while yielding high prediction rates. Therefore, this methodology becomes an alternative to the limitation on the expression and architecture of Fuzzy Cognitive Maps to handle pattern classification tasks, even when the problem domain cannot be interpreted. Likewise, the notion of rough cognitive mapping opens new research avenues toward solving more complex classification problems in which each observation may be associated with multiple decisions.

Chapter 1

Introduction

Pattern classification (Duda et al., 2012) is one of the most ubiquitous real-world problems and certainly one at which humans really excel. It consists of identifying the right decision class (among those in a predefined set) to which an observation belongs. These objects are described by a set of predictive attributes $\Psi = \{\psi_1, \psi_2, \ldots, \psi_M\}$ of numerical and/or nominal nature. More formally, the pattern classification problem (Duda et al., 2012) is about building a mapping $f : \mathcal{U} \rightarrow \mathcal{D}$ that assigns to each object of the universe of discourse $\mathcal{U}$ a decision class D_k from the K possible ones in $\mathcal{D} = \{D_1, D_2, \ldots, D_K\}$. The mapping is frequently learned in a supervised fashion, that is, by relying on an existing set of previously labeled examples, which is used to train a classification model. More complex classification problems include learning in presence of class imbalance (Sun et al., 2009) (López et al., 2013), class noise (Frénay and Verleysen, 2014), semi-supervised scenarios (Chapelle et al., 2010) (Cohen et al., 2004) or multiple decision classes per object (Tsoumakas and Katakis, 2007) (Cheng et al., 2010) (Nápoles et al., 2016a), among other scenarios.

The literature on classification models (henceforth simply called "*classifiers*") is vast and offers a myriad of techniques that approach the classification problem from multiple angles. Decision trees (Quinlan, 1986), rule-based models (Ishibuchi et al., 1999), k-nearest neighbors learners (Cover and Hart, 1967), neural networks (Zhang, 2000), ensemble models (Dietterich, 2000), Bayesian networks (Friedman et al., 1997) or support vector machines (Hearst et al., 1998) stand among the most popular classifiers, each having its own advantages and limitations. This motivates the scientific community to put forth new (or modified) algorithms to cover a wide spectrum of classification tasks coming from real-world problems.

1.1 Motivation and challenges

Due to the inherent complexity of real-world classification problems, researchers in Statistics and Machine Learning have developed a wide range of models (hereinafter referred to as classifiers) with the goal of achieving the best possible accuracy. Some classifiers like Artificial Neural Networks, Support Vector Machines, Ensemble Techniques or Random Forests are well-known to be the most likely successful algorithms for addressing a real-world problem in terms of prediction rates (Witten and Frank, 2005). Regrettably, most accurate algorithms do not provide any mechanism to explain how they arrived at a particular conclusion and behave like “black-boxes”. This means that their reasoning mechanism is not *transparent*. The research community has then resorted to *Granular Computing* (Pedrycz, 2001) (Bargiela and Pedrycz, 2012) in order to develop classifiers that acquire, process and interpret the problem data at the level of (more symbolic) information granules.

Granular classifiers and, more generally, *granular systems* (Al-Hmouz et al., 2014) (Balamash et al., 2015) encompass those complex, intelligent methods that lean on information granularity at their core in order to deal with general, regularly vague and imprecise specifications (Szczuka et al., 2015). Such methods are good representatives of Computational Intelligence techniques given their adherence to the Computational Intelligence underpinnings, i.e., the exploitation of tolerance, imprecision and partial truth in order to achieve tractability, robustness and resemblance with human-centric decision making (Kacprzyk and Pedrycz, 2015). Such features provide a strong framework to design classification models able to deal with highly inconsistent scenarios where similar observations could lead to quite different outcomes.

On the other hand, in recent years, Fuzzy Cognitive Maps (FCMs) have become a suitable knowledge-based methodology for modeling and simulating complex systems (Kosko, 1986). FCM-based systems can be understood as *interpretable recurrent neural networks* (Tsadiras, 2008) (Kreinovich and Stylios, 2015) (Nápoles et al., 2016), comprising a collection of processing entities called concepts or simply neurons, which are connected by signed and weighted arrows. Concepts denote variables, objects, entities or states describing the system under investigation, whereas the edges denote causal relations between these neural processing entities.

Despite the advantages of using FCMs for modeling dynamic systems, their application in solving pattern classification tasks has been less studied. This is motivated by their poor prediction rates when compared to other classifiers in more generic scenarios (Papakostas and Koulouriotis, 2010). However, sometimes, FCM-based models perform well in specific domains, even using very simple architectures. Kreinovich and Stylios (Kreinovich and Stylios, 2015) conjectured about the empirical success of FCMs by considering that human's subjective opinions follow *Miller's seven plus minus two law* (Miller, 1956). Another likely explanation for this interesting behavior is that the performance of FCM-based classifiers is subject to their ability to represent and generalize the domain knowledge reflected in the historical data. But, could this issue be effectively faced using Granular Computing models?

1.2 Granular cognitive mapping

Within the Granular Computing, the information granulation concerns the processing of complex information entities called *granules*, which arise in the process of data abstraction and derivation of knowledge from information. Thus, an information granule can be defined as a collection of objects sharing a specific property. Several methods have been augmented with different types of information granules in order to enhance their performance. FCMs are one of the more recent models that have benefited from the interplay with Granular Computing. The ensuing models are captured under the term *Granular Cognitive Maps* (GCMs) and provide several advantages including the high-level interpretability of the physical system under analysis.

Pedrycz and Homenda (Pedrycz and Homenda, 2014) evoke the allocation of information granularity as a pivotal driving force behind the development of these types of granular structures and describe five concise protocols as its realization mechanisms. Pedrycz (Pedrycz, 2010) and his collaborators (Pedrycz et al., 2016) put forth a granular representation of time series in which FCM concepts (i.e., neurons) denote cluster prototypes induced by the well-known fuzzy c-means algorithm over the space of amplitude and change of amplitude of the signal.

Homenda et al. (Homenda et al., 2014) adopted numeric intervals as the granulation vehicle for their GCM weight matrix. More precisely, each causal weight is no longer a number but an interval. The authors elaborate on three methodologies for building a GCM from scratch by maintaining an adequate balance between specificity and generality in the design of the interval-based FCM weights and the ensuing map operations. The numerical simulations showed that the resulting granular model had a good degree of coverage without a loss in precision.

Inspired on the approaches discussed in (Pedrycz, 2010) and (Pedrycz et al., 2016), we proposed a partitive GCM model to solve graded multi-label classification problems (Nápoles et al., 2016a) instead of time series prediction. Three different FCM topologies were studied and several convergence features were included into the learning scheme driven by Particle Swarm Optimization. Numerical experiments confirmed the capability of these GCMs to accurately estimate the degree of association between an object and each decision class using synthetic datasets.

The above granular models certainly illustrate the advantages of using Granular Computing and cognitive mapping to deal with different problems, which range from time series forecasting to graded multi-label classification scenarios. However, their application in solving standard pattern classification problems has been less investigated. On the other hand, these approaches fail in producing truly causal high-level models. This is a result of using stochastic search methods to determine the sign and intensity of causal relations defining the semantics between the discovered information granules. Determining these weights in a comprehensive, accurate and theoretically sound way remains an open problem (Felix et al., 2017).

1.3 Scope and research goals

The goal of this research is *to put forth a transparent GCM-based classifier capable of elucidating its reasoning mechanism*, as an alternative to black-boxes. This suggests that our algorithm should be capable of computing high-quality prediction rates when compared to traditional classifiers, and providing an introspection mechanism into its decision process. In this thesis, the term *transparency* refers to the classifier's ability to explain its reasoning mechanism, whereas *interpretability* refers to the classifier's ability to explain the problem domain at the attribute level.

This general objective can be divided into several research goals, each comprising interesting challenges that range from the theoretical contributions to the exhaustive empirical evaluation. Such goals can be formalized as follows:

1. To develop a transparent GCM-based classifier without requiring the intervention of experts to define the network topology.

2. To overcome the parametric requirement related to the information granulation stage through further modifications to the proposed classifier.

3. To evaluate the prediction capability of the GCM-based classifiers using different configurations, operators and distance functions.

4. To compare the prediction capability of the GCM-based classifiers against state-of-the-art methods across benchmark problems.

1.4 Technical roadmap

In order to address the above research goals, we introduce the notion of *rough cognitive mapping* in the pattern classification context. The proposed granular methodology comprises three steps, namely 1) the information granulation, 2) the network design, and 3) the network exploitation for unlabeled objects.

In this research, we employ Rough Set Theory (Pawlak, 1982) to discover the information constructs that will be used to build the granular classifier. Using rough sets as a vehicle to granulate the input space allows making decisions using both the certain and the hesitant information. More explicitly, we use extended rough sets to replace the equivalence classes with similarity ones, thus equipping the model with the capability of dealing with mixed-attributes objects. On the other hand, determining a suitable heterogeneous distance function to measure dissimilarity between objects is a pivotal issue to achieve good prediction rates.

The second step is concerned with building the network topology by using the Three-way Decision Rules (Yao, 2009). In the proposed scheme, information granules are denoted as input neurons in the GCM-based network, while output neurons represent decision classes. The well-defined semantics of rough granules seems suitable to define the causal relations characterizing the network topology, without requiring neither the expert intervention nor a further learning stage.

In the third step, we provide an inference model to exploit the granular classifier for unlabeled (new) objects. In principle, this is achieved by quantifying the inclusion degree of a similarity class into each information granule.

Determining the granularity degree leading to high prediction rates is a fascinating challenge that may be addressed in different ways. An intuitive solution for this issue is to estimate the value of this parameter using a search method; however, this strategy may become non-practical for large datasets since it requires rebuilding the information granules. In order to overcome the parametric requirement the proposed classifier in a more elegant way, this thesis presents two additional models: the *Rough Cognitive Ensembles* and the *Fuzzy-Rough Cognitive Networks.*

The reader can observe that the *Fuzzy Cognitive Maps*, the *Extended Rough Sets* and the *Three-way Decision Rules* are the building blocks supporting the proposed granular classifier. These theoretical resources will be opportunely revised in subsequent chapters to ensure a coherent readability. Likewise, the parametric requirement, the effect of using different distance functions and the comparison against state-of-the-art classifiers are issues to be discussed in this thesis. Finally, we characterize the problems on which our algorithm stands as the best choice.

1.5 Main contributions

The *Rough Cognitive Networks* comprise four contributions, namely: 1) the automatic construction of high-level cognitive networks from historical data, therefore freeing up the classifier from the subjectivity of human intervention, 2) the exploitation of negative, positive and hesitant information during the reasoning stage, 3) the transparency on the inference process, and 4) the model refinement by suppressing the parametric requirement in the information granulation stage, leading to a free-parameter classifier. Such theoretical contributions are detailed as follows:

1. Rough cognitive mapping becomes a suitable alternative to deal with the limitation on expression and architecture of cognitive mapping. Being more explicit, it is well-known that FCM-based models are problem-dependent since domain experts must define the network topology (i.e., concepts and causal relations). However, the granular approach proposed in this research allows automatically constructing the network structure from historical data, thus suppressing the need for human intervention in the construction phase.

2. Most traditional classifiers derive their decision models based on the available positive information, without taking into account the negative or hesitant knowledge. Nevertheless, using rough sets to granulate the information space allows exploiting the positive, negative and hesitant information in order to improve the prediction rates. This seems convenient to solve classification problems on which “quite similar” situations lead to different outcomes.

3. Unlike black-box classifiers, the proposed granular classifiers allow explaining their decision process at a high-level using inclusion degree equations and causal relations. In point of fact, the lack of *transparency* is the key drawback of most successful black-boxes (e.g., Random Forests, Multilayer Perceptron, Support Vector Machines). The model transparency however does not necessarily ensure that we can *interpret* the semantics behind the physical system under analysis. Even so, the transparency (often called interpretability at the model level) is a necessary condition to build truly interpretable classifiers.

4. As a last contribution, we propose two approaches to overcome the parametric requirement during the information granulation step. Therefore, we obtain two parameterless granular classifiers that allow solving mixed-attribute classification problems while preserving the prediction rates.

1.6 Thesis organization

The rest of this thesis is organized as follows. Chapter 2 is devoted to the first building block of rough cognitive algorithms: the *Rough Set Theory*. It describes the rough set models based on both equivalence and similarity relations, the role of the distance function and the derivation of rough classification rules. Chapter 3 presents the *Fuzzy Cognitive Mapping* as the second building block of the proposed classification models. This chapter goes over the foundations of such structures, their reasoning process and behavior, the transfer functions, among other aspects.

Chapter 4 introduces the notion of *rough cognitive mapping* in the context of pattern recognition, which refers to the automatic construction of FCM-based classifiers from rough information granules. More explicitly, this chapter describes the stages related to the information granulation, the automatic construction of the underlying FCM topology from rough constructs and the classifier exploitation. The convergence properties of the inferred causal network is also discussed.

Chapters 5 and 6 propose two approaches to deal with the parametric requirement coming from the information granulation stage. The first model, called *Rough Cognitive Ensembles*, uses a bootstrap aggregation scheme in an attempt to suppress the dependence of user-specified parameters. The second model, called *Fuzzy-Rough Cognitive Networks*, is more refined since it replaces the crisp granules with fuzzy ones, thus hyperparameter learning is no longer required.

Chapters 7 introduces the statistical analysis and its ensuing discussion. As a first simulation, we illustrate how the proposed classifier works once the network has been activated. Moreover, we study the role of the distance function and fuzzy operators over the prediction rates. To conclude, we carry out an extensive comparative study against state-of-the-art classifiers across benchmark datasets.

Chapters 8 outlines the concluding remarks and future research directions to be accomplished as a future work. Furthermore, Appendix A illustrates how the proposed classifier operates by using the Iris datatset as an example, Appendix B outlines the properties of benchmark problems used in our experiments, while Appendix C, D and E provide the full results achieved during simulations.

Chapter 2

Rough Set Theory

This chapter is devoted to the theoretical foundations of *rough sets* and the derived rough classification rules (e.g., three-way decisions). These notions comprise the core of the granular cognitive classifiers introduced in this research.

2.1 Rough sets based on equivalence relations

The Rough Set Theory (Pawlak, 1982) is a methodology proposed in the early 1980's for handling uncertainty that is manifested in the form of inconsistent data (Bello et al., 2008). Let $DS = (\mathcal{U}, \Psi \cup \{d\})$ denote a decision system where $\mathcal{U}$ is a non-empty finite set of objects called the *universe*, Ψ is a non-empty finite set of attributes describing any object in $\mathcal{U}$ and $d \notin \Psi$ represents the decision attribute. Any subset $X \subseteq \mathcal{U}$ can be approximated by two crisp sets, which are referred to as its *lower* and *upper approximations* and denoted by $\underline{\Phi}X = \{x \in \mathcal{U} \mid [x]_\Phi \subseteq X\}$ and $\overline{\Phi}X = \{x \in \mathcal{U} \mid [x]_\Phi \cap X \neq \emptyset\}$, respectively. In this classic formulation, the equivalence class $[x]_\Phi$ comprises the set of objects in $\mathcal{U}$ that are deemed inseparable from x according to the information contained in the attribute subset $\Phi \subseteq \Psi$.

The lower and upper approximations are the basis for computing the positive, negative and boundary regions of any set X. The *positive region* $POS(X) = \underline{\Phi}X$ includes those objects that are certainly related to X; the *negative region* $NEG(X) = \mathcal{U} - \overline{\Phi}X$ denotes those objects that are certainly not related to X, while the *boundary region* $BND(X) = \overline{\Phi}X \setminus \underline{\Phi}X$ captures the objects whose membership to the set X is uncertain, but they might be members of X. These three regions are information granules that will be used to design our granular classifiers.

Figure 2.1 portrays how to compute the lower and upper approximations for an arbitrary set X, which regularly is comprised of those objects belonging to a particular decision class. In this example, each rectangle represents an equivalence class, thus generating a partition of the universe of discourse.

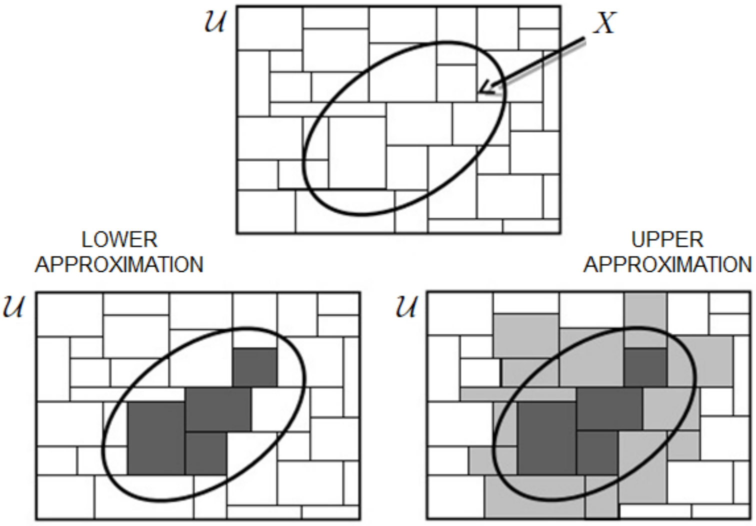


Figure 2.1: Approximation space using equivalence relations.

In most RST problems, $\underline{\Phi}X \subseteq \overline{\Phi}X$ but if $\underline{\Phi}X = \overline{\Phi}X$ then the boundary region will be empty, which means that the set is crisp. Of course, whether $\underline{\Phi}X = \overline{\Phi}X$ will strongly depend on the subset of attributes Φ used to compare the objects comprised into the universe of discourse. This set often denotes a *reduct* that may be determined either by experts or using an attribute selection method.

2.2 Rough sets based on similarity relations

In the classical RST formalism, two objects are deemed indiscernible if they have identical values for the selected attributes. This definition works well with nominal attributes but is not suitable for numerical ones, as negligible differences between two numerical ones could cause two nearly identical objects to lie in two distinct inseparability classes. To relax this stringent assumption, the equivalence requirement on the inseparability relation is replaced with a similarity relation.

Equation (2.1) shows an indiscernibility relation, where $0 \leq \varphi(x, y) \leq 1$ is a similarity function. This weaker binary relation claims that two objects x and y are inseparable as long as their similarity degree $\varphi(x, y)$ goes above a similarity threshold $0 \leq \xi \leq 1$. This user-specified parameter establishes the degree of granularity in the granular space. Determining the precise granularity degree becomes a central issue when designing high-performing rough classifiers.

$$R : xRy \Longleftrightarrow \varphi(x, y) \geq \xi \tag{2.1}$$

The similarity function could be formulated in a variety of ways. In this research, we assume that $\varphi(x, y) = 1 - \delta(x, y)$, where $0 \leq \delta(x, y) \leq 1$ is a function comprising the distance between objects x and y. In the next section, we describe there heterogeneous functions (Wilson and Martinez, 1997) that allow quantifying the dissimilarity degree between objects having either numerical and/or nominal attributes.

Once we have determined the set of objects with a similarity degree greater than the similarity threshold, we can determine the lower and upper approximations for the kth decision class. If the object x is only related with objects belonging to the kth decision class, then x will be included in the kth lower approximation. In contrast, if the object x is related with objects that belong to other decision classes, then the object will included in the kth upper approximation.

It should be mentioned that a similarity relation does not necessarily induce a *partition* of $\mathcal{U}$ into a set of equivalence classes but rather a *covering* of $\mathcal{U}$ into multiple similarity classes. This implies that an object could belong to several similarity classes at the same time, thus introducing some flexibility degree. However, this also suggests that the decision-making process is no longer straightforward.

2.3 Heterogeneous distance functions

As mentioned, the distance function plays a pivotal role when designing the similarity relation. In this section, we outline the mathematical formulation of three extensively used distance functions taken from (Wilson and Martinez, 1997) that allow comparing objects comprising both numerical and nominal attributes.

Let $\Phi = \{\phi_1, \ldots, \phi_M\}$ be the attribute set, where ϕ_j can be either numerical or nominal, and it attaches a weight $0 \leq \omega_j \leq 1$ that quantifies its relevance. Assuming that numerical attributes have been normalized, the distance between two objects x and y can be computed using one of the following functions:

- **The Heterogeneous Euclidean-Overlap Metric (HEOM)**. This function computes the normalized Euclidean distance between numerical attributes and an overlap metric for nominal ones. Equations (2.2) and (2.3) define the HEOM distance function, where $x(j)$ and $y(j)$ denote the values for the jth attribute in the heterogeneous objects x and y, respectively.

$$\delta_{HEOM}(x,y) = \sqrt{\frac{\sum_{j=1}^{M} \omega_j \sigma_j(x,y)}{\sum_{j=1}^{M} \omega_j}} \tag{2.2}$$

where

$$\sigma_j(x,y) = \begin{cases} 0 & \text{if } \phi_j \text{ is nominal } \wedge x(j) = y(j) \\ 1 & \text{if } \phi_j \text{ is nominal } \wedge x(j) \neq y(j) \\ (x(j) - y(j))^2 & \text{if } \phi_j \text{ is numerical} \end{cases} \tag{2.3}$$

- **The Heterogeneous Manhattan-Overlap Metric (HMOM)**. This variant is similar to the HEOM function since it replaces the Euclidean distance with the Manhattan distance when computing the dissimilarity between two numerical values. Equations (2.4) and (2.5) display the HMOM function, whose calculation requires less computational effort compared to HEOM's.

$$\delta_{HMOM}(x,y) = \frac{\sum_{j=1}^{M} \omega_j \rho_j(x,y)}{\sum_{j=1}^{M} \omega_j} \tag{2.4}$$

where

$$\rho_j(x,y) = \begin{cases} 0 & \text{if } \phi_j \text{ is nominal } \wedge x(j) = y(j) \\ 1 & \text{if } \phi_j \text{ is nominal } \wedge x(j) \neq y(j) \\ |x(j) - y(j)| & \text{if } \phi_j \text{ is numerical} \end{cases} \tag{2.5}$$

- **The Heterogeneous Value Difference Metric (HVDM)**. This function involves a stronger strategy for quantifying the dissimilarity between discrete attribute values. Instead of computing the matching rate, it measures the correlation between attributes and decision classes. Equations (2.6) and (2.7) show the HVDM function variant adopted in this research.

$$\delta_{HVDM}(x,y) = \sqrt{\frac{\sum_{j=1}^{M} \omega_j \tau_j(x,y)}{\sum_{j=1}^{M} \omega_j}} \tag{2.6}$$

where

$$\tau_j(x,y) = \begin{cases} \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\beta_{\phi_j,x(j),k}}{\beta_{\phi_j,x(j)}} - \frac{\beta_{\phi_j,y(j),k}}{\beta_{\phi_j,y(j)}}\right)^2 & \text{if } \phi_j \text{ is nominal} \\ (x(j) - y(j))^2 & \text{if } \phi_j \text{ is numerical} \end{cases} \tag{2.7}$$

and K is the number of decision classes, $\beta_{\phi_j,x(j)}$ denotes the number of objects in the training set for which $\phi_j = x(j)$, whereas $\beta_{\phi_j,x(j),k}$ is the number of objects that have output class k and additionally $\phi_j = x(j)$. The reader can observe that we could use a matching approach in conjunction with the correlation strategy in those cases where a nominal attribute has the same values in both objects.

In the above distance functions, estimating the relevance of each attribute may result in improved prediction rates. To overcome this issue, we compute the *gain ratio* (Quinlan, 1986) associated with the jth attribute:

$$G(\phi_j, X) = \frac{I(X) - E(\phi_j, X)}{IC(\phi_j)} \tag{2.8}$$

where

$$I(X) = -\sum_{k=1}^{K} \frac{|X \cap D_k|}{|X|} log_2 \frac{|X \cap D_k|}{|X|} \tag{2.9}$$

measures the randomness of the distribution of available objects in X over K decision classes, whereas $E(\phi_j, X)$ is defined as follows:

$$E(\phi_j, X) = \sum_{l=1}^{L_j} \frac{|X_l|}{|X|} I(X_l) \tag{2.10}$$

where L_j represents the number of possible values for the jth attribute and X_l denotes the object set in X having value v_l for the jth attribute. Likewise, in order to counter this metric's bias in favor of attributes with a larger number of values, the following entropy-based normalization factor is introduced:

$$IC(\phi_j) = -\sum_{l=1}^{L_j} \frac{|X_l|}{|X|} log_2 \frac{|X_l|}{|X|} \tag{2.11}$$

The gain ratio associated to the jth attribute replaces the weight ω_j when computing the dissimilarity degree between two heterogeneous objects. Of course, we can use more sophisticated measures as the distance between partitions, although it could significantly increase the computational cost.

2.4 Deriving classification rules

The positive, negative and boundary regions can be effectively employed to derive classification rules when facing decision-making problems.

For instance, in (Grzymala-Busse, 1988) the author defined two categories of rules: certain rules from the lower approximations and possible rules from the upper approximations. Nevertheless, since the lower approximation is in fact a subset of the upper approximation, there is an overlap between these rules (Yao, 2010) leading to confusing interpretations for the decision model. In (Wong and Ziarko, 1986) the authors proposed two types of decision rules: deterministic decision rules for positive regions and nondeterministic decision rules for boundary regions. Since the three regions are mutually exclusive, the derived rule sets no longer have an overlap (Yao, 2010) and consequently, a higher degree of transparency is achieved.

Likewise, one may associate probabilistic measures, such as accuracy and confidence to derived rules (Tsumoto, 2002) where the accuracy and confidence of a deterministic rule is 1, whereas for nondeterministic rules they can take values between 0 and 1. In (Pawlak, 2002) the author referred to them as certain and uncertain decision rules, respectively. In point of fact, Pawlak focused on the positive region and certain rules (Pawlak, 1992) since they characterize objects on which we can make confident decisions, so they ensure highest consistency when selecting the decision. Despite this fact, uncertain rules often comprise relevant information that could be used in order to improve the accuracy of the decision/classification model.

As an alternative, Yao introduced the three-way decision model (Yao, 2009). Rules constructed from rough regions are associated to different actions. A *positive rule* suggests a decision of acceptance, a *negative rule* implies a decision of rejection, whereas a *boundary rule* advocates for abstaining. The three-way rules play a central role in decision-making problems in the sense that experts usually make a decision based on available knowledge and evidence. If the evidence is insufficient, then they cannot make a positive or negative decision, instead experts could make a non-commitment decision (e.g., in the discussion of policies at the United Nations sessions, member can abstain if they are not convinced about the proposal).

2.5 Three-way decision rules

Let $X \subseteq \mathcal{U}$ denote the set to be approximated. The notion of three-way decision rules (Yao, 2009) is inspired by the three disjoint regions produced by the lower and upper approximations. Such exact sets divide the universe $\mathcal{U}$ into three regions, the positive region $POS(X)$, the boundary region $BND(X)$, and the negative region $NEG(X)$. Based on these regions we can definitely affirm that any element $x \in POS(X)$ belongs to X, and that an object $x \in NEG(X)$ does not belong to the X set. These principles are the basis of the three-way decision rules:

- $Des([x]_\Phi) \rightarrow_P Des(X)$, for $[x]_\Phi \subseteq POS(X)$
- $Des([x]_\Phi) \rightarrow_B Des(X)$, for $[x]_\Phi \subseteq BND(X)$
- $Des([x]_\Phi) \rightarrow_N Des(X)$, for $[x]_\Phi \subseteq NEG(X)$

In these rules, $Des([x]_\Phi)$ denotes the logic formula defining the equivalence class $[x]_\Phi$ and $Des(X)$ is the name of the concept. If $[x]_\Phi \subseteq POS(X)$, we accept x to be a member of the target concept X. If $[x]_\Phi \subseteq NEG(X)$, we reject x to be a member of X. If $[x]_\Phi \subseteq BND(X)$, we neither accept nor reject x to be a member of X, instead we make a decision of deferment, abstaining or non-commitment (Yao, 2011). These decision rules recognize and model our inability to make a definite acceptance or rejection decision in scenarios with weak information.

Within the classic rough set model (i.e., based on equivalence relations) the above rules are unnecessarily restricted, but one can generalize them to revoke these limitations. Hence, the Decision-theoretic Rough Set (DTRS) model (Yao and Zhou, 2016) emerged as a general probabilistic approach that uses two states and three actions to characterize the decision process. The states refer to the pertinence of an object to a set, whereas the three actions are the three-way decisions.

Let $\Omega = X, \sim X$ be the set of states indicating that an object is enclosed in the set X or not, respectively. Let $\Pi = \{\pi_P, \pi_N, \pi_B\}$ denote the set of actions, where π_P, π_N, and π_B are the three decision actions, that is, deciding $x \in POS(X)$, deciding $x \in NEG(X)$, and deciding $x \in BND(X)$, respectively. In this approach, the probabilities $Pr(X|[x]_\Phi)$ and $Pr(\sim X|[x]_\Phi)$ denote the probability that the equivalence class $[x]_\Phi$ belongs to the set X and the set $\sim X$, respectively. Let $\lambda_{iP}(\pi_i|X)$ and $\lambda_{iN}(\pi_i|\sim X)$, $\forall i = \{P, N, B\}$, denote the loss for taking the action π_i when the state is X and $\sim X$. Accordingly, the expected loss $\mathcal{L}(\pi_i|[x]_\Phi)$ associated with taking each individual action can be expressed as follows:

- $\mathcal{L}_P = \mathcal{L}(\pi_P|[x]_\Phi) = \lambda_{PP} Pr(X|[x]_\Phi) + \lambda_{PN} Pr(\sim X|[x]_\Phi)$
- $\mathcal{L}_N = \mathcal{L}(\pi_N|[x]_\Phi) = \lambda_{NP} Pr(X|[x]_\Phi) + \lambda_{NN} Pr(\sim X|[x]_\Phi)$
- $\mathcal{L}_B = \mathcal{L}(\pi_B|[x]_\Phi) = \lambda_{BP} Pr(X|[x]_\Phi) + \lambda_{BN} Pr(\sim X|[x]_\Phi)$

Likewise, the Bayesian decision procedure (Yao and Zhou, 2016) leads to minimum-risk decisions, that are summarizes in the following rules:

- $(P1)$ IF $\mathcal{L}_P \leq \mathcal{L}_N$ AND $\mathcal{L}_P \leq \mathcal{L}_B$ THEN $x \in POS(X)$
- $(N1)$ IF $\mathcal{L}_N \leq \mathcal{L}_P$ AND $\mathcal{L}_N \leq \mathcal{L}_B$ THEN $x \in NEG(X)$
- $(B1)$ IF $\mathcal{L}_B \leq \mathcal{L}_P$ AND $\mathcal{L}_B \leq \mathcal{L}_N$ THEN $x \in BND(X)$

Since $Pr(X|[x]_\Phi) + Pr(\sim X|[x]_\Phi) = 1$, we can simplify the above rules based on the probability $Pr(X|[x]_\Phi)$ and the loss function. Being more explicit, let us consider a reasonable cost function where $\lambda_{PP} \leq \lambda_{BP} < \lambda_{NP}$ and $\lambda_{NN} \leq \lambda_{BN} < \lambda_{PN}$, then the cost of classifying an object x belonging to X into the positive region is less than or equal to the cost of classifying x into the boundary region, and these costs are strictly less than the cost of classifying x into the negative region. The minimum-risk probabilistic decision rules can be expressed as follows:

- $(P2)$ IF $Pr(X|[x]_\Phi) \geq \alpha$ AND $Pr(X|[x]_\Phi) \geq \gamma$ THEN $x \in POS(X)$
- $(N2)$ IF $Pr(X|[x]_\Phi) \leq \beta$ AND $Pr(X|[x]_\Phi) < \gamma$ THEN $x \in NEG(X)$
- $(B2)$ IF $Pr(X|[x]_\Phi) < \alpha$ AND $Pr(X|[x]_\Phi) > \beta$ THEN $x \in BND(X)$

where parameters α, β and γ are defined as:

$$\alpha = \frac{\lambda_{PN} - \lambda_{BN}}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})},$$

$$\beta = \frac{\lambda_{BN} - \lambda_{NN}}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})},$$

$$\gamma = \frac{\lambda_{PN} - \lambda_{NN}}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}.$$

In the DTRS model, an object in the probabilistic positive region does not surely belong to the decision class, but it belongs with a high probability. It implies that the acceptance and rejection decisions are made in light of certain error tolerance levels (Yao, 2011). However, the practical usability of classical three-way decisions may be reduced when facing scenarios with numerical attributes. More explicitly, the presence of numerical attributes requires replacing the equivalence class with a similarity class; thus an object could simultaneously belong to multiple decision classes. This implies that the decision process is no longer straightforward.

2.6 Concluding note

Using rough sets as a vehicle for granulating the available information brings several advantages towards building our granular classifier. For instance, rough information granules have a precise meaning and allow differentiating between the positive-certain, the negative-certain and the hesitant information. Determining the relation between granules coming from other theories (e.g., fuzzy sets or shadowed sets) becomes less intuitive. On the other hand, it is noticeable the potential behind the abstract semantics of three-way decisions to derive sound classification rules. But could a Machine Learning algorithm handle such rules? The next chapter introduces a kind of cognitive neural network that seems suitable for this purpose.

Chapter 3

Fuzzy Cognitive Maps

This chapter introduces the foundations behind *Fuzzy Cognitive Maps*, the second building block attached to our granular cognitive classifier. Moreover, we revise some predefined architectures to deal with pattern classification scenarios.

3.1 Fuzzy cognitive mapping

Fuzzy Cognitive Maps (FCMs) have increased their popularity within the scientific community during the last years. Such networks (Kosko, 1986) have become a suitable tool for the design of knowledge-based systems, where one of the most relevant characteristics is the network interpretability. Not many computer science techniques can claim this valuable feature. FCMs incorporated many aspects from Soft Computing (Kecman, 2001) (Pratihar, 2015) that provide further flexibility. From the structural perspective, an FCM can be defined as a fuzzy digraph that describes the behavior of an intelligent system in terms of concepts (i.e., objects, states, variables or entities). Such concepts (or simply neurons) involve a precise meaning for the problem domain and are connected by signed and weighted causal relationships.

The sign and intensity of causal relations can involve the quantification of a fuzzy linguistic variable which can be assigned by experts during a knowledge acquisition phase (Kosko, 1997). These elements iteratively interact when updating the activation value of each neural processing entity, thus conferring to the network its recurrent behavior. Consequently, an FCM-based network will produce a state vector at each discrete-time step until a stopping criterion is satisfied.

FCMs can be seen as interpretable *recurrent neural networks* (Nápoles et al., 2016) that allow capturing the semantics of a physical system in terms of concepts and causal relations. As mentioned, concepts are denoted by neural processing units that define a set $\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$ where M is the number of neurons in the causal network. The strength of the causal relation between two neurons C_i and C_j is quantified by a weight $w_{ij} \in [-1, 1]$ and denoted via an edge from C_i to C_j. This weight is defined by a function $\mathcal{W} : \mathcal{C} \times \mathcal{C} \rightarrow [-1, 1] : (C_i, C_j) \mapsto w_{ij}$. There are three possible types of causal relationships between neural processing units that express the type of influence from one neuron to another, which are detailed as follows:

- If $w_{ij} > 0$ then an increment (decrement) in the cause neuron C_i produces an increment (decrement) of the effect neuron C_j with intensity $|w_{ij}|$.

- If $w_{ij} < 0$ then an increment (decrement) in the cause neuron C_i produces a decrement (increment) of the effect neuron C_j with intensity $|w_{ij}|$.

- If $w_{ij} = 0$ then there is no causal relation between C_i and C_j.

In these knowledge-based models, both neurons and causal relations allow elucidating the semantics behind the system under analysis and performing WHAT-IF simulations. Such simulations are based on the following rule: *the stronger the activation value of a neuron, the greater its impact on the system.* Clearly, weights attached to the neurons are also relevant since they define the direction (i.e., inverse or direct) and the intensity on which a neuron influences another. Notice that weights comprise causal relations instead of correlation ones. It is well-known that causality does surely imply correlation, but the opposite does not necessarily hold.

Defining authentic causal relations between neural entities is a key aspect towards designing truly interpretable FCM-based systems. Otherwise, the model will produce misleading results when performing WHAT-IF simulations. It should be highlighted that we can model different levels of interpretability using the cognitive mapping principle, which depend on the abstraction degree. Neurons denoting entities with high abstraction level (i.e., information granules) lead to high-level interpretable networks. If the level of abstraction is too high, then the physical system under investigation is difficult to analyze, but the reasoning process is still transparent.

On the other hand, defining attribute-level entities allow interpreting the system behavior at a low level. However, sometimes the domain experts are unable to define authentic causal relations with such specificity level.

3.2 Neural updating rules

During the neural reasoning process, an FCM exploits an activation vector by using a rule similar to the standard McCulloch-Pitts scheme (McCulloch and Pitts, 1988). The activation degree of each neuron is given by the value of the transformed weighted sum that this processing unit receives from connected neurons on the causal network. Next, we discuss three widely used activation rules.

Equation (3.1) shows the Kosko's activation rule (Kosko, 1986), where $A_i^{(t)}$ is the activation value of the C_i neuron at the tth iteration, w_{ji} is the causal weight connecting the neurons C_j and C_i, $A^{(0)} = \left[A_1^{(0)}, A_2^{(0)}, \ldots, A_M^{(0)}\right]$ is the activation vector, while $f(.)$ is a monotonically non-decreasing transfer function. This updating mechanism is repeated until a stopping condition is satisfied, thus producing a state vector $A^{(t)} = \left[A_1^{(t)}, A_2^{(t)}, \ldots, A_M^{(t)}\right]$ at each iteration. Aiming at preserving the concordance between the problem domain and the modeled neural network, the Kosko's rule states that a neural entity should not be influenced by itself.

$$A_i^{(t+1)} = f\left(\sum_{j=1}^{M} w_{ji} A_j^{(t)}\right), i \neq j \tag{3.1}$$

Equation (3.1) describes a neural updating rule that has derived other (slightly different) reasoning procedures. In (Stylios and Groumpos, 2004), the authors proposed a modified inference rule (see Equation (3.2)) where neurons take into account its own past value and the corresponding weights when performing the inference process. This reasoning rule is preferred to update the activation value of neurons that are not influenced by other neural entities. The reader can notice that this variant is equivalent to suppress the $i \neq j$ constraint in the Kosko's rule.

$$A_i^{(t+1)} = f\left(\sum_{j=1}^{M} w_{ji} A_j^{(t)} + A_i^{(t)}\right), i \neq j \tag{3.2}$$

Another modified updating rule was proposed in (Papageorgiou, 2011) to avoid the conflicts emerging in the case of non-active concepts. Being more explicit, the rescaled inference depicted in Equation (3.3) allows dealing with the scenarios where there is no information about an initial concept-state and helps preventing the saturation of neural entities. Of course, we could face the aforementioned issues by using the proper parametric settings in the $f(.)$ transfer function.

$$A_i^{(t+1)} = f\left(\sum_{j=1}^{M} w_{ji}(2A_j^{(t)} - 1) + (2A_i^{(t)} - 1)\right), i \neq j \tag{3.3}$$

Selecting the proper updating rule depends on the problem domain and regularly requires a strong understanding of the physical system under analysis. As a further valuable remark, in (Papakostas and Koulouriotis, 2010) the authors concluded that removing the $i \neq j$ restriction in Equations (3.1) and (3.2) does not necessarily lead to improved prediction rates of FCM-based classifiers.

3.3 Network dynamics

As mentioned before, FCMs are recurrent neural networks that produce a state vector at each iteration. This procedure is repeated until either the FCM stabilizes or meets a predefined stopping criterion (e.g., reaching a maximum number of iterations). The former implies that a hidden pattern was discovered (Kosko, 1988) while the latter suggests that the responses are cyclic or completely chaotic.

If the FCM is able to converge, then the cognitive model will produce the same output towards the end, and thus the activation degree of neurons will remain without change (or the changes are infinitesimal). On the other hand, a cyclic FCM produces dissimilar responses with the exception of a few states that are periodically produced along the reasoning process. The last possible scenario is related to chaotic outcomes on which the network continues producing different outputs without any fixed pattern. Such situations can be mathematically defined as follows:

- **Fixed-point** ($\exists t_\alpha \in \{1, 2, \ldots, (T-1)\} : A^{(t+1)} = A^{(t)}, \forall t \geq t_\alpha$): the network produces the same state vector after the t_αth iteration (Nápoles et al., 2016). This suggests that $A^{(t_\alpha)} = A^{(t_\alpha+1)} = A^{(t_\alpha+2)} = \cdots = A^{(T)}$.

- **Limit cycle** ($\exists t_\alpha, P \in \{1, 2, \ldots, (T-1)\} : A^{(t+P)} = A^{(t)}, \forall t \geq t_\alpha$): the network periodically produces the same state vector after the t_αth iteration (Nápoles et al., 2016). Therefore, $A^{(t_\alpha)} = A^{(t_\alpha+P)} = A^{(t_\alpha+2P)} = \cdots = A^{(t_\alpha+jP)}$ where $t_\alpha + jP \leq T$, such that $j \in \{1, 2, \ldots, (T-1)\}$.

- **Chaos** (Wang et al., 1990): the network produces different outputs for successive iterations, thus being difficult to make decisions.

If the FCM is unable to converge, then the model will produce confusing responses and thus a pattern cannot be concluded (Nápoles et al., 2016), being impossible to made suitable decisions. In presence of chaos or cyclic states, the reasoning rule stops once a maximal number of iterations T is reached. If so, the state vector is calculated from the last response. However, this output is partially unreliable due to the lack of convergence. The convergence is often desirable since the responses become consistent and the expert may understand the system behavior.

3.4 The transfer function

The transfer function $f : \Re \to I$ in Equations (3.1), (3.2) and (3.3) is a monotonically non-decreasing function that confines the activation value of each neuron into the allowed interval, where $I = [0, 1]$ or $I = [-1, 1]$ depending on the problem domain. According to the cardinality of the state space, transfer functions may be gathered into two groups (Tsadiras, 2008): *discrete* and *continuous.* The most widely used functions are the bivalent, the trivalent, the hyperbolic tangent and the sigmoid function. Next, we outline their key advantages and limitations.

1. **The bivalent function** (see Equation (3.4)). It is a discrete transfer function that only produces binary responses leading to a finite number of states vectors. This happens because an FCM is a deterministic system and so, if it reaches a state to which it has been previously, the FCM will enter in a closed orbit that will always repeat itself (Tsadiras, 2008). Therefore, a binary FCM will always converge to a fixed-point or produce cyclic patterns (with an exponential period in the worst scenario) but it will never produce chaos.

$$f_1(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{3.4}$$

2. **The trivalent function** (see Equation (3.5)). It is a discrete function that produces a finite number of different outputs, and thus the system will always converge to a fixed-point or produce cyclic patterns; chaos is not possible. The disadvantage of discrete functions relies on their poor representation capability where only qualitative scenarios can be modeled.

$$f_2(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{3.5}$$

3. **The sigmoid function** (see Equation (3.6)). It is a continuous transfer function that produces infinite different states freely distributed in the space defined by the $[0, 1]^M$ hypercube, with M being the number of neurons. In this nonlinear function, $\lambda > 0$ and $h > 0$ are two user-specified parameters controlling the function slope and the offset, respectively. Higher values of λ increase the steepness and make it more sensitive to the neuron's changes, hence the derivative grows as the activation value is increased.

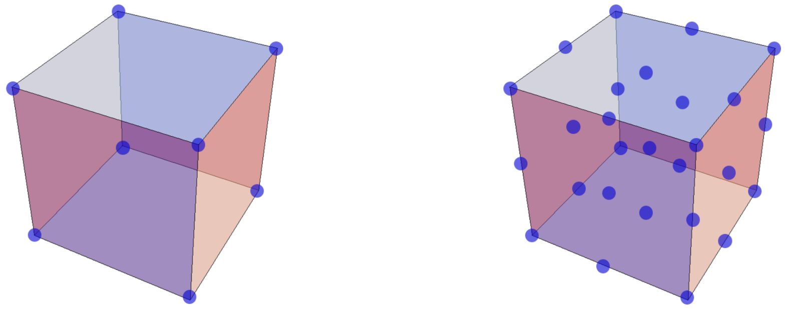
$$f_3(x) = \frac{1}{1 + e^{-\lambda(x-h)}} \tag{3.6}$$

4. **The hyperbolic function** (see Equation (3.7)). It is another continuous function that produces infinite state vectors but distributed in the $[-1, 1]^M$ hypercube. Besides the fixed-points and the cyclic states, continuous functions may additionally produce chaos (Tsadiras, 2008). As an advantage, they can be used for modeling both qualitative and quantitative scenarios.

$$f_4(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{3.7}$$

Figure 3.1 displays the state space of an FCM-based network with three neurons, for both bivalent and trivalent functions. For the bivalent function, the states are located at the corners of the $[0, 1]^2$ hypercube, whereas in the case of the trivalent function, the states are located at the corners, at the middle of the edges, at the center of the sides and, at the center of the $[-1, 1]^2$ hypercube.

The number of patterns that the network is able to recognize increases with the number of different outputs that $f(.)$ produces. However, this also increases the risk of producing chaos or cyclic states with larger exponential periods. This risk may be reduced (or completely suppressed) if the causal weight matrix fulfills some conditions ensuring the convergence to a fixed-point attractor.



(a) Bivalent transfer function.

(b) Trivalent transfer function.

Figure 3.1: State space for an FCM with three neurons.

Likewise, in (Bueno and Salmeron, 2009) the authors conducted a simulation comparing the inference capability of the above functions. Results showed that the sigmoid function attains the highest predictive capacity when compared with the other alternatives. In point of fact, this nonlinear function will be adopted in this research when designing our GCM-based classifiers. It should be however emphasized that the selection of the threshold function is frequently conditioned by the problem requirements, i.e., the role each neuron plays into the causal network.

As a side remark, although FCMs inherited many aspects from well-known neural systems, there are important differences regarding to other types of *Artificial Neural Networks*. Classical neural models regularly perform like *black-boxes*, where both the neurons and the connections do not have a clear meaning for the problem itself, or results cannot easily be explained. Nevertheless, all neurons in an FCM have a precise meaning for the physical system being modeled and correspond to specific variables, objects, states or entities. Furthermore, FCM-based networks do not comprise hidden neurons since these entities could not be interpreted nor help at explaining whether a solution is suitable for a given scenario or not.

3.5 FCM-based classifiers

Despite the noticeable advantages of using FCMs for modeling complex systems, their application in classification scenarios has been less studied. This is motivated by their poor prediction rates when compared to traditional classifiers in more generic problems. In spite of this fact, sometimes, FCM-based models are capable of producing high-quality solutions, even using very simple architectures.

The first attempt to use FCMs in pattern classification environments was implemented in (Papakostas et al., 2008) and later extended in (Papakostas and Koulouriotis, 2010). In these papers, Papakostas and his collaborators defined the basic principles of *FCM-based classifiers*. The most prominent challenge to be faced when constructing an FCM-based classifier lies in how to connect the neural processing entities in order to preserve the coherence on the network topology. In such causal structures, input neurons denote features, while decision classes are represented by output neurons. It should be noticed that the activation value of an input neuron may depend on the other neural entities or may be totally independent.

In (Papakostas and Koulouriotis, 2010), the authors defined two network topologies to define the interaction between input and output neurons. The first of these generic models connects the input neurons to each other, whereas the second model only connects the input neurons with the output ones.

None of the above approaches confidently reflect the underlying behavior behind the physical system under investigation. In the first case, we cannot always suppose that all features are dependent each other. For example, let us assume three concepts: "C_1 - *precipitation*", "C_2 - *take a bus*" and "C_3 - *take a bike*". It is reasonable to expect that an increase on C_1 leads to an increase on the value of C_2 and a decrease on the value of C_3. This suggests that C_2 and C_3 are dependent from C_1, although they have their own initial activation values. On the other hand, an increase on C_2 and C_3 will not increase/decrease any chance of precipitation. In the second case, the model fails in capturing the dependence between features when performing the classification process, therefore leading to poor prediction rates.

Another important issue attached to construction of an FCM-based classifier is how to compute the decision class. Roughly speaking, a standard FCM-based classifier can work based on two types of generic architectures:

- **Class-per-output architecture**. Each decision class is mapped as an output neuron. After performing the inference process, the predicted class corresponds to the output neuron with the highest activation value.

- **Single-output architecture**. Decision classes are determined from the activation space associated to a single decision neuron:

1. *Using a clustering approach.* During the training phase, each decision class is associated with a cluster center (to be determined by a clustering algorithm). Afterwards, the center having the closest distance to the projected activation value is assigned to the input object.
2. *Using a thresholding approach.* During the training phase, each decision class is associated with a partition (defined by two thresholds) of the activation space. In the testing phase, the interval comprising the projected activation value is then assigned to the input object.

Figure 3.2 and 3.3 show two hybrid typologies (Papakostas and Koulouriotis, 2010) that include a *black-box* classifier to improve the prediction rates. In the first hybrid FCM-based classifier, the black-box produces a confidence degree per decision class. Sequentially, the confidence vector is used as initial configuration for the FCM model that corrects the outputs produced by the black-box. In the second model, the input neurons (i.e., features) are also connected to output ones, so the predictions computed by the black-box classifier can be understood as a bias.

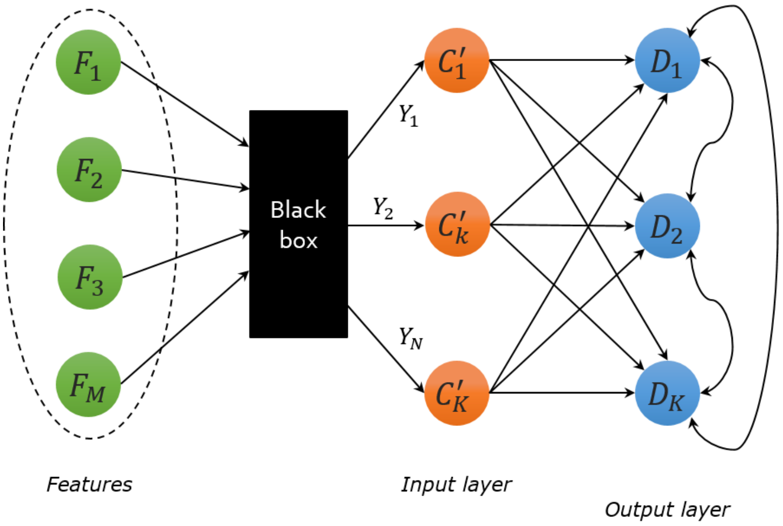


Figure 3.2: Hybrid FCM-based classifier type-1.

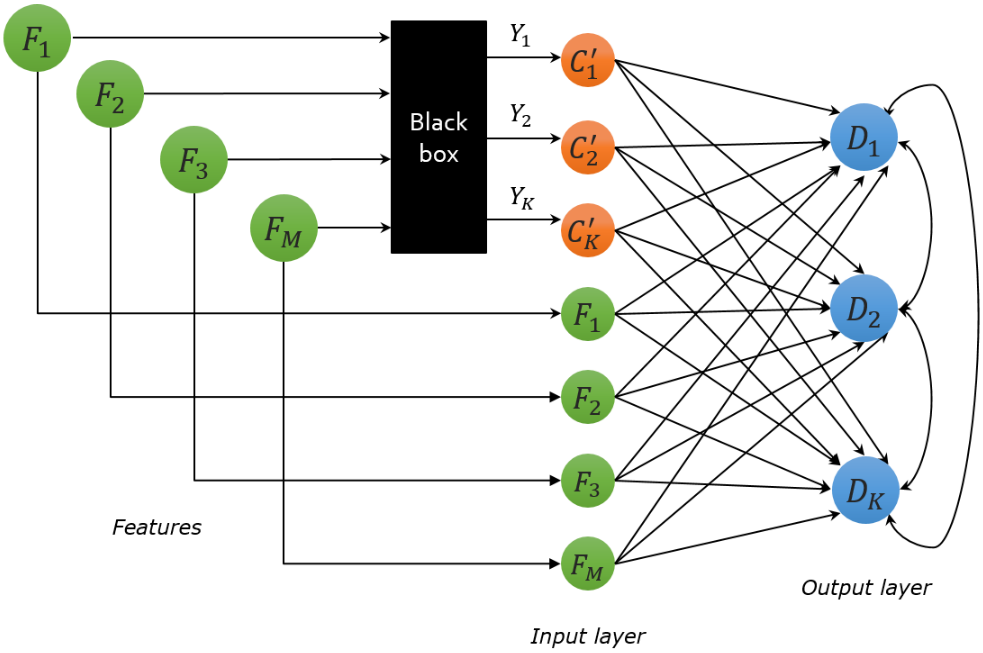


Figure 3.3: Hybrid FCM-based classifier type-2.

In these hybrid models, both the *transparency* and *interpretability* are irremediably damaged as the classifier is no longer capable of clearly explaining its decision process nor elucidating the causal relations related to the physical system.

3.6 Concluding note

The fuzzy cognitive mapping seems a suitable Soft Computing technique to cope with pattern classification problems in a comprehensible fashion. Nevertheless, establishing the dependency relations between input neurons at the attribute level often requires the intervention of domain experts to preserve the model coherence. Otherwise, we cannot ensure that the cognitive network describes the physical system under analysis. As an alternative, we can rely on Granular Computing techniques to derive accurate FCM-based classifiers that allow understanding the system behavior at a symbolic level. Thereby, a low-level reasoning is no longer possible, but the classifier's decision process remains transparent and comprehensible.

Chapter 4

Rough Cognitive Networks

This chapter presents the *Rough Cognitive Networks*, the first granular classifier introduced in this research, which emerge from the synergy between the abstract semantics of three-way decision rules and cognitive mapping.

4.1 Preliminaries about the algorithm

As discussed in Chapter 2, an equivalence relation induces a partition of the universe, but this does not necessarily hold if we use similarity relations since an object could simultaneously belong to different similarity classes. This implies that the similarity class of an object may activate several decision rules. In such scenarios, making precise decisions based on the available evidence could be challenging.

For example, let us assume that the similarity class $\bar{R}(x)$ comprises objects associated with two decision classes D_1 and D_2. According to the standard three-way decision rules, we cannot confidently decide D_1 nor D_2 since $\bar{R}(x)$ is not entirely contained into neither $POS(X_1)$ nor $POS(X_2)$, with X_1 and X_2 being the set of objects labeled as D_1 and D_2, respectively. Of course, we can calculate confidence levels to determine the most likely decision class. If the available knowledge is enough and there are a few decision classes then determining the most likely class may be relatively easy to perform by considering the positive information. However, real-world classification problems are often characterized by complex features (e.g., imbalance, inconsistency). Besides, estimating the parameters of more advanced three-way decisions rules may be as complex as solving the classification problem itself.

4.2 Rough cognitive mapping

Essentially, a Rough Cognitive Network (RCN) can be defined as a GCM where input neurons represent rough approximation regions and the output neurons denote the set of class labels for a pattern classification problem. The process of building and exploiting an RCN, referred to as *rough cognitive mapping*, involves three well-defined steps, namely: 1) the information granulation, 2) the network construction, and 3) the network exploitation for unlabeled objects.

4.2.1 Information granulation

Similarly to the three-way decision rules (Yao, 2009), the first step of our proposal is oriented to determine the positive, negative and boundary regions related to each decision class. Let $X = \{X_1, \dots, X_k, \dots, X_K\}$ be a partition of the universe $\mathcal{U}$ according to the values of the decision attribute, where each subset X_k comprises those objects labeled as D_k. It should be highlighted that attributes may be either nominal and/or numerical as our approach uses similarity relations to compute the lower and upper approximations. Therefore, the *information granulation* refers to the process of computing the lower and upper approximations for each subset X_k, and next determining the related positive, negative and boundary regions. Algorithm 1 displays the steps related to the information granulation stage in its simplified form since rough regions are determined from lower and upper approximations.

Algorithm 1. Information granulation procedure.

```
FOREACH subset $X_k$ DO
   Compute the positive region $POS(X_k)$
   Compute the negative region $NEG(X_k)$
   Compute the boundary region $BND(X_k)$
END
```

The granulation of the information space is a central step towards building a granular classifier since constructs carry the knowledge to perform the classification process. One cannot expect achieving high-quality predictions from deficient information granules. It is reasonable to suppose that deriving information granules with higher negative and positive regions will surely lead to higher prediction rates as well. But this is not true in all situations. Sometimes, the boundary regions help to increase the classifier's discriminatory power (Nápoles et al., 2016).

During the information granulation stage, two important components must be determined. The first of them is the *distance function* used to measure the dissimilarity between objects, whereas the second one is the *similarity threshold* that establishes whether two objects are inseparable or not. In the context of rough sets based on similarity relations, two objects are deemed inseparable if they are identical or reasonably similar. That is why a similarity threshold is required.

As mentioned, the similarity threshold defines the granularity level in the rough constructs. Smaller threshold values lead to a lower granularity degree, while larger values lead to a higher granularity level. Regrettably, determining the exact granularity level for the problem at hand is quite challenging. For example, using smaller similarity values increases the risk of claiming the inseparability between objects that are in fact separable. In contrast, larger values increases the risk of ignoring authentic inseparability relations between objects. In both scenarios the prediction rates computed by the classifier will probably be negatively affected. Chapter 5 and 6 will approach this problem using other Machine Learning resources.

On the other hand, it is noticeable the advantages of the three-way decision model to derive classification rules from rough granules. Nevertheless, in numerical domains, a similarity class $\bar{R}(x)$ might activate multiple decision rules. If $\bar{R}(x)$ only contains objects related to a single decision class, then selecting the correct decision is relatively easy since the similarity class will completely be contained within a specific decision region. But what happens if $\bar{R}(x)$ activates the positive region of multiple decisions? Likewise, what happens if only negative and boundary regions are activated? In such non-trivial scenarios further inference strategies are required.

In the first scenario, we could calculate confidence levels for each active rule and then compute the likelihood of each decision. The second scenario seems to be more complex and depends on the available knowledge. For example, sometimes it is not possible to infer a decision class only based on the positive regions (e.g., they remain inactive or have low cardinality). The reader can fairly claim that we can reject the hypothesis that an object belongs to a decision class if a positive or boundary evidence cannot be established. Nevertheless, in multiclass classification problems, the negative evidence will not be redundant; instead it will increase the discriminatory power of the rough classifier in rather inconsistent problems.

In the next section, we explain how to automatically construct a cognitive neural network (e.g., FCM-based system with sigmoid neurons) from rough granules by using the abstract semantics of three-way decision rules. This cognitive network will be used to conduct the classification process at a higher level.

4.2.2 Network construction

Without loss of generality, we can define *Rough Cognitive Networks* as FCMs where each rough approximation region is mapped to an input neural entity whereas each decision class gives rise to an output concept. In order to boost the inference capability of this granular classifier, we adopt neurons equipped with sigmoid transfer functions instead of using discrete ones. It should be remarked that output neurons do not influence each other since they are receiving concepts (i.e., their activation value only depends on the connected neurons). This design is similar to the *class-per-output architecture* presented in (Papakostas and Koulouriotis, 2010) where input neurons represent attributes instead of denoting information granules.

Once the neurons have been determined, we need to compute the causal relations between them. Such causal values are either estimated by domain experts during the construction stage or automatically computed from historical data using a learning algorithm. Aiming at overcoming this issue, we propose a set of rules inspired on the three-way decision model, which is depicted as follows:

- (R_1) IF C_i is P_k AND C_j is D_k THEN $w_{ij} = 1.0$
- (R_2) IF C_i is P_k AND C_j is $D_{v \neq k}$ THEN $w_{ij} = -1.0$
- (R_3) IF C_i is P_k AND C_j is $P_{v \neq k}$ THEN $w_{ij} = -1.0$
- (R_4) IF C_i is N_k AND C_j is D_k THEN $w_{ij} = -1.0$

In such rules, C_i and C_j are two neurons, P_k and N_k are the positive and negative regions related to the kth decision class, respectively, whereas $w_{ij} \in [-1, 1]$ denotes the weight between the cause C_i and the effect C_j. Neurons can be gathered into four categories, namely: positive, negative, boundary and decision neurons.

Rules R_1 and R_2 define the relation between positive regions and decision neurons. If the P_k positive region is activated, then the D_k class will be stimulated as well, since we surely know that objects belonging to the kth positive region will categorically be members of the kth subset. On the contrary, decisions $D_{v \neq k}$ will be inhibited in order to increase the classifier's discriminatory power. Activating the kth positive region (i.e., $\bar{R}(x) \cap POS(X_k) \neq \emptyset$) does not necessarily imply accepting the object x to be a member of the subset X_k, but it suggests a positive causal influence over the kth decision class. The D_k decision class will surely be accepted if $\bar{R}(x) \subseteq POS(X_k)$, which is in concordance with the three-way decision rules.

The R_3 rule follows an equivalent reasoning: if the P_k positive region is activated, then positive regions unrelated to the kth class (i.e. $P_{v \neq k}$) will be inhibited as well. This suggests that the probability of accepting a decision class is inversely dependent on the others, thus avoiding the neurons' saturation phenomenon.

The R_4 rule describes the causal relation between the N_kth negative region and the kth decision class: if the N_k negative region is activated, then the kth class must be inhibited, since the evidence $\bar{R}(x) \cap NEG(X_k)$ suggests rejecting this decision. However, we cannot conclude anything about the other decisions, unless the classification problem has only two classes. If $\bar{R}(x) \subseteq NEG(X_k)$, then the kth class must be rejected as the three-way decision rules suggest.

According to the three-way rules, boundary regions suggest a decision of abstaining, but an object $x \in BND(X_k)$ could be still associated with the kth alternative. Let us suppose a problem having three decision classes D_1, D_2 and D_3, an unlabeled object x such that $x \in BND(X_1)$, $x \in BND(X_2)$ and $x \notin BND(X_3)$. This means that x could be labeled as either D_1 or D_2 to the same extent, but there is no evidence supporting the third decision class. The following rule (R_5) includes the knowledge about boundary regions during the network construction stage.

- (R_5) IF (C_i is B_k AND C_j is D_v) AND ($BND(X_k) \cap BND(X_v) \neq \emptyset$) THEN $w_{ij} = 0.5$

Neurons representing empty boundary regions (i.e., $BND(X_k) = \emptyset$) will not be included into the model to keep the topology as simpler as possible. This implies that an RCN will comprise at most $|\mathcal{D}|$ output neurons, $3|\mathcal{D}|$ input neurons and $3|\mathcal{D}|(1 + |\mathcal{D}|)$ causal edges, where $\mathcal{D}$ is the set of decision classes. Algorithm 2 shows the procedure that allows building the network from granular regions.

Algorithm 2. Network construction procedure.

```
FOREACH subset $X_k$ DO
    Add a neuron $P_k$ as the $POS(X_k)$ region
    Add a neuron $N_k$ as the $NEG(X_k)$ region
    Add a neuron $B_k$ as the $BND(X_k)$ region
END
FOREACH decision $D_k$ DO
    Add a neuron $D_k$ as the $k$th decision
END
```

```
FOREACH neuron $C_i$ DO
    FOREACH neuron $C_j$ DO
        Configure $w_{ij}$ according to rules $R_1 - R_5$
    END
END
```

Figure 4.1 displays an RCN to solve any pattern classification problem with two decision classes. In this example, we assume that $BND(X_1) \cap BND(X_2) \neq \emptyset$, which means that the available information (i.e., the training dataset) involves some degree of inconsistency. The reader can perceive that we added a self-reinforcement positive causal edge to each input neuron with the goal of preserving its initial excitation level when performing the FCM inference process.

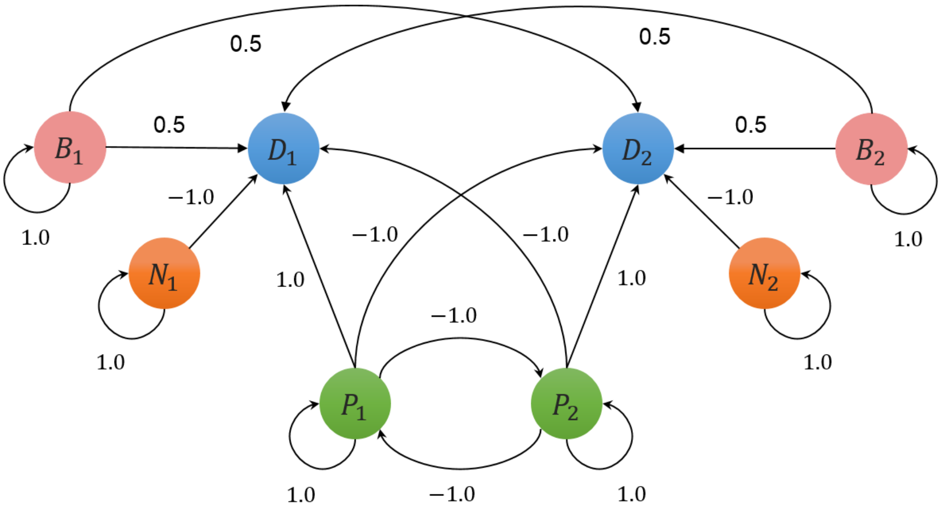


Figure 4.1: Rough Cognitive Network for 2-class problems.

It should be highlighted that Kosko's formulation does not allow neurons to be influenced by themselves. The purpose of this constraint is to preserve the coherent interpretation of the modeled system. However, under this restriction, the activation value of input neurons (i.e., concepts that are not influenced by other neural entities) will be overwritten in the second iteration step. As an alternative, we could adopt the Equation (2.6) to perform the neural inference process, but it is equivalent to use the Kosko's rule with self-reinforced neurons. This feature may be understood as the memory that input neurons have about their own excitation value.

4.2.3 Network exploitation

The last step is related to the network exploitation. Let x be an unlabeled object and $\mathcal{A}_x(\mathcal{D}) = \big[\mathcal{A}_x(D_1), \ldots, \mathcal{A}_x(D_k), \ldots, \mathcal{A}_x(D_K)\big]$ is the preference vector comprising the likelihood of x to be a member of each decision class. Therefore, the classification process is equivalent to computing the preference vector and determining the most likely decision class. The unlabeled object is presented to the granular network as the activation vector $A^{(0)}$ to perform the FCM inference process. This vector encodes the initial activation value of input neurons as follows:

$$A^{(0)} = \big[\mathcal{A}_x^{(0)}(P_1), \ldots, \mathcal{A}_x^{(0)}(P_k), \ldots, \mathcal{A}_x^{(0)}(P_K), \mathcal{A}_x^{(0)}(N_1), \ldots, \mathcal{A}_x^{(0)}(N_k), \ldots, \mathcal{A}_x^{(0)}(N_K),$$
$$\mathcal{A}_x^{(0)}(B_1), \ldots, \mathcal{A}_x^{(0)}(B_k), \ldots, \mathcal{A}_x^{(0)}(B_K), \mathcal{A}_x^{(0)}(D_1), \ldots, \mathcal{A}_x^{(0)}(D_k), \ldots, \mathcal{A}_x^{(0)}(D_K)\big]$$

where $\mathcal{A}(.)$ refers to the activation of each neuron. It should be remarked that output neurons remain inactive when activating the network since their activation value is computed from the recurrent propagation of the evidence over the granular network. This implies that $\mathcal{A}_x^{(0)}(D_1) = \ldots = \mathcal{A}_x^{(0)}(D_k) = \ldots = \mathcal{A}_x^{(0)}(D_K) = 0$.

In order to compute the activation vector $A^{(0)}$, we use the similarity class $\bar{R}(x)$ and the *inclusion degree* to gauge the extent to which $\bar{R}(x)$ is included into each rough approximation region. Equations (4.1) shows this inclusion measure, which essentially provides a relative measure to exhibit information concentration.

$$Pr(X|\bar{R}(x)) = \frac{|\bar{R}(x) \cap X|}{|X|} \tag{4.1}$$

For instance, let us suppose that $|POS(X_1)| = 20$, $|\bar{R}(x)| = 10$, while the number of objects that belong to the positive region is $|\bar{R}(x) \cap POS(X_k)| = 7$. This implies that the activation degree of the P_1 neuron is given by $\mathcal{A}_x^{(0)}(P_1) = 7/20 = 0.35$, which actually denotes the conditional probability of accepting D_1 given the similarity class $\bar{R}(x)$ associated to the input object, that is $Pr(D_k|\bar{R}(x))$. Therefore, the kth positive neuron will affect the D_1 decision neuron with excitation degree $\mathcal{A}_x^{(0)}(P_1) = 0.35$ and causal weight $w_{ij} = 1$. Rules $R_6 - R_8$ generalize this activation method for all input neurons representing non-empty granular regions.

- (R_6) IF C_i is P_k THEN $\mathcal{A}_x^{(0)}(P_k) = \frac{|\bar{R}(x) \,\cap\, POS(X_k)|}{|POS(X_k)|}$

- (R_7) IF C_i is N_k THEN $\mathcal{A}_x^{(0)}(N_k) = \frac{|\bar{R}(x) \,\cap\, NEG(X_k)|}{|NEG(X_k)|}$

- (R_8) IF C_i is B_k THEN $\mathcal{A}_x^{(0)}(B_k) = \frac{|\bar{R}(x) \,\cap\, BND(X_k)|}{|BND(X_k)|}$

In short, the exploitation of the proposed granular classifier is composed of two main stages related to the activation of input neurons and the classification process itself. Algorithm 3a shows the steps related to the first stage.

Algorithm 3a. Network activation procedure.

```
FOREACH decision $D_k$ DO
    Calculate $\mathcal{A}_x^{(0)}(P_k)$ according to rule $R_6$
    Calculate $\mathcal{A}_x^{(0)}(N_k)$ according to rule $R_7$
    Calculate $\mathcal{A}_x^{(0)}(B_k)$ according to rule $R_8$
END
```

Once the $A^{(0)}$ vector has been computed, the FCM reasoning rule is performed until either the network converges to a fixed-point or a maximal number of iterations T is reached. Next, the decision class with the highest activation value is assigned to the target object. Algorithm 3b summarizes this process.

Algorithm 3b. Network reasoning procedure.

```
FOR $t = 0$ TO $T$ DO
    $converged \leftarrow TRUE$
    FOREACH neuron $C_i$ DO
        Compute $A_i^{(t+1)}$ using the FCM rule
        IF $A_i^{(t)} \neq A_i^{(t+1)}$ THEN
            $converged \leftarrow FALSE$
        END
    END
    IF $converged$ THEN
        RETURN $argmax_k\{\mathcal{A}_x^{(t+1)}(D_k)\}$
    END
END
IF not $converged$ THEN
    RETURN $argmax_k\{\mathcal{A}_x^{(T)}(D_k)\}$
END
```

Sometimes we could identify scenarios on which multiple decision classes have the highest probability of being produced (i.e., two aggregated neurons bear the same maximal activation value). These situations are common in highly inconsistent problems where there is a lack of conclusive knowledge. To solve this issue, we select the decision class D_k associated with the closest neighbor of the test instance $x \in X$, such that D_k has the highest output probability. This heuristic provides a fair compromise between the decision class computed by the granular classifier and the intrinsic relation among those instances that represent the same pattern.

4.3 Classifier convergence

In this section, we briefly discuss some aspects related to the classifier convergence. Unlike feed-forward neural networks, FCM-based systems produce a state vector at each iteration step due to their recurrent nature. This iterative updating mechanism is repeated until either the network converges to a *fixed-point attractor* or a maximal number of iterations is reached. The former scenario implies that a hidden pattern was discovered (Kosko, 1988) while the latter suggests that the system outputs are either *cyclic* or *chaotic*. Ensuring the FCM convergence is a key aspect, otherwise the model will produce unreliable (non-interpretable) responses.

In the context of FCM-based classifiers, the term *convergence* refers to the network ability to eventually produce the same decision class over successive iteration steps (Nápoles et al., 2016). Aiming at formalizing this scenario, we introduced the notion of *slightly-stable FCM-based classifiers*. It refers to classifiers that produce numerically dissimilar responses that lead to the same decision class.

The lack of stability in FCM-based systems could be caused by three key factors (Nápoles et al., 2017): i) the weight matrix, ii) the strategy for updating the neurons' values and iii) the transfer function used in the reasoning rule.

Several studies (Hopfield, 1982) (Bruck, 1990) (Baran and Coughlin, 1990) have shown that a symmetric zero-diagonal matrix in conjunction with an asynchronous updating strategy lead to fixed-points. A perfectly symmetric weight matrix implies the existence of a large number of cycles with positive feedback loops that amplify any initial change, thus leading to an exponential growth or decline (Tsadiras, 2008). An antisymmetric weight matrix encourages the existence of negative cycles with an odd number of connections, thus providing the system with negative feedback loops that counteract any stimulus. As a result, the recurrent system creates limit cycles wherein some state vectors are periodically produced.

This implies that RCN-based classifiers are likely to converge since the proposed topology has rather symmetric properties. More explicitly, the network topology is comprised of three kind of neural entities: independent input neurons (i.e., negative and boundary entities), dependent input neurons (i.e., positive entities) and output neurons. Definitions 1, 2 and 3 formalize the semantics behind such categories in a pattern classification context, which actually can be used to briefly analyze the overall convergence properties of rough cognitive models.

Definition 1. *We say that a concept C_i is an **independent input neuron** if its activation value does not depend on the other input neurons.*

Definition 2. *We say that a concept C_i is a **dependent input neuron** if its activation value is influenced by other input neurons.*

Definition 3. *We say that a neural entity C_i is an **output neuron** if its activation value only depends on the connected input neurons.*

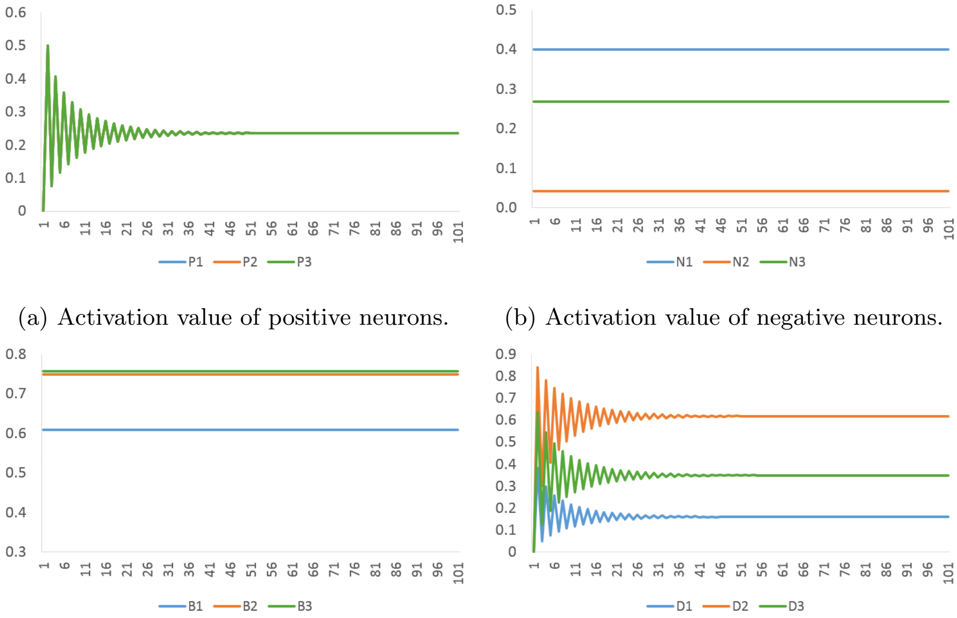
From the above definitions we can conclude that independent input neurons will converge by definition as their activation values remain unaltered. Likewise, dependent input neurons are likely to converge because they are symmetrically connected to each other. This implies that output neurons will converge as well because their activation values are computed from stable input neurons.

As a tangential research, we introduced a learning algorithm to improve the convergence of FCM-based classifiers without modifying the weight set estimated during the construction phase (Nápoles et al., 2013) (Nápoles et al., 2014). This algorithm computes the sigmoid function parameters associated with each neural processing unit that leads to improved convergence features. More recently, we published several variations of this learning method (Nápoles et al., 2016) (Nápoles et al., 2016) (Nápoles et al., 2017) that produce stronger results. This suggests that we have the proper tools to deal with the lack of convergence of continuous FCM-based systems without altering the network structure. However, in the context of rough cognitive mapping, the use of these learning procedures is not required since the proposed topology seems to naturally converge to multiple fixed-point attractors.

4.4 How the classifier works?

Before introducing the simulations, it seems convenient to illustrate how the proposed classifier operates. Let us consider a classification problem comprising three decision classes D_1, D_2 and D_3 such that the values of the decision attribute induces a partition $X = \{X_1, X_2, X_3\}$ of the universe. In this partition, each subset X_k comprises those objects labeled as D_k. Likewise, let us suppose a hypothetical scenario where there is no positive information (i.e., $\mathcal{A}_x^{(0)}(P_k) = 0, \forall k$), $\mathcal{A}_x^{(0)}(N_1) = 0.4$, $\mathcal{A}_x^{(0)}(N_2) = 0.04$, $\mathcal{A}_x^{(0)}(N_3) = 0.26$, $\mathcal{A}_x^{(0)}(B_1) = 0.6$, $\mathcal{A}_x^{(0)}(B_2) = 0.74$, $\mathcal{A}_x^{(0)}(B_3) = 0.75$, while decision neurons are initially inactive (i.e., $\mathcal{A}_x^{(0)}(D_k) = 0, \forall k$).

Figure 4.2 illustrates the neural (recurrent) reasoning process of an RCN-based classifier for the above configuration. The vertical axis represents the the activation degree of positive, negative, boundary and decision neurons at each iteration, whereas the horizontal axis denotes the iterations. Observe that boundary and negative neurons do not change their activation value when performing the recurrent reasoning process as they were conceived as independent input neurons.



(a) Activation value of positive neurons.

(b) Activation value of negative neurons.

(c) Activation value of boundary neurons.

(d) Activation value of decision neurons.

Figure 4.2: Rough cognitive reasoning: an example.

From the above simulation, we can perceive that the decision process is based on the hesitant (boundary) and certain-negative information. The negative information suggests rejecting the decision classes in the following order: $D_1 \succ D_3 \succ D_2$, therefore implying that D_2 is most likely to be produced. Likewise, the boundary information suggests accepting D_2 and D_3 with the same likelihood, while it greatly rejects the hypothesis of x to be a member of X_1. Overall, the available evidence coming from the rough information granules advocates for the strong rejection of D_1 while accepting D_2 over the D_3 decision class. Of course, the confidence of the FCM inference process is subject to the quality of information granules.

It is worth mentioning the *transparency* on the decision process attached to the proposed granular classifier. Not too many classification algorithms can claim this valuable feature without harming their predictive capability. In RCN-based classifiers we can easily understand the whole decision process by using inclusion equations and causal relations. What is more, we can *interpret* the physical system under analysis at a *high-level* by relying on the causal relations between rough information constructs. However, a low-level interpretability is no longer possible.

4.5 Concluding note

This chapter introduced the concept of rough cognitive mapping in the context of pattern classification. One of the advantages of this classifier relies on its transparency when performing the reasoning process. This is a result of combining Soft Computing techniques that allow elucidating the inference process at different transparency levels. However, numerical simulations reported in (Nápoles et al., 2016b) (Nápoles et al., 2016) (Nápoles et al., 2017a) have shown that the RCNs' performance is quite sensitive to the similarity threshold defining whether two objects are considered similar or not. The next chapter focuses on addressing the parametric requirement of the proposed algorithm using other Machine Learning approaches.

Chapter 5

Rough Cognitive Ensembles

This chapter presents the *Rough Cognitive Ensembles*, the second classifier introduced in this research, which attempts addressing the parametric requirement when granulating the information space in the RCN construction stage.

5.1 Preliminaries about the algorithm

A well-established trend within Machine Learning is that of combining the output of several classifiers to predict the class of a given object. These models are called *multi-classifier systems* (Dietterich, 2000) and often perform better than any single classifier, especially if some diversity (either parametric or structural) is present among the set of base classifiers. This heterogeneity may come in the form of independent samples of the training data (bootstrapping), multiple parametric configurations, different types of base classifiers used as building blocks, etc. *Bayesian voting*, *bagging*, *boosting* and *stacking* are common ensemble methods (Ren et al., 2016). AdaBoost (Fan et al., 1999) and Random Forests (Breiman, 2001) are ensemble learners that have shown great promise in solving pattern classification problems.

A recent survey (Ren et al., 2016) explores trends in this field, including: a) using multiobjective optimization algorithms to derive several structural elements of the ensembles; b) employing decomposition techniques; c) resorting to negative correlation learning schemes; d) incorporating elements from fuzzy logic and multiple-kernel learning and e) endowing the models with deep learning features. The broad research possibilities that bring ensemble learning techniques seem adequate to increase the practical usability of the proposed rough classifier.

5.2 Motivation and challenges

A pivotal issue when constructing an RCN is related to the precise estimation of the similarity threshold. This parameter determines whether two objects are similar or not, which then influences the construction of the similarity classes upon which the rough approximations regions are built. Regrettably, the RCN performance is highly sensitive to this user-specified parameter, therefore small variations on the granularity degree may lead to quite different outcomes. Aiming at alleviating this problem, in (Nápoles et al., 2016b) (Nápoles et al., 2016) we proposed a hyperparameter learning method to estimate the similarity threshold value from historical data. The drawback of this procedure relies on its computational complexity due to the fact that a single evaluation requires recalculating the lower and upper approximations of all decision classes, which could be time-consuming for large-scale datasets.

Let us assume that $\mathcal{U}_1 \subset \mathcal{U}$ is the training set and $\mathcal{U}_2 \subset \mathcal{U}$ is the hold-out test (validation) set such that $\mathcal{U}_1 \cap \mathcal{U}_2 = \emptyset$. The computational complexity of building the upper and lower approximations is $O(|\Phi||\mathcal{U}_1|^2)$, with Φ being the attribute set, whereas the complexity of building the network topology is $O(|\mathcal{D}|^2)$, with $\mathcal{D}$ being the set of decision classes. Besides, the complexity of exploiting the granular network for $|\mathcal{U}_2|$ instances is $O(|\mathcal{U}_2||\Phi||\mathcal{U}_1|^2)$. This implies that the overall temporal complexity of evaluating a single parameter value is $O(\max\{|\Phi||\mathcal{U}_1|^2, |\mathcal{D}|^2, |\mathcal{U}_2||\Phi||\mathcal{U}_1|^2\})$. Due to the fact that $|\mathcal{U}_1| \geq |\mathcal{U}_2|$ in most Machine Learning scenarios, we can conclude that the overall complexity of this learning method is $O(T|\Phi||\mathcal{U}_1|^3)$, where T is the number of iterations. Unfortunately, this may negatively affect the practical usability of RCNs in solving real-world pattern classification problems.

On the other hand, estimating a suitable similarity threshold for the hold-out test does not necessarily ensure that the granular network will produce optimal predictions in other scenarios. This happens because the validation and test sets must be disjoint, and so the parameter learning method fits the model to the examples in the validation set. Furthermore, it might occur that $\bar{R}(x) = \emptyset$ for some testing object as a result of estimating a very strict threshold for validation examples. If this happens, the model uses the k-nearest neighbors to activate the network.

This chapter is devoted to designing a granular ensemble model using RCNs as base classifiers in order to suppress the hyperparameter learning requirement of rough cognitive mapping. Before presenting the proposed ensemble classifier, we first revise the foundations of some ensemble learning techniques.

5.3 Ensemble learning approaches

A suitable approach to increase the global realibility on the classification process is to combine the output of multiple classifers into a single response. Several algorithms use this approach by learning an ensemble of models when performing the classification process. Relevant ensemble learners include *bagging*, *boosting* and *stacking*; they can, more often than not, increase the predictive capability of a single classification model (Witten and Frank, 2011). Besides, they involve general learning schemes that can be adopted to solve both classification and regression problems.

- **Bagging**. Combining the decisions of different models means amalgamating the various outputs into a single prediction. The simplest way to accomplish that in the case of classification is to use a voting strategy. Bagging and boosting both adopt this approach, although they derive the individual learners in different ways. In bagging the models receive equal weight, while in boosting weighting is used to give more influence to the more successful ones.

  Let us suppose that several training datasets of the same size are randomly chosen from a common problem domain. Bagging uses a collection of classifiers of the same type to construct the ensemble, each built over a different training dataset. As a rule, the base classifier must be sensitive to small perturbations on the training dataset. This implies that there will be test objects for which some of the base classifiers will produce incorrect decision classes, whereas other will successfully predict the expected decision class.

  Voting is an adequate strategy to produce (hopefully more accurate) consensus decisions. If one decision class receives more votes than any other, it is taken as the correct one (Witten and Frank, 2011). Generally, the more the merrier: the consensus is more reliable as more votes are taken into account. Decisions rarely deteriorate, but improvements cannot be ensured.

- **Boosting**. Bagging exploits the sensitivity of the base classifier to perturbations in the training set, where each model is built separately. Similarly to bagging, boosting combines base classifiers of the same type and uses voting to produce the consensus decision. However, boosting is iterative since each base classifier is influenced by the performance of previously built models (Witten and Frank, 2011). Moreover, boosting encourages the new model to be focused on those objects that have been incorrectly classified by the previous ones. This is done by assigning greater weights to harder instances; thus base classifiers complement each other by exploiting their learned model.

- **Stacking** Unlike bagging and boosting, stacking is not normally used to combine classifiers of the same type; instead it uses models having different nature. Moreover, voting is only effective if most classifiers perform well, otherwise the inferred prediction class will be a wrong one. Stacking replaces the voting strategy by a *metalearner* that attempts predicting which classifiers are the reliable ones when combining their outputs. Although introduced some years ago, stacking is not widely used since there is no a generalized best way to construct the ensemble learner (Witten and Frank, 2011). Furthermore, its foundations are difficult to analyze in a theoretically sound way.

Ensemble learning models are likely to produce stronger prediction rates, although certainly such improvements cannot be ensured in all scenarios. However, how could we take advantage of this approach to face the limitations coming from the parametric requirement of the proposed rough cognitive classifier?

5.4 Rough ensemble mapping

In this section, we build upon the ensemble learning formalism by developing a granular ensemble to deal with the RCNs' parametric sensitivity. The proposed classifier, named *Rough Cognitive Ensembles* (RCEs), uses a collection of RCNs as base classifiers, each operating at a different granularity level. It is worth mentioning that the concept of *granular ensembles* seems to still be in its infancy though. A good starting point is the use of granular classifiers as base classifiers, given the momentum they are presently enjoying. The model presented in this section follows this idea, and it may well be one of the first granular ensemble techniques ever put forth. As a result, the hyperparameter learning is no longer required, which notably increases the practical usability of rough cognitive mapping in real-world applications.

5.4.1 Information granulation

Without loss of generality, an RCE can be understood as a collection of RCNs, each operating at a different granularity degree. To induce different granularity levels, we use a random similarity threshold when constructing the three approximation regions associated with each base classifier. This approach attempts removing the parameter learning requirement, which in fact does not necessarily lead to the best predictions when testing the learned classifier. Definition 4 summarizes the semantics behind the granular cognitive ensembles presented in this research.

Definition 4. *An* RCE $\Gamma_{(\mathcal{R})}$ *is composed of a set* $\mathcal{R} = \{\mathcal{R}_{(\xi_1)}, \ldots, \mathcal{R}_{(\xi_i)}, \ldots, \mathcal{R}_{(\xi_N)}\}$ *of* N *different RCNs as base classifiers, where the* i*th granular network* $\mathcal{R}_{(\xi_i)}$ *is built by using a (randomly) selected similarity threshold* ξ_i*.*

The reader can observe that the granulation process does not change itself. This means that the granulation of the information space in the $\mathcal{R}_{(\xi_i)}$ classifier is subject to the (randomly) selected similarity threshold.

Theoretical and empirical results (Tumer and Ghosh, 1995) (Breiman, 1996) (Turner and Oza, 1999) have shown that combining multiple classifiers leads to optimal performance if these classifiers are not strongly correlated each another. In the proposed ensemble, the diversity among base classifiers is promoted by using different similarity thresholds $(\xi_1, \ldots, \xi_i, \ldots, \xi_N)$ instead of using a single value.

However, the fact that $\xi_i \leq \xi_j \implies \bar{R}_{(\xi_j)}(x) \subseteq \bar{R}_{(\xi_i)}(x)$ may lead to correlated base classifiers if they operate under the same conditions. In order to increase diversity we can introduce randomization. Instance-based learners depend mostly on attributes used to compute the distance between objects. Therefore, we can promote diversity by using a random subset of attributes (i.e., random subspace method). In the context of rough classification, using random attributes may not be the best choice since rough approximations are often built upon a *reduct* of the attribute set.

Moreover, unlike nearest-neighbor classifiers, RCN-based models are sensitive to perturbations on the training dataset due to the presence of the similarity threshold, even when using the same subset of attributes. As an alternative, we can perform *instance bagging* (Breiman, 1996) in order to counter the correlation coming from the fact that $\xi_i \leq \xi_j \implies \bar{R}_{(\xi_j)}(x) \subseteq \bar{R}_{(\xi_i)}(x)$. Instance bagging attempts neutralizing the instability of the base classifier by modifying the original training set (i.e., deleting some instances and replicating others). During this process, instances are randomly sampled with replacement from the original dataset to create a new one with the same size (Witten and Frank, 2011). This allows establishing a reasonable trade-off between ensemble diversity and accuracy (Nápoles et al., 2017a).

5.4.2 The ensemble architecture

Once the information granulation stage is complete, we aggregate the individual outputs to determine the most likely decision class. With this goal in mind, we adopt a standard voting scheme where the decision class for each test object is obtained by voting on all the N granular classifiers. Actually, the voting scheme can be naturally modeled by using an extended FCM-based network.

Figure 5.1 illustrates an RCE comprised of a set $\mathcal{R} = \{\mathcal{R}_{(\xi_1)}, \dots, \mathcal{R}_{(\xi_i)}, \dots, \mathcal{R}_{(\xi_N)}\}$ of different RCNs as base classifiers, where the ith granular network $\mathcal{R}_{(\xi_i)}$ is built by using the ξ_i similarity threshold. In this example, $K = |\mathcal{D}|$ denotes the number of decision classes, $D_k^{(i)}$ is a sigmoid neuron comprising the preference degree of producing the kth class according to the $\mathcal{R}_{(\xi_i)}$ network, whereas D_k is a neuron that gathers the voting preference related with the kth decision class. The reader can perceive that the classification process is now performed from the activation value of aggregated-type neurons. Hopefully, this bagging granular model will produce higher prediction rates with regard to RCNs using fixed similarity thresholds.

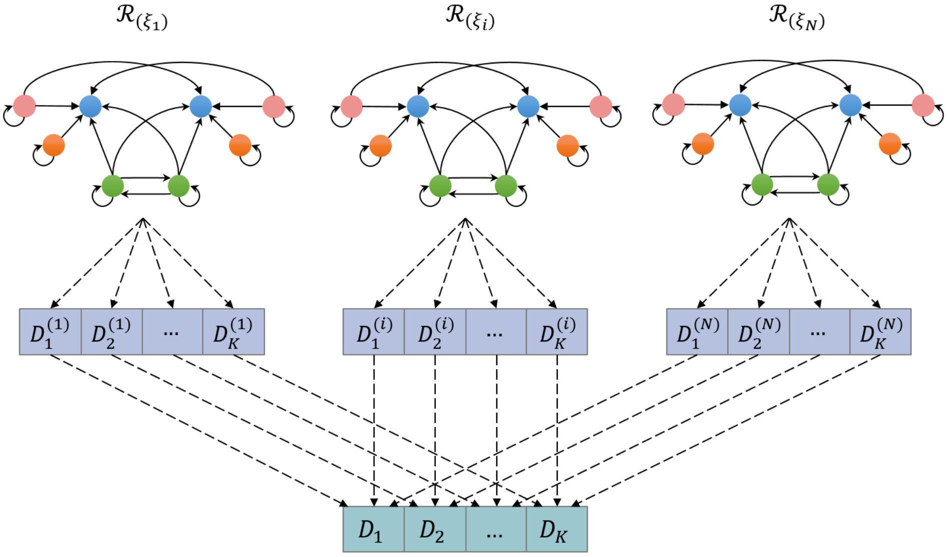


Figure 5.1: RCE-based classifier for K-class problems.

It should be mentioned that weights connecting $D_k^{(i)}$ and D_k neurons are set to one (i.e., $w_{ij} = 1$) in order to perform a standard voting process. Similarly, we can implement a weighted voting process by assigning greater causal weights to relations coming from more confident granular classifiers.

Moreover, we can estimate such weights in a supervised fashion to achieve higher prediction rates. Notice that this learning approach does not require computing the rough approximation regions for each error function evaluation since it is performed after the networks have been constructed. This procedure should be efficient due to the low cardinality of the search space. However, this alternative is not explored in our research, instead it becomes a future research work.

5.4.3 The exploitation scheme

The ensemble's exploitation phase is focused on determining the decision classes for unlabeled objects. This is equivalent to activating each RCN comprised in the ensemble, perform the inference process and select the most likely decision class according to the voting scheme. Therefore, the $\mathcal{R}_{(\xi_i)}$ network will produce a response vector $\mathcal{A}_x(\mathcal{D}^{(i)}) = \big[\mathcal{A}_x(D_1^{(i)}),\, \ldots,\, \mathcal{A}_x(D_k^{(i)}),\, \ldots,\, \mathcal{A}_x(D_K^{(i)})\big]$ where $\mathcal{A}_x(D_k^{(i)})$ represents the activation degree of the kth output neuron according to the ith granular network, with x being the test object. This allows computing the aggregated response vector $\mathcal{A}_x(\mathcal{D}) = \big[\mathcal{A}_x(D_1),\, \ldots,\, \mathcal{A}_x(D_k),\, \ldots,\, \mathcal{A}_x(D_K)\big]$ by combining the N response vectors produced by the base classifiers over the K decision classes.

In order to activate the rough cognitive ensemble, we need to compute N excitation vectors $\{A_{[x|\xi_i]}^{(0)}\}_{i=1}^{N}$ where $A_{[x|\xi_i]}^{(0)}$ is used to perform the neural reasoning process in the ith cognitive network. It should be remarked that the ith activation vector denotes the inclusion degree of the similarity class $\bar{R}_{(\xi_i)}(x)$ into each rough granule, which are built using the ξ_i similarity threshold. Formally, the ith activation vector associated with the unlabeled object x could be defined as follows:

$$A_{[x|\xi_i]}^{(0)} = \big[\mathcal{A}_{[x|\xi_i]}^{(0)}(P_1), \ldots, \mathcal{A}_{[x|\xi_i]}^{(0)}(P_k), \ldots, \mathcal{A}_{[x|\xi_i]}^{(0)}(P_K), \ldots, \mathcal{A}_{[x|\xi_i]}^{(0)}(N_1), \ldots, \mathcal{A}_{[x|\xi_i]}^{(0)}(N_k),$$
$$\ldots, \mathcal{A}_{[x|\xi_i]}^{(0)}(N_K), \ldots, \mathcal{A}_{[x|\xi_i]}^{(0)}(B_1), \ldots, \mathcal{A}_{[x|\xi_i]}^{(0)}(B_k), \ldots, \mathcal{A}_{[x|\xi_i]}^{(0)}(B_K)\big]$$

where $\mathcal{A}_{[x|\xi_i]}^{(0)}(P_k)$, $\mathcal{A}_{[x|\xi_i]}^{(0)}(N_k)$ and $\mathcal{A}_{[x|\xi_i]}^{(0)}(B_k)$ denote the activation degree of the kth positive, negative and boundary neuron, respectively. These initial activation values are computed using the rules R_6 - R_8 depicted in Section 4.2.3.

When activating the granular ensemble classifier, output (decision) neurons remain inactive. Sometimes, there is further knowledge about the probability of producing a specific class given a new object (e.g., resulting from an intermediate classification process). If so, we could use this information to activate the output neurons in order to improve the performance. However, in this research we assume that the available evidence is coming from the information granulation stage.

Once the neural reasoning process is completed (i.e., either an equilibrium point is discovered or a maximal number of iterations T is reached), the predicted decision class is derived from the $\mathcal{A}_x(D)$ aggregated output vector.

5.5 Numerical simulations

In this section, we evaluate the performance of the ensemble approach using the three distance functions described in Chapter 2. More explicitly, we compare the prediction capability of the RCN-based classifiers using a fixed, reasonable similarity threshold (i.e., $\xi = 0.98$), the ensemble model without bagging (RCE) and the ensemble model performing instance bagging (RCB). In the case of ensemble variants, we use $N = 10$ base classifiers to keep the computational complexity low.

Appendix B outlines the properties of the 140 datasets adopted for simulation purposes (e.g., imbalance ratio, number of instances, attributes and decision classes). Each dataset has been partitioned using a 10-fold cross-validation procedure where the dataset has been split into ten folds, each containing 10% of the available objects. For each fold, an algorithm is trained with the objects contained into the training partition (90% of the available data) and then tested with the current fold. It should be mentioned that test partitions are kept aside to evaluate the performance of the learned hypothesis. An object will never be used for training and testing purposes at the same time, otherwise the prediction rates will be inflated.

Figure 5.2 summarizes the average Cohen's Kappa coefficient (Smeeton, 1985) achieved by each model. This coefficient measures the inter-rater agreement for categorical items and it is often deemed a more robust measure than the standard accuracy since it takes into account the agreement occurring by chance. Appendix C provides the full Kappa results attached to this simulation.

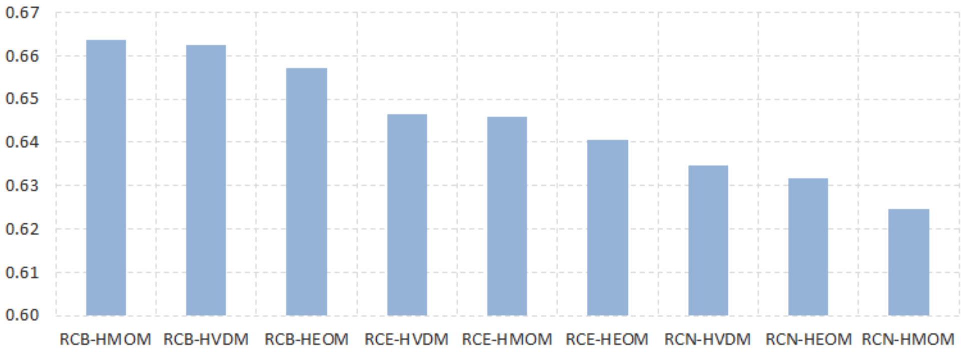


Figure 5.2: Average Kappa measure for RCE-based classifiers.

The preliminary results suggest that using an ensemble approach that performs instance bagging is actually convenient towards increasing the prediction rates of the rough cognitive models. In order to examine the existence of statistically significant differences in performance, we perform the Friedman two-way analysis of variances by ranks introduced in (Friedman, 1937). The Friedman test is a multiple-comparison nonparametric statistical method that detects whether at least two of the samples in a group represent populations with different median values or not.

This test suggests rejecting the null hypothesis (p-value $= 7.989819E-11 < 0.05$) using a confidence interval of 95%. This implies that there exist significant performance differences between at least two algorithms across all the selected datasets. Figure 5.3 shows the Friedman's rank values (the lower the better), where RCB using the HVDM distance function emerged as the best model.

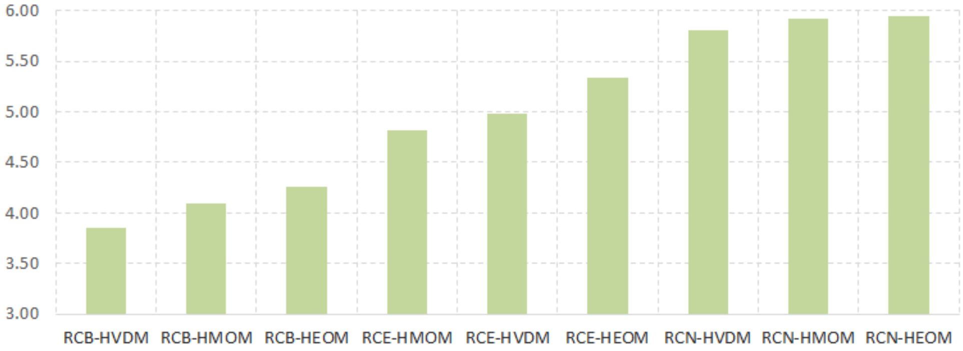


Figure 5.3: Friedman's rank values for RCE-based classifiers.

The next step is orienetd to determine whether the superiority of the RCB-HVDM classifier is statistically significant or not. By doing so, we resorted to the Wilcoxon signed rank test (Wilcoxon, 1945) and multiple post-hoc procedures that allow correcting the p-values, as was recently suggested in (Benavoli et al., 2016). The post-hoc procedures are required since in pairwise analysis, if we try to draw a conclusion involving more than one pairwise comparison, we accumulate an error coming from their combination. Therefore, we are losing control on the *Family-Wise Error Rate*, defined as the probability of making one or more false discoveries among the set of hypotheses associated to multiple pairwise tests.

Table 5.1 reports the p-value computed by the Wilcoxon signed rank test as well as the corrected ones associated with each pairwise comparison using RCB-HVDM as the control method. We assume that a null hypothesis H_0 can be rejected if at least one of the adopted post-hoc procedures supports the rejection. The statistical analysis supports the superiority of the RCB-HVDM classifier as all the conservative hypotheses were rejected, save for the one concerning to the RCB-HMOM vs. RCB-

HVDM pair. This suggests that RCB-HMOM and RCB-HVDM perform comparably although the RCB-HMOM algorithm is ranked first.

Table 5.1: Adjusted p-values using RCB-HVDM as the control method.

Algorithm	p-value	Bonferroni	Holm	Holland	Hypothesis
RCN-HEOM	6.67E-10	5.33E-09	5.33E-09	5.33E-09	Rejected
RCN-HVDM	3.44E-08	2.75E-07	2.40E-07	2.40E-07	Rejected
RCN-HMOM	5.59E-08	4.47E-07	3.35E-07	3.35E-07	Rejected
RCE-HEOM	1.75E-06	1.40E-05	8.76E-06	8.76E-06	Rejected
RCE-HVDM	0.001107	0.008861	0.004430	0.004423	Rejected
RCE-HMOM	0.003157	0.025261	0.009473	0.009443	Rejected
RCB-HEOM	0.010440	0.083525	0.020881	0.020772	Rejected
RCB-HMOM	0.848561	1.000000	0.848561	0.848561	Accepted

Figures 5.4 and 5.5 display the impact of using different heterogeneous distance functions and different ensemble strategies, respectively. Thus, we can conclude that performing instance bagging is convenient regardless of the adopted distance function. Second, it seems that selecting an ensemble scheme may reduce the negative effects of using a non-optimal distance function for a particular dataset.

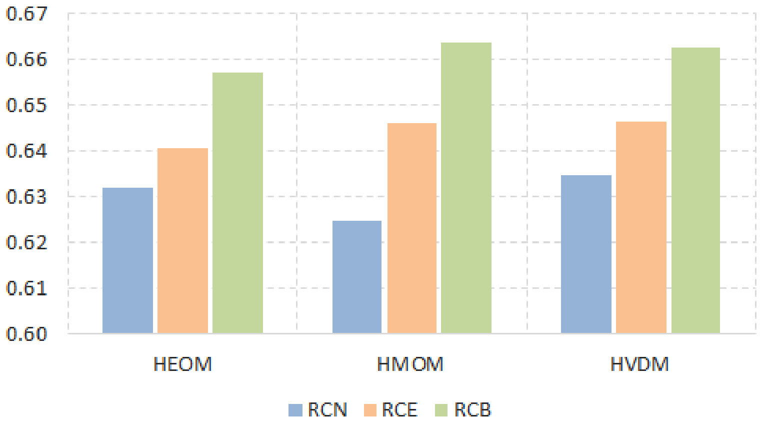


Figure 5.4: RCEs' average Kappa measure: *distance function.*

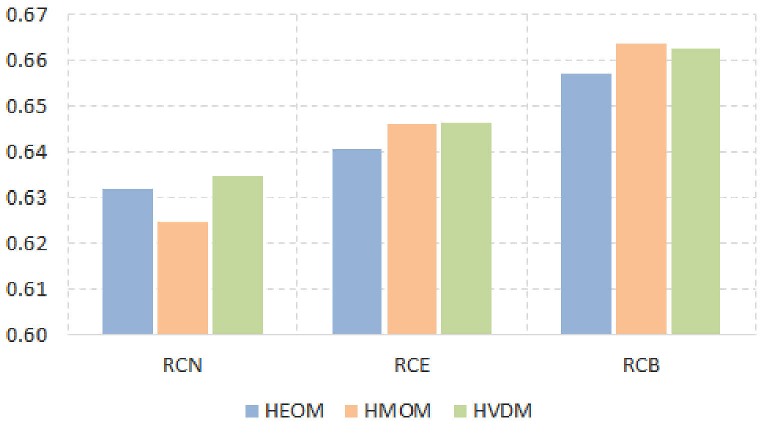


Figure 5.5: RCEs' average Kappa measure: *ensemble approach*.

It should be noticed that tuning the similarity threshold parameter (Nápoles et al., 2016) could lead to higher prediction rates, but it will excessively increase the computational complexity of building the (optimized) granular classifier. That is why this learning procedure is not included in this thesis.

5.6 Concluding note

The numerical simulations confirm the superiority of the RCE-based variants over the RCN algorithm discussed in the previous chapter. The proposed granular ensemble removed the need for a parameter tuning stage, which became the key motivation to explore ensemble techniques. By doing so, each base classifier operates at a different granularity degree while performing instance bagging to increase the overall diversity among the individual classifiers. In point of fact, the simulations have shown that the HMOM distance function in conjunction with the instance bagging strategy lead to higher prediction rates across benchmark problems.

Chapter 6

Fuzzy-Rough Cognitive Networks

This chapter presents the *Fuzzy-Rough Cognitive Networks*, the third method introduced in this research, which attempts addressing the parametric requirement when granulating the information space in the construction stage.

6.1 Preliminaries about the algorithm

Rough sets (Pawlak, 1982) and fuzzy sets (Zadeh, 1965) are two natural computing paradigms to deal with characteristics of imperfect data and knowledge in a human-like fashion. The former model provides approximations of a target set in the presence of incomplete information, characterizing those objects that certainly, and possibly, belong to the concept. The latter (often linguistic) model establishes that an object can belong to several sets or relations with different degrees.

Fuzzy sets have become a pivotal piece within the Soft Computing paradigm (Kecman, 2001) (Pratihar, 2015). This paradigm comprises a collection of techniques that are tolerant to typical characteristics of imperfect data and knowledge. During the last decades, new approaches generalizing the original fuzzy set theory (often called type-1 fuzzy set theory) have been developed. Type-2 fuzzy sets, intuitionistic fuzzy sets, interval-valued fuzzy sets and fuzzy-rough sets have in common that they can all be formally characterized by membership functions taking values in a partially ordered set (Cock et al., 2007), which is no longer the same (but an extension of) the set of membership degrees used in fuzzy set theory.

6.2 Motivation and challenges

In the previous chapter, we introduced the notion of *rough ensemble mapping* in an attempt to suppress the hyperparameter learning requirement related to the information granulation stage. Voting several classifiers by using an instance bagging scheme produces a combined classification model that often performs significantly better than the single model built from the original training data, and is never substantially worse. However, whether it is the best alternative is questionable.

For example, bagging follows the *bias-variance* decomposition principle (Witten and Frank, 2005) for an infinite number of training datasets. In real-world classification problems, there is only one training set, and acquiring more data is either impossible or expensive. To face this limitation, bagging performs a random sampling (with replacement) from the original training dataset to create a new one with the same size. The new (artificial) datasets are different from one another but are certainly not independent because they are all based on the same data source. Sometimes, we need to generate larger ensemble structures to address the lack of fresh training sets. Therefore, whether such larger structures lead to a good balance between prediction rates and computational efficiency may be questionable. In a nutshell, if the ensemble size is equal to the number of iteration required to estimate the similarity threshold, then there is no such claimed ensemble efficiency.

On the other hand, these combined models share the disadvantage of being rather hard to analyze: they can comprise several individual models, and although they perform well, it is not easy to understand in intuitive terms what factors are contributing to the improved decisions. This suggests that an ensemble approach may negatively affect the transparency of rough cognitive mapping.

This chapter explores the inclusion of fuzzy sets into the proposed granular classifier as an alternative to the ensemble scheme. Though, the goal remains unchangeable: to suppress the hyperparameter learning requirement when granulating the information space. Next, we review the foundations of fuzzy-rough sets.

6.3 Fuzzy-Rough Set Theory

In fuzzy-rough sets, rather than assessing objects' indiscernibility, we may measure their approximate equality. As a result, objects are categorized into classes, or granules, with *soft* boundaries based on their similarity to one another.

6.4 Fuzzy-rough cognitive mapping

The hybridization between rough sets and fuzzy sets was originally investigated by Dubois and Prade (Dubois and Prade, 1990), and later extended and/or modified by several authors. In this research, we adopt the approach proposed by Inuiguchi et al. (Inuiguchi et al., 2015) since it includes some mathematical properties that may be convenient when designing our classifier. Under these fuzzy conditions, objects are categorized into information granules with soft boundaries, and therefore, a strict similarity threshold is no longer required.

Let us assume a fuzzy set $X \in \mathcal{U}$ and a fuzzy binary relation R, where $\mu_X(x)$ and $\mu_R(y,x)$ are their membership functions, respectively. The function $\mu_X : \mathcal{U} \rightarrow [0,1]$ computes the degree to which $x \in \mathcal{U}$ is a member of X, whereas $\mu_R : \mathcal{U} \times \mathcal{U} \rightarrow [0,1]$ quantifies the degree to which y is presumed to be a member of X from the evidence that x is a member of the fuzzy set X. For the sake of simplicity, $R(x)$ is defined by its membership function, that is, $\mu_{R(x)}(y) = \mu_R(y,x)$.

Aiming at defining the fuzzy lower and upper approximation, we should consider the degree to which x is a member of X under the knowledge R. This can be measured by the truth value of statement '$y \in R(x)$ implies $y \in X$' under fuzzy sets $R(x)$ and X. More explicitly, the truth value can be computed by using a necessity measure $\inf_{y \in \mathcal{U}} \mathcal{I}(\mu_R(y,x), \mu_{X_k}(y))$ with an implication function $\mathcal{I} : [0,1] \times [0,1] \rightarrow [0,1]$ such that $\mathcal{I}(0,0) = \mathcal{I}(0,1) = \mathcal{I}(1,0) = \mathcal{I}(1,1) = 0$, where $\mathcal{I}(.,a)$ decreases and $\mathcal{I}(a,.)$ increases, $\forall a \in [0,1]$. Equation (6.1) displays the membership function defining the fuzzy lower approximation $R_*(X)$ associated to the fuzzy set X.

$$\mu_{R_*(X_k)}(x) = \min \left\{ \mu_{X_k}(x), \inf_{y \in \mathcal{U}} \mathcal{I}(\mu_R(y,x), \mu_{X_k}(y)) \right\} \tag{6.1}$$

Following the same reasoning, we can obtain a membership function for the upper approximation, assuming that X is a fuzzy set and R is a fuzzy binary relation. By doing so, we should measure the truth value of statement '$\exists y \in \mathcal{U}$ such that $x \in R(y)$' under fuzzy sets $R(x)$ and X. The true value of this statement can be obtained by a possibility measure $\sup_{y \in \mathcal{U}} \mathcal{T}(\mu_R(x,y), \mu_{X_k}(y))$ with a conjunction function $\mathcal{T} : [0,1] \times [0,1] \rightarrow [0,1]$ such that $\mathcal{T}(0,0) = \mathcal{T}(0,1) = \mathcal{T}(1,0) = \mathcal{T}(1,1) = 0$, where both $\mathcal{T}(.,a)$ and $\mathcal{T}(a,.)$ increase, $\forall a \in [0,1]$. Equation (6.2) shows the membership function for the upper approximation $R^*(X)$ associated to X.

$$\mu_{R^*(X_k)}(x) = \max \left\{ \mu_{X_k}(x), \sup_{y \in \mathcal{U}} \mathcal{T}(\mu_R(x,y), \mu_{X_k}(y)) \right\} \tag{6.2}$$

It should be remarked that the intersection of two fuzzy sets X and Y is regularly defined as $\mu_{X\cap Y} = \min\{\mu_X(x), \mu_Y(x)\}, \forall x \in \mathcal{U}$, while the union is commonly defined as $\mu_{X\cup Y} = \max\{\mu_X(x), \mu_Y(x)\}, \forall x \in \mathcal{U}$. However, some researchers replace the *min* operator with a t-norm and the *max* operator with a t-conorm (Inuiguchi et al., 2015). Besides, the Inuiguchi's model does not assume that $\mu_R(x,x) = 1, \forall x \in \mathcal{U}$. Instead, we compute the minimum between $\mu_X(x)$ and $\inf_{y\in\mathcal{U}} \mathcal{I}(\mu_R(y,x), \mu_{X_k}(y))$ when computing $\mu_{R_*(X_k)}(x)$, and the maximum between $\mu_X(x)$ and $\sup_{y\in\mathcal{U}} \mathcal{T}(\mu_R(x,y), \mu_{X_k}(y))$ when computing $\mu_{R^*(X_k)}(x)$. This allows preserving the inclusiveness of $R_*(X)$ into the fuzzy set X and the inclusiveness of X into $R^*(X)$.

Based on the above elements, one can easily define the three fuzzy-rough regions. Equation (6.3), (6.4) and (6.5) formalize the membership functions associated to the positive, negative and boundary regions, respectively.

$$\mu_{POS(X_k)}(x) = \mu_{R_*(X_k)}(x) \tag{6.3}$$

$$\mu_{NEG(X_k)}(x) = 1 - \mu_{R_*(X_k)}(x) \tag{6.4}$$

$$\mu_{BND(X_k)}(x) = \mu_{R^*(X_k)}(x) - \mu_{R_*(X_k)}(x) \tag{6.5}$$

These memberships functions allow completely removing the threshold, therefore leading to the fuzzy-rough modeling. In this model, abrupt transitions between classes are replaced by gradual ones, allowing that an element can belong to several classes with different degrees. In the next section, we explain how to exploit such fuzzy-rough granules by using a cognitive neural network.

6.4.1 Information granulation

Fuzzy-Rough Cognitive Networks (FRCNs) transform the attribute space into a fuzzy-rough one. This result in fuzzy-rough information granules that will be exploited by a recurrent neural network. Similarly to the RCN model, the first step when building an FRCN is related to the granulation of available information.

Let $X = \{X_1, \ldots, X_k, \ldots, X_K\}$ be a partition of the universe $\mathcal{U}$ according to the values of the decision attribute, such that the subset X_k comprises those objects labeled as D_k. Based on this partition, we can straightforwardly define the membership degree of $y \in \mathcal{U}$ to a subset X_k (see Equation 6.6). Observe that we assume that all objects labeled as D_k have maximal membership degree to the kth subset; however, more sophisticated variants can be formalized as well.

$$\mu_{X_k}(y) = \begin{cases} 1 & \text{if } y \in X_k \\ 0 & \text{if } y \notin X_k \end{cases} \tag{6.6}$$

Another element to be defined is the membership function $\mu_R(y, x)$ associated to the fuzzy binary relation. Equation 6.7 displays the function adopted in this thesis, which depends on the membership degree of object x to X, and the similarity degree between x and y. In this paper, the similarity degree $\varphi(x, y)$ denotes the complement of a normalized distance $\delta(x, y)$ between objects x and y.

$$\mu_R(y, x) = \mu_{X_k}(x)(1 - \delta(x, y)) \tag{6.7}$$

Granulating the information space in this approach is equivalent to i) computing the membership functions attached to the lower and upper fuzzy approximations (i.e., $\mu_{R_*(X_k)}(x)$ and $\mu_{R^*(X_k)}(x)$) and consequently ii) computing the membership functions associated to the positive, negative and boundary fuzzy regions (i.e., $\mu_{POS(X_k)}(x)$, $\mu_{NEG(X_k)}(x)$ and $\mu_{BND(X_k)}(x)$). The fuzzy granulation step produces a covering of the universe of discourse defined by membership functions.

6.4.2 Network construction

After granulating the information space, the resultant fuzzy-rough constructs are used to build an FCM-based neural network. Similarly to RCN models, input neurons denote positive or negative fuzzy-rough regions, whereas output neurons comprise the decision classes for the problem at hand. In this granular model, including the fuzzy-rough boundary regions into the model does not significantly increase the classifier's discriminatory ability (Nápoles et al., 2017b). This behavior is not surprising because in crisp-rough environments the hesitant evidence is more conclusive when compared to the evidence coming from fuzzy-rough granules. Thus, the neural network topology can be designed by using the following construction rules:

- (R_1^*) IF $C_i = P_k^*$ AND $C_j = D_k$ THEN $w_{ij} = 1.0$
- (R_2^*) IF $C_i = N_k^*$ AND $C_j = D_k$ THEN $w_{ij} = -1.0$
- (R_2^*) IF $C_i = P_k^*$ AND $C_j = D_{v \neq k}$ THEN $w_{ij} = -1.0$
- (R_4^*) IF $C_i = P_k^*$ AND $C_j = P_{v \neq k}$ THEN $w_{ij} = -1.0$

where C_i is the ith neural processing entity, D_k denotes the kth decision class, while P_k^* and N_k^* are neurons denoting the positive and negative rough-fuzzy region associated to the kth decision class, respectively.

Figure 6.1 portrays the network topology associated to the FRCN classifier for any binary classification problem. More generically, any FRCN comprises $2|\mathcal{D}|$ input neurons, $|\mathcal{D}|$ output neurons and $|\mathcal{D}|(4+|\mathcal{D}|)$ causal relations, with $\mathcal{D}$ being the set of all possible decision classes. Observe that, unlike neural networks that their complexity depends on the number of features, the number of neurons in an FRCN network is determined by the number of decision classes.

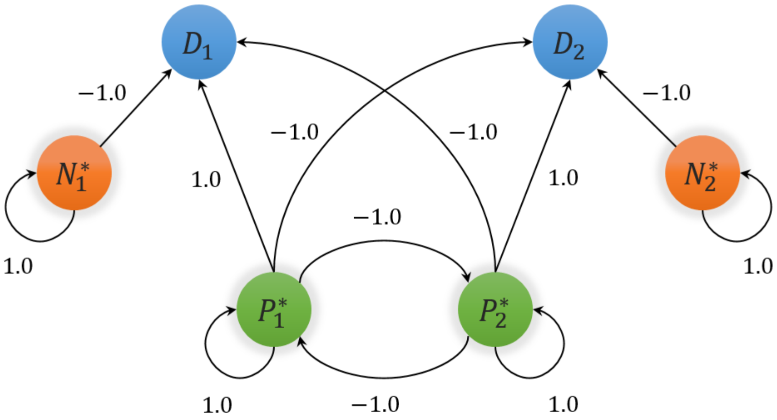


Figure 6.1: Fuzzy-Rough Cognitive Network for 2-class problems.

6.4.3 Network exploitation

Once the granular network has been constructed, we can determine the decision class for unlabeled objects by activating the input neurons and performing the reasoning process. To activate the neurons, we use the similarity degree between the unlabeled object x and $y \in \mathcal{U}$ as well as the membership degree of y to each fuzzy-rough granular region. Figure 6.2 and 6.3 show the semantics behind this activation mechanism for the kth positive and negative granule, respectively.

More explicitly, such figures illustrate how to determine the degree to which the unlabeled object belongs to the fuzzy intersection defined from the membership functions $\mu_{POS(X_k)}(y)$ (or $\mu_{NEG(X_k)}(y)$), and the fuzzy similarity relation between x and $y \in X$. The excitation of the kth input neuron is given by the inclusion degree of the fuzzy intersection set into the kth fuzzy-rough region.

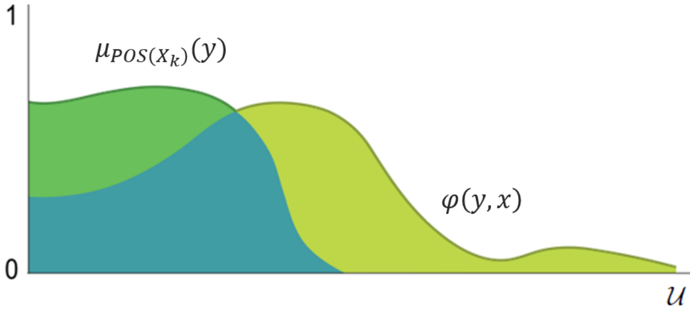


Figure 6.2: Inclusion degree of x into the kth positive region.

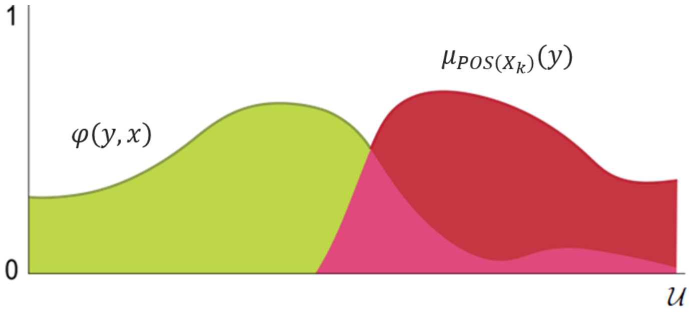


Figure 6.3: Inclusion degree of x into the kth negative region.

Equation (6.8) formalizes how to compute the excitation degree of the kth positive neuron, where $\mathcal{T}_2$ denotes a t-norm[1], $\varphi(y,x)$ is the similarity degree between y and x whereas $\mu_{POS(X_k)}(y)$ represents the membership grade of y to the kth rough-fuzzy positive granule. Using an equivalent equation, we can activate input neurons denoting fuzzy-rough negative regions. Output neurons remain inactive at the outset of the reasoning process since they are used to collect the evidence coming from propagating the initial information though the causal network.

$$\mathcal{A}(P_k^*) = \frac{\int \mathcal{T}_2(\varphi(y,x), \mu_{POS(X_k)}(y))dy}{\int \mu_{POS(X_k)}(y)dy} \tag{6.8}$$

[1] A t-norm operator is a conjunction function $\mathcal{T} : [0,1] \times [0,1] \to [0,1]$ that must fulfill three conditions: i) $\forall a \in [0,1], \mathcal{T}(a,1) = \mathcal{T}(1,a) = a$, (ii) $\forall a,b \in [0,1], \mathcal{T}(a,b) = \mathcal{T}(b,a)$, and (iii) $\forall a,b,c \in [0,1], \mathcal{T}(a,\mathcal{T}(b,c)) = \mathcal{T}(\mathcal{T}(a,b),c)$.

Due to the fact that the universe of discourse $\mathcal{U}$ is finite in real-world problems, the use of integrals may not be convenient. Rules (R_5^*) and (R_6^*) display a more practical mechanism to activate the granular neural network.

- (R_5^*) IF $C_i = P_k^*$ THEN $A_i^{(0)} = \frac{\sum_{y \in \mathcal{U}} \mathcal{T}_2(\varphi(y,x), \mu_{POS(X_k)}(y))}{\sum_{y \in \mathcal{U}} \mu_{POS(X_k)}(y)}$
- (R_6^*) IF $C_i = N_k^*$ THEN $A_i^{(0)} = \frac{\sum_{y \in \mathcal{U}} \mathcal{T}_2(\varphi(y,x), \mu_{NEG(X_k)}(y))}{\sum_{y \in \mathcal{U}} \mu_{NEG(X_k)}(y)}$

Once the initial activation vector $A^{(0)}$ associated to the unlabeled object y has been computed, we perform the neural reasoning process until (i) a fixed-point attractor is discovered, or alternatively (ii) a maximal number of iterations is reached. Afterwards, the label of the output neuron having the highest activation value is assigned to the object. In this classification scheme, ties are broken using a nearest neighbor approach: the model chooses the closest instance (i.e., an object from the universe) to the test object and returns the attached decision class.

6.5 Numerical simulations

This section is focused on determining the combination of fuzzy operators leading to improved prediction rates for FRCN-based models. Tables 6.1 and 6.2 display the t-norms and fuzzy implicators used in our simulations.

Table 6.1: Fuzzy operators explored in the FRCN algorithm: *t-norms*.

T-norm	**Formulation**
Standard intersection	$\mathcal{T}(x,y) = \min\{x,y\}$
Algebraic product	$\mathcal{T}(x,y) = xy$
Lukasiewicz	$\mathcal{T}(x,y) = \max\{0, x+y-1\}$
Drastic product	$\mathcal{T}(x,y) = \begin{cases} x & , y = 1 \\ y & , x = 1 \\ 0 & , otherwise \end{cases}$

In a nutshell, the FRCN algorithm requires to define a fuzzy implicator and two t-norms. The $\mathcal{I}$ implicator is used to compute the membership degree of an object to

Table 6.2: Fuzzy operators explored in the FRCN algorithm: *implicators*.

Implicator	**Formulation**
Standard	$\mathcal{I}(x,y) = \begin{cases} 1 & , x \leq y \\ 0 & , x > y \end{cases}$
Kleene-Dienes	$\mathcal{I}(x,y) = \max\{1-x, y\}$
Lukasiewicz	$\mathcal{I}(x,y) = \min\{1-x+y, 1\}$
Zadeh	$\mathcal{I}(x,y) = \max\{1-x, \min\{x,y\}\}$
Godel	$\mathcal{I}(x,y) = \begin{cases} 1 & , x \leq y \\ y & , x > y \end{cases}$
Larsen	$\mathcal{I}(x,y) = xy$
Mamdani	$\mathcal{I}(x,y) = \min\{x,y\}$
Reichenbach	$\mathcal{I}(x,y) = 1-x+xy$
Yager	$\mathcal{I}(x,y) = \begin{cases} 1 & , x = y = 0 \\ y^x & , otherwise \end{cases}$
Goguen	$\mathcal{I}(x,y) = \begin{cases} 1 & , x \leq y \\ y/x & , otherwise \end{cases}$

the lower approximations, the $\mathcal{T}_1$ t-norm is used to compute the membership degree of an object to the upper approximations, whereas the $\mathcal{T}_2$ t-norm is used to activate the neural processing entities. For the sake of simplicity, we use the same t-norm to compute the membership degree to the upper approximations as well as to exploit the neural network. Similarly to the ensemble simulations, we adopt the Kappa coefficient to measure the performance of the fuzzy-rough models.

Figure 6.4 portrays the average Kappa coefficient achieved by each fuzzy-rough granular classifier for different combinations of t-norms and fuzzy implicators, using the HMOM distance as the standard dissimilarity function. Appendix D sows the full Kappa results attached to this numerical simulation.

It should be noticed that the FRCN classifier computes the best prediction rates when using the Lukasiewicz t-norm to activate the network, regardless of the operator attached to the membership functions $\mu_{R_*(X_k)}(x)$ and $\mu_{R^*(X_k)}(x)$. Thus, hereinafter we will use the Lukasiewicz t-norm and the Lukasiewicz implicator as standard fuzzy operators in all simulations conducted in this research.

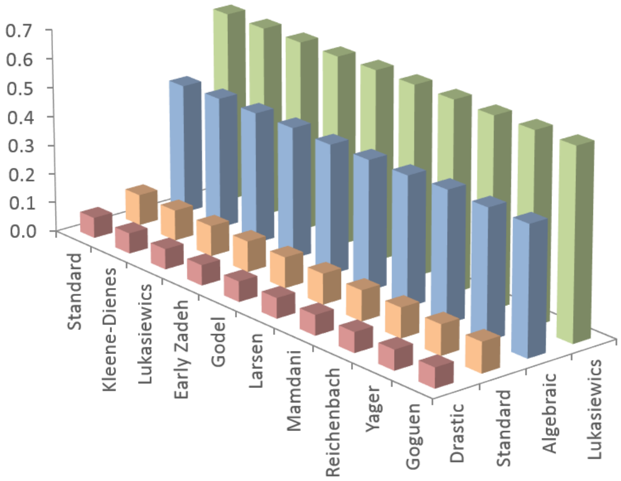


Figure 6.4: Average Kappa measure for FRCN-based classifiers.

Next, we compare the prediction capability of the proposed fuzzy-rough classifier with regards to the crisp model using the distance functions presented in Chapter 2. Figures 6.5 and 6.6 summarize the average Kappa measure attained by each variant. The results confirm that the fuzzy-rough models always report higher prediction rates regardless of the adopted distance function. Besides, in the fuzzy configuration, the HMOM distance function seems to stand as the best choice.

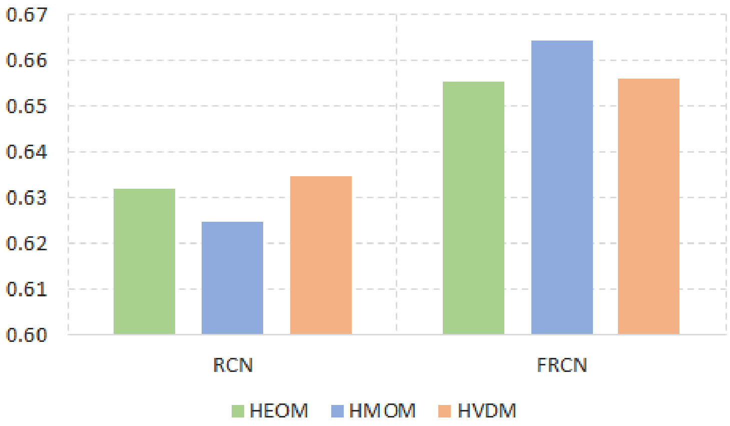


Figure 6.5: FRCNs' average Kappa measure: *distance function.*

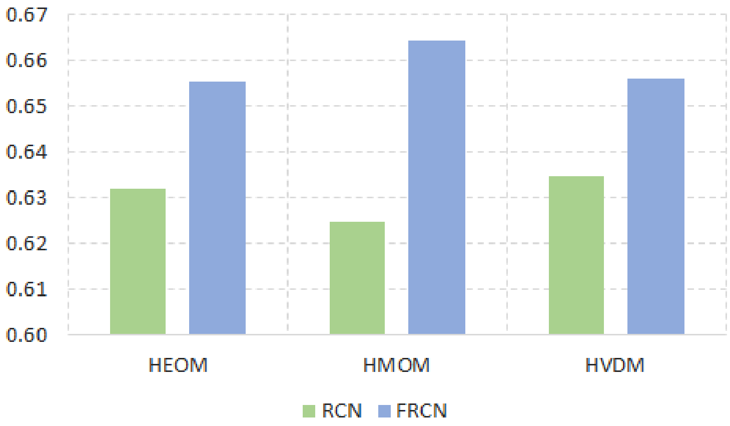


Figure 6.6: FRCNs' average Kappa measure: *fuzzy approach*.

Aiming at performing a more rigorous statistical analysis, we compute the Friedman two-way analysis of variances by ranks. The test suggests rejecting the conservative hypothesis (p-value $= 8.12680478E - 10 < 0.05$) using a confidence interval of 95%. Therefore, we conclude that there exist significant differences between at least two algorithms. Figure 6.7 reports the rank values (the lower the better) where the FRCN-HMOM variant stands as the best-performing classifier.

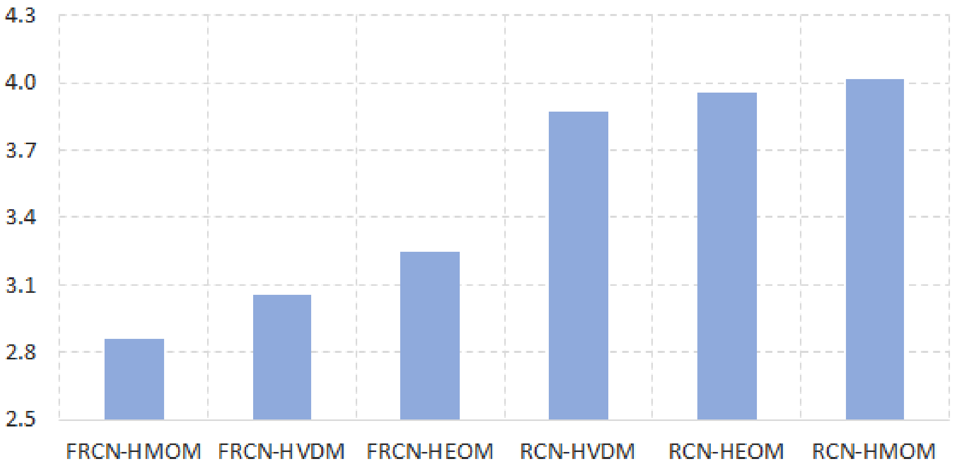


Figure 6.7: Friedman's rank values for RCE-based classifiers.

Table 6.3 reports the p-value computed by the Wilcoxon test and the corrected p-values associated with each pairwise comparison using FRCN-HMOM as the control method. Similarly, we assume that a null hypothesis can be rejected if at least one of the post-hoc procedures advocates for the rejection. The statistical analysis confirms FRCN-HMOM's superiority as all the null hypotheses were rejected.

Table 6.3: Adjusted p-values using FRCN-HMOM as the control method.

Algorithm	p-value	Bonferroni	Holm	Holland	Hypothesis
RCN-HEOM	2.15E-07	0.000001	0.000001	0.000001	Rejected
RCN-HMOM	2.50E-07	0.000001	0.000001	0.000001	Rejected
RCN-HVDM	0.000003	0.000015	0.000009	0.000009	Rejected
FRCN-HEOM	0.000076	0.000380	0.000152	0.000152	Rejected
FRCN-HVDM	0.007897	0.039485	0.007897	0.007897	Rejected

Likewise, the fuzzy-rough classifier also suppresses the requirement of estimating a similarity threshold when comparing to the base RCN model. This implies that the parameter tuning step is no longer needed, thus significantly increasing the practical usability of *rough cognitive mapping* in classification scenarios.

6.6 Concluding note

The simulations confirm that FRCNs are a suitable alternative to deal with the parametric requirement of rough cognitive mapping. In this model, we can understand the decision process by using fuzzy inclusion equations and causal relations. The results also suggest that the HMOM distance function in conjunction with the Lukasiewicz operators lead to improved prediction rates across datasets adopted for simulation. It is worth mentioning that the fuzzy classifier seems to be more convenient than the ensemble model because its simpler structure. In the next chapter we provide further reasons to adopt the fuzzy-rough classifier in the remaining simulations, which include a statistical comparison between both models.

Chapter 7

Further simulations

This chapter is devoted to evaluating the discriminatory power of the proposed classification model. More explicitly, we carry out an exhaustive comparative study against state-of-the-art classifiers across benchmark datasets.

7.1 Benchmark datasets summary

Similarly to the previous simulations, we leaned upon 140 classification datasets taken from the KEEL (Alcalá et al., 2010) and UCI Machine Learning (Lichman, 2013) repositories. These problems comprise different characteristics and allow evaluating the predictive capability of both state-of-the-art and granular classifiers under consideration. Appendix B outlines the properties of these datasets (e.g., imbalance ratio, number of instances, attributes and decision classes).

In these ML problems, the number of attributes ranges from 2 to 262, the number of decision classes from 2 to 100, and the number of objects from 14 to 12,906. The benchmark set includes 13 noisy and 47 imbalanced datasets, with the imbalance ratio fluctuating between 5:1 and 2160:1. In order to avoid out-of-range situations in the distance functions, the numerical attributes have been normalized. On the other hand, we replaced missing values with the mean or the mode depending on whether the attribute was numerical or nominal, respectively.

7.2 Comparison against traditional classifiers

The above simulations have shown that both RCEs and FRCNs are capable of outperforming RCNs using a fixed (reasonable) threshold. As a result, the estimation of the similarity threshold parameter is no longer a concern. But are they similar in performance? To answer this question we applied the Wilcoxon signed rank test for pairwise comparisons. The test suggests accepting the conservative hypothesis (p-value=0.7387 $>$ 0.05) using a confidence interval of 95%. Therefore, we can conclude that both approaches perform similarly for adopted datasets.

However, the fuzzy approach is preferred since it fits the parsimony principle: *the simpler the better*. Besides, the bagging scheme and the ensemble model itself make the RCE algorithm less transparent than the fuzzy variant, thus notably reducing one of the main contributions attached to the rough cognitive mapping. That is why we decided to use the FRCN-HMOM algorithm (hereinafter simply called FRCN) in the remaining numerical simulations performed in this chapter.

7.2.1 Classifiers adopted for comparison

As a further simulation, we compare the FRCN algorithm against several well-known state-of-the-art classifiers in terms of prediction rates. The traditional classifiers are gathered in six categories that are depicted as follows:

- **Rule-based models**
  - *Decision Table (DT)* (Kohavi, 1995). The algorithm searches for matches in the body using a subset of attributes. If no instances are found, the majority class in the table is returned; otherwise, the majority class of all matching instances is returned.
- **Bayesian models**
  - *Naïve Bayes (NB)* (John and Langley, 1995). A probabilistic classification algorithm using estimator classes, where numeric estimator precision values are chosen based on the analysis of the training data.
  - *Naïve Bayes Updateable (NBU)* (John and Langley, 1995). Implements an incremental NB classifier that learns one instance at a time. Instead of using normal density measures for numerical attributes, this algorithm employs a kernel estimator without discretization.

- **Function-based models**
  - *Simple Logistic (SL)* (Sumner et al., 2005). A classifier building linear logistic regression models. LogitBoost with simple regression functions as base learners is used for fitting the logistic models.
  - *Multilayer Perceptron (MLP)* (Hecht-Nielsen, 1989). Neural network that uses the backpropagation algorithm to train the model.
  - *Support Vector Machines (SMO).* (Keerthi et al., 2001) Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier. In our research, we adopted a quadratic polynomial kernel to perform the numerical simulations.
- **Tree-based models**
  - *Decision Tree (J48)* (Quinlan, 1986). Induces classification rules in the form of a pruned/unpruned decision tree.
  - *Random Tree (RT)* (Amit and Geman, 1997). Decision tree without pruning that considers k randomly chosen attributes at each node.
  - *Random Forest (RF)* (Breiman, 2001). Bagging of random trees.
  - *Fast Decision Tree (FDT)* (Su and Zhang, 2006). Builds a tree using information gain and prunes it using reduced-error pruning.
  - *Best-first Decision Tree (BFT)* (Shi, 2007). Classification trees that use binary split for both nominal and numeric attributes.
  - *Logistic Model Tree (LMT)* (Landwehr et al., 2005). Decision trees for classification that use logistic regression functions at the leaves.
- **Instance-based models**
  - *Nearest Neighbor (NN)* (Aha et al., 1991). Instance-based (lazy) classifier that simply chooses the closest instance (i.e., an object from the universe) to the test instance and returns its class.
  - *k-Nearest Neighbors (kNN)* (Aha et al., 1991). Lazy learner that computes the class based upon the classes of the k training instances that are most similar to the test instance, as determined by a similarity function.

  - *K^* classifier (K^*)* (Cleary et al., 1995). Instance-based classifier similar to kNN that uses an entropy-based distance function.
  - *Locally Weighted Learning (LWL)* (Atkeson et al., 1997). An instance-based algorithm that assigns weights to the problem instances, which are then used by a specified weight instance handler.

- **Fuzzy-rough models**
  - *Fuzzy-Rough k-Nearest Neighbors (FRNN)* (Jensen and Cornelis, 2008). Nearest neighbor model that utilizes the lower and upper approximations from fuzzy rough set theory to classify test instances.
  - *Vaguely-quantified k-Nearest Neighbors (VQNN)* (Jensen and Cornelis, 2011). Fuzzy-rough model that emulates the linguistic quantifiers *some* and *most* when performing the classification process.

It should be mentioned that we retain the default parameter settings implemented in Weka v3.6.11 (Hall et al., 2009) during the simulations, as no algorithm performs parameter tuning. In (Triguero et al., 2015) the authors correctly stated that a proper parametric setting increases the algorithms' performance over different data sources. Nevertheless, a robust classifier should produce good results even when its parameters might not have been optimized for a specific problem.

7.2.2 Statistical analysis and discussion

Analogously to the previous simulations, we utilize the Kappa coefficient to quantify the algorithms' performance. Figure 7.1 shows the average Kappa measure attained by each algorithm across benchmark problems. Likewise, the reader can find the full Kappa results attached to this simulation in Appendix E.

The results have shown that LMT is the best-performing classifier, whereas FRCN arises as the second-best ranked algorithm. The LMT classifier is another ensemble that use boosting to build very effective decision trees with logic functions in the leaves. More explicitly, it uses the LogitBoost algorithm to induce trees with linear-logistic regression models at the leaves. However, it uses an internal cross-validation process at each subset to determine the proper number of iterations for the boosting procedure. This process is equivalent to find the optimal similarity threshold value in rough cognitive models through a hyperlearning algorithm.

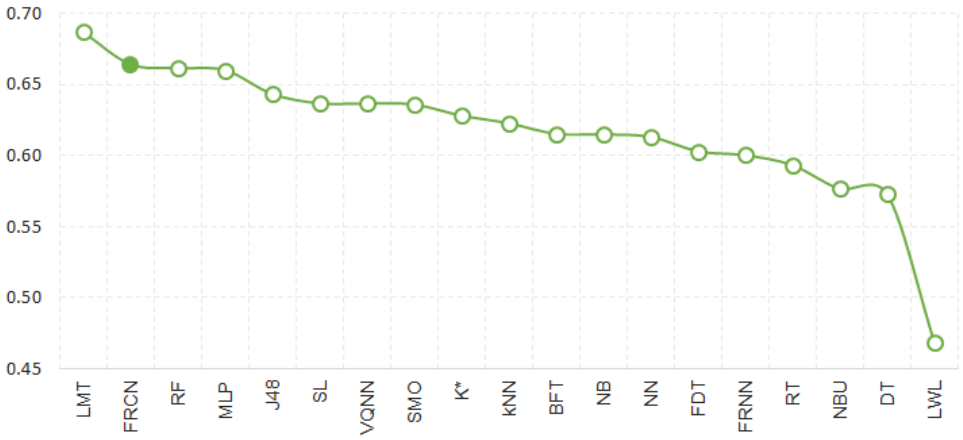


Figure 7.1: Average Kappa measure for traditional classifiers.

In this experiment, the Friedman test advocates for rejection of the conservative hypothesis (p-value $= 1.4396928E - 10 < 0.1$) using a confidence level of 90%. This suggests that there are significant differences between at least two algorithms across the selected datasets. Figure 7.2 portrays the rank values computed by the Friedman test. It is clear that LMT is the best-ranked algorithm, FRCN is the fourth-best ranked, while the LWL method is ranked at the last position.

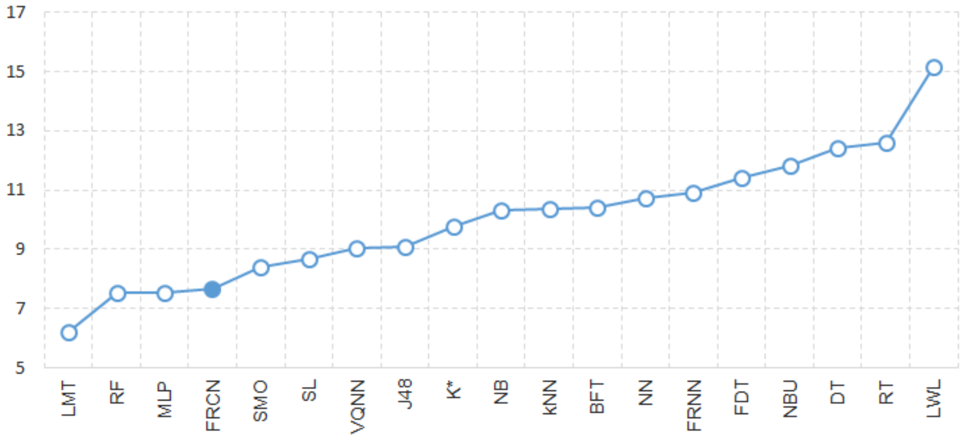


Figure 7.2: Friedman's rank values for traditional classifiers.

Table 7.1 summarizes the p-values reported by the Wilcoxon signed rank test and the corrected p-values according to the post-hoc procedures using the FRCN model as the control method. The adjusted p-values confirm that the LMT ensemble is the best-performing classifier in our study, with no significant differences spotted between the proposed fuzzy-rough cognitive model and MLP, RF, SMO and SL, as the null hypothesis was accepted in each of these pairwise comparisons. Observe however that none of these high-performance models provide an introspection mechanism into their decision model. In the case of the FRCN algorithm, the transparency is achieved by using fuzzy inclusion equations and causal relations.

Table 7.1: Adjusted p-values using FRCN as the control method.

Algorithm	p-value	Bonferroni	Holm	Holland	Hypothesis
LWL	7.69E-18	1.38E-16	1.38E-16	0.000000	Rejected
RT	1.29E-11	2.34E-10	2.21E-10	2.21E-10	Rejected
DT	4.94E-11	8.91E-10	7.92E-10	7.91E-10	Rejected
NBU	3.79E-08	6.82E-07	5.68E-07	5.68E-07	Rejected
FDT	6.31E-07	1.13E-05	0.000008	8.83E-06	Rejected
FRNN	8.94E-07	0.000016	0.000011	1.16E-05	Rejected
NN	2.15E-06	0.000038	0.000025	2.57E-05	Rejected
NB	5.51E-06	0.000099	0.000060	6.06E-05	Rejected
BFT	3.20E-05	0.000576	0.000320	0.000319	Rejected
kNN	7.13E-05	0.001285	0.000642	0.000642	Rejected
K^*	0.005752	0.103543	0.046019	0.045103	Rejected
LMT	0.006376	0.114778	0.046019	0.045103	Rejected
J48	0.010528	0.189511	0.063170	0.061531	Rejected
VQNN	0.010947	0.197052	0.063170	0.061531	Rejected
SL	0.109578	1.000000	0.438314	0.371388	Accepted
SMO	0.273587	1.000000	0.820761	0.616689	Accepted
RF	0.940694	1.000000	1.000000	0.996482	Accepted
MLP	1.000000	1.000000	1.000000	1.000000	Accepted

From the above experiments, we can conclude that the FRCN algorithm allows suppressing the parameter learning method of rough cognitive mapping while outperforming most state-of-the-art classifiers used for comparison. It is equally noticeable the superiority of FRCN when compared with other instance-based learners such as kNN, K^*, LWL or FRNN. On the other hand, the complexity of constructing and exploiting an FRCN is $O(|\mathcal{U}_2||\mathcal{U}_1||\Phi|\})$, where $|\Phi|$ is the attribute set, whereas $\mathcal{U}_1$ and $\mathcal{U}_2$ denote the training and test set, respectively. This means that the proposed model requires less computational effort to produce accurate predictions with regards to the crisp variant constructed over a fixed similarity threshold.

7.3 When is our classifier the best choice?

In our last experiment, we attempt characterizing the scenarios on which the granular classifier outperforms a baseline model. In this case, we select the k-nearest neighbors classifier (Aha et al., 1991) as both algorithms are instance-based models. In order to discover these patterns, we learn a low-level interpretable classifier from a generated dataset. Being more explicit, we construct a synthetic dataset with instances comprising five descriptive attributes (i.e., number of instances, number of attributes, number of classes, whether the problem is noisy or not, and the imbalance ratio) and two decision classes (P - *positive*, N - *negative*). In a nutshell, an object describes the features of a particular benchmark dataset and whether our model was capable of outperforming the baseline model for that problem.

We labeled an instance as *positive* if $\kappa(RCN) - \kappa(kNN) \geq 0.05$, with $\kappa(.)$ being the kappa coefficient computed by a particular problem. If $\kappa(RCN) - \kappa(kNN) \leq -0.05$ then the instance is labeled as *negative*, while problems where $-0.05 < |\kappa(RCN) - \kappa(kNN)| < 0.05$ were not included in the study to achieve a better separability between classes. As a result, we obtained a dataset comprised of 63 valid instances (i.e., 47 labeled as positive and 16 labeled as negative).

As practical example, the *anneal* problem (see Appendix B) includes 898 objects, 38 attributes, 6 decision classes. Moreover, this dataset has no noise but a high degree of imbalance (85:1). Due to the fact that $\kappa(RCN) = 0.983$ and $\kappa(kNN) = 0.920$ we can infer the following synthetic instance: “898, 38, 6, no, high, P”. These instances have three numerical attributes, two nominal attributes and the (nominal) decision class. Of course, other features can be considered as well.

Once the dataset has been created, we train a decision tree model (Quinlan, 1986) to predict whether the proposed classifier will be suitable for solving a new classification problem or not. Decision tree classifiers derive a rule set that allow interpreting the prediction results at a low-level. Figure 7.3 displays the unpruned decision tree obtained after running the underlying learning scheme.

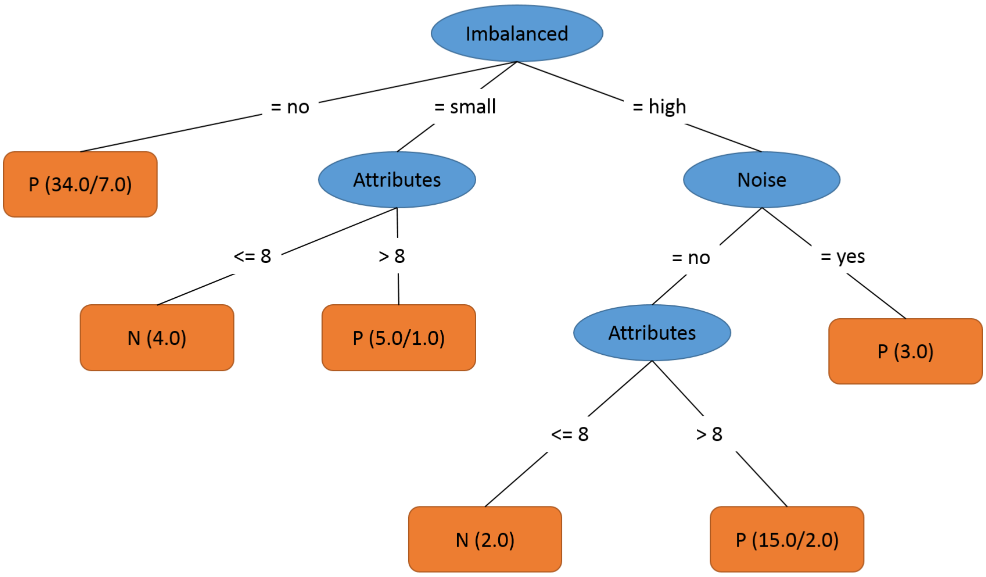


Figure 7.3: Decision tree characterizing the model performance.

The decision tree classifier yielded an 86% accuracy, which can be interpreted as the confidence of the derived rule set. These rules have shown that the granular classifier outperforms the kNN algorithm in *rather balanced problems*, or alternatively in those *problems that exhibit a certain imbalance degree but are defined by a reasonable number of attributes*. Furthermore, the derived rules have shown that our classifier seems to be less sensitive to noise effects. Despite such promising findings, one future research direction will be focused on determining other potentialities behind this novel rough classifier. In that sense, further work need to be done.

Chapter 8

Concluding remarks

In this chapter, we outline the main contributions, the key results and future research directions to be accomplished in a near future

8.1 Contributions and discussion

In recent years, the increasing amount of available data has motivated researchers to introduce new classification models to acquire, process and interpret the problem data using more symbolic approaches. Another problem that notably affects the performance of traditional classifiers is the quality of data (i.e., partial truth, inconsistency, noise, vague and imprecise information). The inherent difficulty attached to the above issues becomes more evident in modern decision-making problems where experts are interested on both *accuracy* and *transparency*. Regrettably, most accurate classifiers (e.g., Random Forests, Neural Networks or Support Vector Machines) do not provide an intrinsic introspection mechanism into its inference process, thus performing like black-boxes. That is why further approaches are required.

In this research, we have presented the notion of *rough cognitive mapping* in the context of pattern classification. *Rough Cognitive Networks* are granular neural networks that allow elucidating their reasoning process using inclusion degree equations and causal relations. While we were focused on developing a granular classifier capable of attaining competitive prediction rates, other relevant contributions came to the light. For example, our approach comprises a suitable alternative to automatically construct high-level FCM-based systems from historical records, hence overcoming the construction issues of cognitive mapping. This suggests that the human intervention is not required when deriving the classifier structure.

The first rough classifier presented in this thesis comprised a suitable model towards solving classification problems in a comprehensible way. Nevertheless, its performance was highly conditioned by the threshold defining whether two objects can be considered similar or not. The first (perhaps instuitive) alternative presented in (Nápoles et al., 2016b) (Nápoles et al., 2016) was to estimate the value of this parameter in a supervised fashion. Although it proved to be effective in terms of accuracy, this harsh approach increased the computational time required to build the network, thus reducing the practical usability of our classifier.

Aiming at overcoming this problem from a more elegant perspective, we introduced two RCN-based classifiers: *Rough Cognitive Ensembles* and *Fuzzy-Rough Cognitive Networks*. The former is a granular ensemble model where each base RCN operates at a different granularity degree, whereas the latter replaced the crisp-rough constructs with fuzzy-rough ones. Numerical simulations have shown that both approaches are capable of outperforming the primary RCN model using a (reasonable) fixed similarity threshold. Likewise, the results also showed that the Lukasiewicz operators and the HMOM distance function report the best prediction rates.

Although both variants perform comparably, the reader can easily perceive why the fuzzy approach is preferred: we can achieve the same accuracy using an ensemble composed of ten crisp networks that using a single fuzzy-rough classifier! Furthermore, the fuzzy-rough classifier allows smoothly elucidating its decision process based on its foundations: fuzzy inclusion degrees and causal relations.

The comparison against state-of-the-art classifiers have shown that there are no significant differences between our model and the best-performing methods (i.e., SL, MLP, SVM and RF), with the exception of the LMT ensemble that stands as the best classifier in the study. Nonetheless, as already mentioned, these best-performing algorithms perform like black-boxes, and thus they are unable to plainly explain why a specific decision class is assigned to an unlabeled object.

It is worth mentioning that the proposed granular classifier performs better than other instance-based learners (e.g., kNN or K*) across selected datasets. This may be a direct result of its ability to infer a decision class when no positive evidence is available, i.e., based on the negative and hesitant information. Further simulations have shown that our algorithm outperforms the kNN algorithm in 1) rather balanced problems, or alternatively in 2) those problems that exhibit some imbalance degree but are defined by a reasonable number of attributes.

8.2 Future research lines

Of course, rough cognitive mapping is *no panacea*. While the theoretical foundations behind FRCNs (the best-performing model) seem quite intuitive for mathematicians, it may not be intuitive enough for decision-makers with no background in computer science or related areas. Therefore, as a future research work, we will explore new strategies to automatically construct low-level FCM-based classifiers from historical data. Deriving FCM-based models with lower abstraction levels leads to more transparent classifiers, but the accuracy may be compromised.

Furthermore, the automatic construction of low-level FCM-based models from data is challenging and remains an open problem. Existing construction procedures for cognitive mapping are unable to derive authentic causal structures. This happens because they are oriented to fit the network to the historical data, without considering the domain semantics. Some authors attempted overcoming this drawback using correlation measures, which fail in capturing the conceptual meaning of causal relations. Being more explicit, it is well-known that causality does surely imply the existence of correlation, but the opposite does not necessarily hold.

The notion of rough cognitive mapping opens new research avenues toward solving more complex classification problems in which each observation may be associated with multiple decisions. Therefore, as a continuity of our research, we are currently implementing an RCN-based model for *multilabel classification problems*, which uses a pair of equivalence/similarity relations to granulate a multilabel information space. The former indiscernibility relation determines whether two objects are considered inseparable according to their descriptive attributes, whereas the latter examines the inseparability based on the decision attributes. While this approach is straightforward from the Granular Computing viewpoint, further strategies to construct the network topology and activate the neurons should be investigated.

Appendix A

Classifying a new instance: the Iris dataset

This appendix describes how the proposed granular classifier operates by using the well-known Iris datatset (see Appendix B) as an example. This classification problem comprises 150 examples from which each 50 of them belong to the following decision classes: *Iris-Setosa*, *Iris-Versicolor* and *Iris-Virginica*. Aiming at exemplify how the classifier works, we separate one object of each class for testing the classification model and the remaining 147 instances will be used for training.

The first step when constructing the classifier is related to the granulation of the information space. By doing so, the method internally computes a symmetric matrix storing the similarity between all instances in the dataset. In this example, we adopt a similarity threshold equals to 0.95 and the HEOM distance function to determine those objects that are deemed inseparable to each other. By relying on this similarity matrix, we can compute the positive, negative and boundary regions associated to each decision class (see Algorithm 1, 2.1).

Table A.1 reports the regions attached to each decision class, where an object is represented by its row index into the training set. The negative region $NEG(X_{Setosa})$ for *Iris-Setosa* contains those objects that do not fulfill the similarity relation for that decision class, the positive region $POS(X_{Setosa})$ comprises those objects that are similar to each other and they are labeled as *Iris-Setosa*, while the boundary region $BND(X_{Setosa})$ includes those objects that fulfill the similarity relation but some of them are labeled as *Iris-Versicolor* or *Iris-Virginica*. Analogously, we can understand the semantics behind the remaining information granules.

Table A.1: Positive, negative and boundary regions for each decision class.

Class	Region	Resulting Set
Setosa	Positive	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48}
	Negative	{49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146}
	Boundary	{}
Versicolor	Positive	{49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 69, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97}
	Negative	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 125, 126, 127, 128, 129, 131, 132, 133, 134, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145}
	Boundary	{68, 70, 81, 124, 130, 135, 146}
Virginica	Positive	{98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 125, 126, 127, 128, 129, 131, 132, 133, 134, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145}

Continued on next page

Table A.1 – *Continued from previous page*

Class	Region	Resulting Set
	Negative	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 69, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97}
	Boundary	{68, 70, 81, 124, 130, 135, 146}

After granulating the information space, the network structure is built following the Algorithm 2, Section 4.2.2. At this point the Rough Cognitive Networks is ready to perform the classification process. Table A.2 shows a testing object x taken from the Iris dataset, which was not included into the training set.

Table A.2: Example instance from Iris dataset to be classified by the RCN.

Attributes	sepallength	sepalwidth	petallength	petalwidth	class
Values	0.75	0.5	0.627119	0.541667	Iris-versicolor

In order to activate the neurons, we compute the similarity class associated to the target object, that is $\bar{R}(x) = \{3, 6, 9, 10, 16, 19, 26, 27, 34, 35, 38, 39, 45, 47, 48\}$. Equations A.7 and A.2, and A.3 display the use of the inclusion degree expression for computing the activation value of neurons representing the positive regions of each decision class, according to Algorithm 3a, Section 4.2.3:

$$\begin{aligned}\mathcal{A}_x^{(0)}(P_{Setosa}) &= \frac{|\bar{R}(x)\ \cap\ POS(X_{Setosa})|}{|POS(X_{Setosa})|}\\ &= \frac{|\bar{R}(x)|}{|POS(X_{Setosa})|} = \frac{15}{49} = 0.306\end{aligned} \tag{A.1}$$

$$\begin{aligned}\mathcal{A}_x^{(0)}(P_{Versicolor}) &= \frac{|\bar{R}(x)\ \cap\ POS(X_{Versicolor})|}{|POS(X_{Versicolor})|}\\ &= \frac{|\emptyset|}{|POS(X_{Versicolor})|} = \frac{0}{49} = 0.0\end{aligned} \tag{A.2}$$

$$\begin{aligned}\mathcal{A}_x^{(0)}(P_{Virginica}) &= \frac{|\bar{R}(x) \ \cap \ POS(X_{Virginica})|}{|POS(X_{Virginica})|} \\ &= \frac{|\emptyset|}{|POS(X_{Virginica})|} = \frac{0}{49} = 0.0\end{aligned} \tag{A.3}$$

Equations A.4, A.5 and A.8 show how to compute the activation value for neurons representing the negative regions, whereas A.7 and A.8 neurons denoting boundary regions for *Iris-Versicolor* and *Iris-Virginica*, respectively.

$$\begin{aligned}\mathcal{A}_x^{(0)}(N_{Setosa}) &= \frac{|\bar{R}(x) \ \cap \ NEG(X_{Setosa})|}{|NEG(X_{Setosa})|} \\ &= \frac{|\emptyset|}{|NEG(X_{Setosa})|} = \frac{0}{98} = 0.0\end{aligned} \tag{A.4}$$

$$\begin{aligned}\mathcal{A}_x^{(0)}(N_{Versicolor}) &= \frac{|\bar{R}(x) \ \cap \ NEG(X_{Versicolor})|}{|NEG(X_{Versicolor})|} \\ &= \frac{|\bar{R}(x)|}{|NEG(X_{Versicolor})|} = \frac{15}{94} = 0.159\end{aligned} \tag{A.5}$$

$$\begin{aligned}\mathcal{A}_x^{(0)}(N_{Virginica}) &= \frac{|\bar{R}(x) \ \cap \ NEG(X_{Virginica})|}{|NEG(X_{Virginica})|} \\ &= \frac{|\bar{R}(x)|}{|NEG(X_{Virginica})|} = \frac{15}{95} = 0.157\end{aligned} \tag{A.6}$$

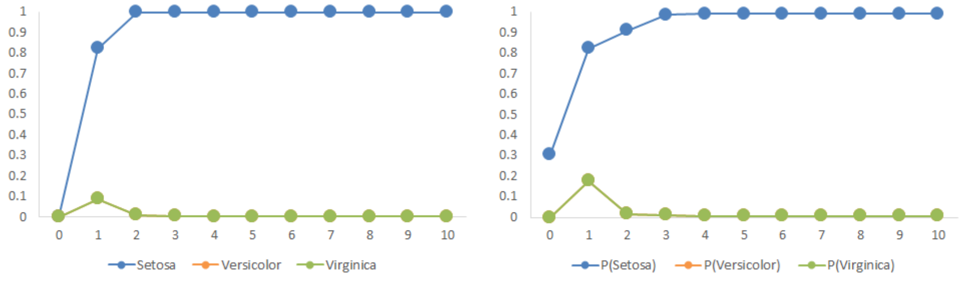
$$\begin{aligned}\mathcal{A}_x^{(0)}(B_{Versicolor}) &= \frac{|\bar{R}(x) \ \cap \ BND(X_{Versicolor})|}{|BND(X_{Versicolor})|} \\ &= \frac{|\emptyset|}{|BND(X_{Versicolor})|} = \frac{0}{7} = 0.0\end{aligned} \tag{A.7}$$

$$\begin{aligned}\mathcal{A}_x^{(0)}(B_{Virginica}) &= \frac{|\bar{R}(x) \ \cap \ BND(X_{Virginica})|}{|BND(X_{Virginica})|} \\ &= \frac{|\emptyset|}{|BND(X_{Virginica})|} = \frac{0}{7} = 0.0\end{aligned} \tag{A.8}$$

Observe the case where the boundary region for the class value “Setosa” is empty, therefore the activation value for this neuron is automatically set to zero. Similarly, the decision neurons are set to zero in the activation vector:

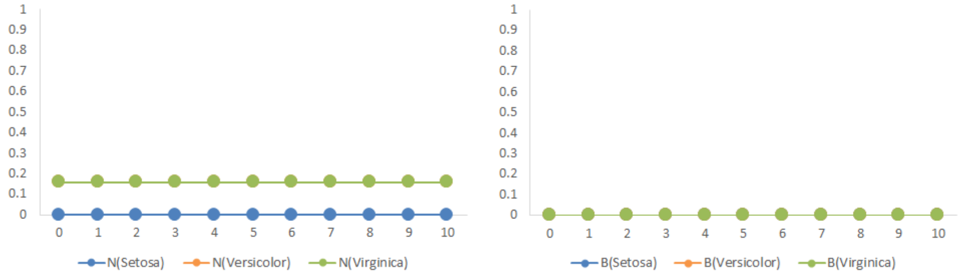
$$A^{(0)} = [0.306, 0.0, 0.0, 0.0, 0.159, 0.157, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]$$

Once the $A^{(0)}$ vector is obtained, we perform the neural inference process until a maximum number of iteration $T = 10$ is reached. Figure A.1 displays the activation value of neurons comprised into the network at each iteration step.



(a) Activation value of decision neurons.

(b) Activation value of positive neurons.



(c) Activation value of negative neurons.

(d) Activation value of boundary neurons.

Figure A.1: Activation value of neurons for Iris dataset over 10 iterations.

As a final step, the decision class with the highest activation value is assigned to the object x according to Algorithm 3b, Section 4.2.3. In this case, the decision class predicted by the granular classifier for x is *Iris-Setosa* [1].

[1] Thanks to Marilyn Bello and Isel Grau for running this example.

Appendix B

Description of benchmark datasets

Table B.1: Characterization of benchmark problems.

Dataset	ID	Instances	Attributes	Classes	Noisy	Imbalance
abalone	1	4174	8	28	no	689:1
acute-inflammation	2	120	6	2	no	no
acute-nephritis	3	120	6	2	no	no
anneal	4	898	38	6	no	85:1
anneal.orig	5	898	38	6	no	85:1
appendicitis	6	106	7	2	no	no
arrhythmia	7	452	262	13	no	122:1
audiology	8	226	69	24	no	57:1
australian	9	690	14	2	no	no
autos	10	205	25	7	no	22:1
balance-noise	11	625	4	3	yes	5:1
balance-scale	12	625	4	3	no	5:1
ballons	13	16	4	2	no	no
banana	14	5300	2	2	no	no
bank	15	4521	16	2	no	7:1
blood	16	748	4	2	no	no
breast	17	277	9	2	no	no
bc-wisconsin-diag	18	569	31	2	no	no
bc-wisconsin-prog	19	198	34	2	no	no
bridges-version1	20	107	12	6	no	no
bridges-version2	21	107	12	6	no	no
car	22	1728	6	4	no	17:1
cardiotocography-10	23	2126	35	10	no	10:1
cardiotocography-3	24	2126	35	3	no	9:1
chess	25	3196	36	2	no	no
cleveland	26	297	13	5	no	12:1
colic	27	368	22	2	no	no
colic.orig	28	368	27	2	no	no

Continued on next page

Table B.1 – *Continued from previous page*

Dataset	ID	Instances	Attributes	Classes	Noisy	Imbalance
collins	29	500	23	15	no	13:1
contact-lenses	30	24	4	3	no	no
contraceptive	31	1473	9	3	no	no
credit-a	32	690	15	2	no	no
credit-g	33	1000	20	2	no	no
crx	34	653	15	2	no	no
csj	35	653	34	6	no	no
cylinder-bands	36	540	39	2	no	no
dermatology	37	358	34	6	no	5:1
echocardiogram	38	131	11	2	no	5:1
ecoli	39	336	7	8	no	71:1
ecoli0	40	220	7	2	no	no
ecoli-0vs1	41	220	7	2	no	no
ecoli1	42	336	7	2	no	no
ecoli2	43	336	7	2	no	5:1
ecoli3	44	336	7	2	no	8:1
ecoli-5an-nn	45	336	7	8	yes	71:1
energy-y1	46	768	8	38	no	no
energy-y2	47	768	8	38	no	no
eucalyptus	48	736	19	5	no	no
flags	49	194	28	8	no	15:1
glass	50	214	9	6	no	8:1
glass0	51	214	9	2	no	no
glass-0123vs456	52	214	9	2	no	no
glass1	53	214	9	2	no	no
glass-10an-nn	54	214	9	6	yes	8:1
glass2	55	214	9	2	no	no
glass-20an-nn	56	214	9	6	yes	8:1
glass3	57	214	9	2	no	6:1

Continued on next page

Table B.1 – *Continued from previous page*

Dataset	ID	Instances	Attributes	Classes	Noisy	Imbalance
glass-5an-nn	58	214	9	6	yes	8:1
glass6	59	214	9	2	no	6:1
haberman	60	306	3	2	no	no
hayes-roth	61	160	4	3	no	no
heart-5an-nn	62	270	13	2	yes	no
heart-statlog	63	270	13	2	no	no
hypothyroid	64	3772	29	4	no	1740:1
ionosphere	65	351	34	2	no	no
iris	66	150	4	3	no	no
iris0	67	150	4	2	no	no
iris-10an-nn	68	150	4	3	yes	no
iris-20an-nn	69	150	4	3	yes	no
iris-5an-nn	70	150	4	3	yes	no
labor	71	57	16	2	no	no
led7digit	72	500	7	10	no	no
libras	73	360	90	15	no	no
liver-disorders	74	345	6	2	no	no
lung-cancer	75	32	56	3	no	no
lymph	76	148	18	4	no	40:1
mammographic	77	830	5	2	no	no
mfeat-factors	78	2000	216	10	no	no
mfeat-fourier	79	2000	76	10	no	no
mfeat-karhunen	80	2000	64	10	no	no
mfeat-morpho	81	2000	6	10	no	no
mfeat-pixel	82	2000	240	10	no	no
mfeat-zernike	83	2000	47	10	no	no
molecular-biology	84	106	57	2	no	no
monk-2	85	432	6	2	no	no
mushroom	86	5644	22	2	no	no

Continued on next page

Table B.1 – *Continued from previous page*

Dataset	ID	Instances	Attributes	Classes	Noisy	Imbalance
musk-1	87	476	167	2	no	no
musk-2	88	6598	167	2	no	5:1
new-thyroid	89	215	5	2	no	5:1
nursery	90	12960	8	5	no	2160:1
optdigits	91	5620	64	10	no	no
ozone	92	2536	72	2	no	33:1
page-blocks	93	5473	10	5	no	175:1
parkinsons	94	195	22	2	no	no
pendigits	95	10992	16	10	no	no
phoneme	96	5404	5	2	no	no
pima	97	768	8	2	no	no
pima-10an-nn	98	768	8	2	yes	no
pima-20an-nn	99	768	8	2	yes	no
pima-5an-nn	100	768	8	2	yes	no
planning	101	182	12	2	no	no
plant-margin	102	1600	64	100	no	no
plant-shape	103	1600	64	100	no	no
plant-texture	104	1599	64	100	no	no
postoperative	105	90	8	3	no	32:1
primary-tumor	106	339	17	22	no	84:1
saheart	107	462	9	2	no	no
segment	108	2310	19	7	no	no
solar-flare-1	109	323	5	6	no	11:1
solar-flare-2	110	1066	12	6	no	7:1
sonar	111	208	60	2	no	no
soybean	112	683	35	19	no	11:1
spambase	113	4601	57	2	no	no
spectfheart	114	267	44	2	no	no
spectrometer	115	531	101	48	no	29:1

Continued on next page

Table B.1 – *Continued from previous page*

Dataset	ID	Instances	Attributes	Classes	Noisy	Imbalance
splice	116	3190	60	3	no	no
sponge	117	76	44	3	no	23:1
tae	118	151	5	3	no	no
tic-tac-toe	119	958	9	2	no	no
vehicle	120	846	18	4	no	no
vehicle0	121	846	18	2	no	no
vehicle1	122	846	18	2	no	no
vehicle2	123	846	18	2	no	no
vehicle3	124	846	18	2	no	no
vertebral2	125	310	6	2	no	no
vertebral3	126	310	6	3	no	no
vote	127	435	16	2	no	no
vowel	128	990	13	11	no	no
wall-following	129	5456	24	4	no	6:1
waveform	130	5000	40	3	no	no
weather	131	14	4	2	no	no
wine	132	178	13	3	no	no
wine-5an-nn	133	178	13	3	yes	no
winequality-red	134	1599	11	6	no	68:1
winequality-white	135	4898	11	7	no	439:1
wisconsin	136	683	9	2	no	no
yeast	137	1484	8	10	no	92:1
yeast1	138	1484	8	2	no	no
yeast3	139	1484	8	2	no	8:1
zoo	140	101	16	7	no	10:1

Appendix C

Detailed Kappa results: *Section 5.5*

Table C.1: Full Kappa values attached to *Section 5.5*.

	RCN-	RCN-	RCN-	RCE10-	RCE10-	RCE10-	RCB10-	RCB10-	RCB10-
ID	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM
1	0.126	0.128	0.126	0.137	0.141	0.137	0.144	0.150	0.143
2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4	0.983	0.978	0.970	0.983	0.975	0.970	0.970	0.965	0.962
5	0.659	0.561	0.654	0.789	0.732	0.777	0.751	0.747	0.724
6	0.351	0.384	0.351	0.351	0.384	0.351	0.395	0.439	0.395
7	0.360	0.305	0.360	0.360	0.305	0.360	0.343	0.311	0.343
8	0.766	0.754	0.793	0.766	0.755	0.793	0.756	0.751	0.783
9	0.624	0.540	0.609	0.659	0.707	0.697	0.690	0.691	0.673
10	0.709	0.753	0.780	0.709	0.753	0.780	0.693	0.722	0.742
11	0.585	0.584	0.585	0.585	0.587	0.585	0.495	0.479	0.495
12	0.607	0.607	0.607	0.607	0.607	0.607	0.516	0.515	0.516
13	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
14	0.728	0.733	0.728	0.774	0.787	0.774	0.799	0.794	0.799
15	0.317	0.310	0.317	0.303	0.359	0.303	0.373	0.427	0.373
16	0.051	0.062	0.051	0.145	0.121	0.145	0.169	0.146	0.169

Continued on next page

Table C.1 – *Continued from previous page*

	RCN-	RCN-	RCN-	RCE10-	RCE10-	RCE10-	RCB10-	RCB10-	RCB10-
ID	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM
17	0.905	0.921	0.905	0.905	0.921	0.905	0.902	0.917	0.902
18	0.020	-0.026	0.020	-0.001	-0.008	-0.001	0.171	0.253	0.171
19	0.249	0.259	0.239	0.249	0.240	0.239	0.278	0.272	0.241
20	0.478	0.503	0.501	0.478	0.503	0.501	0.471	0.490	0.475
21	0.528	0.528	0.553	0.528	0.528	0.553	0.552	0.552	0.574
22	0.854	0.789	0.789	0.854	0.789	0.918	0.902	0.891	0.908
23	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
24	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.969	0.974
25	0.932	0.887	0.908	0.935	0.921	0.925	0.945	0.918	0.939
26	0.317	0.300	0.300	0.324	0.315	0.302	0.284	0.313	0.309
27	0.590	0.600	0.606	0.590	0.600	0.606	0.604	0.608	0.641
28	0.314	0.389	0.426	0.314	0.389	0.426	0.354	0.439	0.451
29	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	0.600	0.600	0.550	0.600	0.600	0.550	0.750	0.750	0.750
31	0.121	0.093	0.156	0.131	0.173	0.162	0.157	0.178	0.167
32	0.631	0.578	0.606	0.636	0.649	0.635	0.687	0.719	0.665
33	0.283	0.297	0.272	0.283	0.292	0.272	0.317	0.337	0.341

Continued on next page

Table C.1 – *Continued from previous page*

	RCN-	RCN-	RCN-	RCE10-	RCE10-	RCE10-	RCB10-	RCB10-	RCB10-
ID	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM
34	0.642	0.509	0.614	0.664	0.653	0.675	0.698	0.725	0.710
35	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
36	0.537	0.597	0.534	0.537	0.597	0.534	0.550	0.592	0.523
37	0.963	0.959	0.959	0.963	0.959	0.959	0.973	0.969	0.973
38	0.560	0.525	0.587	0.522	0.564	0.542	0.544	0.590	0.548
39	0.941	0.970	0.941	0.941	0.951	0.941	0.951	0.950	0.951
40	0.628	0.656	0.628	0.628	0.656	0.628	0.614	0.646	0.614
41	0.709	0.668	0.709	0.709	0.668	0.709	0.721	0.695	0.721
42	0.941	0.970	0.941	0.941	0.951	0.941	0.951	0.950	0.951
43	0.470	0.607	0.470	0.475	0.477	0.475	0.605	0.549	0.605
44	0.758	0.722	0.758	0.758	0.733	0.758	0.766	0.749	0.766
45	0.566	0.567	0.566	0.532	0.547	0.532	0.538	0.570	0.538
46	0.583	0.583	0.583	0.583	0.583	0.583	0.621	0.629	0.621
47	0.528	0.528	0.528	0.528	0.528	0.528	0.478	0.489	0.478
48	0.389	0.458	0.485	0.387	0.456	0.485	0.396	0.447	0.489
49	0.418	0.373	0.418	0.448	0.438	0.448	0.456	0.433	0.456
50	0.819	0.889	0.819	0.819	0.879	0.819	0.875	0.894	0.875

Continued on next page

Table C.1 – *Continued from previous page*

	RCN-	RCN-	RCN-	RCE10-	RCE10-	RCE10-	RCB10-	RCB10-	RCB10-
ID	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM
51	0.428	0.405	0.428	0.470	0.459	0.470	0.459	0.507	0.459
52	0.283	0.342	0.283	0.275	0.354	0.275	0.326	0.427	0.326
53	0.516	0.523	0.516	0.522	0.577	0.522	0.505	0.498	0.505
54	0.623	0.545	0.623	0.657	0.670	0.657	0.624	0.614	0.624
55	0.596	0.507	0.596	0.734	0.759	0.734	0.671	0.724	0.671
56	0.477	0.445	0.477	0.518	0.438	0.518	0.572	0.643	0.572
57	0.819	0.889	0.819	0.819	0.879	0.819	0.875	0.894	0.875
58	0.853	0.871	0.853	0.853	0.854	0.853	0.871	0.871	0.871
59	0.853	0.871	0.853	0.853	0.854	0.853	0.871	0.871	0.871
60	-0.016	-0.016	-0.016	-0.054	-0.054	-0.054	0.099	0.066	0.099
61	0.665	0.676	0.665	0.665	0.676	0.665	0.752	0.729	0.752
62	0.534	0.525	0.534	0.542	0.542	0.542	0.575	0.589	0.575
63	0.565	0.570	0.565	0.565	0.592	0.565	0.520	0.585	0.520
64	0.504	0.608	0.635	0.665	0.822	0.792	0.552	0.788	0.725
65	0.726	0.797	0.726	0.726	0.797	0.726	0.741	0.780	0.741
66	0.790	0.830	0.790	0.790	0.830	0.790	0.820	0.860	0.820
67	0.730	0.740	0.730	0.730	0.740	0.730	0.730	0.730	0.730

Continued on next page

Table C.1 – *Continued from previous page*

	RCN-	RCN-	RCN-	RCE10-	RCE10-	RCE10-	RCB10-	RCB10-	RCB10-
ID	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM
68	0.830	0.820	0.830	0.830	0.820	0.830	0.830	0.840	0.830
69	0.940	0.930	0.940	0.940	0.930	0.940	0.950	0.930	0.950
70	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
71	0.735	0.666	0.709	0.735	0.666	0.709	0.738	0.711	0.754
72	0.488	0.488	0.492	0.488	0.488	0.492	0.592	0.592	0.588
73	0.848	0.815	0.848	0.851	0.821	0.851	0.860	0.827	0.860
74	0.017	0.022	0.017	0.011	0.047	0.011	0.310	0.215	0.310
75	0.483	0.483	0.363	0.483	0.483	0.363	0.214	0.214	0.288
76	0.717	0.729	0.699	0.717	0.729	0.699	0.750	0.757	0.680
77	0.396	0.193	0.407	0.509	0.448	0.517	0.618	0.595	0.618
78	0.954	0.954	0.954	0.954	0.954	0.954	0.956	0.952	0.956
79	0.814	0.813	0.814	0.814	0.813	0.814	0.814	0.812	0.814
80	0.963	0.963	0.963	0.963	0.963	0.963	0.967	0.965	0.967
81	0.624	0.507	0.624	0.694	0.693	0.694	0.695	0.695	0.695
82	0.958	0.958	0.967	0.958	0.958	0.967	0.957	0.957	0.966
83	0.768	0.766	0.768	0.769	0.767	0.769	0.769	0.763	0.769
84	0.809	0.809	0.811	0.809	0.809	0.811	0.789	0.789	0.867

Continued on next page

Table C.1 – *Continued from previous page*

	RCN-	RCN-	RCN-	RCE10-	RCE10-	RCE10-	RCB10-	RCB10-	RCB10-
ID	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM
85	0.945	0.945	0.945	0.945	0.945	0.945	0.991	0.963	0.991
86	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
87	0.901	0.796	0.974	0.901	0.796	0.974	0.893	0.764	0.974
88	1.000	0.991	1.000	1.000	0.991	1.000	0.995	0.977	0.999
89	0.938	0.914	0.938	0.938	0.914	0.938	0.899	0.913	0.899
90	0.916	0.843	0.839	0.916	0.918	0.957	0.924	0.914	0.948
91	0.987	0.983	0.987	0.987	0.983	0.987	0.986	0.984	0.986
92	0.132	0.191	0.132	0.131	0.191	0.131	0.232	0.243	0.232
93	0.674	0.611	0.674	0.745	0.728	0.745	0.777	0.779	0.777
94	0.893	0.821	0.893	0.893	0.821	0.893	0.907	0.862	0.907
95	0.993	0.994	0.993	0.993	0.994	0.993	0.993	0.994	0.993
96	0.685	0.588	0.685	0.752	0.744	0.752	0.748	0.728	0.748
97	0.294	0.291	0.294	0.290	0.300	0.290	0.308	0.347	0.308
98	0.256	0.273	0.256	0.254	0.268	0.254	0.283	0.276	0.283
99	0.275	0.276	0.275	0.276	0.279	0.276	0.354	0.343	0.354
100	0.326	0.324	0.326	0.326	0.306	0.326	0.417	0.394	0.417
101	0.000	0.000	0.000	0.000	0.000	0.000	-0.045	-0.041	-0.045

Continued on next page

Table C.1 – *Continued from previous page*

	RCN-	RCN-	RCN-	RCE10-	RCE10-	RCE10-	RCB10-	RCB10-	RCB10-
ID	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM
102	0.739	0.771	0.739	0.739	0.771	0.739	0.742	0.777	0.742
103	0.594	0.555	0.594	0.594	0.543	0.594	0.551	0.485	0.551
104	0.806	0.823	0.806	0.806	0.827	0.806	0.800	0.817	0.800
105	-0.112	-0.092	0.014	-0.130	-0.092	0.014	-0.169	-0.161	-0.179
106	0.204	0.201	0.216	0.204	0.204	0.216	0.229	0.229	0.234
107	0.160	0.188	0.160	0.156	0.183	0.156	0.144	0.221	0.149
108	0.965	0.942	0.965	0.968	0.972	0.968	0.965	0.965	0.965
109	0.603	0.599	0.623	0.603	0.603	0.623	0.589	0.596	0.597
110	0.616	0.596	0.610	0.616	0.616	0.610	0.655	0.653	0.658
111	0.756	0.718	0.756	0.756	0.718	0.756	0.747	0.796	0.747
112	0.920	0.923	0.936	0.920	0.918	0.934	0.918	0.915	0.923
113	0.758	0.451	0.758	0.835	0.795	0.835	0.833	0.799	0.833
114	0.111	0.213	0.111	0.111	0.213	0.111	0.264	0.276	0.264
115	0.468	0.457	0.482	0.466	0.478	0.498	0.435	0.488	0.469
116	0.870	0.870	0.918	0.870	0.870	0.918	0.862	0.862	0.915
117	0.500	0.500	0.500	0.500	0.500	0.500	0.600	0.600	0.500
118	0.030	0.030	0.030	0.030	0.030	0.030	0.294	0.293	0.294

Continued on next page

Table C.1 – *Continued from previous page*

	RCN-	RCN-	RCN-	RCE10-	RCE10-	RCE10-	RCB10-	RCB10-	RCB10-
ID	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM
119	0.768	0.768	0.811	0.768	0.768	0.811	0.809	0.811	0.886
120	0.593	0.605	0.593	0.593	0.604	0.593	0.618	0.614	0.618
121	0.900	0.888	0.900	0.900	0.894	0.900	0.885	0.900	0.885
122	0.312	0.292	0.312	0.312	0.283	0.312	0.347	0.324	0.347
123	0.933	0.943	0.933	0.933	0.943	0.933	0.925	0.934	0.925
124	0.269	0.254	0.269	0.269	0.239	0.269	0.323	0.261	0.323
125	0.614	0.590	0.614	0.629	0.623	0.629	0.642	0.645	0.642
126	0.678	0.667	0.678	0.706	0.668	0.706	0.703	0.663	0.703
127	0.893	0.868	0.869	0.893	0.868	0.879	0.904	0.895	0.894
128	0.992	0.992	0.989	0.992	0.992	0.989	0.990	0.987	0.986
129	0.826	0.878	0.826	0.834	0.899	0.834	0.833	0.892	0.833
130	0.697	0.684	0.697	0.697	0.684	0.697	0.729	0.719	0.729
131	0.500	0.500	0.500	0.500	0.500	0.500	0.600	0.600	0.600
132	0.882	0.942	0.882	0.882	0.942	0.882	0.865	0.942	0.865
133	0.442	0.383	0.442	0.477	0.488	0.477	0.444	0.441	0.444
134	0.941	0.949	0.941	0.941	0.949	0.941	0.958	0.965	0.958
135	0.442	0.425	0.442	0.457	0.465	0.457	0.450	0.456	0.450

Continued on next page

Table C.1 – *Continued from previous page*

	RCN-	RCN-	RCN-	RCE10-	RCE10-	RCE10-	RCB10-	RCB10-	RCB10-
ID	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM	HEOM	HMOM	HVDM
136	0.888	0.888	0.921	0.888	0.888	0.921	0.901	0.901	0.918
137	0.357	0.354	0.357	0.396	0.464	0.396	0.423	0.443	0.423
138	0.273	0.280	0.273	0.309	0.315	0.309	0.309	0.341	0.309
139	0.668	0.713	0.668	0.697	0.744	0.697	0.721	0.744	0.721
140	0.960	0.960	0.960	0.960	0.960	0.960	0.960	0.960	0.973

Appendix D

Detailed Kappa results: *Section 6.5*

Table D.1: Full Kappa values attached to *Section 6.5*.

ID	RCN-HEOM	RCN-HMOM	RCN-HVDM	FRCN-HEOM	FRCN-HMOM	FRCN-HVDM
1	0.126	0.128	0.126	0.136	0.145	0.139
2	1.000	1.000	1.000	1.000	1.000	1.000
3	1.000	1.000	1.000	1.000	1.000	1.000
4	0.983	0.978	0.970	0.980	0.983	0.966
5	0.659	0.561	0.654	0.797	0.813	0.798
6	0.351	0.384	0.351	0.498	0.484	0.498
7	0.360	0.305	0.360	0.340	0.317	0.340
8	0.766	0.754	0.793	0.711	0.700	0.739
9	0.624	0.540	0.609	0.664	0.691	0.670
10	0.709	0.753	0.780	0.641	0.743	0.720
11	0.585	0.584	0.585	0.729	0.743	0.729
12	0.607	0.607	0.607	0.811	0.819	0.811
13	1.000	1.000	1.000	1.000	1.000	1.000
14	0.728	0.733	0.728	0.781	0.788	0.781
15	0.317	0.310	0.317	0.320	0.364	0.320
16	0.051	0.062	0.051	0.164	0.159	0.164
17	0.905	0.921	0.905	0.917	0.933	0.917

Continued on next page

Table D.1 – *Continued from previous page*

ID	RCN-HEOM	RCN-HMOM	RCN-HVDM	FRCN-HEOM	FRCN-HMOM	FRCN-HVDM
18	0.020	-0.026	0.020	0.092	0.114	0.092
19	0.249	0.259	0.239	0.267	0.283	0.225
20	0.478	0.503	0.501	0.477	0.535	0.544
21	0.528	0.528	0.553	0.513	0.524	0.546
22	0.854	0.789	0.789	0.875	0.886	0.913
23	1.000	1.000	1.000	1.000	1.000	1.000
24	0.974	0.974	0.974	0.976	0.973	0.976
25	0.932	0.887	0.908	0.928	0.929	0.932
26	0.317	0.300	0.300	0.301	0.344	0.348
27	0.590	0.600	0.606	0.627	0.646	0.605
28	0.314	0.389	0.426	0.378	0.417	0.163
29	1.000	1.000	1.000	1.000	0.998	1.000
30	0.600	0.600	0.550	0.700	0.750	0.650
31	0.121	0.093	0.156	0.157	0.192	0.173
32	0.631	0.578	0.606	0.688	0.723	0.694
33	0.283	0.297	0.272	0.317	0.342	0.347
34	0.642	0.509	0.614	0.709	0.714	0.703
35	1.000	1.000	1.000	1.000	0.999	1.000

Continued on next page

Table D.1 – *Continued from previous page*

ID	RCN-HEOM	RCN-HMOM	RCN-HVDM	FRCN-HEOM	FRCN-HMOM	FRCN-HVDM
36	0.537	0.597	0.534	0.583	0.569	0.609
37	0.963	0.959	0.959	0.949	0.949	0.955
38	0.560	0.525	0.587	0.595	0.644	0.595
39	0.941	0.970	0.941	0.970	0.750	0.759
40	0.628	0.656	0.628	0.680	0.970	0.970
41	0.709	0.668	0.709	0.759	0.970	0.970
42	0.941	0.970	0.941	0.970	0.674	0.680
43	0.470	0.607	0.470	0.586	0.592	0.586
44	0.758	0.722	0.758	0.790	0.717	0.790
45	0.566	0.567	0.566	0.593	0.586	0.593
46	0.583	0.583	0.583	0.583	0.583	0.583
47	0.528	0.528	0.528	0.528	0.528	0.528
48	0.389	0.458	0.485	0.451	0.507	0.522
49	0.418	0.373	0.418	0.459	0.521	0.459
50	0.819	0.889	0.819	0.834	0.854	0.834
51	0.428	0.405	0.428	0.479	0.487	0.479
52	0.283	0.342	0.283	0.283	0.311	0.283
53	0.516	0.523	0.516	0.514	0.521	0.514

Continued on next page

Table D.1 – *Continued from previous page*

ID	RCN-HEOM	RCN-HMOM	RCN-HVDM	FRCN-HEOM	FRCN-HMOM	FRCN-HVDM
54	0.623	0.545	0.623	0.620	0.644	0.620
55	0.596	0.507	0.596	0.687	0.681	0.687
56	0.477	0.445	0.477	0.444	0.465	0.444
57	0.819	0.889	0.819	0.834	0.854	0.834
58	0.853	0.871	0.853	0.843	0.860	0.843
59	0.853	0.871	0.853	0.843	0.860	0.843
60	-0.016	-0.016	-0.016	-0.045	-0.045	-0.045
61	0.665	0.676	0.665	0.700	0.691	0.700
62	0.534	0.525	0.534	0.532	0.540	0.532
63	0.565	0.570	0.565	0.599	0.644	0.599
64	0.504	0.608	0.635	0.700	0.807	0.789
65	0.726	0.797	0.726	0.906	0.885	0.906
66	0.790	0.830	0.790	0.850	0.840	0.850
67	0.730	0.740	0.730	0.760	0.790	0.760
68	0.830	0.820	0.830	0.850	0.860	0.850
69	0.940	0.930	0.940	0.940	0.950	0.940
70	1.000	1.000	1.000	1.000	1.000	1.000
71	0.735	0.666	0.709	0.835	0.852	0.809

Continued on next page

Table D.1 – *Continued from previous page*

ID	RCN-HEOM	RCN-HMOM	RCN-HVDM	FRCN-HEOM	FRCN-HMOM	FRCN-HVDM
72	0.488	0.488	0.492	0.499	0.503	0.497
73	0.848	0.815	0.848	0.767	0.755	0.767
74	0.017	0.022	0.017	0.056	0.056	0.056
75	0.483	0.483	0.363	0.500	0.500	0.250
76	0.717	0.729	0.699	0.621	0.663	0.595
77	0.396	0.193	0.407	0.515	0.521	0.526
78	0.954	0.954	0.954	0.939	0.936	0.939
79	0.814	0.813	0.814	0.810	0.801	0.810
80	0.963	0.963	0.963	0.953	0.949	0.953
81	0.624	0.507	0.624	0.663	0.671	0.663
82	0.958	0.958	0.967	0.938	0.937	0.941
83	0.768	0.766	0.768	0.737	0.719	0.737
84	0.809	0.809	0.811	0.863	0.863	0.888
85	0.945	0.945	0.945	0.945	0.945	0.945
86	1.000	1.000	1.000	1.000	1.000	1.000
87	0.901	0.796	0.974	0.901	0.771	0.991
88	1.000	0.991	1.000	0.998	0.928	1.000
89	0.938	0.914	0.938	0.947	0.960	0.947

Continued on next page

Table D.1 – *Continued from previous page*

ID	RCN-HEOM	RCN-HMOM	RCN-HVDM	FRCN-HEOM	FRCN-HMOM	FRCN-HVDM
90	0.916	0.843	0.839	0.931	0.915	0.963
91	0.987	0.983	0.987	0.973	0.968	0.973
92	0.132	0.191	0.132	0.206	0.235	0.206
93	0.674	0.611	0.674	0.766	0.788	0.766
94	0.893	0.821	0.893	0.862	0.736	0.862
95	0.993	0.994	0.993	0.984	0.982	0.984
96	0.685	0.588	0.685	0.735	0.731	0.735
97	0.294	0.291	0.294	0.323	0.364	0.323
98	0.256	0.273	0.256	0.379	0.430	0.379
99	0.275	0.276	0.275	0.339	0.355	0.339
100	0.326	0.324	0.326	0.267	0.314	0.267
101	0.000	0.000	0.000	0.000	0.105	0.000
102	0.739	0.771	0.739	0.648	0.659	0.648
103	0.594	0.555	0.594	0.528	0.517	0.528
104	0.806	0.823	0.806	0.633	0.659	0.633
105	-0.112	-0.092	0.014	-0.065	-0.065	-0.097
106	0.204	0.201	0.216	0.243	0.236	0.247
107	0.160	0.188	0.160	0.189	0.207	0.189

Continued on next page

Table D.1 – *Continued from previous page*

ID	RCN-HEOM	RCN-HMOM	RCN-HVDM	FRCN-HEOM	FRCN-HMOM	FRCN-HVDM
108	0.965	0.942	0.965	0.965	0.962	0.965
109	0.603	0.599	0.623	0.618	0.610	0.630
110	0.616	0.596	0.610	0.608	0.605	0.603
111	0.756	0.718	0.756	0.689	0.680	0.689
112	0.920	0.923	0.936	0.894	0.893	0.905
113	0.758	0.451	0.758	0.848	0.853	0.848
114	0.111	0.213	0.111	0.192	0.303	0.192
115	0.468	0.457	0.482	0.470	0.524	0.534
116	0.870	0.870	0.918	0.912	0.913	0.922
117	0.500	0.500	0.500	0.500	0.500	0.500
118	0.030	0.030	0.030	0.030	0.110	0.030
119	0.768	0.768	0.811	0.796	0.785	0.804
120	0.593	0.605	0.593	0.609	0.580	0.609
121	0.900	0.888	0.900	0.878	0.860	0.878
122	0.312	0.292	0.312	0.360	0.316	0.360
123	0.933	0.943	0.933	0.921	0.923	0.921
124	0.269	0.254	0.269	0.302	0.247	0.302
125	0.614	0.590	0.614	0.650	0.683	0.650

Continued on next page

Table D.1 – *Continued from previous page*

ID	RCN-HEOM	RCN-HMOM	RCN-HVDM	FRCN-HEOM	FRCN-HMOM	FRCN-HVDM
126	0.678	0.667	0.678	0.728	0.727	0.728
127	0.893	0.868	0.869	0.895	0.895	0.880
128	0.992	0.992	0.989	0.966	0.942	0.959
129	0.826	0.878	0.826	0.828	0.884	0.828
130	0.697	0.684	0.697	0.749	0.758	0.749
131	0.500	0.500	0.500	0.500	0.500	0.500
132	0.882	0.942	0.882	0.900	0.933	0.900
133	0.442	0.383	0.442	0.513	0.511	0.513
134	0.941	0.949	0.941	0.949	0.957	0.949
135	0.442	0.425	0.442	0.497	0.503	0.497
136	0.888	0.888	0.921	0.934	0.934	0.934
137	0.357	0.354	0.357	0.463	0.466	0.463
138	0.273	0.280	0.273	0.350	0.386	0.350
139	0.668	0.713	0.668	0.706	0.730	0.706
140	0.960	0.960	0.960	0.934	0.947	0.934

Appendix E

Detailed Kappa results: *Section 7.2*

Table E.1: Full Kappa values attached to *Section 7.2*.

ID	DT	MLP	SL	SMO	NB	NBU	J48	FDT	BFT	RT	RF	LMT	NN	kNN	K*	FRCN	FRNN	VQNN	LWL
1	0.16	0.16	0.17	0.14	0.15	0.14	0.11	0.14	0.12	0.11	0.13	0.17	0.10	0.11	0.15	0.14	0.10	0.14	0.10
2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00	1.00	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	0.87	0.97	0.99	0.99	0.91	0.73	0.96	0.95	0.96	0.96	0.98	0.99	0.98	0.92	0.92	0.98	0.97	0.93	0.89
5	0.75	0.72	0.71	0.74	0.82	0.44	0.85	0.81	0.82	0.79	0.87	0.89	0.82	0.67	0.77	0.81	0.82	0.67	0.27
6	0.48	0.50	0.50	0.57	0.53	0.57	0.51	0.31	0.49	0.62	0.48	0.50	0.48	0.48	0.43	0.48	0.49	0.54	0.48
7	0.46	0.42	0.55	0.45	0.57	0.44	0.46	0.52	0.45	0.31	0.40	0.55	0.24	0.20	0.16	0.32	0.26	0.19	0.20
8	0.68	0.78	0.81	0.78	0.70	0.65	0.75	0.67	0.67	0.56	0.69	0.81	0.70	0.59	0.73	0.70	0.66	0.59	0.32
9	0.69	0.67	0.75	0.70	0.70	0.52	0.70	0.70	0.72	0.59	0.72	0.75	0.60	0.70	0.49	0.69	0.62	0.71	0.71
10	0.59	0.72	0.69	0.71	0.62	0.46	0.76	0.51	0.60	0.62	0.77	0.69	0.70	0.62	0.76	0.74	0.75	0.54	0.30
11	0.54	0.79	0.74	0.79	0.49	0.77	0.57	0.60	0.58	0.59	0.63	0.77	0.58	0.69	0.74	0.74	0.55	0.74	0.20
12	0.50	0.84	0.77	0.84	0.49	0.82	0.58	0.58	0.63	0.63	0.69	0.81	0.63	0.76	0.79	0.82	0.57	0.78	0.17
13	1.00	1.00	1.00	0.80	1.00	1.00	1.00	0.80	1.00	1.00	1.00	1.00	0.60	1.00	1.00	1.00	0.40	1.00	1.00
14	0.49	0.41	0.04	0.00	0.42	0.17	0.78	0.79	0.76	0.74	0.77	0.78	0.74	0.76	0.78	0.79	0.74	0.80	0.44
15	0.36	0.35	0.29	0.25	0.34	0.30	0.36	0.35	0.28	0.32	0.31	0.34	0.22	0.20	0.17	0.36	0.24	0.15	0.00
16	0.00	0.28	0.08	0.04	0.34	0.14	0.34	0.24	0.30	0.18	0.18	0.36	0.15	0.21	0.25	0.16	0.12	0.25	0.00
17	0.87	0.91	0.94	0.95	0.89	0.84	0.84	0.84	0.85	0.84	0.91	0.94	0.92	0.93	0.88	0.93	0.92	0.94	0.83

Continued on next page

Table E.1 – *Continued from previous page*

ID	DT	MLP	SL	SMO	NB	NBU	J48	FDT	BFT	RT	RF	LMT	NN	kNN	K*	FRCN	FRNN	VQNN	LWL
18	0.03	0.16	0.31	0.32	-0.01	0.19	0.24	0.06	0.05	-0.03	0.16	0.31	0.18	0.11	0.11	0.11	0.26	0.19	0.00
19	0.21	0.28	0.32	0.19	0.32	0.33	0.26	0.20	0.21	0.10	0.24	0.32	0.20	0.22	0.30	0.28	0.20	0.24	0.27
20	0.39	0.60	0.56	0.54	0.48	0.57	0.38	0.00	0.00	0.22	0.28	0.56	0.55	0.43	0.50	0.53	0.50	0.52	0.50
21	0.38	0.52	0.53	0.50	0.50	0.54	0.45	0.00	0.00	0.29	0.20	0.53	0.46	0.50	0.53	0.52	0.45	0.47	0.35
22	0.81	0.99	0.85	0.98	0.67	0.67	0.83	0.74	0.94	0.63	0.87	0.97	0.50	0.85	0.70	0.89	0.02	0.82	0.00
23	0.97	1.00	1.00	1.00	0.99	0.96	1.00	1.00	1.00	0.97	1.00	1.00	1.00	1.00	0.84	1.00	0.99	1.00	0.76
24	0.96	0.98	0.97	0.98	0.83	0.77	0.96	0.97	0.96	0.91	0.95	0.97	0.97	0.98	0.84	0.97	0.97	0.97	0.75
25	0.94	0.99	0.94	0.99	0.76	0.76	0.99	0.98	0.99	0.92	0.97	0.99	0.80	0.93	0.94	0.93	0.68	0.92	0.45
26	0.18	0.30	0.28	0.33	0.30	0.31	0.25	0.23	0.19	0.29	0.29	0.28	0.32	0.27	0.20	0.34	0.33	0.31	0.26
27	0.63	0.55	0.61	0.50	0.59	0.53	0.62	0.67	0.62	0.42	0.67	0.64	0.47	0.52	0.44	0.65	0.47	0.60	0.61
28	0.31	0.23	0.40	0.40	0.45	0.48	0.55	0.07	0.08	0.16	0.32	0.53	0.25	0.26	0.38	0.42	0.30	0.27	0.02
29	1.00	1.00	1.00	1.00	0.98	0.92	1.00	1.00	1.00	0.87	1.00	1.00	1.00	1.00	0.44	1.00	0.96	0.98	0.71
30	0.65	0.60	0.60	0.60	0.60	0.60	0.75	0.80	0.70	0.60	0.70	0.60	0.45	0.70	0.60	0.75	0.30	0.35	0.60
31	0.29	0.26	0.22	0.22	0.26	0.26	0.25	0.26	0.28	0.16	0.22	0.27	0.14	0.14	0.23	0.19	0.15	0.21	0.17
32	0.69	0.69	0.70	0.71	0.72	0.53	0.71	0.71	0.69	0.61	0.70	0.70	0.61	0.70	0.47	0.72	0.60	0.73	0.71
33	0.20	0.30	0.39	0.33	0.39	0.39	0.25	0.27	0.32	0.20	0.27	0.39	0.32	0.32	0.27	0.34	0.28	0.28	0.00
34	0.70	0.69	0.71	0.69	0.72	0.53	0.70	0.71	0.71	0.58	0.68	0.71	0.63	0.73	0.48	0.71	0.63	0.74	0.71
35	1.00	1.00	1.00	1.00	0.97	0.57	1.00	1.00	1.00	0.92	1.00	1.00	0.99	0.98	0.91	1.00	0.96	0.97	0.29

Continued on next page

Table E.1 – *Continued from previous page*

ID	DT	MLP	SL	SMO	NB	NBU	J48	FDT	BFT	RT	RF	LMT	NN	kNN	K*	FRCN	FRNN	VQNN	LWL
36	0.31	0.00	0.47	0.70	0.55	0.50	0.26	0.06	0.10	0.35	0.45	0.55	0.57	0.43	0.59	0.57	0.61	0.50	0.26
37	0.83	0.95	0.97	0.95	0.97	0.96	0.92	0.89	0.92	0.85	0.93	0.97	0.93	0.95	0.93	0.95	0.93	0.95	0.85
38	0.63	0.57	0.58	0.52	0.63	0.66	0.58	0.63	0.63	0.51	0.63	0.58	0.59	0.68	0.39	0.64	0.59	0.54	0.59
39	0.97	0.97	0.97	0.95	0.97	0.95	0.97	0.98	0.98	0.97	0.97	0.97	0.95	0.97	0.93	0.75	0.93	0.96	0.97
40	0.61	0.64	0.63	0.68	0.69	0.59	0.63	0.61	0.63	0.61	0.71	0.64	0.61	0.66	0.66	0.97	0.65	0.74	0.42
41	0.62	0.73	0.73	0.73	0.73	0.72	0.72	0.68	0.70	0.70	0.75	0.71	0.73	0.75	0.70	0.97	0.68	0.77	0.44
42	0.97	0.97	0.97	0.95	0.97	0.95	0.97	0.98	0.98	0.97	0.97	0.97	0.95	0.97	0.93	0.67	0.93	0.96	0.97
43	0.69	0.71	0.66	0.68	0.64	0.63	0.66	0.71	0.69	0.65	0.69	0.69	0.64	0.68	0.66	0.59	0.64	0.71	0.69
44	0.62	0.79	0.61	0.64	0.67	0.79	0.75	0.77	0.80	0.75	0.80	0.76	0.79	0.85	0.77	0.72	0.76	0.86	0.56
45	0.58	0.62	0.47	0.52	0.46	0.55	0.51	0.49	0.47	0.51	0.55	0.50	0.48	0.45	0.49	0.59	0.50	0.60	0.06
46	0.63	0.56	0.37	0.46	0.51	0.33	0.69	0.61	0.68	0.60	0.64	0.59	0.27	0.35	0.61	0.58	0.55	0.64	0.09
47	0.54	0.49	0.33	0.39	0.49	0.43	0.55	0.53	0.50	0.49	0.52	0.52	0.27	0.34	0.54	0.53	0.49	0.52	0.16
48	0.52	0.52	0.56	0.53	0.40	0.40	0.51	0.27	0.55	0.39	0.45	0.58	0.39	0.39	0.50	0.51	0.44	0.47	0.29
49	0.54	0.34	0.39	0.36	0.57	0.35	0.51	0.52	0.47	0.40	0.55	0.46	0.31	0.22	0.39	0.52	0.38	0.40	0.30
50	0.82	0.81	0.79	0.79	0.80	0.72	0.82	0.75	0.75	0.84	0.82	0.83	0.85	0.82	0.75	0.85	0.84	0.78	0.81
51	0.44	0.43	0.32	0.33	0.45	0.27	0.47	0.42	0.48	0.44	0.56	0.42	0.41	0.39	0.55	0.49	0.51	0.49	0.22
52	0.32	0.40	0.18	0.22	0.37	0.18	0.39	0.34	0.32	0.36	0.45	0.32	0.25	0.36	0.45	0.31	0.35	0.42	0.25
53	0.42	0.42	0.35	0.34	0.61	0.19	0.58	0.46	0.46	0.52	0.60	0.50	0.44	0.43	0.63	0.52	0.51	0.51	0.23

Continued on next page

Table E.1 – *Continued from previous page*

ID	DT	MLP	SL	SMO	NB	NBU	J48	FDT	BFT	RT	RF	LMT	NN	kNN	K*	FRCN	FRNN	VQNN	LWL
54	0.55	0.55	0.49	0.47	0.60	0.32	0.55	0.54	0.59	0.53	0.66	0.58	0.60	0.61	0.66	0.64	0.64	0.57	0.24
55	0.53	0.61	0.44	0.18	0.58	0.32	0.58	0.57	0.60	0.51	0.72	0.54	0.65	0.61	0.69	0.68	0.62	0.62	0.52
56	0.41	0.37	0.03	0.09	0.39	0.28	0.46	0.46	0.45	0.56	0.60	0.47	0.62	0.55	0.64	0.47	0.62	0.60	0.38
57	0.82	0.81	0.79	0.79	0.80	0.72	0.82	0.75	0.75	0.84	0.82	0.83	0.85	0.82	0.75	0.85	0.84	0.78	0.81
58	0.83	0.85	0.82	0.84	0.72	0.79	0.78	0.84	0.78	0.79	0.88	0.82	0.82	0.76	0.84	0.86	0.89	0.82	0.84
59	0.83	0.85	0.82	0.84	0.72	0.79	0.78	0.84	0.78	0.79	0.88	0.82	0.82	0.76	0.84	0.86	0.89	0.82	0.84
60	0.07	0.18	0.09	0.00	0.10	0.21	0.17	0.05	0.07	0.08	0.16	0.09	0.11	0.07	0.06	-0.04	0.05	0.07	0.19
61	0.17	0.52	0.20	0.30	0.37	0.55	0.73	0.74	0.70	0.66	0.69	0.65	0.48	0.03	0.39	0.69	0.45	-0.04	0.56
62	0.61	0.56	0.65	0.65	0.65	0.62	0.53	0.50	0.51	0.47	0.52	0.61	0.44	0.54	0.54	0.54	0.51	0.65	0.31
63	0.69	0.56	0.66	0.60	0.62	0.66	0.53	0.53	0.51	0.45	0.63	0.66	0.50	0.57	0.49	0.64	0.51	0.65	0.43
64	0.96	0.53	0.73	0.32	0.89	0.59	0.97	0.96	0.97	0.88	0.93	0.97	0.37	0.34	0.54	0.81	0.40	0.28	0.72
65	0.76	0.80	0.74	0.79	0.77	0.64	0.81	0.77	0.78	0.74	0.81	0.84	0.68	0.69	0.63	0.88	0.82	0.76	0.57
66	0.84	0.82	0.79	0.79	0.88	0.82	0.85	0.81	0.85	0.82	0.87	0.79	0.85	0.83	0.82	0.84	0.81	0.87	0.81
67	0.72	0.81	0.80	0.74	0.78	0.71	0.67	0.73	0.72	0.66	0.74	0.80	0.73	0.79	0.74	0.79	0.73	0.84	0.66
68	0.88	0.83	0.86	0.89	0.85	0.89	0.82	0.84	0.83	0.80	0.85	0.84	0.85	0.85	0.85	0.86	0.84	0.90	0.85
69	0.89	0.96	0.91	0.94	0.89	0.92	0.94	0.91	0.92	0.90	0.91	0.91	0.93	0.93	0.92	0.95	0.91	0.91	0.90
70	1.00	1.00	0.99	1.00	1.00	1.00	0.98	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
71	0.46	0.72	0.71	0.75	0.60	0.70	0.55	0.55	0.52	0.54	0.71	0.71	0.70	0.73	0.92	0.85	0.82	0.74	0.44

Continued on next page

Table E.1 – *Continued from previous page*

ID	DT	MLP	SL	SMO	NB	NBU	J48	FDT	BFT	RT	RF	LMT	NN	kNN	K*	FRCN	FRNN	VQNN	LWL
72	0.62	0.67	0.72	0.68	0.70	0.70	0.68	0.68	0.68	0.67	0.67	0.72	0.63	0.70	0.71	0.50	0.53	0.70	0.64
73	0.38	0.79	0.74	0.84	0.58	0.60	0.68	0.53	0.60	0.67	0.75	0.75	0.85	0.79	0.83	0.76	0.85	0.71	0.40
74	0.09	0.40	0.29	0.04	0.03	0.18	0.34	0.24	0.27	0.30	0.37	0.28	0.24	0.22	0.33	0.06	0.21	0.22	0.13
75	0.36	0.12	0.30	0.08	0.33	0.28	0.20	0.09	0.38	0.15	-0.08	0.30	0.08	0.22	0.13	0.50	0.03	0.21	0.15
76	0.56	0.69	0.67	0.73	0.72	0.67	0.56	0.46	0.54	0.53	0.63	0.67	0.63	0.60	0.70	0.66	0.64	0.69	0.48
77	0.64	0.61	0.66	0.67	0.63	0.62	0.65	0.65	0.64	0.54	0.56	0.65	0.51	0.57	0.63	0.52	0.49	0.63	0.63
78	0.64	0.97	0.97	0.98	0.92	0.92	0.88	0.86	0.87	0.82	0.94	0.97	0.95	0.96	0.95	0.94	0.95	0.96	0.79
79	0.58	0.80	0.80	0.82	0.74	0.73	0.73	0.73	0.72	0.60	0.78	0.81	0.78	0.80	0.73	0.80	0.78	0.80	0.50
80	0.59	0.95	0.95	0.96	0.91	0.93	0.81	0.76	0.80	0.69	0.91	0.95	0.95	0.96	0.93	0.95	0.95	0.95	0.60
81	0.65	0.72	0.71	0.68	0.65	0.66	0.69	0.69	0.69	0.60	0.66	0.71	0.62	0.66	0.66	0.67	0.63	0.69	0.67
82	0.49	0.89	0.95	0.98	0.93	0.93	0.76	0.71	0.84	0.62	0.89	0.95	0.96	0.96	0.96	0.94	0.96	0.96	0.76
83	0.50	0.79	0.81	0.82	0.69	0.70	0.68	0.65	0.67	0.58	0.72	0.81	0.76	0.78	0.72	0.72	0.75	0.76	0.39
84	0.43	0.83	0.83	0.83	0.79	0.81	0.61	0.47	0.50	0.38	0.55	0.83	0.68	0.58	0.70	0.86	0.64	0.66	0.43
85	0.94	1.00	0.57	0.94	0.94	0.84	1.00	1.00	1.00	0.98	0.99	1.00	0.28	0.94	0.97	0.94	0.27	0.96	0.95
86	1.00	1.00	1.00	1.00	0.92	0.91	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.94
87	0.89	0.99	0.98	1.00	0.72	0.53	0.99	0.99	0.99	0.68	0.85	0.98	0.84	0.79	0.48	0.77	0.71	0.68	0.39
88	0.99	1.00	1.00	1.00	0.71	0.55	1.00	1.00	1.00	0.92	0.98	1.00	0.98	0.97	0.66	0.93	0.79	0.87	0.32
89	0.88	0.92	0.89	0.69	0.90	0.94	0.93	0.85	0.80	0.93	0.91	0.89	0.96	0.88	0.87	0.96	0.97	0.90	0.67

Continued on next page

Table E.1 – *Continued from previous page*

ID	DT	MLP	SL	SMO	NB	NBU	J48	FDT	BFT	RT	RF	LMT	NN	kNN	K*	FRCN	FRNN	VQNN	LWL
90	0.92	1.00	0.89	1.00	0.86	0.86	0.96	0.94	0.99	0.92	0.97	0.99	0.69	0.98	0.95	0.91	0.25	0.95	0.82
91	0.58	0.98	0.97	0.99	0.91	0.90	0.90	0.87	0.89	0.85	0.96	0.97	0.98	0.99	0.98	0.97	0.98	0.98	0.71
92	0.00	0.30	0.04	0.03	0.13	0.09	0.20	0.00	0.02	0.15	0.05	0.04	0.16	0.17	0.13	0.23	0.20	0.14	0.00
93	0.75	0.78	0.80	0.58	0.71	0.56	0.83	0.83	0.82	0.80	0.85	0.83	0.77	0.76	0.83	0.79	0.78	0.76	0.58
94	0.47	0.76	0.55	0.65	0.54	0.39	0.47	0.61	0.66	0.64	0.72	0.66	0.91	0.81	0.79	0.74	0.85	0.74	0.55
95	0.73	0.94	0.95	0.99	0.87	0.84	0.96	0.95	0.96	0.95	0.99	0.98	0.99	0.99	0.99	0.98	0.99	0.99	0.61
96	0.54	0.55	0.36	0.43	0.51	0.47	0.68	0.67	0.69	0.69	0.76	0.68	0.76	0.72	0.80	0.73	0.77	0.70	0.45
97	0.33	0.37	0.35	0.40	0.40	0.35	0.32	0.34	0.28	0.23	0.40	0.35	0.29	0.28	0.25	0.36	0.24	0.34	0.31
98	0.35	0.25	0.29	0.35	0.32	0.32	0.33	0.31	0.36	0.29	0.30	0.36	0.19	0.22	0.26	0.43	0.23	0.31	0.38
99	0.42	0.39	0.42	0.42	0.43	0.41	0.34	0.41	0.41	0.29	0.37	0.42	0.32	0.35	0.28	0.36	0.24	0.41	0.33
100	0.42	0.46	0.47	0.47	0.49	0.45	0.42	0.42	0.42	0.34	0.40	0.47	0.36	0.44	0.33	0.31	0.31	0.38	0.37
101	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	-0.03	0.09	0.11	0.00	0.12	0.06	0.10	0.11	0.03	-0.08	0.02
102	0.21	0.70	0.79	0.84	0.79	0.85	0.47	0.40	0.44	0.42	0.64	0.80	0.74	0.76	0.79	0.66	0.77	0.83	0.51
103	0.28	0.56	0.63	0.57	0.51	0.53	0.45	0.35	0.42	0.43	0.54	0.62	0.64	0.60	0.63	0.52	0.65	0.55	0.17
104	0.25	0.71	0.81	0.82	0.70	0.74	0.52	0.40	0.48	0.41	0.67	0.81	0.80	0.78	0.78	0.66	0.83	0.80	0.51
105	-0.04	-0.15	0.00	-0.03	-0.09	-0.03	-0.02	-0.02	-0.02	-0.19	-0.11	0.00	-0.10	0.00	-0.06	-0.07	-0.04	-0.01	-0.02
106	0.27	0.29	0.41	0.33	0.40	0.40	0.31	0.28	0.27	0.26	0.30	0.41	0.28	0.30	0.32	0.24	0.24	0.34	0.25
107	0.25	0.24	0.34	0.31	0.31	0.38	0.33	0.21	0.25	0.21	0.18	0.34	0.16	0.25	0.11	0.21	0.16	0.17	0.34

Continued on next page

Table E.1 – *Continued from previous page*

ID	DT	MLP	SL	SMO	NB	NBU	J48	FDT	BFT	RT	RF	LMT	NN	kNN	K*	FRCN	FRNN	VQNN	LWL
108	0.87	0.95	0.94	0.94	0.91	0.77	0.96	0.95	0.96	0.95	0.97	0.95	0.97	0.95	0.97	0.96	0.97	0.96	0.76
109	0.63	0.61	0.61	0.63	0.58	0.56	0.64	0.63	0.63	0.57	0.61	0.61	0.57	0.55	0.61	0.61	0.54	0.59	0.63
110	0.68	0.65	0.68	0.66	0.67	0.67	0.67	0.68	0.68	0.65	0.67	0.68	0.60	0.67	0.67	0.61	0.60	0.65	0.57
111	0.38	0.62	0.54	0.64	0.60	0.37	0.42	0.51	0.43	0.46	0.54	0.56	0.73	0.72	0.69	0.68	0.71	0.56	0.47
112	0.84	0.93	0.93	0.92	0.93	0.91	0.92	0.81	0.92	0.86	0.92	0.93	0.91	0.91	0.91	0.89	0.90	0.90	0.62
113	0.79	0.80	0.84	0.58	0.78	0.60	0.85	0.85	0.85	0.81	0.89	0.87	0.81	0.79	0.80	0.85	0.80	0.80	0.54
114	0.08	0.27	0.35	0.36	0.41	0.39	0.24	0.19	-0.01	0.29	0.20	0.35	0.18	0.19	0.19	0.30	0.22	0.27	0.00
115	0.37	0.39	0.49	0.57	0.44	0.37	0.45	0.36	0.38	0.40	0.51	0.53	0.34	0.34	0.46	0.52	0.44	0.46	0.10
116	0.67	0.92	0.94	0.94	0.93	0.93	0.91	0.88	0.91	0.58	0.85	0.94	0.63	0.66	0.68	0.91	0.58	0.76	0.64
117	0.22	0.60	0.50	0.60	0.20	0.55	0.40	0.40	0.30	0.62	0.50	0.50	0.40	0.40	0.50	0.50	0.40	0.40	0.60
118	0.05	0.31	0.31	0.32	0.00	0.30	0.39	0.31	0.36	0.50	0.47	0.30	0.45	0.25	0.46	0.11	0.47	0.25	0.30
119	0.44	0.93	0.96	1.00	0.28	0.28	0.70	0.66	0.83	0.65	0.81	0.96	0.57	0.97	0.94	0.78	0.08	0.90	0.34
120	0.54	0.76	0.69	0.74	0.47	0.27	0.63	0.63	0.60	0.62	0.67	0.77	0.60	0.62	0.62	0.58	0.60	0.62	0.29
121	0.70	0.93	0.89	0.91	0.48	0.33	0.81	0.81	0.76	0.83	0.91	0.93	0.83	0.81	0.89	0.86	0.89	0.89	0.00
122	0.26	0.59	0.34	0.41	0.33	0.30	0.33	0.28	0.33	0.32	0.43	0.54	0.34	0.33	0.30	0.32	0.30	0.34	0.09
123	0.79	0.94	0.89	0.90	0.66	0.44	0.92	0.88	0.89	0.89	0.94	0.96	0.89	0.90	0.95	0.92	0.91	0.94	0.46
124	0.29	0.58	0.30	0.35	0.31	0.28	0.41	0.25	0.37	0.35	0.39	0.55	0.32	0.33	0.26	0.25	0.29	0.31	0.04
125	0.56	0.63	0.67	0.50	0.52	0.55	0.57	0.55	0.58	0.54	0.65	0.67	0.59	0.49	0.64	0.68	0.55	0.57	0.56

Continued on next page

Table E.1 – *Continued from previous page*

ID	DT	MLP	SL	SMO	NB	NBU	J48	FDT	BFT	RT	RF	LMT	NN	kNN	K*	FRCN	FRNN	VQNN	LWL
126	0.67	0.76	0.78	0.63	0.60	0.73	0.72	0.69	0.74	0.70	0.74	0.78	0.65	0.54	0.69	0.73	0.64	0.66	0.62
127	0.88	0.88	0.92	0.88	0.80	0.80	0.92	0.90	0.90	0.85	0.90	0.93	0.84	0.86	0.85	0.89	0.85	0.85	0.91
128	0.54	0.92	0.80	0.97	0.57	0.60	0.80	0.65	0.78	0.82	0.94	0.93	0.99	0.97	0.99	0.94	0.99	0.76	0.30
129	0.83	0.82	0.55	0.86	0.83	0.36	0.99	0.99	0.99	0.97	0.99	0.94	0.82	0.80	0.92	0.88	0.90	0.89	0.50
130	0.61	0.76	0.80	0.79	0.70	0.70	0.63	0.65	0.63	0.59	0.72	0.80	0.60	0.67	0.60	0.76	0.60	0.72	0.36
131	0.40	0.70	0.40	0.70	0.40	0.30	0.50	0.50	0.10	0.50	0.40	0.40	0.70	0.20	0.10	0.50	0.20	0.40	0.10
132	0.76	0.95	0.94	0.96	0.97	0.93	0.90	0.81	0.81	0.90	0.94	0.94	0.89	0.90	0.96	0.93	0.97	0.93	0.83
133	0.26	0.28	0.24	0.19	0.25	0.22	0.38	0.31	0.37	0.44	0.51	0.41	0.49	0.33	0.48	0.51	0.48	0.35	0.02
134	0.82	0.97	0.98	0.95	0.98	0.97	0.91	0.90	0.81	0.87	0.97	0.98	0.93	0.95	0.99	0.96	0.97	0.97	0.85
135	0.30	0.37	0.34	0.28	0.32	0.31	0.40	0.33	0.36	0.40	0.48	0.36	0.45	0.34	0.47	0.50	0.45	0.34	0.24
136	0.81	0.91	0.89	0.92	0.94	0.94	0.82	0.88	0.83	0.83	0.91	0.89	0.90	0.88	0.89	0.93	0.85	0.88	0.87
137	0.42	0.47	0.47	0.45	0.45	0.47	0.44	0.45	0.40	0.36	0.46	0.47	0.39	0.40	0.39	0.47	0.39	0.47	0.19
138	0.35	0.36	0.31	0.20	0.37	0.37	0.36	0.38	0.32	0.29	0.40	0.40	0.30	0.31	0.28	0.39	0.28	0.36	0.15
139	0.76	0.75	0.68	0.68	0.73	0.75	0.76	0.73	0.73	0.68	0.74	0.74	0.65	0.71	0.62	0.73	0.68	0.73	0.76
140	0.82	0.93	0.95	0.95	0.92	0.91	0.89	0.79	0.89	0.95	0.97	0.95	0.95	0.93	0.95	0.95	0.96	0.88	0.87

Publications

1. **G. Nápoles**, C. Mosquera, R. Falcon, I. Grau, R. Bello, K. Vanhoof, Fuzzy-Rough Cognitive Networks, *Neural Networks* (2017, submitted), IF 3.216.

2. **G. Nápoles**, L. Concepción, R. Falcon, R. Bello, K. Vanhoof, On the accuracy-convergence trade-off in sigmoid Fuzzy Cognitive Maps, *IEEE Transactions on Fuzzy Systems* (2017, submitted), IF 6.701.

3. G. Felix, **G. Nápoles**, R. Falcon, W. Froelich, K. Vanhoof, R. Bello, A review on methods and software for Fuzzy Cognitive Maps, *Artificial Intelligence Review* (2017, submitted), IF 1.731.

4. M. Bello, **G. Nápoles**, I. Fuentes, I. Grau, R. Falcon, R. Bello, K. Vanhoof, A fuzzy activation mechanism for Rough Cognitive Ensembles, *2nd International Symposium on Fuzzy and Rough Sets* (ISFUROS 2017), 2017.

5. A. Alghzawi, **G. Nápoles**, G. Sammour, K. Vanhoof, Forecasting social security revenues in Jordan using Fuzzy Cognitive Maps. *Intelligent Decision Technologies* (IDT 2017), 2017.

6. F. Vanhoenshoven, **G. Nápoles**, Samantha Bielen, K. Vanhoof, Fuzzy Cognitive Maps employing ARIMA components for time series forecasting. *Intelligent Decision Technologies* (IDT 2017), 2017.

7. **G. Nápoles**, R. Falcon, E. Papageorgiou, R. Bello, K. Vanhoof, Rough Cognitive Ensembles, *International Journal of Approximate Reasoning* 85 (2017) 7996, IF 2.696.

8. **G. Nápoles**, E. Papageorgiou, R. Bello, K. Vanhoof, Learning and convergence of Fuzzy Cognitive Maps used in pattern recognition, *Neural Processing Letters* 45 (2017) 431444, IF 1.747.

9. **G. Nápoles**, R. Falcon, Z. Dikopoulou, E. Papageorgiou, R. Bello, K. Vanhoof, Weighted aggregation of partial rankings using Ant Colony Optimization, *Neurocomputing* (2017, accepted for publication), IF 2.392.

10. Z. Zoumpolia, **G. Nápoles**, E. Papageorgiou, K. Vanhoof, A modified fuzzy TOPSIS method aggregating 8.921 partial rankings for companies attractiveness. *The Application of Fuzzy Logic for Managerial Decision Making Processes*, 2017, pp. 59-71.

11. **G. Nápoles**, Isel Grau, E. Papageorgiou, R. Bello, K. Vanhoof, Rough Cognitive Networks, *Knowledge-Based Systems* 91 (2016) 46-61, IF 3.325.

12. **G. Nápoles**, E. Papageorgiou, R. Bello, K. Vanhoof, On the convergence of Fuzzy Cognitive Maps, *Information Sciences* 349 (2016) 154-171, IF 3.364.

13. **G. Nápoles**, Z. Dikopoulou, E. Papageorgiou, R. Bello, K. Vanhoof, Prototypes construction from partial rankings to characterize the attractiveness of companies in Belgium, *Applied Soft Computing* 42 (2016) 276-289, IF 2.857.

14. M. León, **G. Nápoles**, Modeling and experimentation framework for Fuzzy Cognitive Maps. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (AAAI-16), 2016, pp. 4361-4362.

15. **G. Nápoles**, R. Falcon, E. Papageorgiou, R. Bello, K. Vanhoof, Partitive granular cognitive maps to graded multilabel classification. *IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE), 2016, pp. 1363-1370.

16. F. Vanhoenshoven, **G. Nápoles**, R. Falcon, K. Vanhoof, M. Koppen, Detecting malicious URLs using Machine Learning techniques. *IEEE Symposium on Computational Intelligence for Security and Defense Applications* (CISDA 2016), 2016, pp. 6-9.

17. F. Vanhoenshoven, **G. Nápoles**, M. Creemers, M. Leon, K. Vanhoof, Analyzing the impact of the adaptive clearing mechanism on algorithm accuracy in Variable Mesh Optimization. *IEEE Swarm Intelligence Symposium* (SIS 2016), 2016.

18. **G. Nápoles**, I. Grau, R. Falcon, R. Bello, K. Vanhoof, A granular intrusion detection system using Rough Cognitive Networks. *Studies in Computational Intelligence*, vol. 621, 2015, pp 169-191.

19. **G. Nápoles**, I. Grau, M. Len, R. Bello, E. Papageorgiou, K. Vanhoof, A computational tool for simulation and learning of Fuzzy Cognitive Maps.*IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE), 2015, pp. 1-8.

20. Z. Dikopoulou, **G. Nápoles**, E. Papageorgiou, K. Vanhoof, Ranking and Aggregation of factors affecting companies attractiveness. *International Symposium on Knowledge Acquisition and Modeling* (KAM 2015), 2015, p. 288-292.

21. **G. Nápoles**, Z. Dikopoulou, E. Papageorgiou, R. Bello, K. Vanhoof, Aggregation of partial rankings - an approach based on the Kemeny Ranking Problem. *Lecture Notes in Computer Science*, vol. 9095, 2015, pp. 343-355.

22. **G. Nápoles**, I. Grau, K. Vanhoof, R. Bello, Hybrid model based on Rough Set Theory and Fuzzy Cognitive Maps for decision-making. *Lecture Notes in Computer Science*, vol. 8537, 2014, pp. 169-178.

23. **G. Nápoles**, R. Bello, K. Vanhoof, How to improve the convergence on sigmoid Fuzzy Cognitive Maps? *Intelligent Data Analysis* 18 (2014) S77-S88, IF 0.631.

24. **G. Nápoles**, R. Bello, K. Vanhoof, Learning stability features on sigmoid Fuzzy Cognitive Maps through a Swarm Intelligence approach. *Lecture Notes in Computer Science*, vol. 8258, 2013, pp. 270-277.

Bibliography

Aha, D. W., Kibler, D., Albert, M. K., 1991. Instance-based learning algorithms. Machine learning 6 (1), 37–66.

Al-Hmouz, R., Pedrycz, W., Balamash, A., Morfeq, A., 2014. From data to granular data and granular classifiers. In: 2014 IEEE international conference on fuzzy systems (FUZZ-IEEE). IEEE, pp. 432–438.

Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F., 2010. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing 17 (255-287), 11.

Amit, Y., Geman, D., 1997. Shape quantization and recognition with randomized trees. Neural computation 9 (7), 1545–1588.

Atkeson, C. G., Moore, A. W., Schaal, S., 1997. Locally weighted learning for control. In: Lazy learning. Springer, pp. 75–113.

Balamash, A., Pedrycz, W., Al-Hmouz, R., Morfeq, A., 2015. Granular classifiers and their design through refinement of information granules. Soft Computing, 1–15.

Baran, R., Coughlin, J., 1990. Convergence rates in symmetric neural networks with glauber dynamics. Mathematical and Computer Modelling 14, 325–327.

Bargiela, A., Pedrycz, W., 2012. Granular computing: an introduction. Vol. 717. Springer Science & Business Media.

Bello, R., Falcon, R., Pedrycz, W., Kacprzyk, J., 2008. Granular Computing: at the Junction of Rough Sets and Fuzzy Sets. Springer Verlag, Berlin-Heidelberg, Germany.

Benavoli, A., Corani, G., Mangili, F., 2016. Should we really use post-hoc tests based on mean-ranks? Journal of Machine Learning Research 17, 1–10.

Breiman, L., 1996. Bagging predictors. Machine learning 24 (2), 123–140.

Breiman, L., 2001. Random forests. Machine learning 45 (1), 5–32.

Bruck, J., 1990. On the convergence properties of the hopfield model. Proceedings of the IEEE 78 (10), 1579–1585.

Bueno, S., Salmeron, J. L., 2009. Benchmarking main activation functions in fuzzy cognitive maps. Expert Systems with Applications 36 (3), 5221–5229.

Chapelle, O., Schlkopf, B., Zien, A., 2010. Semi-Supervised Learning, 1st Edition. The MIT Press.

Cheng, W., Hüllermeier, E., Dembczynski, K. J., 2010. Graded multilabel classification: the ordinal case. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 223–230.

Cleary, J. G., Trigg, L. E., et al., 1995. K*: An instance-based learner using an entropic distance measure. In: Proceedings of the 12th International Conference on Machine learning. Vol. 5. pp. 108–114.

Cock, M. D., Cornelis, C., Kerre, E. E., 2007. Fuzzy rough sets: The forgotten step. IEEE TRansactions on Fuzzy Systems 15 (1), 121–129.

Cohen, I., Cozman, F. G., Sebe, N., Cirelo, M. C., Huang, T. S., 2004. Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (12), 1553–1566.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13 (1), 21–27.

Dietterich, T. G., 2000. Ensemble methods in machine learning. In: Multiple classifier systems. Springer, pp. 1–15.

Dubois, D., Prade, H., 1990. Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems 17, 91209.

Duda, R. O., Hart, P. E., Stork, D. G., 2012. Pattern classification, 2nd Edition. John Wiley & Sons.

Fan, W., Stolfo, S. J., Zhang, J., 1999. The application of adaboost for distributed, scalable and on-line learning. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 362–366.

Felix, G., Nápoles, G., Falcon, R., Vanhoof, K., Froelich, W., Bello, R., 2017. A review on methods and software for fuzzy cognitive maps. Artificial Intelligence Review.

Frénay, B., Verleysen, M., 2014. Classification in the presence of label noise: a survey. Neural Networks and Learning Systems, IEEE Transactions on 25 (5), 845–869.

Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the american statistical association 32 (200), 675–701.

Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. Machine learning 29 (2-3), 131–163.

Grzymala-Busse, J. W., 1988. Knowledge acquisition under uncertainty - a rough set approach. Journal of Intelligent and Robotic Systems 1, 3–16.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H., 2009. The weka data mining software: an update. ACM SIGKDD explorations newsletter 11 (1), 10–18.

Hearst, M. A., Dumais, S. T., Osman, E., Platt, J., Scholkopf, B., 1998. Support vector machines. Intelligent Systems and their Applications, IEEE 13 (4), 18–28.

Hecht-Nielsen, R., 1989. Theory of the backpropagation neural network. In: Neural Networks, 1989. IJCNN., International Joint Conference on. IEEE, pp. 593–605.

Homenda, W., Jastrzebska, A., Pedrycz, W., 2014. Granular cognitive maps reconstruction. In: Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on. IEEE, pp. 2572–2579.

Hopfield, J., 1982. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences 79 (8), 2554–2558.

Inuiguchi, M., Wu, W.-Z., Cornelis, C., Verbiest, N., 2015. Fuzzy-Rough Hybridization. Springer Berlin Heidelberg, pp. 425–451.

Ishibuchi, H., Nakashima, T., Morisawa, T., 1999. Voting in fuzzy rule-based systems for pattern classification problems. Fuzzy sets and systems 103 (2), 223–238.

Jensen, R., Cornelis, C., 2008. A new approach to fuzzy-rough nearest neighbour classification. In: Proceedings of the 6th International Conference on Rough Sets and Current Trends in Computing. Vol. 5. p. 310319.

Jensen, R., Cornelis, C., 2011. Fuzzy-rough nearest neighbour classification and prediction. Theoretical Computer Science 412, 58715884.

John, G. H., Langley, P., 1995. Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence. Morgan Kaufmann, pp. 338–345.

Kacprzyk, J., Pedrycz, W., 2015. Springer handbook of Computational Intelligence. Springer.

Kecman, V., 2001. Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. MIT Press, Cambridge, MA, USA.

Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., Murthy, K. R. K., 2001. Improvements to platt's smo algorithm for svm classifier design. Neural Computation 13 (3), 637–649.

Kohavi, R., 1995. The power of decision tables. In: Machine Learning: ECML-95. Springer, pp. 174–189.

Kosko, B., 1986. Fuzzy cognitive maps. International Journal of Man-Machine Studies 24 (1), 65–75.

Kosko, B., 1988. Hidden patterns in combined and adaptive knowledge networks. International Journal of Approximate Reasoning 2 (4), 377–393.

Kosko, B., 1997. Fuzzy Engineering. Prentice Hall.

Kreinovich, V., Stylios, C. D., 2015. Why fuzzy cognitive maps are efficient. International Journal of Computers, Communications and Control 10 (6), 825–833.

Landwehr, N., Hall, M., Frank, E., 2005. Logistic model trees. Machine Learning 59 (1-2), 161–205.

Lichman, M., 2013. Uci machine learning repository.
URL `http://archive.ics.uci.edu/ml`

López, V., Fernández, A., García, S., Palade, V., Herrera, F., 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences 250, 113–141.

McCulloch, W. S., Pitts, W., 1988. A logical calculus of the ideas immanent in nervous activity. In: Anderson, J. A., Rosenfeld, E. (Eds.), Neurocomputing: Foundations of Research. MIT Press, Cambridge, MA, USA, pp. 15–27.

Miller, G., 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review 63 (2), 81–97.

Nápoles, G., Bello, R., Vanhoof, K., 2013. Learning stability features on sigmoid fuzzy cognitive maps through a swarm intelligence approach. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Springer Berlin Heidelberg, pp. 270–277.

Nápoles, G., Bello, R., Vanhoof, K., 2014. How to improve the convergence on sigmoid fuzzy cognitive maps? Intell. Data Anal. 18 (6S), S77–S88.

Nápoles, G., Falcon, R., Papageorgiou, E., Bello, R., Vanhoof, K., 2016a. Partitive granular cognitive maps to graded multilabel classification. In: Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on. IEEE Computational Intelligence Society, pp. 1363–1370.

Nápoles, G., Falcon, R., Papageorgiou, E., Bello, R., Vanhoof, K., 2017a. Rough cognitive ensembles. International Journal of Approximate Reasoning 85, 79–96.

Nápoles, G., Grau, I., Falcon, R., Bello, R., Vanhoof, K., 2016b. A granular intrusion detection system using rough cognitive networks. In: Abielmona, R., Falcon, R., Zincir-Heywood, N., Abbass, H. (Eds.), Recent Advances in Computational Intelligence in Defense and Security. Springer Verlag, Ch. 7.

Nápoles, G., Grau, I., Papageorgiou, E., Bello, R., Vanhoof, K., 2016. Rough cognitive networks. Knowledge-Based Systems 91, 46–61.

Nápoles, G., Mosquera, C., Falcon, R., Grau, I., Bello, R., Vanhoof, K., 2017b. Fuzzy-rough cognitive networks. Neural Networks.

Nápoles, G., ón, L. C., Falcon, R., Bello, R., Vanhoof, K., 2017. On the accuracy-convergence trade-off in sigmoid fuzzy cognitive maps. IEEE Transactions on Fuzzy Systems.

Nápoles, G., Papageorgiou, E., Bello, R., Vanhoof, K., 2016. Learning and convergence of fuzzy cognitive maps used in pattern recognition. Neural Processing Letters, 1–14.

Nápoles, G., Papageorgiou, E., Bello, R., Vanhoof, K., 2016. On the convergence of sigmoid fuzzy cognitive maps. Information Sciences 349, 154–171.

Papageorgiou, E. I., 2011. A new methodology for decisions in medical informatics using fuzzy cognitive maps based on fuzzy rule-extraction techniques. Applied Soft Computing 11 (1), 500–513.

Papakostas, G. A., Boutalis, Y. S., Koulouriotis, D. E., Mertzios, B. G., 2008. Fuzzy cognitive maps for pattern recognition applications. International Journal of Pattern Recognition and Artificial Intelligence 22 (8), 1461–1486.

Papakostas, G. A., Koulouriotis, D. E., 2010. Classifying patterns using fuzzy cognitive maps. In: Glykas, M. (Ed.), Fuzzy Cognitive Maps: Advances in Theory, Methodologies, Tools and Applications. Springer Berlin Heidelberg, pp. 291–306.

Pawlak, Z., 1982. Rough sets. International Journal of Computer & Information Sciences 11 (5), 341–356.

Pawlak, Z., 1992. Rough sets-theoretical aspect of reasoning about data, 1st Edition. Kluwer Academic Publishers.

Pawlak, Z., 2002. Rough sets, decision algorithms and bayes theorem. European Journal of Operational Research 136 (1), 181–189.

Pedrycz, W., 2001. Granular computing: an emerging paradigm. Vol. 70. Springer Science & Business Media.

Pedrycz, W., 2010. The design of cognitive maps: A study in synergy of granular computing and evolutionary optimization. Expert Systems with Applications 37 (10), 7288–7294.

Pedrycz, W., Homenda, W., 2014. From fuzzy cognitive maps to granular cognitive maps. Fuzzy Systems, IEEE Transactions on 22 (4), 859–869.

Pedrycz, W., Jastrzebska, A., Homenda, W., 2016. Design of fuzzy cognitive maps for modeling time series. IEEE Transactions on Fuzzy Systems 24 (99), 120–130.

Pratihar, D., 2015. Soft Computing. Alpha Science International Limited.

Quinlan, J., 1986. Induction of decision trees. Machine learning 1 (1), 81–106.

Ren, Y., Zhang, L., Suganthan, P., 2016. Ensemble classification and regression - recent developments, applications and future directions. IEEE Computational Intelligence Magazine 11 (1), 41–53.

Shi, H., 2007. Best-first decision tree learning. Ph.D. thesis, Citeseer.

Smeeton, N. C., 1985. Early history of the kappa statistic. Biometrics 41, 795.

Stylios, C. D., Groumpos, P. P., 2004. Modeling complex systems using fuzzy cognitive maps. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 34 (1), 155–162.

Su, J., Zhang, H., 2006. A fast decision tree learning algorithm. In: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1. AAAI'06. AAAI Press, pp. 500–505.

Sumner, M., Frank, E., Hall, M., 2005. Speeding up logistic model tree induction. In: Knowledge Discovery in Databases: PKDD 2005. Springer, pp. 675–683.

Sun, Y., Wong, A. K., Kamel, M. S., 2009. Classification of imbalanced data: a review. International Journal of Pattern Recognition and Artificial Intelligence 23 (04), 687–719.

Szczuka, M., Jankowski, A., Skowron, A., Ślézak, D., 2015. Building granular systems - from concepts to applications. In: Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. Springer, pp. 245–255.

Triguero, I., García, S., Herrera, F., 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowledge and Information Systems 42 (2), 245–284.

Tsadiras, A. K., 2008. Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps. Information Sciences 178 (20), 3880–3894.

Tsoumakas, G., Katakis, I., 2007. Multi-label classification: an overview. International Journal of Data Warehousing and Mining 3 (3), 1–13.

Tsumoto, S., 2002. Accuracy and coverage in rough set rule induction. In: Alpigini, J. J., Peters, J. F., Skowron, A., Zhong, N. (Eds.), Rough Sets and Current Trends in Computing. Springer Verlag, pp. 373–380.

Tumer, K., Ghosh, J., 1995. Classifier combining: Analytical results and implications. In: In Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms. AAAI Press, pp. 126–132.

Turner, K., Oza, N. C., 1999. Decimated input ensembles for improved generalization. In: Neural Networks, 1999. IJCNN'99. International Joint Conference on. Vol. 5. IEEE, pp. 3069–3074.

Wang, L., Pichler, E. E., Ross, J., 1990. Oscillations and chaos in neural networks: an exactly solvable model. Proceedings of the National Academy of Sciences 87 (23), 9467–9471.

Wilcoxon, F., 1945. Individual comparisons by ranking methods. Biometrics 1, 80–93.

Wilson, D. R., Martinez, T. R., 1997. Improved heterogeneous distance functions. Journal of Artificial Intelligence Research 6 (1), 1–34.

Witten, I. H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Witten, I. H., Frank, E., 2011. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc.

Wong, S., Ziarko, W., 1986. Algorithm for inductive learning. Bulletin of the Polish Academy of Sciences 34, 271–276.

Yao, Y., 2009. Three-way decision: an interpretation of rules in rough set theory. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (Eds.), Rough Sets and Knowledge Technology. Springer Verlag, p. 642649.

Yao, Y., 2010. Three-way decisions with probabilistic rough sets. Information Sciences 180 (3), 341–353.

Yao, Y., 2011. The superiority of three-way decisions in probabilistic rough set models. Information Sciences 181 (1), 1080–1096.

Yao, Y., Zhou, B., 2016. Two bayesian approaches to rough sets. European Journal of Operational Research 251 (3), 904–917.

Zadeh, L., 1965. Fuzzy sets. Information and Control 8, 338–353.

Zhang, G. P., 2000. Neural networks for classification: a survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 30 (4), 451–462.