Dynamic Fuzzy Rule Interpolation

Nitin Naik

Supervisors: Prof. Qiang Shen
Dr. Neal Snooke

Ph.D. Thesis
Department of Computer Science
Institute of Mathematics, Physics and Computer Science
Aberystwyth University

January 12, 2015

Declaration and Statement

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... (candidate)

Date ....

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Where **correction services**[1] have been used, the extent and nature of the correction is clearly marked in a footnote(s).

Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .... (candidate)

Date ....

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .... (candidate)

Date ....

[1]This refers to the extent to which the text has been corrected by others.

Abstract

Designers of effective and efficient fuzzy systems have long recognised the value of inferential hybridity in the implementation of sparse fuzzy rule based systems. Which is to say: such systems should have recourse to fuzzy rule interpolation (FRI) only when no rule matches a given observation; otherwise, when an observation partially or exactly matches at least one of the rules of the sparse rule base, a compositional rule of inference (CRI) should be used in order to avoid the computational overheads of interpolation.

Sparse fuzzy rule bases are constructed by experts or derived from data and may support FRI reasoning in long run. However, two potential problems arise: (1) a system's requirements may change over time leading to rule redundancy; and (2) the system may cease in the long run to provide precise and pertinent results. The need to maintain the concurrency and accuracy of a sparse fuzzy rule base, in order that it generates the most precise and relevant results possible, motivates consideration of a dynamic (real-time) fuzzy rule base.

This thesis therefore presents a framework of dynamic fuzzy rule interpolation (D-FRI), integrated with general fuzzy inference (CRI), which uses the FRI result set itself for the selection, combination and promotion of informative, frequently-used intermediate rules into the existing rule base. Here two versions of the D-FRI approach are presented:*k-means*-based and *GA*-aided. Integration uses the concept of α-cut overlapping between fuzzy sets to decide an exact or partial matching between rules and observation so that CRI can be utilised for reasoning. Otherwise, the best closest rules are selected for FRI by exploiting the centre of gravity (COG), Hausdorff distance (HD) and earth mover's distance (EMD) metrics.

Testing seeks to show that dynamically-promoted rules generate results of greater accuracy and robustness than would be achievable through conventional FRI tout court, and to support the claim that the D-FRI approach results in a more effective interpolative reasoning system. To this end, an implementation of D-FRI is applied to the problem domain of intrusion detection systems (IDS), by integrating it with Snort in order to improve port-scanning detection and increase the level of accuracy of alert predictions.

Dedication

In Loving Memory of My Father

Late Shri. Krishan Kumar Naik

Acknowledgements

I would like to express my enormous gratitude to my supervisors, Professor Qiang Shen and Dr Neal Snooke, for their unfailing motivation and guidance throughout the course of my research. Additionally, I am indebted to the following individuals for their support during this period: Professor Christopher John Price, Professor Reyer Zwiggelaar, Dr Paul Jenkins, Dr Ren Diao, Dr Neil S. Mac Parthaláin, Dr Richard Jensen, Dr. Frédéric Labrosse, Dr Yonghuai Liu and Dr Bernie Tiddeman for providing some of the original impetus for this research; Mr Dave Price, Mrs Meinir Davies, Mrs Bridget Bewick and Mr Mark Batterbee for their encouragement; and Dr David Hanlon for English-writing support.

I would also like to acknowledge my gratitude to the staff—academic, administrative, technical, and support—of the Department of Computer Science at Aberystwyth for providing such a postive and nurturing environment in which to conduct my research. Moreover, the completion of the research would not have been possible without the generous financial support of not only the department, but also the University's Faculty of Science and the IEEE Computational Intelligence Society.

I would like to thank all my fellow researchers in the Advanced Reasoning Group, both past and present, for stimulating discussion, insight, and helpful advice. I am especially grateful in this regard to Pan Su and Jingping Song. I would also like to thank all my colleagues at the Ministry of Defence for their encouragement and support.

Many of the arguments made here were first aired as conference papers. Through these channels, my initial formulations were challenged by, and further developed in response to, the critical scrutiny and commentary of conference organisers and attendees. I record my thanks to these generous interlocutors.

Lastly, the completion of this PhD would not have been possible without the support and encouragement of my family: my parents Shri Krishan Kumar Naik and Smt. Shobha Naik, my siblings Nikita Naik and Nitesh Naik, my dear wife Sarika Naik, my lovely daughters Dishita Naik and Ishita Naik, and all my relatives and friends. My heartfelt thanks to them all.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

FUZZY Logic is a paradigm in soft computing which provides a means of approximate reasoning not found in traditional crisp logic. It helps deal with information arising from computational perception and cognition, that is, information which is uncertain, imprecise, vague, partially true, or without sharp boundaries. It also allows for the inclusion of vague human assessments in computing problems. Fuzzy logic is based on fuzzy set theory which is an extension of classical set theory where elements have varying degrees of membership. A fuzzy inference system (or simply fuzzy system), which is based on fuzzy logic, is an expert system which can mimic the human expert's reasoning process much more realistically than a conventional expert system. The behaviour of a fuzzy system is governed by a combination of fuzzy parameters such as: fuzzy sets, fuzzy variables, fuzzy operators, and fuzzy *If-Then* rules.

Fuzzy systems have been used in a wide variety of applications in science, engineering, business, psychology, medicine and other fields. Examples for commercial applications of fuzzy systems include: fuzzy automatic transmissions developed by Nissan, fuzzy anti-skid braking systems by Nissan, auto-focusing cameras by Canon, digital image stabilizers for camcorders by Matsushita, hand-writing recognition systems by Hitachi, hand-printed character recognition systems by Sony, voice recognition systems by Ricoh and Hitachi, stock-trading portfolio systems used in Tokyo's stock market, Sendai station subway control systems in Japan, and so on [138, 158, 184, 216]. NASA has investigated fuzzy control applications for automated space docking and simulations show that a fuzzy control system can greatly reduce fuel consumption.

In a fuzzy inference system, approximate reasoning is implemented through the execution of fuzzy *If-Then* rules, a collection of which is termed a fuzzy rule base. If a fuzzy inference system has a sufficiently large number of rules to cover all possible input conditions then it is said to contain a dense fuzzy-rule base. In this situation, the inference process is relatively straightforward and any classical inference approach, such as compositional rule of inference (CRI) [208, 209], can be used to infer the results. However, if the number of rules is smaller and do not cover all possible input conditions then such a rule base is termed a sparse fuzzy-rule base. In this situation, the inference process is more complex, and a more intuitive approach such as a fuzzy rule interpolation (FRI) [53, 54, 55, 110, 111, 112, 113] can be used to infer the results. In both cases, the rule base and its rules remain of prime importance and affect the accuracy of the fuzzy inference system.

In most fuzzy inference systems, the provided rule base is typically, initially developed by human experts or derived from data. Over time, due to the static nature of the rule base, these rules may not meet the present or new reasoning requirements of the system and may not be able to provide the correct reasoning results. This is of special concern in online and real-time systems. This gives rise to the need for a dynamic (real-time) rule base and an adaptive fuzzy inference system to ensure correct inference. A great deal of work has been carried out in the study of dense rule-base systems and many adaptive dense fuzzy inference system have been proposed [15, 127, 130, 142, 188, 191, 213]. Unfortunately, sparse fuzzy-rule base systems and their reasoning processes are quite different from those employed in dense fuzzy-rule base systems and an adaptive approach to a dense rule base may not be directly applied. There are, therefore, reasoned grounds for investigating dynamic rule bases for sparse fuzzy inference systems.

It has proved that FRI approaches are quite useful for reasoning in the sparse rule base context. Since FRI methods are already involved in performing inference in sparse fuzzy rule base systems, why not extend their usage in order to handle dynamic problems? This is the motivation behind the development of the proposed Dynamic Fuzzy Rule Interpolation (D-FRI) approach.

The aim of this research is to develop a dynamic fuzzy rule interpolation system that can utilise the interpolation results to modify the sparse rule base. It develops an adaptive system to cope with the need for a dynamic sparse fuzzy rule base and therefore enhance the fuzzy reasoning by the use of existing FRI methods. Without affecting the original functionality of FRI methods, the work also helps improve the accuracy of interpolation results by suggesting and applying more suitable distance metrics. This research also integrates inference (CRI) and interpolation (FRI) so the best of both can be leveraged for a sparse rule base system.

1.1 Fuzzy Logic, Fuzzy Inference and Fuzzy Interpolation

Here, relevant core concepts of fuzzy logic are discussed which are prerequisites to a full understanding of this research work and the terminologies used throughout the thesis. An account of three main concepts will be given: fuzzy logic, fuzzy inference and fuzzy interpolation.

1.1.1 Fuzzy Logic

Fuzzy logic is a mathematical approach to problem solving. It performs exceptionally well in producing exact results from imprecise or incomplete data. Fuzzy logic differs from classical logic in that statements are no longer true or false. In traditional logic a variable takes on a value with a certainty degree or truth measure of either 0 or 1; in fuzzy logic, a variable can assume a value with any degree between 0 and 1, representing the extent (membership) to which an element belongs to a given concept. The human brain can reason with uncertainties, vagueness, and judgements. Computers can only manipulate precise valuations. Fuzzy logic is an attempt to combine the two [59].

In real-life, fuzzy logic provides the means to compute with words. Using fuzzy logic, experts are no longer forced to summarize and express their knowledge in a language that machines or computers can understand. What traditional expert systems failed to achieve may be realized through the use of fuzzy expert systems. Fuzzy logic comprises fuzzy sets and fuzzy (linguistic) variables, which are a way of representing non-statistical uncertainty and performing approximate reasoning, which includes operations that implement knowledge-based inferences.

A fuzzy set is distinct from a crisp set in that it allows its elements to have a degree of membership. The core of a fuzzy set is its membership function that defines

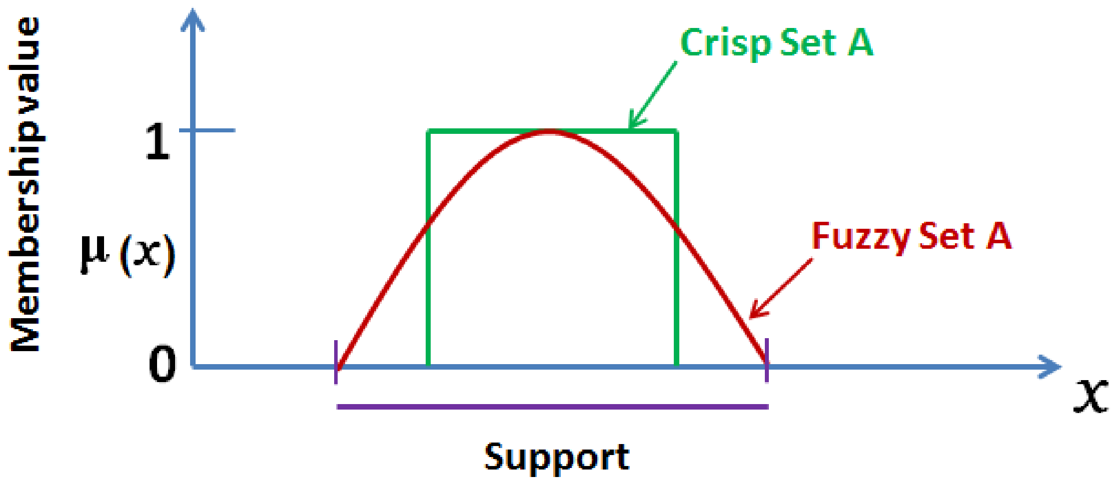


Figure 1.1: Fuzzy Set

the relationship between a value in the set's domain and its degree of membership [65]. In particular, according to the original idea of Zadeh [208], membership of an element x to a fuzzy set A, denoted as $\mu_A(x)$, can vary from 0 (full non-membership) to 1 (full membership), i.e., it can assume all values in the interval [0,1]. The value of $\mu_A(x)$ describes a degree of membership of x in A. Clearly, a fuzzy set is a generalization of the concept of a classical set whose membership function takes on only two values 0 or 1, as shown in Figure 1.1. Each fuzzy set defines a portion of the variable's domain. However, this portion is not uniquely defined. Fuzzy sets overlap as a natural consequence of their elastic boundaries.

Fuzzy (linguistic) variables are variables whose values are not numbers but words or sentences in a natural or artificial language. This concept has been developed as a counterpart to the concept of a numerical variable. Figure 1.2 shows an example of a linguistic variable *Temperature* with three linguistic terms *Low, Medium, and High*. The linguistic values (terms): *low, medium, and high* are fuzzy sets for the *Temperature* linguistic variables [210, 211, 212].

1.1.2 Fuzzy Inference

Inference is the process of deriving logical conclusions from premises known or assumed to be true or partially true. When conclusions are derived based on fuzzy linguistic variables using fuzzy set operators (AND, OR, NOT), then the process is called the approximate reasoning or fuzzy inference. Fuzzy inference is more effective and useful for those systems where a system cannot be defined in precise

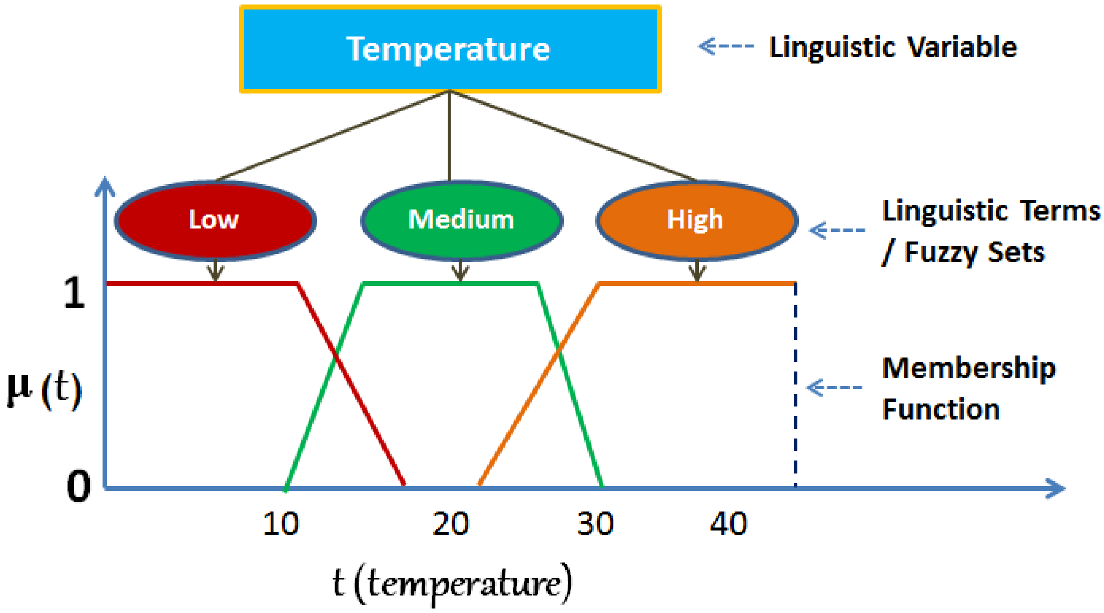


Figure 1.2: Example of linguistic variable *Temperature* with three linguistic terms (fuzzy sets): *Low, Medium, and High*

mathematical terms/models due to uncertainties, unpredicted dynamics and other unknown phenomena. In the real world much of the knowledge is unclear, confusing, ambiguous, imprecise, and vague in nature. Fuzzy inference mimics the crucial ability of the human mind to summarize data and focus on decision-relevant information.

Fuzzy inference is a convenient way to map a fuzzy input space onto a fuzzy output space. This mapping is done using fuzzy *If-Then* rules. These rules together with the fuzzy sets that the rules use serve to partition the input and output spaces into fuzzy regions [3]. A fuzzy rule can be expressed in the form: *If a set of conditions are satisfied Then a set of consequents can be inferred*. The *If* part is called the *antecedent* and the *Then* part is called the *consequent*. The antecedent describes to what degree the rule applies, while the consequent assigns a fuzzy function to each of one or more output variables. A fuzzy *If-Then* rules provide an easy means to express and capture human rule-of thumb type knowledge, because they are expressed using linguistic terms. A collection of fuzzy rules (rule base) can be derived from subject matter experts or extracted from data through a rule induction process.

The rule base is the key component of a fuzzy inference system for which such a system is also known as a fuzzy rule-based system (FRBS). Fuzzy inference is only possible in a dense rule base when the rule base is large enough to cover the complete input space, i.e., every input condition should be covered by at least one rule. The compositional inference and compatibility-modification inference are two main classical approaches to fuzzy inference [48]. Based on these basic inference principles, many inference methods have been proposed such as: Zadeh inference [208, 209], Mamdani inference [136, 137], Larsen inference [123], Sugeno inference [173, 175], and Tsukamoto inference [179, 180].

1.1.3 Fuzzy Interpolation

In most fuzzy inference systems, the completeness of the fuzzy rule base is required to generate meaningful output when classical fuzzy inference methods are applied. This emphasises the need for a dense rule base for fuzzy inference that covers all possible inputs. Regardless of the way in which a rule base is constructed, be it by human experts or by an automated agent, often incomplete or sparse rule bases are generated. A dense rule base is especially impracticable in a multidimensional environment where the number of rules increases exponentially as the input variables and the fuzzy linguistic labels associated with each variable increase. In this situation, the classical fuzzy reasoning techniques cannot generate an acceptable output for such cases. One simple solution to handle incomplete or sparse fuzzy rule bases and to infer reasonable output is through the application of fuzzy rule interpolation (FRI) methods. FRI techniques were originally introduced to generate inference for sparse fuzzy rule bases, and thus to extend the usage of fuzzy inference mechanisms for sparse fuzzy rule base systems [97].

Interpolation is a mathematical term for finding new data points within the range of a discrete set of known data points. Fuzzy rule interpolation (FRI) performs interpolative approximate reasoning by taking into consideration the existing closest fuzzy rules for cases where there is no matching fuzzy rule. Generally, these FRI methods are capable of performing two types of inference operation: fuzzy interpolation and fuzzy extrapolation depending on the location of selected closest rules. If the given input observation lies among the selected closest rules then, a fuzzy interpolation operation is performed; otherwise, if the given input observation lies to one side of all selected closest rules, then extrapolation is performed. A comprehensive overview of FRI techniques will be presented later in the thesis.

1.2 Dynamic Fuzzy Rule Interpolation and its Integration with General Fuzzy Inference

Fuzzy rule interpolation (FRI) is a well established reasoning approach to sparse fuzzy rule bases. In the last two decades, many FRI methods have been introduced and deployed successfully in many application areas. FRI methods infer the approximate conclusion based on the closest rules in the sparse rule base. Such sparse fuzzy rule bases are constructed by experts or derived from data and support the FRI reasoning process in the long run. However, a system's requirements may change over time and certain rules may cease to be useful or may never be applied in the FRI reasoning process. A static rule base may even become a serious problem for the whole expert system and FRI process to work after a period. Therefore, rules should always be contemporary and valid in the light of new observations. That is, the sparse rule base should be concurrent and accurate in order that the reasoning should generate the most precise and relevant results possible.

This leads to a discussion of the requirements of a dynamic (real-time) rule base. One way to ensure and manage a dynamic rule base is to have a procedure that may be applied to update the rule base which was originally used to generate it. Repeated applications of the same procedure may not be a reasonable and feasible solution for obvious reasons like frequent changes in system requirements. Another way is to develop an adaptive (self learning) system that manages the dynamic rule base automatically. There are two possible techniques for designing an adaptive system: learning rules either from the existing rules or from the data. However, in sparse rule bases, the first technique is not practically applicable due to the gaps in the existing rules. The second technique is only applicable when an authentic and concurrent data source is available at all times. For sparse rule bases, even the availability of an appropriate data source may not suffice if the process is complex, time-consuming, and in need of expert supervision.

Hitherto little or no thought has been devoted to another straightforward and simple way to develop this dynamic sparse rule base: using the FRI result set itself. As was previously mentioned, the success and preciseness of the FRI method in many applications is the guarantee of obtaining better results that can be readily utilised in a certain intelligent way in an effect to update the existing sparse rule base. Furthermore, the existing FRI method can play two roles: (1) reasoning; and (2) updating the rule base using the same results. There is a caveat: once FRI reasoning is complete, FRI reasoning results must not be discarded; rather they should be retained. This is because such results may contain potentially useful information, e.g., covering regions that were uncovered by the original sparse rule base. Thus, they should be exploited in order that accumulated data can be used for managing the dynamic rule base. This thesis therefore proposes to develop an approach that utilises FRI both for inference with a sparse rule base and for updating the sparse rule base itself.

This research is focused on enhancing the strength of FRI approaches by providing a dynamic rule base, on the basis of the FRI approach itself. The resulting approach is referred to as Dynamic Fuzzy Rule Interpolation (D-FRI). It seeks to marry the FRI approach with a dynamic rule base rather than a fixed static one. The proposed D-FRI is generalised in nature so that in principle any existing FRI method could be extended by it. However, in this research work, a transformation-based FRI (T-FRI) [85, 86] technique is used throughout for all experimentation. This research does not attempt to modify the original concept of any FRI approach but relies on the correctness of the underlying approach. That said, it does introduce the use of new distance metrics: Hausdorff Distance (HD) and Earth Mover's Distance (EMD) into FRI. This should be helpful in the selection of better closest rules for interpolation, thereby offering improved interpolation results.

The proposed D-FRI approach leads to potential changes in the FRI reasoning technique however, as it provides dynamically updated sparse rule bases that can eventually be converted into dense rule bases depending on the requirements of the given application problem. A dense fuzzy rule base avoids the requirement to interpolate and is less complex. In a sparse rule-based system, fuzzy rule interpolation (FRI) is powerful when no rule observation matches the given observation. However, observation may sometimes match partially or even exactly with at least one of the rules in the rule-base. In these situations, it is natural to avoid the computational overheads of interpolation by firing the best matching rule directly via CRI. If no such match is found then it should be ensured that correct rules are selected for FRI. With this in mind, the research proposed in the thesis also aims to contribute to the field of fuzzy systems in general and FRI literature in particular by integrating inference and interpolation for sparse rule bases. The work innovatively seeks to apply inference in the case of sparse rule bases when this is possible; interpolation is only applied when inference is not possible. Thus, the integrated system offers the best of both inference and interpolation and offers a new way of reasoning in sparse fuzzy rule bases.

1.3 Structure of the Thesis

This section outlines the structure of the remainder of the thesis.

Chapter 2: This chapter provides a background introduction to fuzzy inference systems (FISs) / fuzzy rule-based systems (FRBSs), compositional rule of inference (CRI) and fuzzy rule interpolation (FRI). It provides a comprehensive review of typical FRI methods that have been developed in the last two decades. All the FRI methods are described using a unified style in pseudo code. A framework for systematically evaluating the performance of each approach is provided and each method is then examined and compared accordingly. This helps to demonstrate the strengths and limitations of the respective underlying technologies, thereby revealing their application potentials.

Chapter 3: This chapter presents an initial attempt towards a dynamic fuzzy rule interpolation framework, for the purpose of selecting, combining, and promoting informative, frequently used so-called intermediate rules that are, created during a fuzzy interpolation process into the rule base. It also presents experimental results that demonstrate the performance of the proposed approach, showing improved performance in terms of accuracy than that achievable through conventional FRI that uses just the original sparse rule base. The proposed D-FRI approach is implemented using *k-means* [134] and is presented in this chapter which has been published in [148].

Chapter 4: This chapter presents a more enhanced and intuitive reimplementation of the D-FRI proposed in the preceding chapter, by the use of genetic algorithm. In particular, a genetic algorithm is employed to replace the *k-means* clustering which was used in the previous version. The experimental results show that this version reaffirms the better performance than that achievable by the use of conventional FRI that uses just the original sparse rule base. The proposed GA-based D-FRI approach presented in this chapter has been published in [149].

Chapter 5: This chapter proposes a generalised approach which integrates fuzzy rule interpolation (FRI) and fuzzy rule inference (CRI) effectively. This approach uses the concept of α-cut overlapping between fuzzy sets to decide an exact or partial matching between rules and observation so CRI can be utilised for reasoning. Otherwise, the best closest rules are selected for FRI by exploiting the Hausdorff distance (HD) and earth mover's distance (EMD) metrics. Simulation results are provided to show the performance of these methods, including detailed comparisons over the use of centre of gravity (COG), HD and EMD which demonstrate the impact of different distance metrics upon the FRI performance. The proposed integration of inference and interpolation presented in this chapter has been published in [150].

Chapter 6: This chapter presents an application of the proposed D-FRI approach in network security. Here, the proposed dynamic integrated system (D-FRI and a fuzzy inference system) is used to build a powerful fuzzy intrusion detection system (IDS) called *D-FRI-Snort*. An open-source IDS - *Snort* - has been chosen as the base building block of the application, and D-FRI is integrated with it to develop *D-FRI-Snort*. For this specific application, only port scan attacks are studied as examples of attempted system intrusion, and five computers are used to carry out attacks on a single host in a succession of rounds of increasing aggression. Other network analysis tools (such as Wireshark, NMAP, Basic Analysis and Security Engine (BASE) and WinPcap) are also used for purposes of experimentation. D-FRI-Snort is tested comparatively with the original Snort, and in order to observe and measure dynamic rule promotion processes.

Chapter 7: This chapter summarises the key contributions made within the thesis, together with a discussion of topics which form the basis for future research. Both immediately achievable tasks and long-term projects are considered.

Appendices: Appendix A lists the publications arising from the work presented in this thesis, containing both published papers, and those currently under review for potential journal publication. Appendix B summaries the acronyms employed throughout this thesis.

Chapter 2

Background

FUZZY inference systems, which tend to mimic the behaviour of man, are a vital part of many successful knowledge-based systems in many fields. The basic concepts which underlie fuzzy systems are those of *linguistic variables* and fuzzy *If-Then* rules. A linguistic variable, as its name suggests, is a variable whose values are words rather than numerical numbers, e.g., small, young, very hot and quite slow. Fuzzy *If-Then* rules are of the general form: *If antecedent(s) Then consequent(s)*, where antecedent and consequent are fuzzy propositions that contain linguistic variables. A fuzzy *If-Then* rule is exemplified by *If the temperature is high Then the fan-speed should be high* [65].

The fuzzy rule base (set of fuzzy rules) is the core part of a fuzzy inference system that contains knowledge that is utilised by the reasoning mechanism of the system. If the rule base covers the entire input domain then one of the possible reasoning mechanisms would be compositional rule of inference (CRI). However, if the rule base does not cover the entire input domain then fuzzy rule interpolation (FRI) may offer a potential solution. For fuzzy reasoning, both CRI and FRI have their own importance depending on the nature of the rule base and application area. This section provides a literature review on fuzzy inference systems, compositional rule of inference (CRI) and fuzzy rule interpolation (FRI), where it is assumed that the reader should know the fundamentals of fuzzy logic.

The rest of the chapter is organised as follows. Section 2.1 explains the fuzzy inference systems or fuzzy rule-based systems. Section 2.2 briefly discusses compositional rule of inference (CRI) which is an important part of the proposed integrated dynamic framework. Section 2.3 discusses some typical compositional rule of inference (CRI) methods. Section 2.4 explains fuzzy rule interpolation (FRI) which is the backbone of the proposed integrated dynamic framework. Section 2.5 presents an overview of the most popular methods for fuzzy rule interpolation using a unified representational scheme. Section 2.6 describes several evaluating criteria for fuzzy rule interpolation (FRI) methods. Section 2.7 analyses the introduced FRI methods based on the identified evaluating criteria. This helps to demonstrate the strengths and limitations of the respective underlying technologies, thereby revealing their application potentials. Finally, Section 2.8 summarises this chapter.

2.1 Fuzzy Inference Systems (FISs) / Fuzzy Rule-Based Systems (FRBSs)

A fuzzy inference system or fuzzy rule-based system is a way of mapping from an input space to an output space using fuzzy logic. Such mappings are subsequently used as sources from which decisions are made, or patterns detected. A fuzzy inference system uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data. The mapping is accomplished by a number of fuzzy *If-Then* rules, each of which describes the local behaviour of the mapping. In particular, the antecedent of a rule defines a fuzzy region in the input space, while the consequent specifies the output in the fuzzy region. For example, in a fuzzy rule:

$$If\ x\ is\ low\ and\ y\ is\ high\ Then\ z\ is\ medium.$$

Here *x is low; y is high; z is medium* are fuzzy statements; x and y are input linguistic variables; z is an output linguistic variable. The linguistic terms: *low, medium, and high* are fuzzy sets for their corresponding linguistic variables [210, 211, 212]. These fuzzy rules are either built from expert knowledge or generalized from historical data.

A fuzzy inference system can take either fuzzy inputs or crisp inputs (which are viewed as fuzzy singletons or need to be fuzzified), but the outputs it produces are almost always fuzzy sets [171]. Sometimes it is necessary to have a crisp output, especially in a situation where a fuzzy inference system is used as a controller. Therefore, we need a method of defuzzification to extract a crisp value that best represents a fuzzy set. A fuzzy inference system with a crisp input and output is shown in Figure 2.1. The dashed line indicates a basic fuzzy inference system (FIS) with two major components: knowledge base and inference mechanism (which always processes fuzzy input and produces fuzzy output). The FIS with crisp input and output needs two additional blocks; fuzzification and defuzzification. Fuzzification serves the purpose of transforming a crisp input into a fuzzy value, while defuzzification serves the purpose of transforming an output fuzzy set into a crisp single value. Therefore, the basic structure of a fuzzy inference system consists of the following components:

- *Fuzzifier*: Which converts the crisp input to a linguistic variable using the membership functions stored in the fuzzy knowledge base (the rule base and the database are jointly referred to as the knowledge base as shown in Figure 2.2).
- *Fuzzy Rule Base*: Which contains a selection of fuzzy *If-Then* rules.
- *Database (or Dictionary)*: Which defines the membership functions of fuzzy sets used in the fuzzy rules or in the problem domain in general.
- *Inference Mechanism*: Which performs the inference procedure upon the rules and given facts to derive an inferred output or conclusion.
- *Defuzzifier*: Which converts the fuzzy output of the inference engine to crisp using membership functions analogous to the ones used by the fuzzifier.

Fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis, time series prediction, robotics, and computer vision [129, 158, 184, 216]. Because of its multi-disciplinary nature, the fuzzy inference system is known by a number of names, such as fuzzy-rule-based systems, fuzzy expert systems, fuzzy models, fuzzy associative memories, fuzzy logic controllers, and simply fuzzy systems [103].

The designing of fuzzy rule-bases is the most difficult task in developing fuzzy systems. If the rule-base of a system is too large then it affects efficiency of the system due to redundancy and difficulty in identifying appropriate rules to time; and if the rule-base is too small then it affects the prediction and result of the system due to insufficient information. A fuzzy reasoning system can be implemented using compositional rule of inference (CRI) if the number of rules are sufficiently large

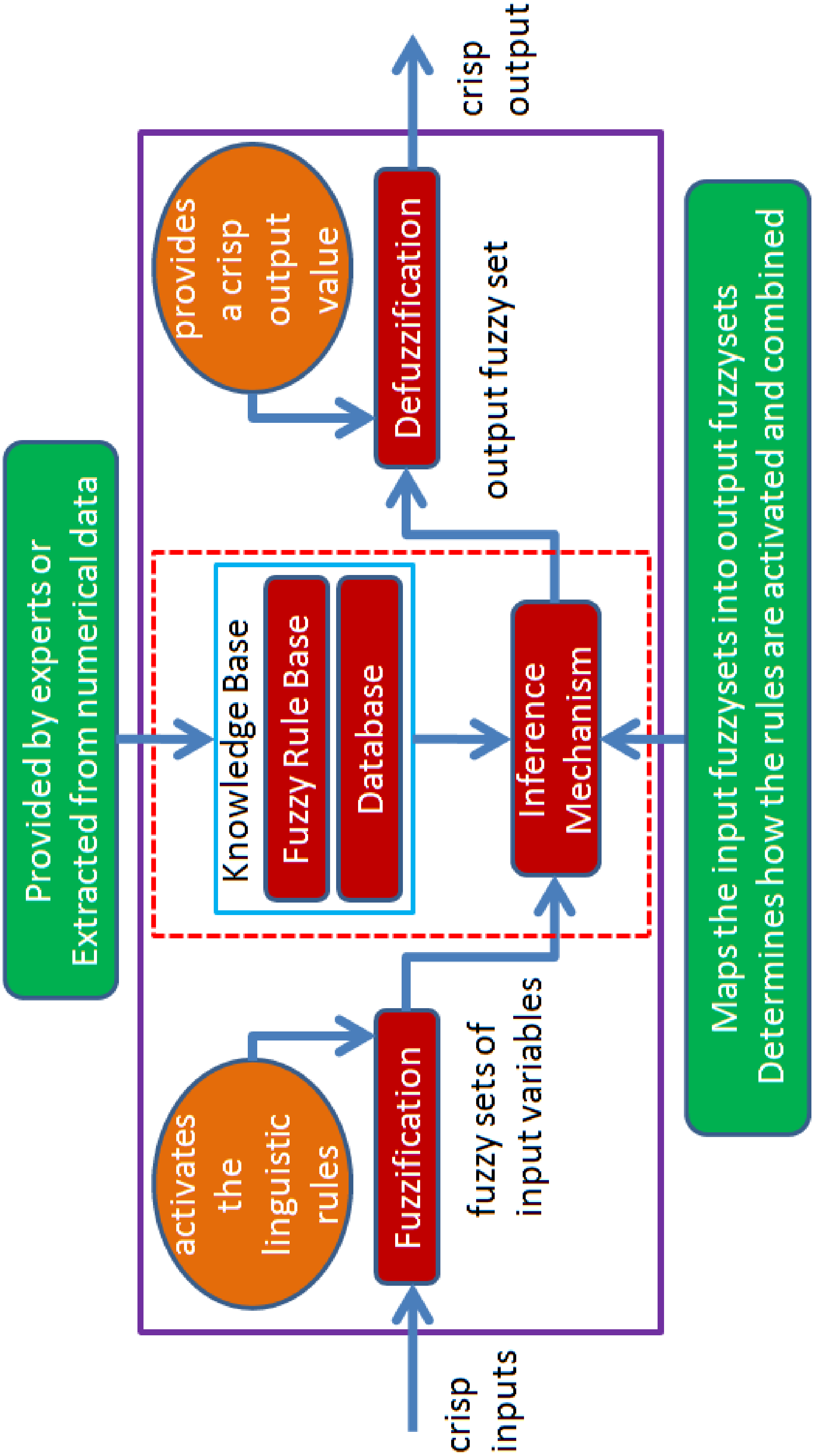


Figure 2.1: Fuzzy inference system

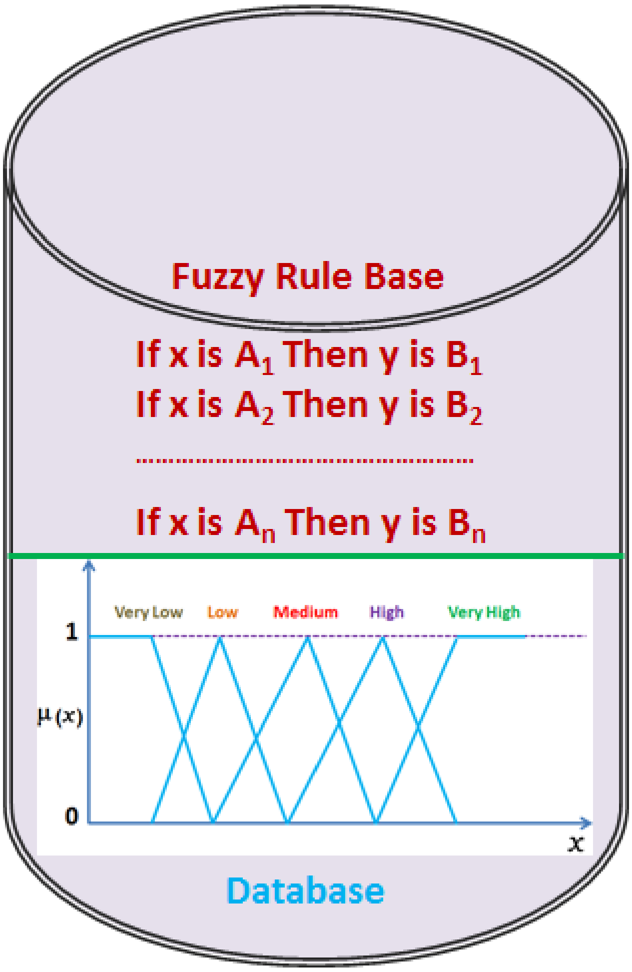


Figure 2.2: Fuzzy knowledge base

to cover the entire problem domain otherwise fuzzy rule interpolation (FRI) would be a reasoning option when rules are sparse. In the next four subsections of this chapter, these two CRI and FRI reasoning methods are explained in detail.

2.2 Compositional Rule of Inference (CRI)

Fuzzy systems use a fuzzy rule base (set of rules) to contain knowledge that is exploited to make inference by the inference mechanism. A fuzzy rule base is fully covered (at level α), if all input universes are covered by rules at level α. Such fuzzy rule bases are also called dense or complete rule bases. In practice, it means that for all the possible observations there exists at least one rule, whose antecedent part overlaps the input data at least partially at level α . If this condition is not satisfied, the rule base is considered sparse, i.e. containing gaps.

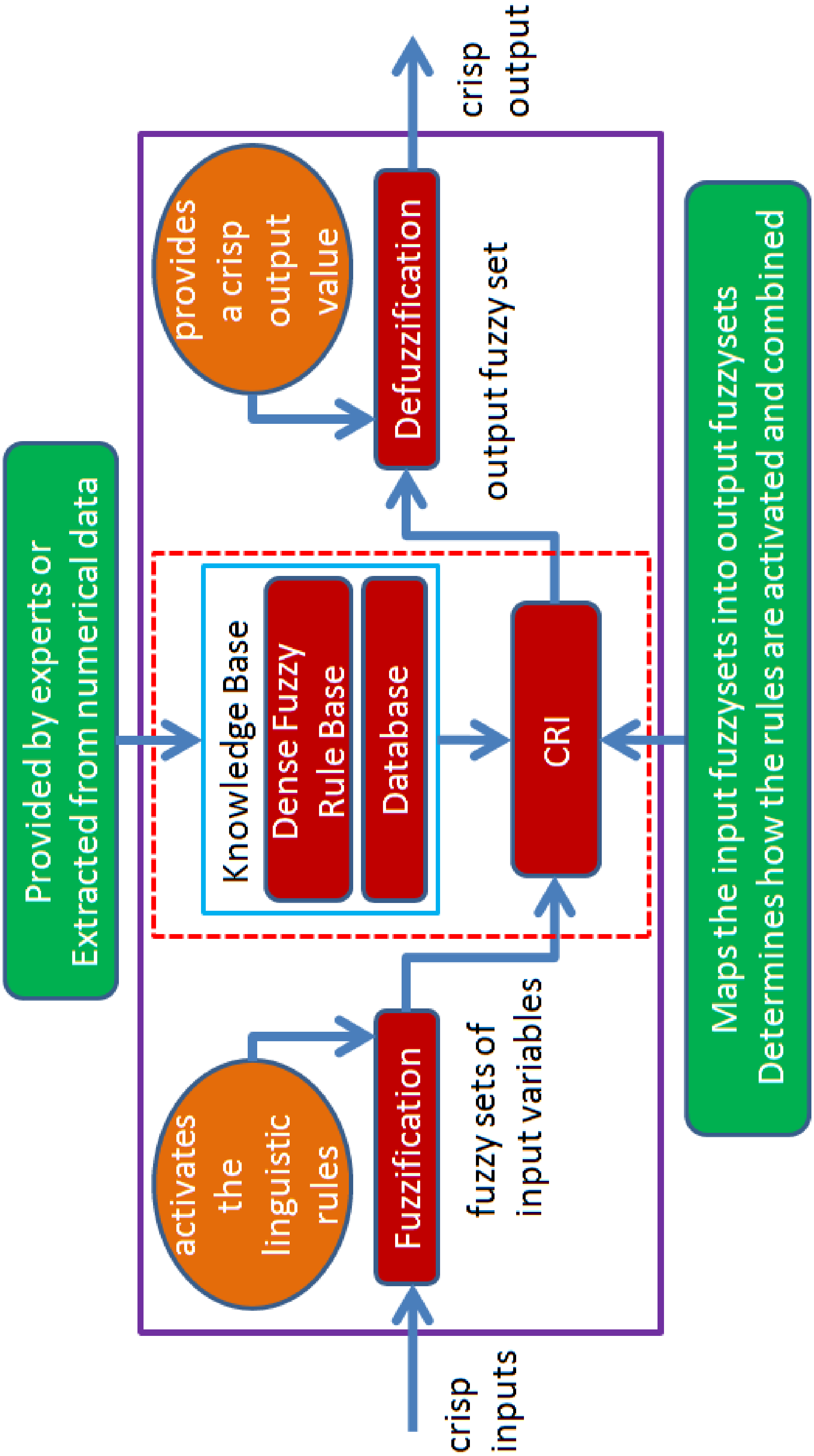


Figure 2.3: Compositional rule of inference system

In order to draw conclusions from a dense rule base, one needs a mechanism that can produce an output from a collection of rules. The most commonly used inference process for dense rule bases is the compositional rule of inference (CRI) [208, 209], as shown in Figure 2.3. For a given observation, in order to obtain a meaningful inference result based on CRI, there are two basic approaches: First Infer - Then Aggregate (FITA) and First Aggregate - Then Infer (FATI). In the FITA approach, for a given observation, we first perform inference using CRI on each of the rules in the rule base, and then combine all these intermediate results. Whereas, in the FATI approach, we first aggregate all the rules by forming an overall fuzzy relation *R* which is the combination of all the fuzzy implication relations and then inference is performed on the given observation.

Inference process based on CRI changes the membership function grades of the right hand sides of the corresponding rules either by reducing or by increasing the membership grades [182]. CRI is also called generalised modus ponens (GMP). For example, here reasoning is performed with one rule using CRI based on FATI. With a single rule and an observation, an inference result can be deduced as follows:

$$\begin{aligned}
&Rule: \ If \ \ x \ is \, A \, Then \, z \ is \ C \\
&Observation: \ x \ is \, A' \\
&Consequence: \ z \ is \ C'
\end{aligned}$$

where A, $A^{'} \subset X$, $C \subset Z$ are fuzzy sets defined in the universes of discourse X and Z, $x \in X$, and $z \in Z$. The fuzzy rule is interpreted as an implication (→):

$$R: A \rightarrow C$$

When input observation A' is given to the inference system, the output consequence would be calculated:

$$C' = A' \ o \ R = A' \ o \ (A \rightarrow C)$$

Where o is the composition operator. This inference procedure is called compositional rule of inference as shown in Figure 2.4. Here the inference mechanism is determined by two factors: 1. implication operators such as *min, product, etc.* and 2. composition operator such as *max-min, max-product, etc*. Therefore, it is clear that an inference process based on CRI includes several stages. More specifically, it includes implication, composition and combination for FITA, and implication, combination and composition for FATI.

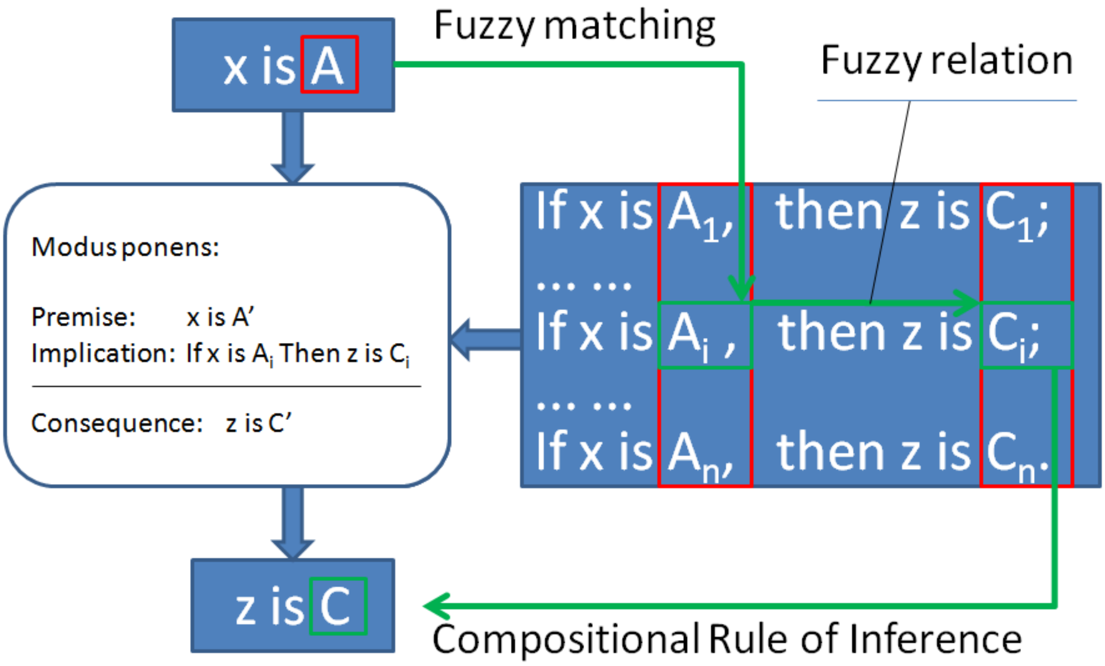


Figure 2.4: Compositional rule of inference example

2.3 Overview of Typical CRI Methods

There are many methods to select from in order to implement the required implication, composition and combination operators to perform compositional rule of inference. The most common fuzzy inference methods based on compositional rule of inference (CRI) are Mamdani, Takagi-Sugeno-Kang (TSK) and Tsukamoto fuzzy inference methods. The Mamdani fuzzy inference is the most commonly seen. This method was introduced by Mamdani and Assilian in 1975 [137]. Another well-known inference method is Takagi-Sugeno-Kang (TSK) method. This inference method was introduced by Takagi,Sugeno and Kang in 1985 [173, 175]. Tsukamoto mehtod is proposed by Tsukamoto in 1979 [179, 180] but it is less common in use.

The main difference between the two popular Mamdani inference method and TSK inference method is the way the crisp output is generated from the fuzzy inputs. While Mamdani system uses the technique of defuzzification of a fuzzy output, TSK system uses weighted average to compute the crisp output. The expressive power and interpretability of Mamdani output is reduced in the TSK systems since the consequents of the rules are not fuzzy [74]. However, TSK has better processing time since the weighted average replaces the time consuming defuzzification process. Due to the interpretable and intuitive nature of the rule base, Mamdani inference systems are widely used in particular for decision support applications [102].

2.3.1 Mamdani Fuzzy Inference Systems

From the introduction of fuzzy sets by Zadeh in 1965 [208], fuzzy logic has become a significant area of interest for researchers in artificial intelligence. In particular, Mamdani was the pioneer who investigated the use of fuzzy logic for interpreting the human derived control rules, and therefore his work was considered a milestone application of this theory [65]. The original Mamdani fuzzy inference system was proposed as the first attempt to control a steam engine and boiler combination by a set of linguistic control rules obtained from experienced human operators. A fuzzy system with two inputs x and y (antecedents) and a single output z (consequent) is described by a linguistic *If-Then* rule in Mamdani form as [137]:

$$If\ x\ is\ A\ and\ y\ is\ B\ Then\ z\ is\ C \tag{2.1}$$

where A and B are fuzzy sets in the antecedent and C is a fuzzy set in the consequent.

Figure 2.5 is an illustration of how a two-rule Mamdani fuzzy inference system derives the overall output z when subjected to two crisp inputs x and y. If we adopt *mim and max* as our choice for the T-norm and T-conorm operators, respectively, and use the original *max-min* composition, then the resulting fuzzy reasoning is shown in Figure 2.5, where the inferred output of each rule is a fuzzy set scaled down by its firing strength via *max*. Other variations are possible if we use different T-norm and T-conorm operators. For example, using *product and max* for T-norm and T-conorm operators, respectively results in the *max-product* composition.

In general, to compute the output for the given input observation, a Mamdani inference system follows the following steps:

1. Determining a set of fuzzy rules
2. Fuzzifying the inputs using the input membership functions
3. Combining the fuzzified inputs according to the fuzzy rules to establish a rule strength
4. Finding the consequence of the rule by combining the rule strength and the output membership function
5. Combining the consequences to get an output distribution

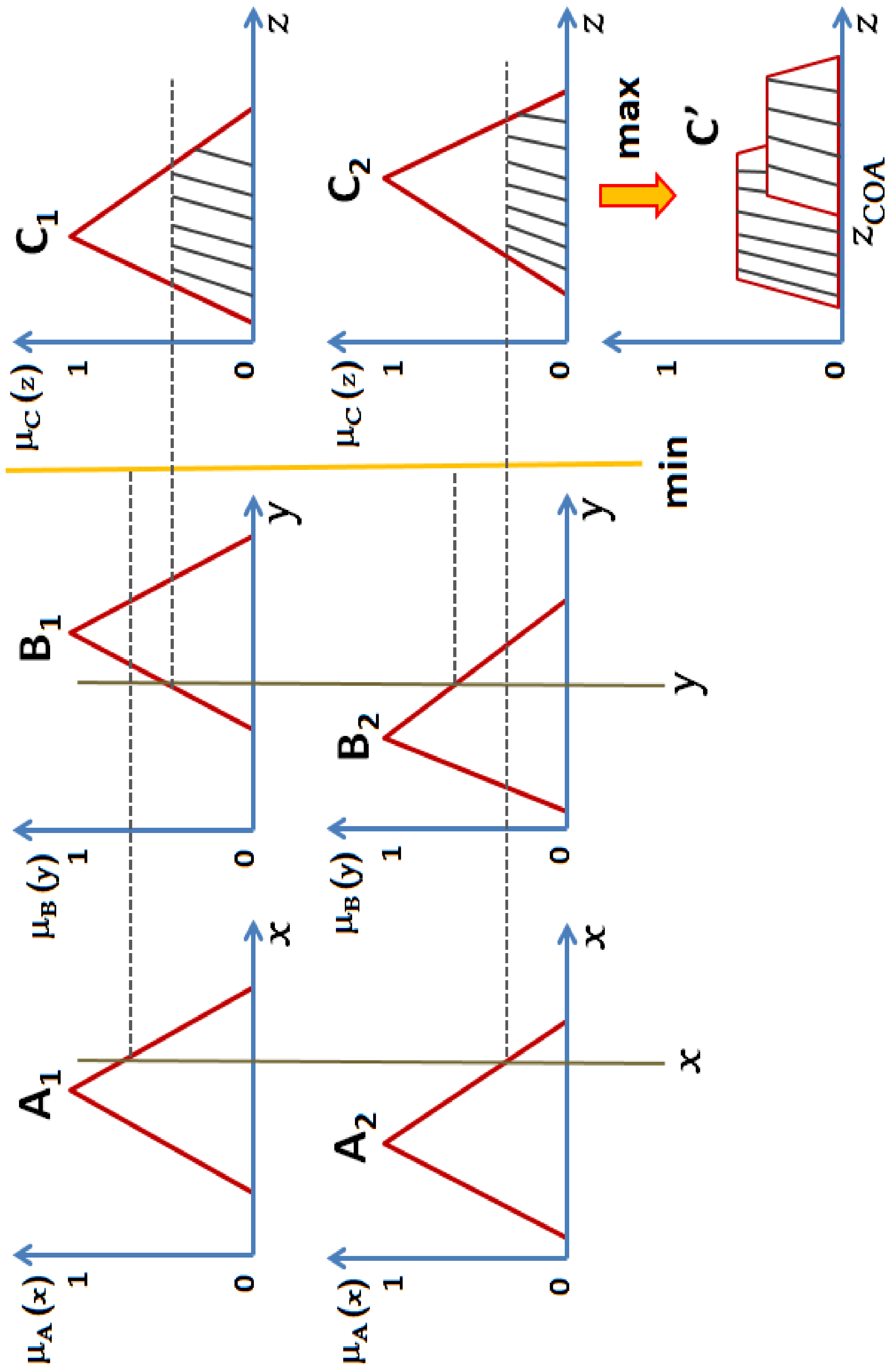


Figure 2.5: Mamdani fuzzy inference system

6. Defuzzifying the output distribution (this step is involved only if a crisp output (class) is needed)

2.3.2 Takagi-Sugeno-Kang (TSK) Fuzzy Inference Systems

Tagaki, Sugeno, and Kang [173, 175] had investigated a new approach to fuzzy inference models with the emphasis upon systematic methods of generating fuzzy rules from given sets of input-output data. A fuzzy system with two inputs x and y (antecedents) and a single output z (consequent) is described by a linguistic *If-Then* rule in Takagi-Sugeno-Kang form as [173, 175]:

$$If\ x\ is\, A\, and\ y\ is\, B\ Then\, z = f(x, y) \tag{2.2}$$

where fuzzy antecedent variables A and B, give rise to the consequent crisp function $z = f(x, y)$. Furthermore, $f(x, y)$, in many cases can be expressed as a polynomial with inputs x and y. These functions are categorized as first order, second order, to n^{th} order, determined by the order of the polynomial describing their behaviour.

First order polynomials of $f(x, y)$, will result in first order TSK fuzzy models proposed by Tagaki, Sugeno, and Kang [173, 175]. If $f(x, y)$ is a constant, then in this instance a zero order TSK fuzzy model is produced. This could be considered as a special case of the Mamdani fuzzy inference, in which the consequent of each rule is specified by a fuzzy singleton (or a pre-defuzzified consequent), or a special case of the Tsukamoto fuzzy inference [179], where the consequent of each rule is specified by a membership function (MF) of a step function centred at the constant. The output of a zero-order TSK inference model is a smooth function of its input variables provided that the neighbouring MFs in the antecedent have sufficient overlap.

The first-order TSK fuzzy inference procedure is shown in Figure 2.6. Here, both rules have crisp outputs so the final output can be calculated through weighted average, therefore Mamdani model's time-consuming process of defuzzification can be avoided. Sometimes, the weighted average operator can also be replaced with the weighted sum operator (that is, $z = w_1 z_1 + w_2 z_2$ in Figure 2.6) so computation would be further reduced. Conversely, this generalization could lead to the loss of MF linguistic meanings except the sum of firing strengths (that is, $\sum_i w_i$) is close to unity [33]. In TSK inference models, the differentiation between a fuzzy rule and non-fuzzy rule is quite obvious as only its antecedent part is fuzzy.

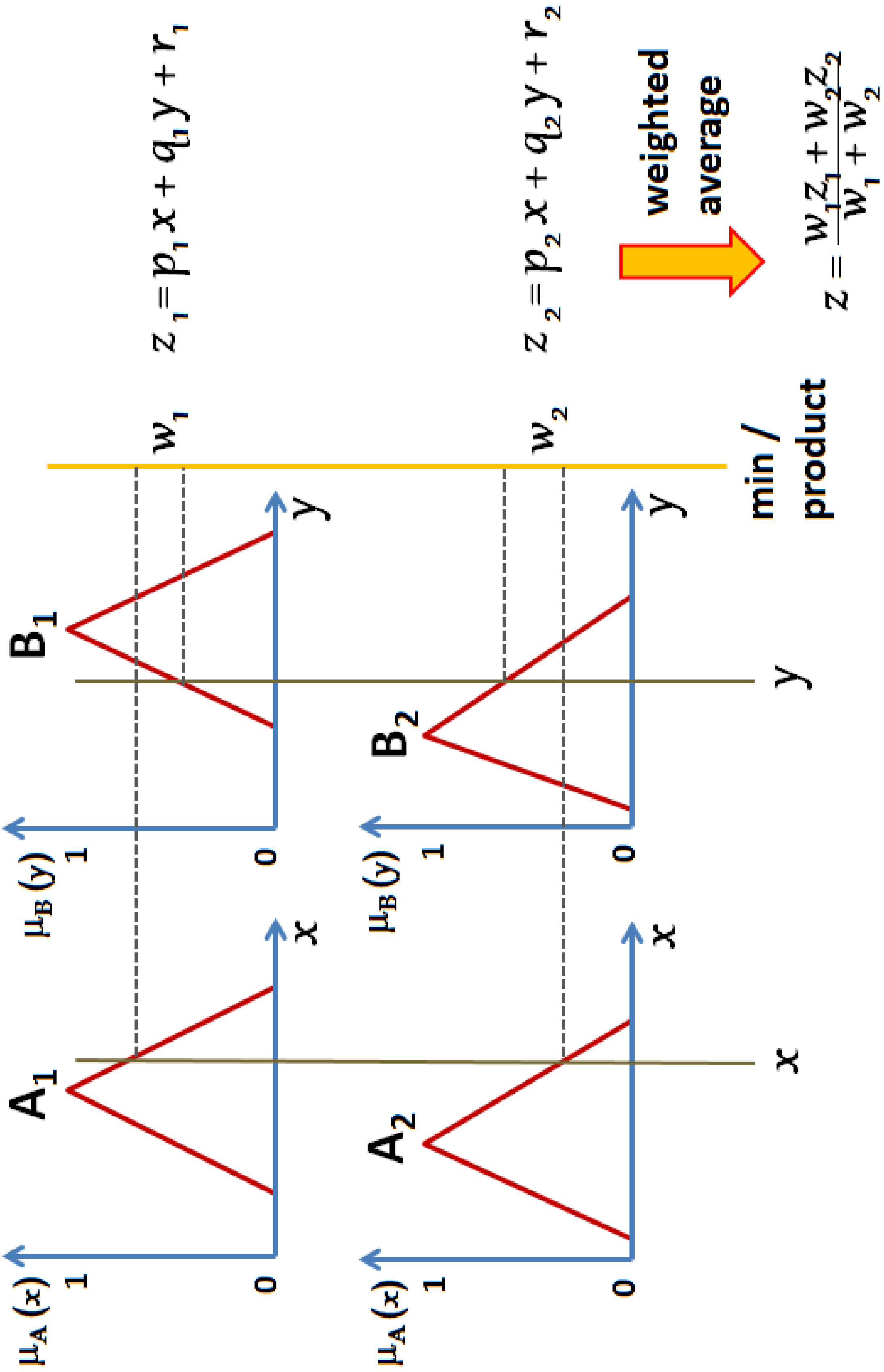


Figure 2.6: Takagi-Sugeno-Kang (TSK) fuzzy inference system

A TSK fuzzy inference model is not a strict compositional rule of inference model. Where, the matching of fuzzy sets can still be used to find the firing strength of each rule [33] which is shown in the antecedent part of Figure 2.6. While the final output is always a crisp output whether it is based on weighted average or weighted sum; this does not seem logically correct because a fuzzy model should be able to transmit the fuzziness from inputs to outputs rationally. Nevertheless, TSK fuzzy inference is a common option for data-oriented fuzzy modelling when simplified defuzzification is required.

2.3.3 Tsukamoto Fuzzy Inference Systems

In the Tsukamoto fuzzy inference model, the consequent of each fuzzy *If-Then* rule is represented by a fuzzy set with a *monotonical* membership function, as shown in Figure 2.7. As a result, the inferred output of each rule is defined as a crisp value induced by the rule's firing strength. The overall output is taken as the weighted average of each rule's output [33]. A fuzzy system with two inputs x and y (antecedents) and a single output z (consequent) is described by a linguistic *If-Then* rule in Tsukamoto form as [179]:

$$If\ x\ is\ A\ and\ y\ is\ B\ Then\ z\ is\ C\ (monotonic) \tag{2.3}$$

where A and B are fuzzy sets in the antecedent and C is a fuzzy set in the consequent (with a monotonical membership function). Figure 2.7 illustrates the Tsukamoto reasoning procedure for a two-input two-rule system. Since each rule infers a crisp output, the Tsukamoto fuzzy model aggregates each rule's output by the method of weighted average and thus avoids the time-consuming process of defuzzification, similar to TSK inference systems [32]. Suppose that the firing degree inferred from the first rule is z_1 such that $w_1 = c_1\ (z_1)$ and the firing degree inferred from the second rule is z_2 such that $w_2 = c_2\ (z_2)$. An overall crisp output in Tsukamoto inference system can therefore be expressed as a weighted average z:

$$z = \frac{w_1 z_1 + w_2 z_2}{w_1 + w_2} \tag{2.4}$$

Since the reasoning mechanism of the Tsukamoto inference fuzzy system does not follow strictly the compositional rule of inference, the output is always crisp even when the inputs are fuzzy. However, the Tsukamoto fuzzy model is not used often as it is not as simplified inference model as either the Mamdani or TSK fuzzy inference systems.

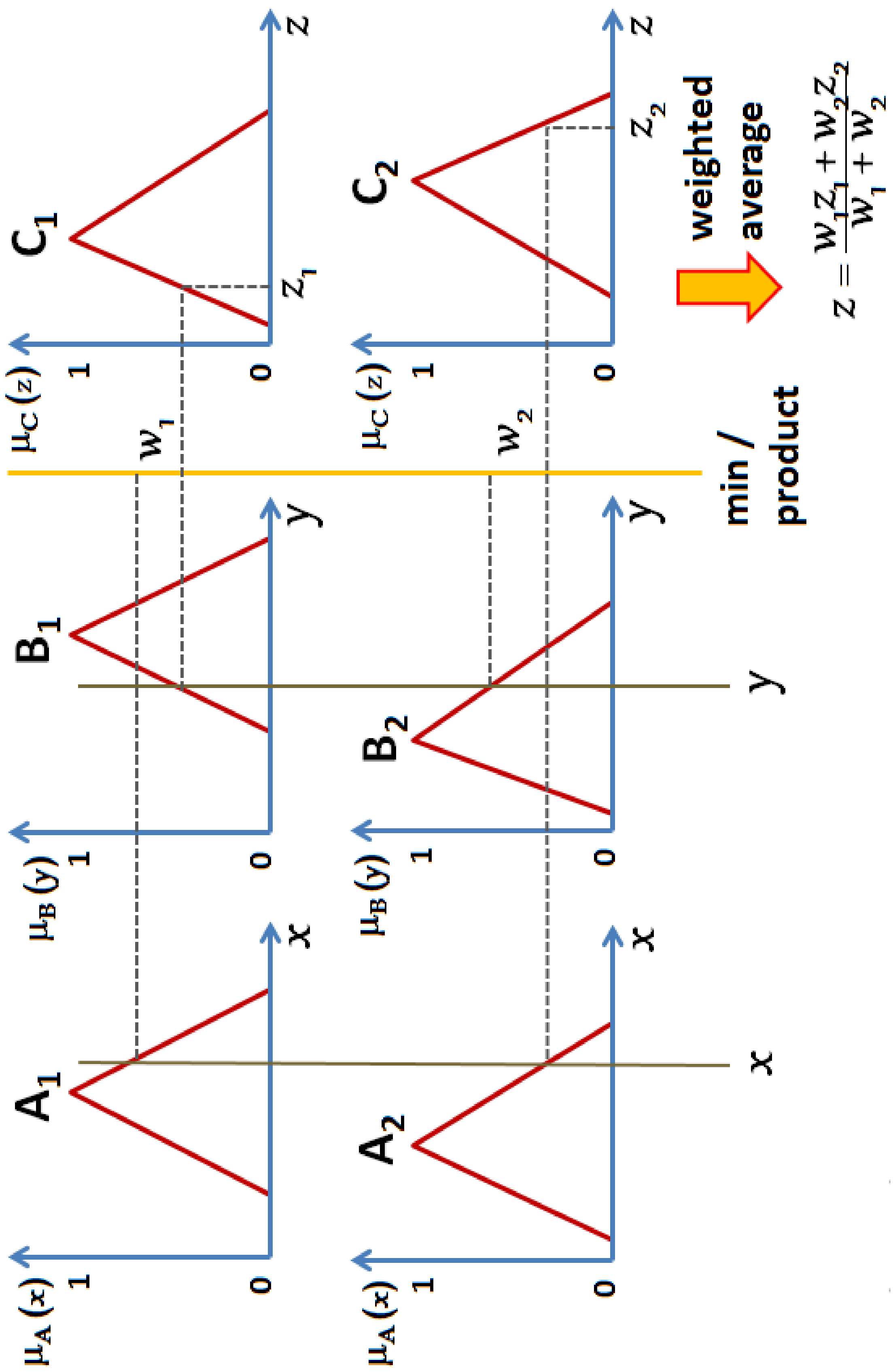


Figure 2.7: Tsukamoto fuzzy inference system

2.4 Fuzzy Rule Interpolation (FRI)

A dense rule-base has all input domains covered by rules completely whereas sparse rule-bases are incomplete, containing gaps among the rules. Thus the reasoning in dense rule-base is relatively easy because every observation matches with a certain number of rules and conclusion can be determined on the basis of such rules directly. However, in sparse fuzzy rule-based systems, certain rules are missing. If the observation arises resemble the missing or unknown rules then no conclusion can be drawn using conventional inference mechanism [53, 54, 55, 110, 111, 112, 113]. This can be explained by the following example:

Suppose in the assumed sparse rule-base there are only two rules, which are given below:

$$Rule-1 = \begin{cases} Antecedent-1 \;\; If \;\; you\, eat\, over\, diet \\ Consequent-1 \;\; Then\, you\, become\, an\, obese\, person \end{cases}$$

$$Rule-2 = \begin{cases} Antecedent-2 \;\; If \;\; you\, eat\, less\, diet \\ Consequent-2 \;\; Then\, you\, become\, a\, lean\, person \end{cases}$$

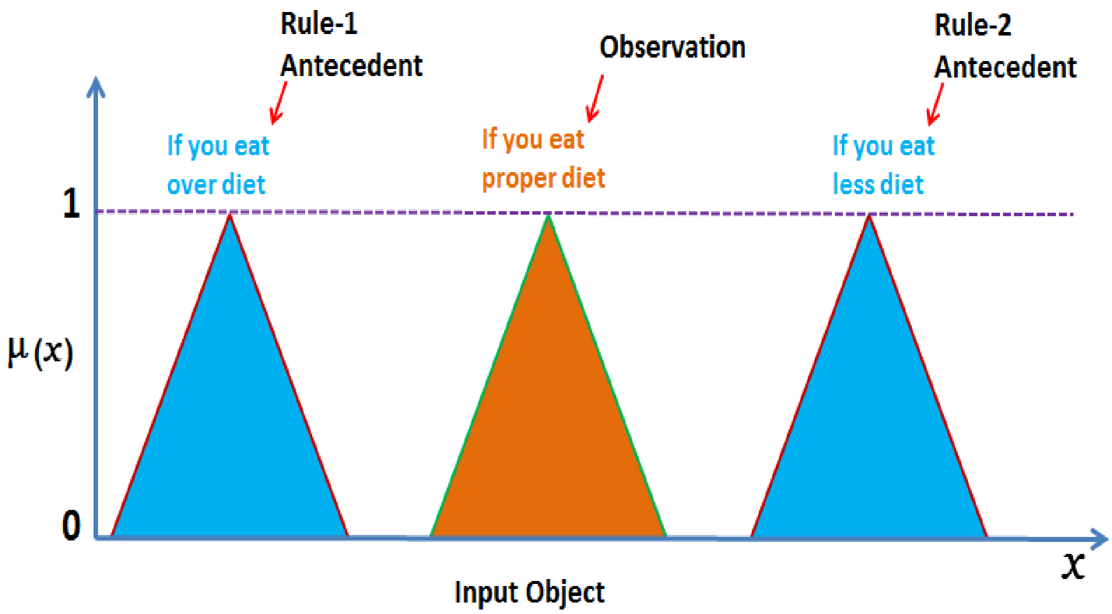
Now the following observation is found somewhere in between these two rules but there will be no result or conclusion could be determined using traditional inference methods:

$$Observation = \Big\{ \; Observed\, Antecedent - \;\; If \;\; you\, eat\, proper\, diet$$

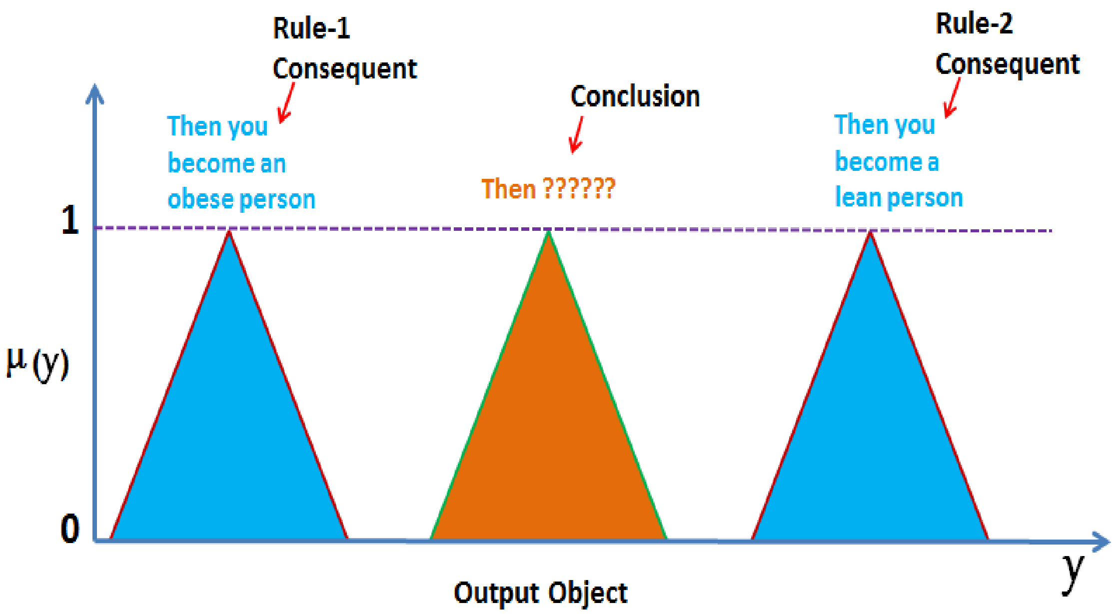
This observation resembles to a missing or unknown rule such as the following:

$$New-Rule = \begin{cases} Observed\, Antecedent - \;\; If \;\; you\, eat\, proper\, diet \\ Inferred\, Consequent - \;\; Then\, ??????????? \end{cases}$$

This is the common state for any sparse fuzzy rule-base system. This can also be represented diagrammatically in terms of fuzzy sets and their membership functions. For instance, these can be represented by triangular membership functions for the sake of simplicity, as shown in Figure 2.8. In this figure, x and y are input and output objects and $\mu(x)$ and $\mu(y)$ are the membership functions for rule antecedent and consequent respectively.



(a) Representation of rule antecedent parts in terms of graphical (triangular) fuzzy sets



(b) Representation of rule consequent parts in terms of graphical (triangular) fuzzy sets

Figure 2.8: General illustration of fuzzy interpolation

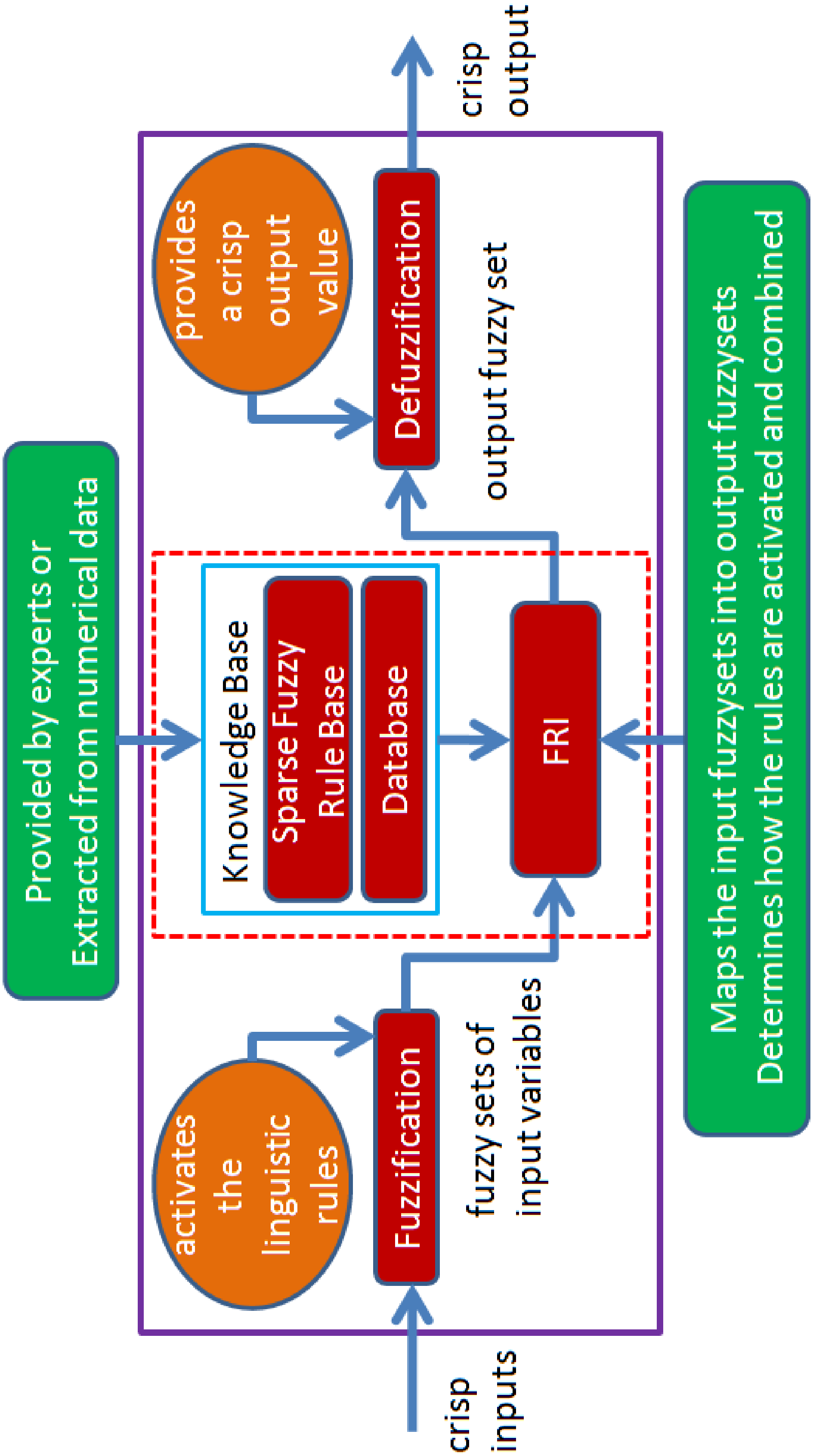


Figure 2.9: Fuzzy rule interpolation system

For such a problem a human being would infer the result - *You become a fit person* but conventional fuzzy inference methods such as CRI could not give this consequence. Fuzzy sparse rule base systems are employed to address this problem, via fuzzy rule interpolation as shown in Figure 2.9.

There are many fuzzy rule interpolation methods (e.g. [55, 99, 107, 108, 110, 111, 112, 113, 162]) available in the literature. A number of methods directly infer the conclusions on the basis of observation, such types of interpolation include: Linear Interpolation Method [110, 111, 112, 113], Cartesian Product Based Interpolation Method [55], Spline Based Interpolation Method [162], Extension KH Central Point Based Interpolation Method [186], Vague Environment Based Interpolation Method [119], Modified Alpha-Cut Interpolation Method [177], Conservation of the Fuzziness Based Interpolation Method [69], Conservation of the Relative Fuzziness Interpolation Method [115], Improved Multidimensional Modified Alpha-Cut Interpolation Method [194], and Slopes of Flaking Edges Interpolation Method [81].

There exist other types of fuzzy rule interpolation also instead of directly devising conclusions, such methods construct an approximated fuzzy rule on the basis of certain similarity principles and then give the conclusion by approximate transformation, these interpolation methods include: Similarity Transfer Based Interpolation Method [205], Spatial Geometric Representation Based Interpolation Method [17, 18], Graduality Based Interpolation Method [28], Flank Function Based Interpolation Method [89], Scale and Move Transformation Based Interpolation Method [85, 86], Cutting and Transformation Based Interpolation Method [109], B-Spline Technique Based Interpolation Method [104], Polar Cuts Based Interpolation Method [93], Least Squares Method Based Interpolation Method [94], Vague Environment Based Two-Step Interpolation Method [96], and Ranking Values of Fuzzy Sets Based Interpolation Method [125]. The following subsection 2.5 presents an overview of such typical approaches for fuzzy rule interpolation.

2.5 Overview of Typical FRI Methods

In this section existing fuzzy rule interpolation technologies are explained to help understand and potentially compare their underlying mechanisms. These methods are described using a unified style in pseudo code. This overview is carried out on the basis of common, simplified assumptions. Suppose that A is the family of rule antecedent fuzzy sets where fuzzy sets A_1, A_2 represent the two adjacent rule antecedents respectively such that $A_1, A_2 \in A \subseteq X$, where X is the input universe of discourse. Similarly B is the family of rule consequent fuzzy sets where fuzzy sets B_1, B_2 represent the two adjacent rule consequents respectively such that $B_1, B_2 \in B \subseteq Y$, where Y is the output universe of discourse. Also A^* is the input observation, and B^* is the approximated conclusion. The $\mu_A(x)$ is the membership function for the whole rule antecedent family A whereas the $\mu_B(y)$ denotes the membership function for the consequent family B. For the sake of simplicity only triangular or trapezoidal fuzzy sets are considered here.

2.5.1 Linear Interpolation Method (KH)

Koczy-Hirota (KH) linear interpolation method [110, 111, 112, 113] is based on the concept that the approximated conclusion divides the distance between the consequent sets of the used rules in the same proportion as the observation does the distance between the antecedents of those rules. This method is developed by using two fundamental principles. The first is the definition of the fuzzy distance using classical Shepard interpolation extension [168], and the second is the fact that fuzzy sets can be decomposed into and composed from α-cuts i.e. the resolution and extension principles [110, 111, 112, 113]. This method can also be applied to multidimensional environment using the Minkowski distance.

KH Algorithm

The algorithm is based on the exploitation of α-cut and fuzzy distance measures. The α-cut sets of rule antecedents A_1 and A_2 are $A_{1\alpha}$ and $A_{2\alpha}$ and the α-cuts sets of rule consequents B_1 and B_2 are $B_{1\alpha}$ and $B_{2\alpha}$ respectively, as shown in the Figure 2.11.

The fundamental requirements [110, 111, 112, 113] of KH linear interpolation method are:
1) All fuzzy sets must be Convex and Normal Fuzzy (CNF) Sets.
2) A partial ordering must hold amongst the CNF sets; meaning that a partial ordering must exist between the elements of the universes of discourses.
3) Fuzzy Distance (d) is calculated in terms of α-cut distance of CNF sets.
4) All the state variables (input and output universes) must be bounded.

The KH algorithm is implemented in the following steps, which is outlined in Figure 2.10:

KHInterpolation(X,Y,α)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
d, the fuzzy distance metric between fuzzy sets;
$inf\ \{A_\alpha\}$, the infimum value of crisp set A_α;
$sup\ \{A_\alpha\}$, the supremum value of crisp set A_α.

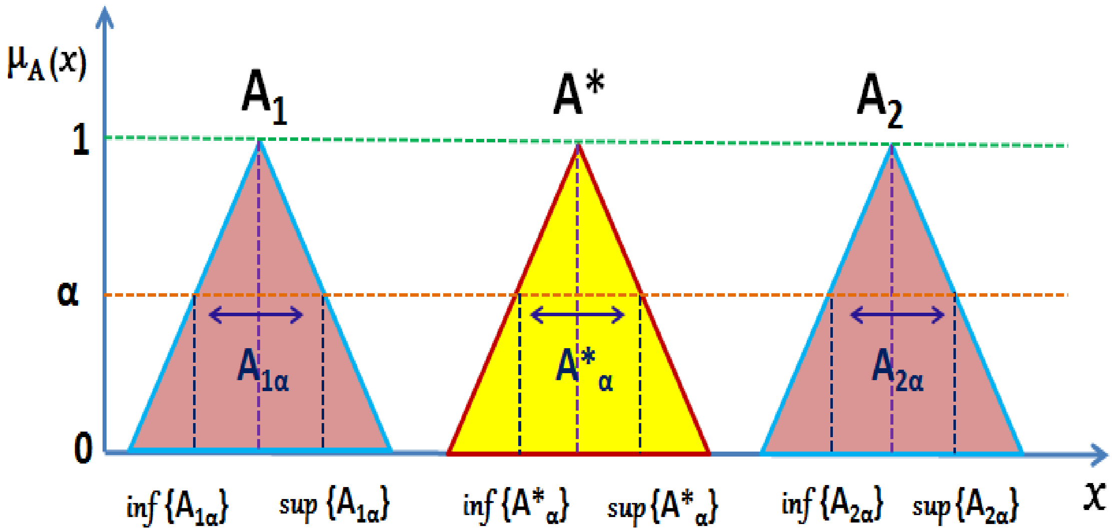
(1) $A_1,A_2 \in X, B_1,B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $d_L\left(A^*_\alpha,A_{1\alpha}\right) \leftarrow d\left(inf\ \left\{A^*_\alpha\right\}, inf\ \{A_{1\alpha}\}\right)$
(5) $d_L\left(A^*_\alpha,A_{2\alpha}\right) \leftarrow d\left(inf\ \left\{A^*_\alpha\right\}, inf\ \{A_{2\alpha}\}\right)$
(6) $d_U\left(A^*_\alpha,A_{1\alpha}\right) \leftarrow d\left(sup\ \left\{A^*_\alpha\right\}, sup\ \{A_{1\alpha}\}\right)$
(7) $d_U\left(A^*_\alpha,A_{2\alpha}\right) \leftarrow d\left(sup\ \left\{A^*_\alpha\right\}, sup\ \{A_{2\alpha}\}\right)$
(8) $d\left(A^*,A_1\right)/d\left(A^*,A_2\right) = d\left(B^*,B_1\right)/d\left(B^*,B_2\right)$
(9) $inf\ \{B^*\} \leftarrow \left[d_L\left(A^*_\alpha,A_{1\alpha}\right) inf\ \{B_{2\alpha}\} + d_L\left(A^*_\alpha,A_{2\alpha}\right) inf\ \{B_{1\alpha}\}\right]$
$/d_L\left(A^*_\alpha,A_{1\alpha}\right) d_L\left(A^*_\alpha,A_{2\alpha}\right)$
(10) $sup\ \{B^*\} \leftarrow \left[d_U\left(A^*_\alpha,A_{1\alpha}\right) sup\ \{B_{2\alpha}\} + d_U\left(A^*_\alpha,A_{2\alpha}\right) sup\ \{B_{1\alpha}\}\right] / d_U\left(A^*_\alpha,A_{1\alpha}\right) +$
$d_U\left(A^*_\alpha,A_{2\alpha}\right)$
(11) $B^* \leftarrow \bigcup_{\alpha\in[0,1]} \alpha B^*_\alpha$

Figure 2.10: The KHInterpolation Pseudocode.

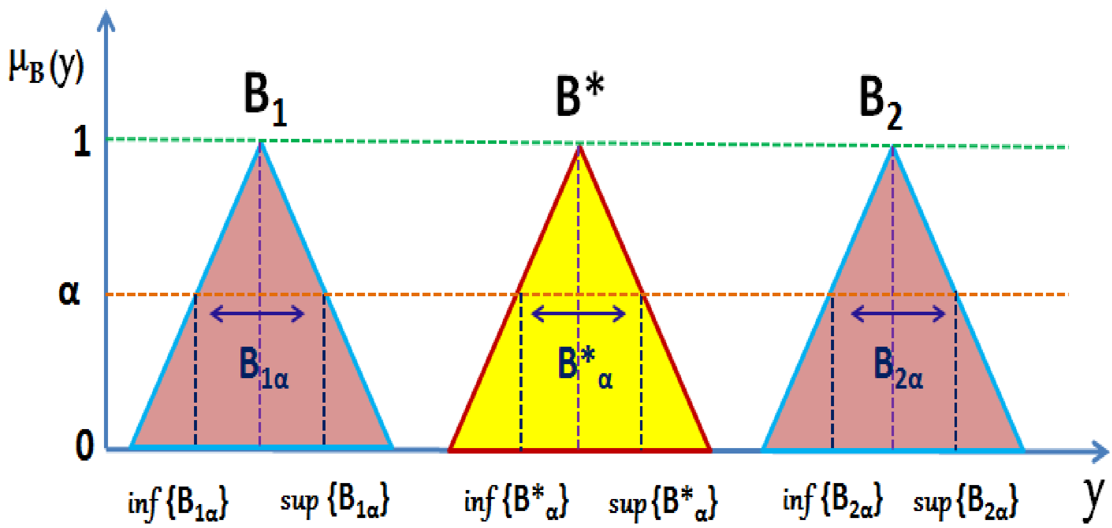
1. Determine the α-cut sets and infimum and supremum values for all the available fuzzy sets as shown in Figure 2.11.
2. Determine the lower and upper distances through infimum and supremum values of two α-cut sets by Euclidean distance (single dimension) and Miskowski distance (multi dimension).
3. Determine the infimum and supremum values of conclusion fuzzy set by applying the linear interpolation formula.

2.5.2 Cartesian Product Based Interpolation Method (CP)

As implied by its name, the method is based on the concept of forming the interpolating relation in the Cartesian product (CP) of input and output space [55]. In terms of



(a) Concepts of Fuzzy Distance and α -cuts for antecedent fuzzy sets in KH method including observation



(b) Concepts of Fuzzy Distance and α -cuts for consequent fuzzy sets in KH method including conclusion

Figure 2.11: KH fuzzy interpolation method

α-cuts, the fuzzy relation of two α-cut sets can be defined by the Cartesian product of the related α-cuts of the fuzzy subsets. Unlike the KH method which has well defined set of conditions for the fuzzy sets this technique is more general and applicable to arbitrary and multidimensional fuzzy sets. However it has problems regarding to subnormal conclusion and numerical estimation for constructing interpolating fuzzy relation.

CP Algorithm

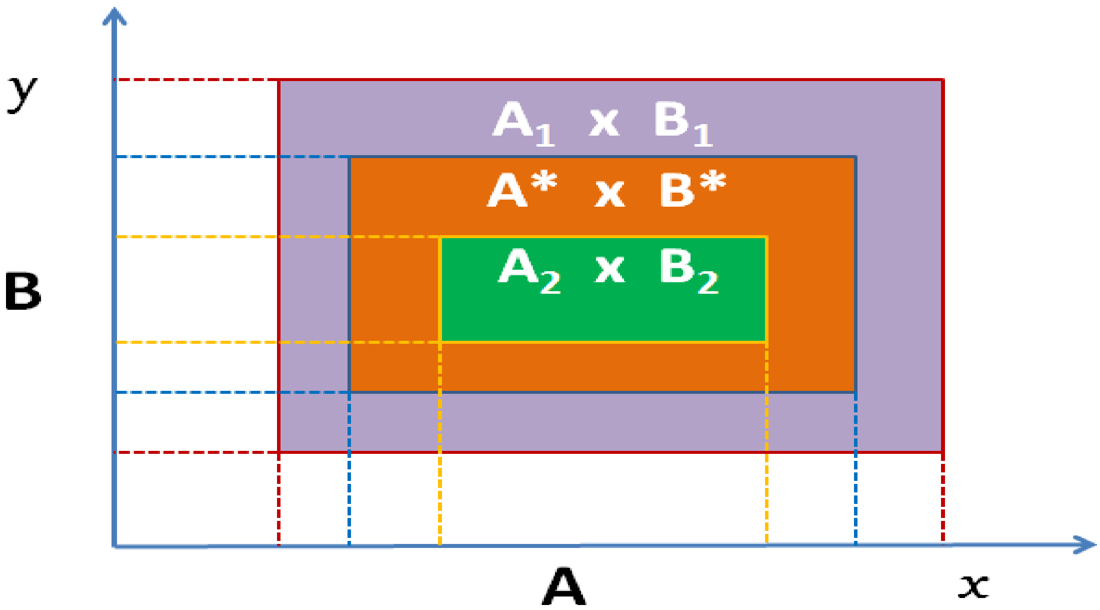
The algorithm is based on the two definitions - Fuzzy Relation and Cartesian Product of fuzzy sets as defined below:

- Fuzzy Relation - A fuzzy relation describes the degree of association of the elements of the two fuzzy sets. Thus by using fuzzy relations the vagueness in the relation between the sets and their elements can be measured.
- Cartesian Product of Fuzzy Sets - If A and B are the two fuzzy sets in the universes of discourse X and Y respectively, then the Cartesian product $(A \times B)$ of these two fuzzy sets is a fuzzy set in the product space $X \times Y$ that is defined by their membership functions $\mu_A(x)$ and $\mu_B(y)$ such that:

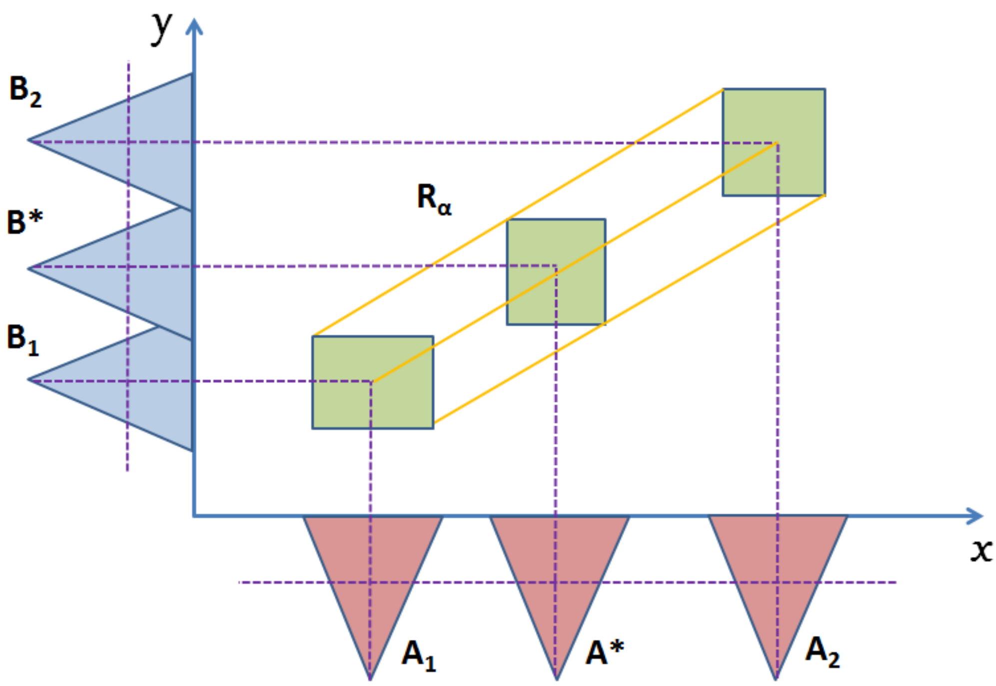
$$\mu_{A\times B}(x, y) = min[\mu_A(x), \mu_B(y)] \tag{2.5}$$

It is obvious that the Cartesian product of the two fuzzy sets is a fuzzy relation which is shown in Figure 2.12.

The CP algorithm is implemented in the following steps, which is outlined in pseudo code as given in Figure 2.13:
1. Determine the α-cuts and infimum and supremum values for all the available fuzzy sets.
2. Determine the interpolating relation in the Cartesian product of the input and output universe based on the resulting α-cuts.
3. Determine the interpolating conclusion as the greatest fuzzy subset by using any systematic technique for solving relational equations.



(a) Cartesian Product of two fuzzy sets in CP method



(b) Representation of Cartesian Products and Relations for antecedent and consequent fuzzy sets in CP method

Figure 2.12: CP fuzzy interpolation method

CPInterpolation(X,Y,α)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
$\diamond$, the product operator;
F, the fuzzy set for input or output universe;
R, the fuzzy relation of antecedent and consequent fuzzy set.

(1) $A_1, A_2 \in X, B_1, B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $\mu_{A\times B}(x,y) \leftarrow min[\mu_A(x), \mu_B(x)]$
(5) $A \times B \leftarrow \{(x,y)|x \in X, y \in Y\}$
(6) $R(x,y) \leftarrow \{(x,y), \mu_R(x,y)|(x,y) \in A \times B, \mu_R(x,y) \in [0,1]\}$
(7) $A_i \diamond B_i \in F(X \times Y)$
(8) $(A_i \diamond B_i)(x,y) \leftarrow min(A_i(x), B_i(x)), for\ i = 1,2$
(9) $min(B^*(y); A^*(x)) \leq R(x,y), \forall (x,y) \in X \times Y$
(10) $B^* \leftarrow max\{F(Y)\}$

Figure 2.13: The CPInterpolation Pseudocode.

2.5.3 Vague Environment Based Interpolation Method (FIVE)

This interpolation method is based on the concept of Vague Environment given in [107, 108]. The vague environment can be defined on the basis of similarity or indistinguishability of the elements. Specially, this technique is developed by exploiting the concept of the scaling function [107, 108], the most critical task is to find the approximate scaling function.

FIVE Algorithm

The algorithm is based on the concept of equality relation $\approx$ (also called similarity relation or indistinguishability operator) as shown in Figure 2.15 which is defined on the set X as a mapping $E: X \times X \to [0,1]$ that satisfies the following three axioms:

$$E_{\approx}(x,x) = 1;\ E_{\approx}(x,y) = E_{\approx}(y,x);\ T(E_{\approx}(x,y), E_{\approx}(y,z)) \leq E_{\approx}(x,z) \tag{2.6}$$

where X is the underlying domain and T is any lower semi-continuous T-norm.

FIVEInterpolation(X,Y,α,ϵ)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
ϵ, the indistinguishability parameter;
δ_s, the vague distance;
S, the scaling function.

(1) $A_1, A_2 \in X, B_1, B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $\delta_s(x_1, x_2) \leftarrow \left| \int_{x_2}^{x_1} S(x)dx \right|$
(5) $If\, \epsilon > \delta_s(x_1, x_2)$
(6) $\mu_A(x) \leftarrow 1 - min\{\delta_s(a,b), 1\}$
(7) $\mu_A(x) \leftarrow 1 - min\left\{ \left| \int_a^b S(x)dx \right|, 1 \right\}$
(8) $\delta_s(a,b) \leftarrow 1 - \alpha$
(9) $S(x) = \left| \mu'(x) \right| \leftarrow |d\mu/dx|$
(10) $min\left\{\mu_i(x), \mu_j(x)\right\} > 0 \Rightarrow |\mu_i(x)| = \left|\mu_j(x)\right|$, $\forall\, i,j \in I$
(11) $S(x) \leftarrow Approximate[S(x)]$
(12) $B^* \leftarrow ClassicalInterpolation(A_1, A_2, A^*, B_1, B_2)$

Figure 2.14: The FIVEInterpolation Pseudocode.

The FIVE algorithm is implemented in the following steps, which is outlined in the pseudo code as given in Figure 2.14:
1. Determine the connection between the similarity of two fuzzy sets and the vague distance of points in a vague environment.
2. Generate vague environment from the fuzzy partitions of the linguistic terms within the fuzzy rules.
3. Decide on the approximate scaling function.
4. Calculate the conclusion by approximating the vague points of the rulebase using any classical interpolation method as shown in Figure 2.16.

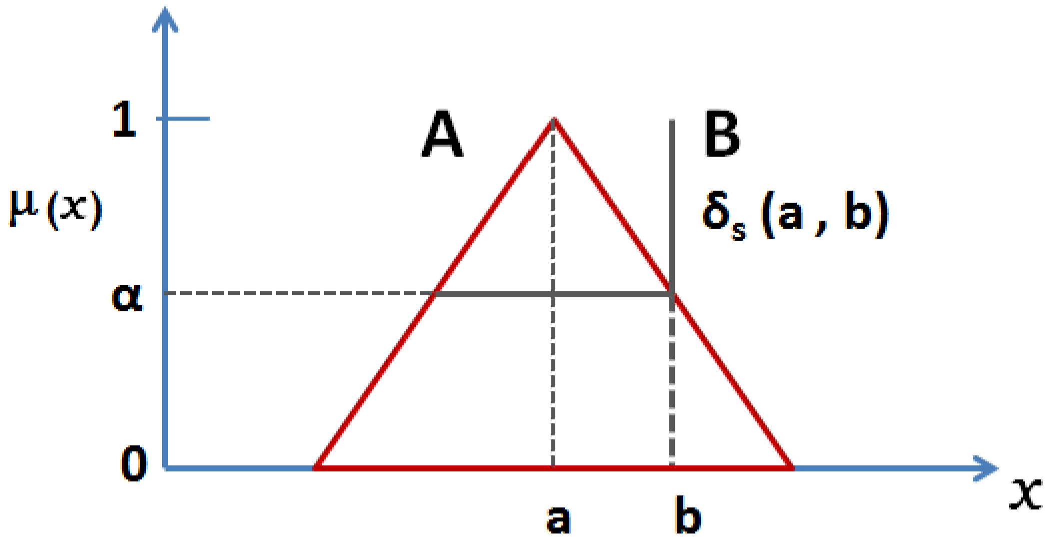


Figure 2.15: Representation of α-cut on the basis of Indistinguishability in FIVE

2.5.4 Modified α-Cut Based Interpolation Method (MACI)

This method is according to its name, a modified version of the KH method in which the vector representation of fuzzy sets and transformation concepts are used [177, 176]. It is specially developed to avoid the problem of abnormal conclusion while maintaining the low computational complexity of the KH method. It transforms the fuzzy sets of the input and output universes to such coordinate system where abnormality is excluded. Then the conclusion is calculated and finally, this conclusion is transformed back to the original coordinate system.

MACI Algorithm

The algorithm is based on the concepts of vector representation of fuzzy set and characteristics points as shown in Figure 2.17. By representing sets as vectors, it means that each fuzzy set is defined by a vector of its characteristics points.

The MACI algorithm is implemented in the following steps, which is outlined in the pseudo code as given in Figure 2.18:
1. Select the appropriate coordinate system for the output space.
2. Apply KH interpolation for inferring conclusion on the transformed fuzzy sets.
3. Transform the conclusion into the original coordinate system.

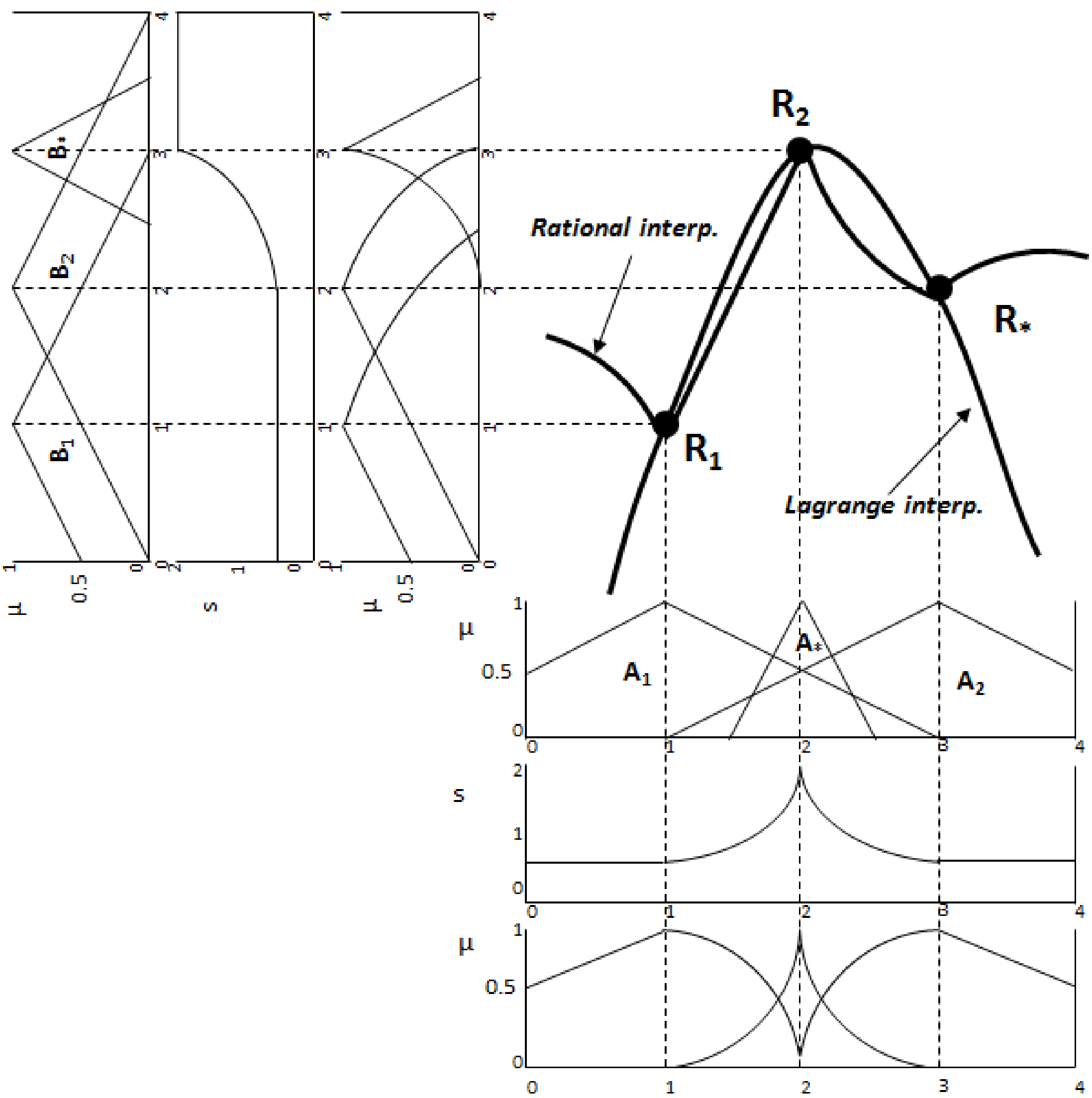


Figure 2.16: Interpolation of fuzzy rules by using Lagrange and Rational method in FIVE [119]

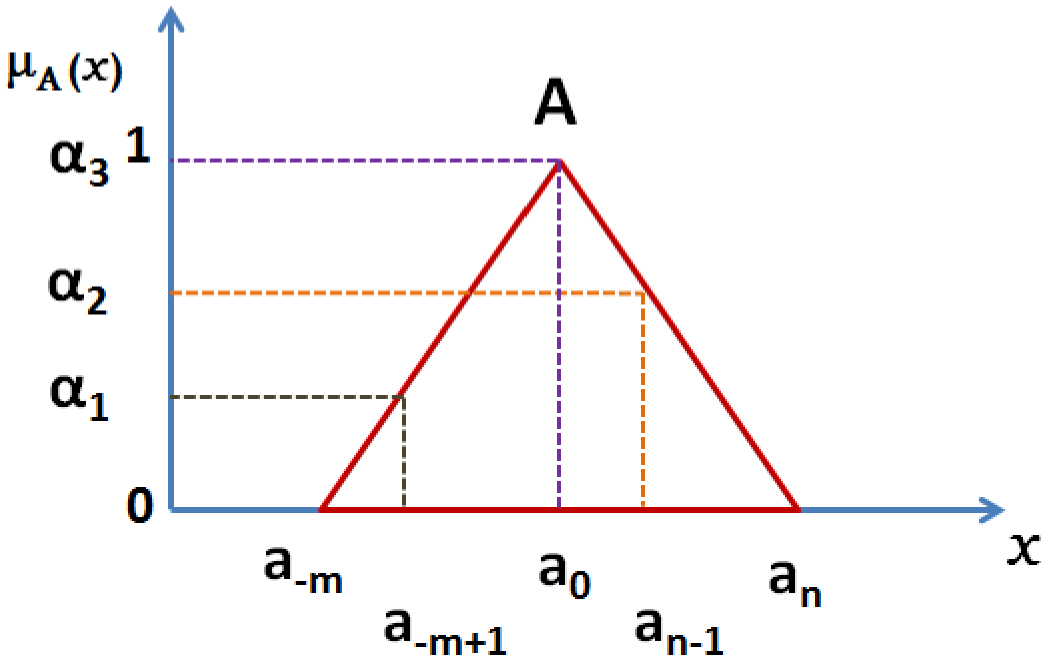


Figure 2.17: Vector representation of fuzzy set in MACI method

2.5.5 Improved Multidimensional Modified α-Cut Based Interpolation Method (IMUL)

The method is specially developed for multidimensional environments by combining characteristics of the two previous methods MACI and Conservation of Relative Fuzziness (CRF) based interpolation [115]. In particular, the vector representation and transformation features of MACI and the relative fuzziness features of CRF method are integrated in this method [177].

IMUL Algorithm

The core of the conclusion is determined by the vector containing characteristics points as shown in Figure 2.17 and the transformation features of MACI, while its flanks are illustrated in Figure 2.19 by computing the fuzziness of the observation and the relative fuzziness of adjacent sets to the observation. The IMUL algorithm is implemented in the following steps, which is outlined in the pseudo code of Figure 2.20:

1. Calculate the Reference Point (RP) of emerging conclusion using the Euclidean distance metric.

2. Determine the Left and Right Cores of the conclusion using its Reference Point

MACIInterpolation(X,Y,α,$\underline{a}$,$\underline{b}$,a_k,b_k)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
$\underline{a}$, the vector representation for antecedent fuzzy set;
$\underline{b}$, the vector representation for consequent fuzzy set;
a_k, the characteristic point for antecedent fuzzy set;
b_k, the characteristic point for consequent fuzzy set;

(1) $A_1, A_2 \in X, B_1, B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $\underline{a} \leftarrow [a_{-m}, .., a_0, .., a_n]$
(5) $\underline{a}_L \leftarrow [a_{-m}, .., a_0]$
(6) $\underline{a}_U \leftarrow [a_0, .., a_n]$
(7) $If\, b_i^* \geq b_j^*, \forall i < j \in [-m, n]$
(8) $\lambda_k \leftarrow a_k^* - a_{1k}/a_{2k} - a_{1k}$
(9) $KHb_k^* \leftarrow (1-\lambda_k)b_{1k} + \lambda_k b_{2k}$
(10) $b_{k(RF)}^* \leftarrow KHb_k^* + \Sigma_{i=0}^{k-1}(\lambda_i - \lambda_{i+1})(b_{2i} - b_{1i})$
(11) $b_{k(LF)}^* \leftarrow KHb_k^* + \Sigma_{k+1}^{i=0}(\lambda_i - \lambda_{i-1})(b_{2i} - b_{1i})$
(12) $B^* \leftarrow b_{k(RF)}^* \cup b_{k(LF)}^*$

Figure 2.18: The MACIInterpolation Pseudocode.

(RP) as shown in Figure 2.21.

3. Calculate the two flanks by using the relative fuzziness concepts and then determine the conclusion.

2.5.6 Slopes of Flaking Edges Based Interpolation Method (SFE)

This method extends the KH method by introducing and manipulating the flanking edges of the fuzzy sets involved. However, it remains to be applicable to triangle-shaped CNF fuzzy sets only [81] as shown in Figure 2.22. The fundamental assumption behind this method is that the slope of the conclusion can be estimated with the linear combination of the respective (left or right) slopes of the consequents of the

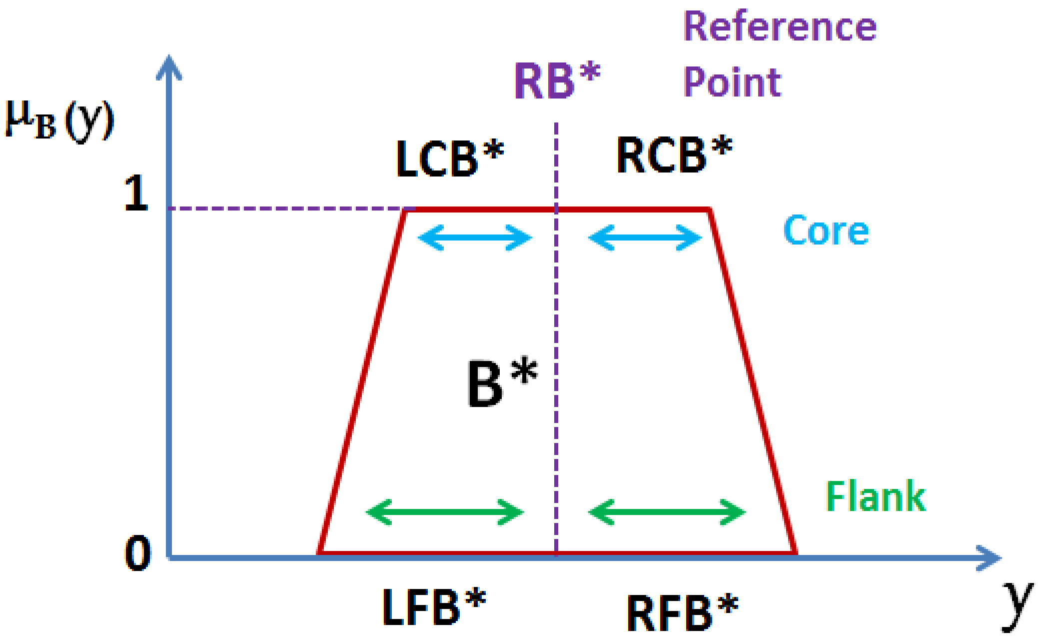


Figure 2.19: Measurement of Reference Point, Two Cores and Two Flanks of conclusion in IMUL method

neighbouring rules, in a similar manner to which the slopes of the observation can be estimated with the linear combination of the respective (left or right) slopes of the antecedents of the neighbouring rules [81].

SFE Algorithm

The SFE algorithm is implemented in the following steps, which is outlined in the pseudo code of Figure 2.23:
1. Employ the KH interpolation method for determining the two endpoints of the support of fuzzy set.
2. Calculate the highest point of the triangle using the slopes of the flanking edges as shown in Figure 2.24.
3. Estimate the conclusion using the highest point, infimum point and supremum point.

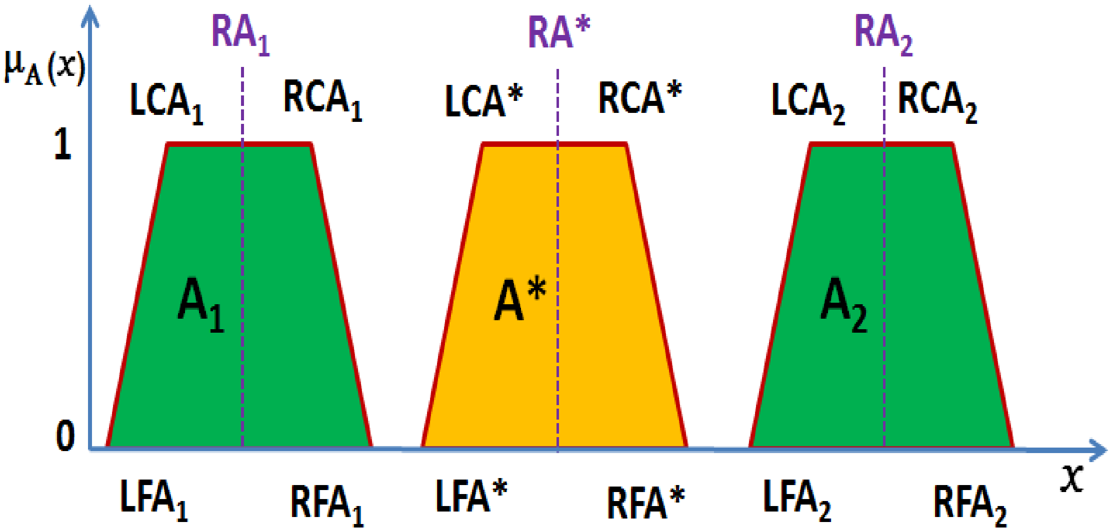
2.5.7 Similarity Transfer Based Interpolation Method (ST)

This technique initially starts from an improvement on the KH method, by introducing the general convex condition for the underlying method based on the concept of the Midpoint of Core as shown in Figure 2.26 and Similarity Transfer functions [205].

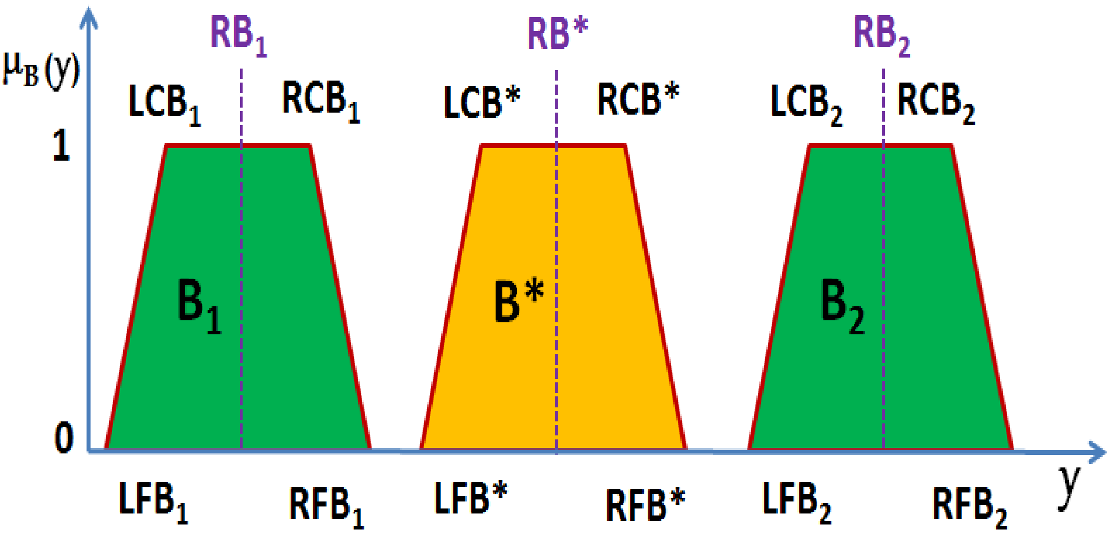
IMULInterpolation(X,Y,α,k)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
R, the reference point;
LC, the left core of fuzzy set;
RC, the right core of fuzzy set;
LF, the left flank of fuzzy set;
RF, the right flank of fuzzy set;
k, the input dimension;
s, the fuzziness of antecedents and consequents;
r, the fuzziness of observation and conclusion;
d, the distance between fuzzy sets.

(1) $A_1,A_2 \in X, B_1, B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $\lambda_{core} \leftarrow \sqrt{\Sigma_{i=1}^{k}(RA_i^* - RA_{i1})^2} / \sqrt{\Sigma_{i=1}^{k}(RA_{i2} - RA_{i1})^2}$
(5) $RB^* \leftarrow (1-\lambda_{core})RB_1 + \lambda_{core}RB_2$
(6) $\lambda_{left} \leftarrow \sqrt{\Sigma_{i=1}^{k}(LCA_i^* - LCA_{i1})^2} / \sqrt{\Sigma_{i=1}^{k}(LCA_{i2} - LCA_{i1})^2}$
(7) $LCB^* \leftarrow (1-\lambda_{left})LCB_1 + \lambda_{left}LCB_2 + (\lambda_{core} - \lambda_{left})(RB_2 - RB_1)$
(8) $\lambda_{right} \leftarrow \sqrt{\Sigma_{i=1}^{k}(RCA_i^* - RCA_{i1})^2} / \sqrt{\Sigma_{i=1}^{k}(RCA_{i2} - RCA_{i1})^2}$
(9) $RCB^* \leftarrow (1-\lambda_{right})RCB_1 + \lambda_{right}RCB_2 + (\lambda_{core} - \lambda_{right})(RB_2 - RB_1)$
(10) $s_i \leftarrow RFA_{i1} - RCA_{i1}$; $s^{'} \leftarrow RFB_1 - RCB_1$
(11) $s \leftarrow \sqrt{\Sigma_{i=1}^{k}(s_i)^2}$
(12) $r_i \leftarrow LCA_i^* - LFA_i^*$; $r^{'} \leftarrow LCB^* - LFB^*$
(13) $r \leftarrow \sqrt{\Sigma_{i=1}^{k}(r_i)^2}$
(14) $d_i \leftarrow RA_i^* - RA_{i1}$; $d^{'} \leftarrow RB^* - RB_1$
(15) $d \leftarrow \sqrt{\Sigma_{i=1}^{k}(d_i)^2}$
(16) $RFB^* \leftarrow RCB^* + r\left(1 + \mid \frac{s^{'}}{d^{'}} - \frac{s}{d} \mid\right)$
(17) $B^* \leftarrow LFB^* \cup RFB^*$

Figure 2.20: The IMULInterpolation Pseudocode.



(a) Representation of various parameters- RP, Cores and Flanks for antecedents in IMUL method



(b) Representation of various parameters- RP, Cores and Flanks for consequents in IMUL method

Figure 2.21: IMUL fuzzy interpolation method

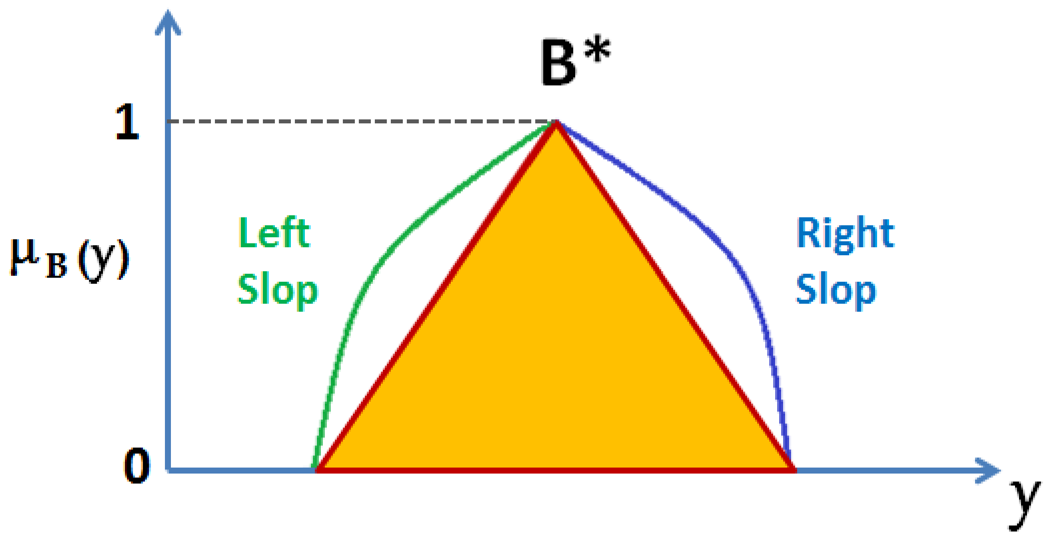


Figure 2.22: Concept of Slopes of Flanking Edges in SFE method

In this method a new rule is first constructed from two given rules in the sparse rule base which are nearest to the observation antecedent fuzzy set. Then on the basis of similarities of fuzzy sets in the antecedent and consequent parts, interpolative reasoning is performed with the new rule using the Similarity Transfer concept.

ST Algorithm

The algorithm is based on the concept of Core of fuzzy set, which is conventionally named the support of the set, i.e. the α-cut set with $\alpha = 1$. The midpoint of the core is denoted by $mcor\ \{A\}$ and $mcor\ \{B\}$ as shown in Figure 2.26. The method assumes the observation of the antecedent defined on the universe of discourse X to be a normal and convex fuzzy set A^* , such that $mcor\ \{A_1\} \leq mcor\ \{A^*\} \leq mcor\ \{A_2\}$.

The ST algorithm is implemented in the following steps, which is outlined in the pseudo code of Figure 2.25:

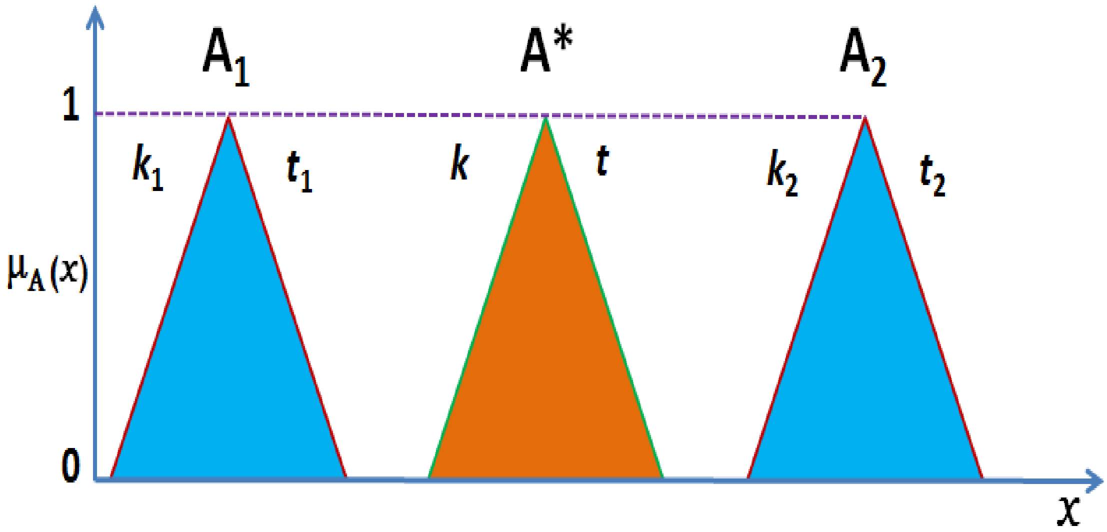
1. New Rule Construction- Constructing a new fuzzy set A^s which has the same midpoint of core to the observation antecedent fuzzy set A^*. So the new fuzzy set A^s will also be a convex and normal fuzzy set. Similarly the consequent fuzzy set B^s can be obtained which is also a convex and normal fuzzy set. Thus the newly created rule $A^s \Rightarrow B^s$ will only involve normal and convex fuzzy sets.

2. Interpolative Reasoning- Perform interpolation on this new rule, giving the same approximated conclusion as the original rule. In the simplest case, if $A^* = A_i$, $i = 1, 2,$

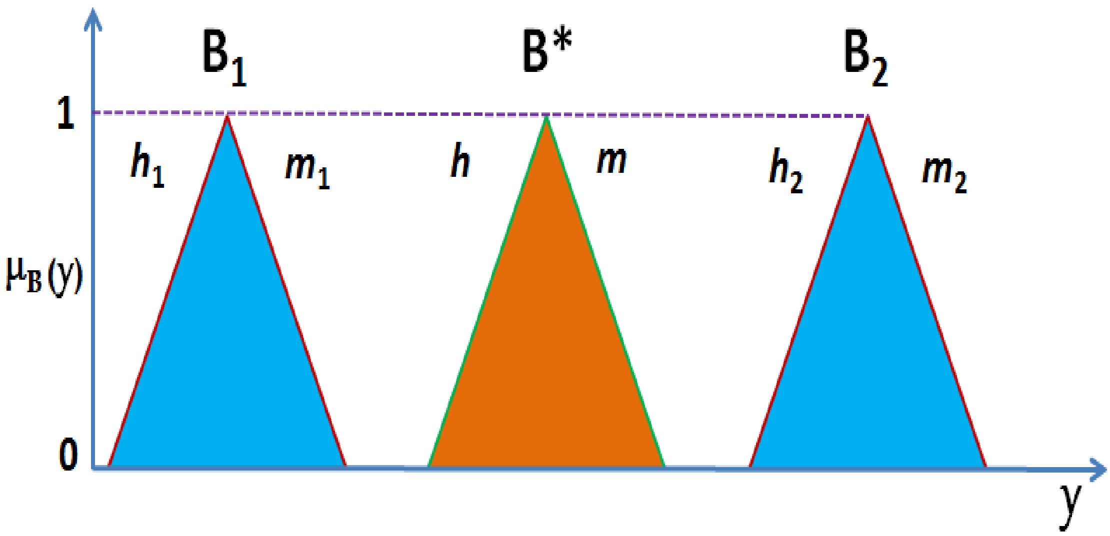
SFEInterpolation(X,Y,α,hst)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
hst, the highest point of normal fuzzy set;
k, t, the left and right slop of observation fuzzy set;
h, m, the left and right slop of conclusion fuzzy set.

(1) $A_1,A_2 \in X, B_1, B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $hstA = \{x|\mu_A(x) = 1, x \in X\}$
(5) $inf \left\{B^*_\alpha\right\} \leftarrow KHB^*$
(6) $sup \left\{B^*_\alpha\right\} \leftarrow KHB^*$
(7) $k \leftarrow k_1 x + k_2 y$
(8) $h \leftarrow |h_1 x + h_2 y| c$
(9) $h \leftarrow kc$
(10) $t \leftarrow t_1 x + t_2 y$
(11) $m \leftarrow -|m_1 x + m_2 y| c$
(12) $m \leftarrow tc$
(13) $\left[(1-\alpha)/hst \left\{B^*_\alpha\right\} - inf \left\{B^*_\alpha\right\}\right] / \left[(\alpha-1)/sup \left\{B^*_\alpha\right\} - hst \left\{B^*_\alpha\right\}\right] \leftarrow h/m$
(14) $hst \left\{B^*_\alpha\right\} \leftarrow m\left(sup \left\{B\right\}^*_\alpha\right) - h\left(inf \left\{B\right\}^*_\alpha\right) /(m-h)$
(15) $B^* \leftarrow inf \left\{B^*_\alpha\right\} \cup hst \left\{B^*_\alpha\right\} \cup sup \left\{B^*_\alpha\right\}$

Figure 2.23: The SFEInterpolation Pseudocode.



(a) Slope Parameters for antecedents in slopes of flanking edges method



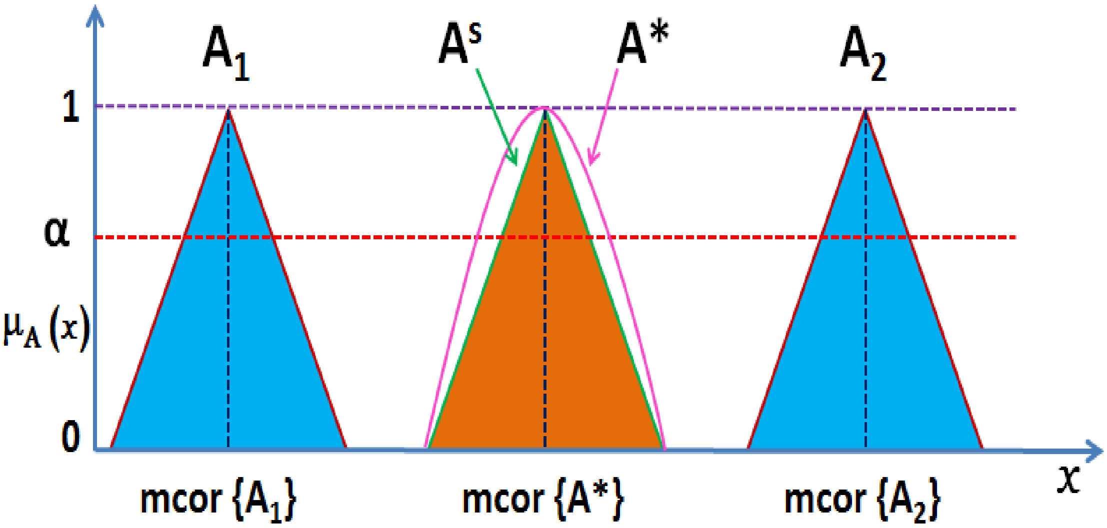
(b) Slope Parameters for consequents in slopes of flanking edges method

Figure 2.24: SFE fuzzy interpolation method

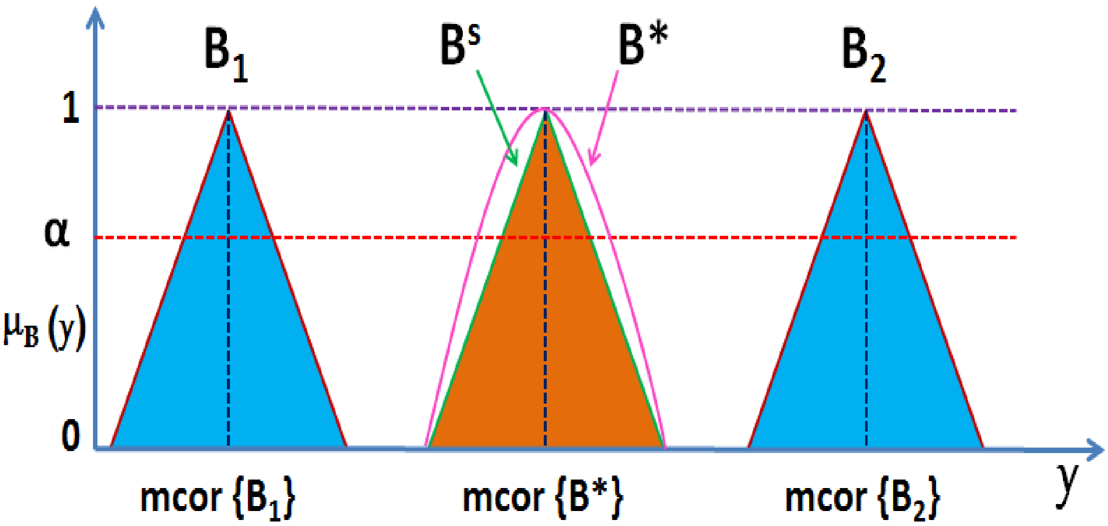
STInterpolation(X,Y,α,$mcor$)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
$mcor$, the mid point of core of fuzzy set;
A^s, the average antecedent fuzzy set;
B^s, the average consequent fuzzy set.
S_L, the lower similarity;
S_U, the upper similarity.

(1) $A_1, A_2 \in X, B_1, B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $mcor\{A^*\} \equiv mcor\{A^s\}$
(5) $\beta \leftarrow d(mcor\{A^*\}, mcor\{A_1\}) / d(mcor\{A_1\}, mcor\{A_2\})$
(6) $A^s \leftarrow \beta A_2 + (1-\beta)A_1, \beta and (1-\beta) \in [0,1]$
(7) $B^s \leftarrow \beta B_2 + (1-\beta) B_1$
(8) $If A^* = A_1 \wedge \beta = 0$
(9) $B^* = B_1$
(10) $If A^* = A_2 \wedge \beta = 1$
(11) $B^* = B_2$
(12) $If A^* > A_1 \wedge A^* < A_2$
(13) $S_L(A^*, A^s)(\alpha) \leftarrow d\left(inf\left\{A^*_\alpha\right\}, mcor\{A^*\}\right) / d\left(inf\left\{A^s_\alpha\right\}, mcor\{A^s\}\right)$
(14) $S_U(A^*, A^s)(\alpha) \leftarrow d\left(sup\left\{A^*_\alpha\right\}, mcor\{A^*\}\right) / d\left(sup\left\{A^s_\alpha\right\}, mcor\{A^s\}\right)$
(15) $S_L(A^*, A^s)(\alpha) \equiv S_L(B^*, B^s)(\alpha)$
(16) $S_U(A^*, A^s)(\alpha) \equiv S_U(B^*, B^s)(\alpha)$
(17) $mcor\{B^*\} \equiv mcor\{B^s\}$
(18) $inf\left\{B^*_\alpha\right\} \leftarrow S_L(A^*, A^s)(\alpha)\, d\left(inf\left\{B^s_\alpha\right\}, mcor\{B^s\}\right) +$
$mcor\{B^s\}$
(19) $sup\left\{B^*_\alpha\right\} \leftarrow S_U(A^*, A^s)(\alpha)\, d\left(sup\left\{B^s_\alpha\right\}, mcor\{B^s\}\right) +$
$mcor\{B^s\}$

Figure 2.25: The STInterpolation Pseudocode.



(a) Representation of Mid Point of Core for antecedents including observation in ST method



(b) Representation of Mid Point of Core for consequents including conclusion in ST method

Figure 2.26: ST fuzzy interpolation method

then $B^* = B_i$. However, when A^* lies between A_1 and A_2 then the similarity between A^* and A^s is maintained. It is assumed that the consequent part B^* and B^s may be kept the same such that B^* retains normality and convexity. In this case, to obtain the conclusion B^* , transform B^s to B^* using the lower and upper similarity function as defined.

2.5.8 Spatial Geometric Representation Based Interpolation Method (SGR)

This method is based on the generation of an appropriate geometric shape of the rule antecedents and consequents in order to obtain an approximated conclusion, by exploiting the semantical and interrelational features of fuzzy sets [17, 18]. Initially this method was developed for one-dimensional real intervals but it has since been modified to deal with multidimensional problem. This is one of the earliest methods which applied the geometric properties of shapes and tried to improve the interpolation solutions.

SGR Algorithm

The algorithm is based on the concept of representative points (centre points or most typical points) as shown in Figure 2.29. The representative point of the conclusion is determined by the ratio of the centres of the observation and the antecedents. Then all fuzzy sets involved are rotated by 90^0 at their centres and connected with regards to the corresponding points of antecedents and consequents. In this way the two solid geometric bodies are formed for the input and output universe. These solid bodies are cut at the centre of observation and at the determined centre of conclusion respectively, to produce the set $A^{*'}$ in the input space as shown in Figure 2.28 and the set $B^{*'}$ in the output space. Finally a transformation is performed to determine the conclusion B^* based on the similarity of the observation A^* and the interpolated observation $A^{*'}$.

The SGR algorithm is implemented in the following steps, which is outlined in the pseudo code of Figure 2.27:
1. Determine the representative (centre) points of involved fuzzy sets.
2. Obtain the solid geometric body by rotating the antecedent fuzzy sets by 90^0 at their representative points.
3. Cut this geometric body at the position of the representative value of the given

SGRInterpolation(X,Y,α,a,b)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
a, the representative point for antecedent of fuzzy set;
b, the representative point for consequent of fuzzy set;
$Rotate()$, the operation for rotating fuzzy set with specific degree and point;
$LTransf()$, the operation for linearly transforming the fuzzy set around specific point;
$GShape()$, the operation for ontaining geometric shape of fuzzy set around specific point.

(1) $A_1,A_2 \in X, B_1, B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $A_1 \leftarrow Rotate(A_1, 90, a_1)$
(4) $A_2 \leftarrow Rotate(A_2, 90, a_2)$
(5) $A^{*''} \leftarrow GShape(A, a^*)$
(6) $A^{*''} \leftarrow Rotate(A^{*''}, 90, a^*)$
(7) $A^{*'} \leftarrow LTransf(A^{*''}, 90, a^*)$
(8) $b^* \leftarrow a_1 + (b_2 - b_1)(a^* - a_1)/(a_2 - a_1)$
(9) $B_1 \leftarrow Rotate(B_1, 90, b_1)$
(10) $B_2 \leftarrow Rotate(B_2, 90, b_2)$
(11) $B^{*''} \leftarrow GShape(B, a^*)$
(12) $B^{*''} \leftarrow Rotate(B^{*''}, 90, b^*)$
(13) $B^{*'} \leftarrow LTransf(B^{*''}, 90, b^*)$

Figure 2.27: The SGRInterpolation Pseudocode.

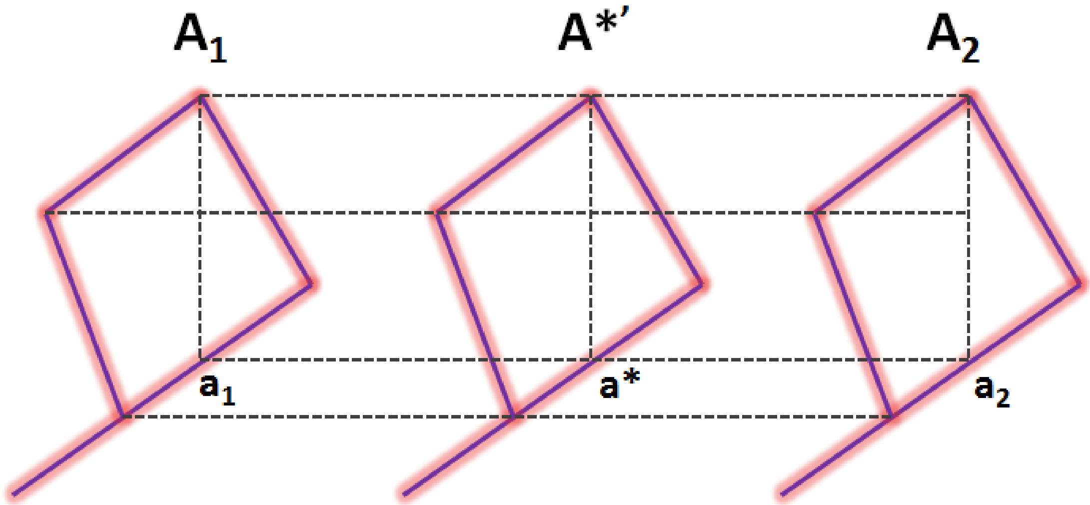


Figure 2.28: Geometric Shape of antecedents in SGR method

new observation vertically then rotate and transform resulting intersection.
4. Calculate the representative point for conclusion.
5. Obtain the conclusion by performing Steps (2) and (3) for consequent fuzzy sets.

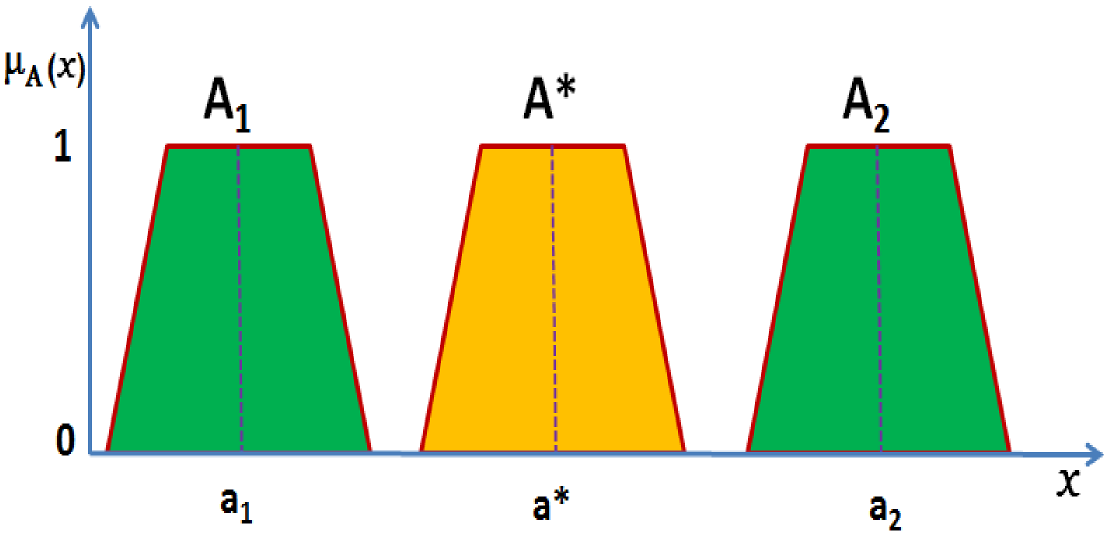
2.5.9 Graduality Based Interpolation Method (IRG)

This method is based on the concepts of the Gradual Behaviour and Analogical Approach [27, 28]. The basic idea is that if an observation lies between two given extreme cases corresponding to the antecedents of two adjacent rules, then the conclusion will lie between the consequents of those two rules. Moreover, the conclusion will be comparable to the consequents in a way analogous to the comparison which can be established between the observation and the antecedents, and this comparison takes into account both extreme cases.

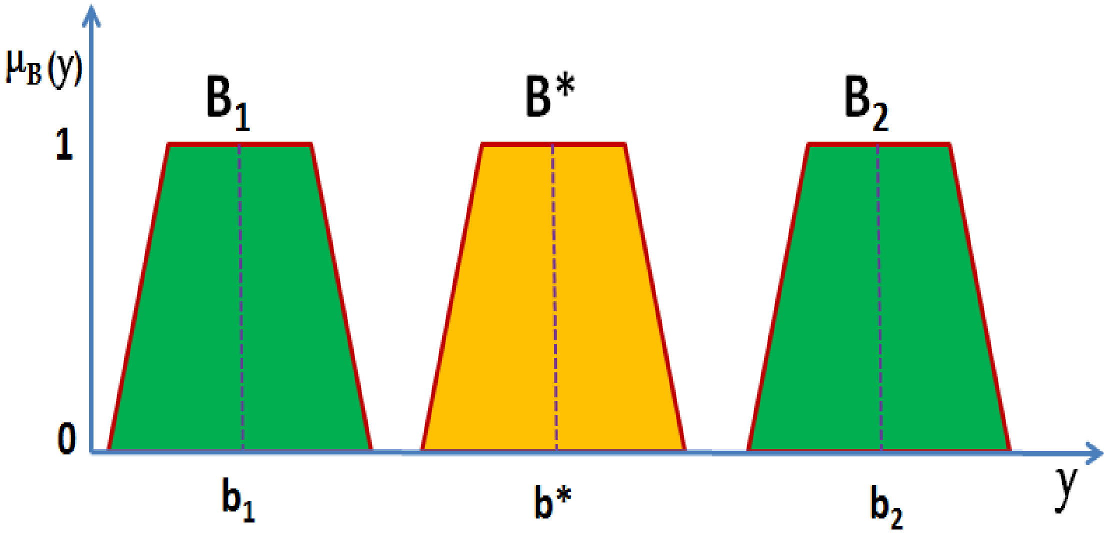
IRG Algorithm

The concepts of Graduality and Analogy in fuzzy sets [28] are defined as follows: Suppose that the input universe X and the output universe Y are defined on the universe R of real numbers and F is a fuzzy set of R.

- Gradual Behaviour - If the rule antecedent variables A_1,A_2 have the gradual relationship ($A_1 \prec A_2$) then rule consequent variables B_1,B_2 have also the gradual behaviour ($B_1 \prec B_2$) with regards to rule antecedent variables.



(a) Representative Points of antecedents including observation in SGR method



(b) Representative Points of consequents including conclusion in SGR method

Figure 2.29: SGR fuzzy interpolation method

IRGInterpolation(X,Y,α,l,T_{sh},m_ρ)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
$l(F)$, the location of fuzzy set F;
$D_{lo}(F,F^{'})$, the l-distinguishability of two fuzzy sets F and $F^{'}$;
$C_{lo}(F,F^{'})$, the relative l-distinguishability of two fuzzy sets F and $F^{'}$;
T_{sh}, the transformation function;
m_ρ, the translated parameter;

(1) $A_1,A_2 \in X, B_1, B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $D_{lo}(F,F^{'}) \leftarrow \left| l(F) - l(F^{'}) \right|$
(5) $F^{'} \leftarrow T_{sh}(F)$
(6) $C_{lo}(A^*,A_1;A_1,A_2) \leftarrow D_{lo}(A^*,A_1)/D_{lo}(A_1,A_2)$
(8) $C_{lo}(B^*,B_1;B_1,B_2) \leftarrow D_{lo}(B^*,B_1)/D_{lo}(B_1,B_2)$
(9) $C_{lo}(A^*,A_1;A_1,A_2) \equiv C_{lo}(B^*,B_1;B_1,B_2)$
(10) $A_1^{'} \leftarrow m_{\rho 1}(A_1), \rho 1 \leftarrow l(A^*) - l(A_1)$
(11) $A_2^{'} \leftarrow m_{\rho 2}(A_2), \rho 1 \leftarrow l(A^*) - l(A_2)$
(12) $A^{*'} \leftarrow T^1_{shx}(A_1^{'})$
(13) $A^{*"} \leftarrow T^2_{shx}(A_2^{'})$
(14) $B_1^{'} \leftarrow m_{\rho 3}(A_1), \rho 3 \leftarrow l(B^*) - l(B_1)$
(15) $B_2^{'} \leftarrow m_{\rho 4}(A_2), \rho 4 \leftarrow l(B^*) - l(B_2)$
(16) $B^{*'} \leftarrow T^1_{shy}(B_1^{'})$
(17) $B^{*"} \leftarrow T^2_{shy}(B_2^{'})$
(18) $B^* \leftarrow B^{*'} \cup B^{*"}$

Figure 2.30: The IRGInterpolation Pseudocode.

- Analogical Approach - Analogical approach tells when any A^* is such that $A_1 \prec A^* \prec A_2$ so study the relationship between A^* and pair (A_1,A_2) then find out the B^* such that an analogous relationship exists between B^* and pair $(B_1 \prec B_2)$.

If F and $F^{'}$ are two fuzzy sets then they can be compared on two components- their locations and their shapes. The location $l(F)$ of any fuzzy set F is a representative of the place of the fuzzy set in R as shown in Figure 2.31. The l-distinguishability $D_{lo}(F, F')$ for the two fuzzy sets F and F' evaluates to which extent the locations of F and F' differ. The shape of a fuzzy set is intrinsically defined from the relative length of its kernel and its support but the shape of F on R is more difficult to define for any form of membership function. The shape-distinguishability of F is represented by $F'(T_{sh}(F))$.

The IRG algorithm is implemented in the following steps, which is outlined in the pseudo code of Figure 2.30:
1. Determine the location $l(B^*)$ of the conclusion fuzzy set B^*.
2. Translate A_1 and A_2 towards A^* to obtain respectively $A_1^{'}$ and $A_2^{'}$.
3. Compare the shapes of $A_1^{'}$ and $A_2^{'}$ with the shape of A^*.
4. Translate B_1 and B_2 to the location $l(B^*)$, to obtain $B_1^{'}$ and $B_2^{'}$ respectively.
5. Construct $B^{*'}$ and $B^{*"}$ with location $l(B^*)$, such that the shape of $B^{*'}$ (resp. $B^{*"}$) can be compared to the shape of $B_1^{'}$ (resp. $B_2^{'}$) in the same way as the shape of A^* can be compared with the shape of $A_1^{'}$ (resp. $A_2^{'}$).
6. Aggregate $B^{*'}$ and $B^{*"}$ to construct B^*.

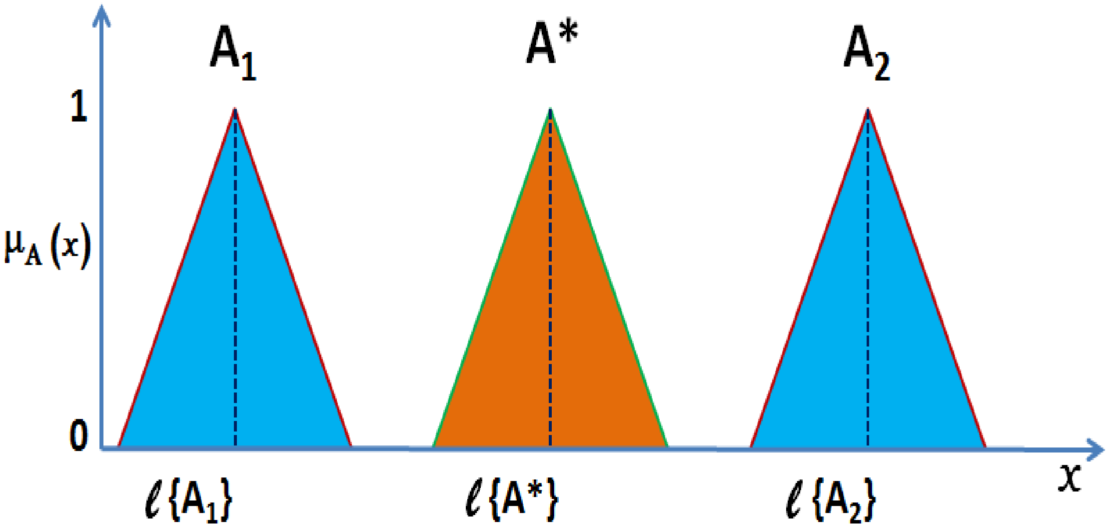
2.5.10 Scale and Move Transformation Based Interpolation Method (TFRI)

The method is based on the concepts of Representative Value (RV), Scale Transformation, Move Transformation, Integrated Transformation and Analogical Approach. In this method a new rule is constructed where the antecedent is closest (or with shortest distance) to and has the same RV as the observed antecedent fuzzy set, and then on the basis of similarities of fuzzy sets in the antecedent and consequent parts, interpolative reasoning is performed using the new rule via the so-called Scale, Move and Integrated Transformations [85, 86].

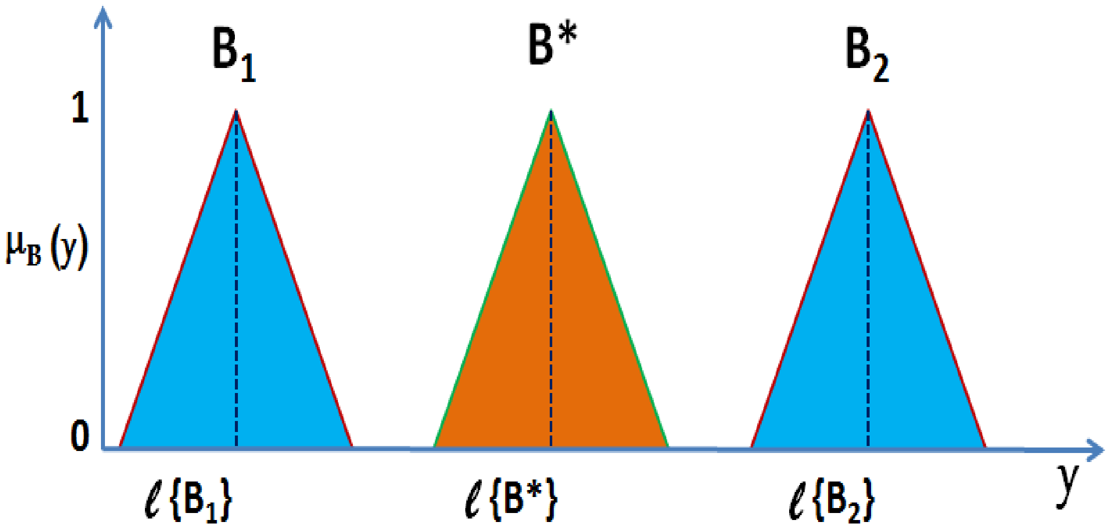
TFRI Algorithm

The algorithm is based on the concept of a representative value which is defined as the average of the x coordinates as shown in Figure 2.33. This representative value captures the centre of gravity which reflects both the location and the shape of a fuzzy set definition.

The TFRI algorithm is implemented in the following steps, which is outlined in the pseudo code of Figure 2.32:



(a) Concept of Location-Indistinguishability of antecedent fuzzy sets in IRG method



(b) Concept of Location-Indistinguishability of consequent fuzzy sets in IRG method

Figure 2.31: IRG fuzzy interpolation method

TFRIInterpolation(X,Y,α,s,l,m,T)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
Rep, the representative value;
s, the scale rate;
l, the move distance;
m, the move rate;
T, the integrated transformation.

(1) $A_1,A_2 \in X, B_1,B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $Rep(A) \leftarrow (a_0+a_1+a_2)/3, (fortriangularfuzzyset)$
(5) $\lambda_{Rep} \leftarrow d(A_1,A^*)/d(A_1,A_2)$
(6) $\lambda_{Rep} \leftarrow d(Rep(A_1),Rep(A^*))/d(Rep(A_1),Rep(A_2))$
(7) $a_0^{'} \leftarrow (1-\lambda_{Rep})a_{10} + \lambda_{Rep}a_{20}$
(8) $a_1^{'} \leftarrow (1-\lambda_{Rep})a_{11} + \lambda_{Rep}a_{21}$
(9) $a_2^{'} \leftarrow (1-\lambda_{Rep})a_{12} + \lambda_{Rep}a_{22}$
(10) $A^{'} \leftarrow (1-\lambda_{Rep})A_1 + \lambda_{Rep}A_2$
(11) $Rep(A^{'}) \equiv Rep(A^*)$
(12) $B^{'} \leftarrow (1-\lambda_{Rep})B_1 + \lambda_{Rep}B_2$
(13) $IfA^* \equiv A_1$
(14) $B^* \leftarrow B_1$
(15) $IfA^* \equiv A_2$
(16) $B^* \leftarrow B_2$
(17) $IfA^* > A_1ANDA^* < A_2$
(18) $a_2 - a_0 \leftarrow (s \times (a_2 - a_0))$
(19) $a_0^{'} \leftarrow a_0(1+2s) + a_1(1-s) + a_2(1-s)$
(20) $a_1^{'} \leftarrow a_0(1-s) + a_1(1+2s) + a_2(1-s)$
(21) $a_2^{'} \leftarrow a_0(1-s) + a_1(1-s) + a_2(1+2s)$
(22) $s \leftarrow (a_2^{'} - a_0^{'})/(a_2 - a_0)$
(23) $a_0 \leftarrow a_0 + l$
(24) $a_0^{'} \leftarrow a_0 + l$
(25) $a_1^{'} \leftarrow a_1 - 2l$
(26) $a_2^{'} \leftarrow a_2 + 2l$
(27) $m \leftarrow l/(a_1 - a_0)$
(28) $B^* \leftarrow T(B^{'})$

Figure 2.32: The TFRIInterpolation Pseudocode.

1. New Rule Construction: Constructing a new fuzzy set $A^{'}$ which is closest (or with shortest distance) to and has the same RV as to the observation $A^{'}$ regarding the antecedent fuzzy set A^{*}. So the new fuzzy set will also be a convex and normal fuzzy set. Similarly the corresponding consequent fuzzy set $B^{'}$ can be obtained which is also a convex and normal fuzzy set. Thus, the newly created rule $A^{'} \Rightarrow B^{'}$ will only involve normal and convex fuzzy sets.

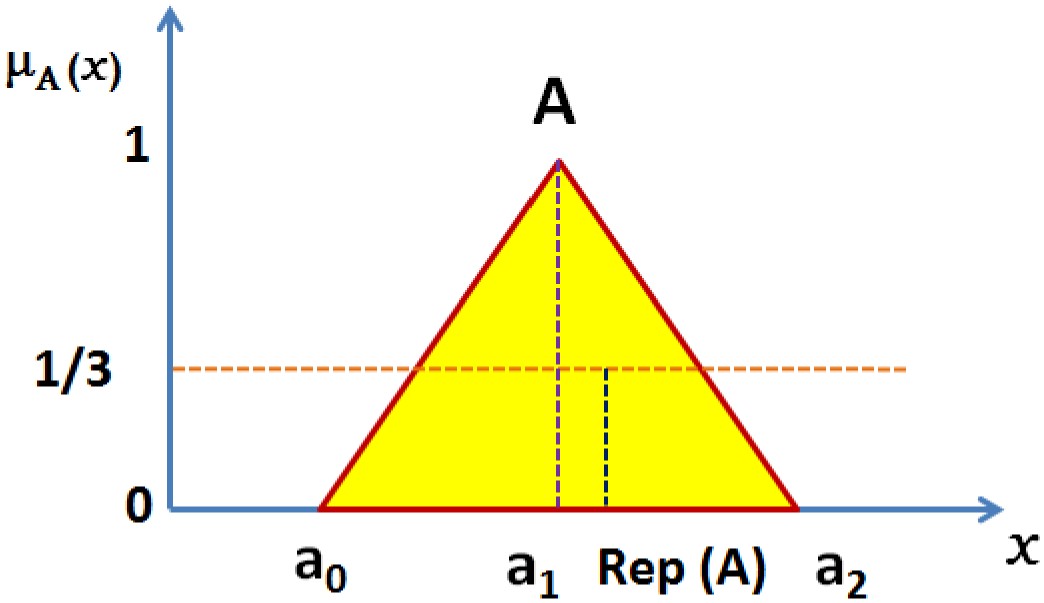
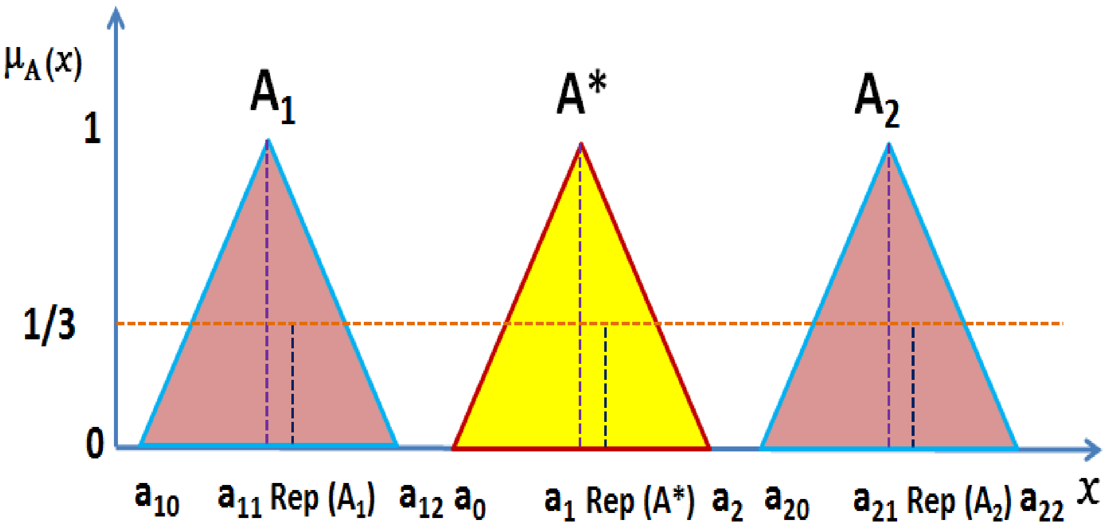


Figure 2.33: Concept of Representative Value (RV) in TFRI method

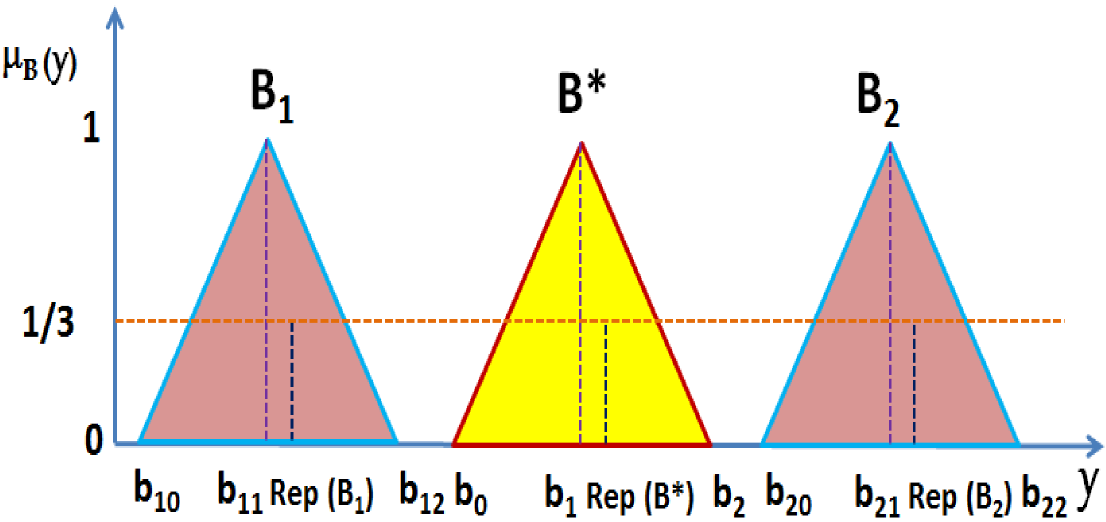
2. Interpolative Reasoning: Now the interpolative reasoning is performed on this new rule. This will result in the same approximated conclusion as the original rule [85, 86]. In the simplest case, if $A^{*} = A_i$, $i = 1,2$, then $B^{*} = B_i$. However, when A^{*} lies between A_1 and A_2 then the similarity between A^{*} and $A^{'}$ is considered such that the consequent part B^{*} and $B^{'}$ may retain the same similarity and that B^{*} is also of normality and convexity. For this to the conclusion B^{*} , the Scale and Move transformations are used to obtain appropriate operators which will allow transformation of $B^{'}$ to B^{*} as mentioned in the pseudo code and in Figure 2.35 .

2.5.11 Cutting and Transformation Based Interpolation Method (IRCT)

This method is based on a number of concepts, including Representative Value as those employed in the TFRI method [85, 86], Collection of Highest Point (Maximum

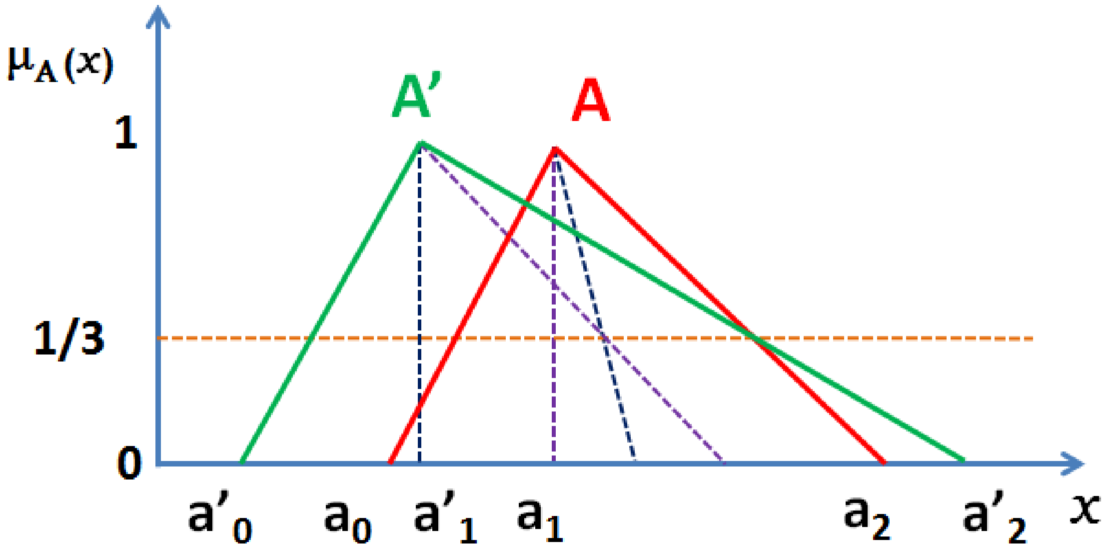


(a) Representation Values for antecedents including observation in TFRI method

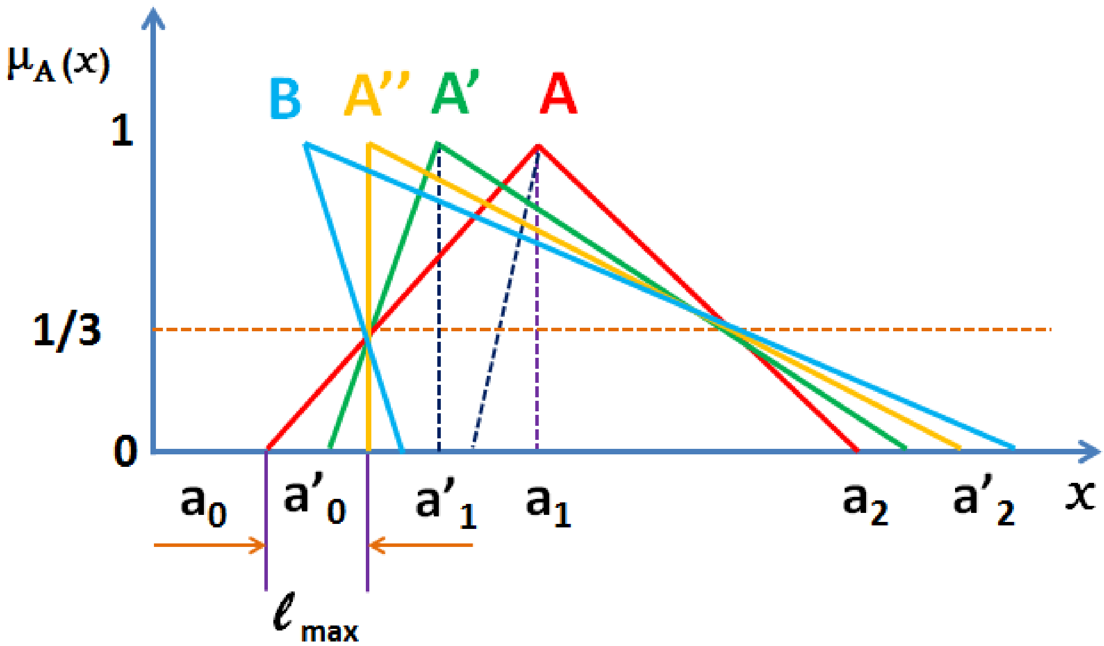


(b) Representation Values for consequents including conclusion in TFRI method

Figure 2.34: TFRI fuzzy interpolation method



(a) Scale Transformation in TFRI method



(b) Move Transformation in TFRI method

Figure 2.35: Scale and Move transformations

IRCTInterpolation(X,Y,α,hst)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
Rep, the representative value;
hst, the highest point of normal fuzzy set;
$ihst$, the highest point of normal fuzzy set;
$shst$, the highest point of normal fuzzy set;
λ, the distance ratio;
l, the length between the two points on fuzzy set;
L, the increment length;
γ, the ratio rate;
$A^{'}$, the average observation fuzzy set;
$B^{'}$, the average conclusion fuzzy set.

Figure 2.36: The IRCTInterpolation Pseudocode-1.

and Minimum Elements), Increment Transformation, Ratio Transformation, and Analogical Approach. In this method a new rule is constructed which is closest (or with shortest distance) to and has the same RV as the observed antecedent fuzzy set. Then on the basis of the similarities of fuzzy sets in the antecedent and consequent parts, interpolative reasoning is performed with the new constructed rule, using Increment and Ratio Transformations.

IRCT Algorithm

The method relies upon the assumption of the so-called highest points as shown in Figure 2.38. If A is a normal fuzzy set in the universe of discourse X, then the collection of the highest points of a fuzzy set A, is denoted as $hstA$. Suppose that the minimum and maximum elements of $hstA$ are represented by $ihstA$ and $shstA$ as shown in Figure 2.39. The IRCT algorithm is implemented in the following steps, which is outlined in the pseudo code of Figures 2.36 and 2.37:

(1) $A_1, A_2 \in X, B_1, B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $hst\{A\} = \{x|\mu_A(x) = 1, x \in X\}$
(5) $ihst\{A\} \leftarrow inf\{hst\{A\}\}$
(6) $shst\{A\} \leftarrow sup\{hst\{A\}\}$
(7) $Rep(A) \leftarrow (a_0 + a_1 + a_2 + a_3)/4,\ (for\ trapezoidalfuzzy\ set)$
(8) $\lambda_{a1} \leftarrow d(A_1, a_1)/d(A_1, A_2)$
(9) $\lambda_{a1} \leftarrow d(Rep(A_1), a_1)/d(Rep(A_1), Rep(A_2))$
(10) $\lambda_{a2} \leftarrow d(A_2, a_2)/d(A_1, A_2)$
(11) $\lambda_{a2} \leftarrow d(Rep(A_2), a_2)/d(Rep(A_1), Rep(A_2))$
(12) $b_1 = isht\{B^*\} \leftarrow (1 - \lambda_{a1}) \times Rep(B_1) + \lambda_{a1} \times Rep(B_2)$
(13) $b_2 = shst\{B^*\} \leftarrow (1 - \lambda_{a2}) \times Rep(B_1) + \lambda_{a2} \times Rep(B_2)$
(14) $\lambda_{Rep} \leftarrow d(A_1, A^*)/d(A_1, A_2)$
(15) $\lambda_{Rep} \leftarrow d(Rep(A_1), Rep(A^*)/d(Rep(A_1), Rep(A_2))$
(16) $l_a^{'} \leftarrow (1 - \lambda_{Rep}) \times l_{a1} + \lambda_{Rep} \times l_{a2}$
(17) $a_0^{'} \leftarrow a_1 - l_a^{'}$
(18) $If\ l_a > l_a^{'}$
(19) $L \leftarrow l_a - l_a^{'}$
(20) $l_b^{'} \leftarrow (1 - \lambda_{Rep}) \times l_{b1} + \lambda_{Rep} \times l_{b2}$
(21) $b_0^{'} \leftarrow b_1 - l_b^{'}$
(22) $l_b \leftarrow L + l_{b1}^{'}$
(23) $inf\{B^*\} = b_0 \leftarrow b_1 - l_b$
(24) $If\ l_a < l_a^{'}$
(25) $\gamma \leftarrow l_a/l_a^{'}$
(26) $l_b^{'} \leftarrow (1 - \lambda_{Rep}) \times l_{b1} + \lambda_{Rep} \times l_{b2}$
(27) $b_0^{'} \leftarrow b_1 - l_b^{'}$
(28) $l_b \leftarrow \gamma \times l_b^{'}$
(29) $inf\{B^*\} = b_0 \leftarrow b_1 - l_b$

Figure 2.37: The IRCTInterpolation Pseudocode-2.

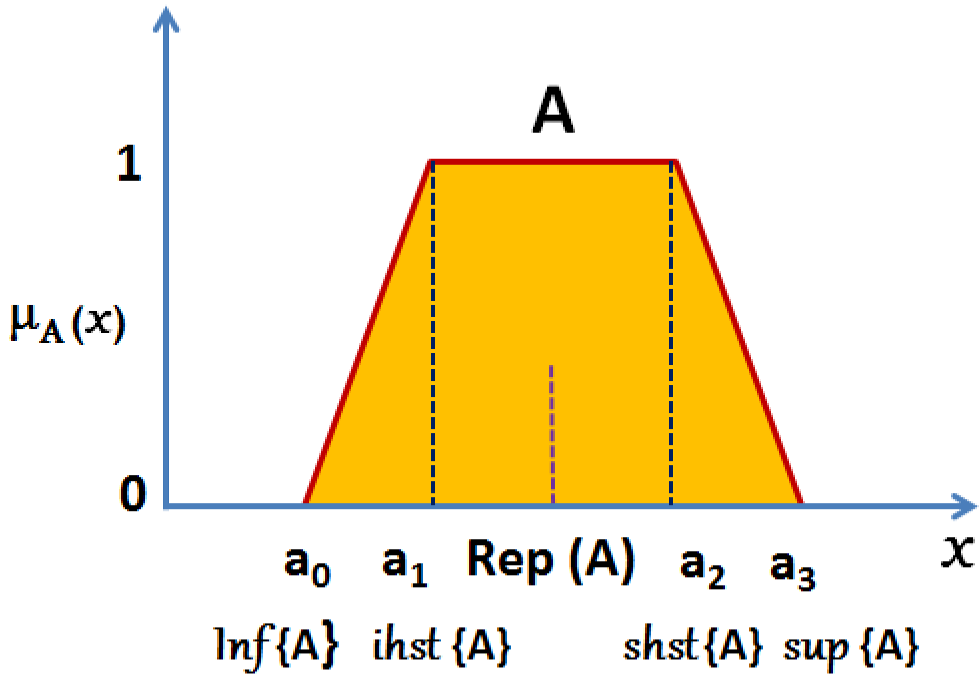


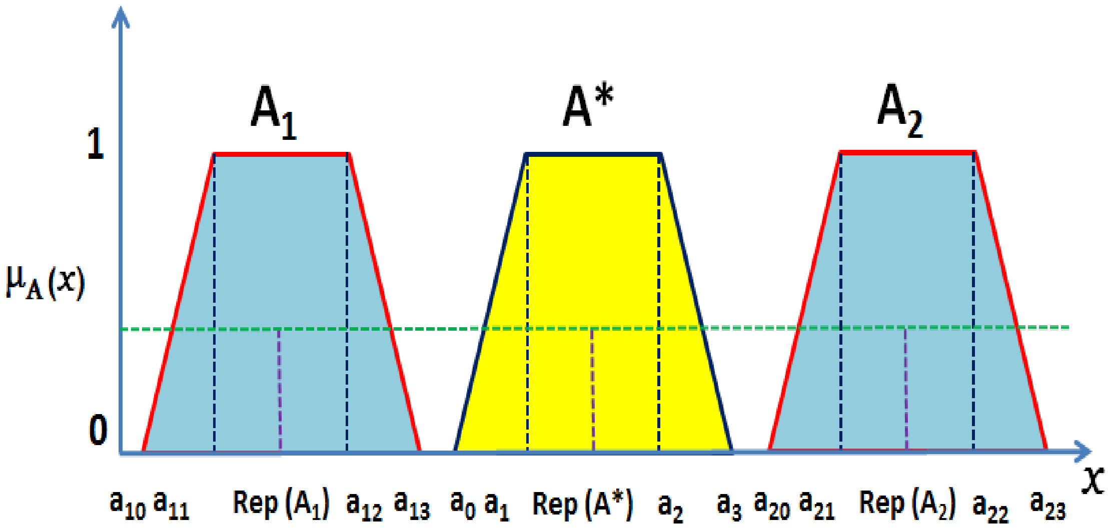
Figure 2.38: Representation of Highest Points and Representation Value in IRCT

1. New Rule Construction: Constructing a new fuzzy set $A^{'}$ which is closest (or with shortest distance) to and has the same RV as the observation A^{*} antecedent fuzzy set. The new fuzzy set $A^{'}$ is ensured to be convex and normal. Similarly the consequent fuzzy set $B^{'}$ can be obtained which is also convex and normal. Thus the newly created rule $A^{'} \Rightarrow B^{'}$ will only involve normal and convex fuzzy sets.

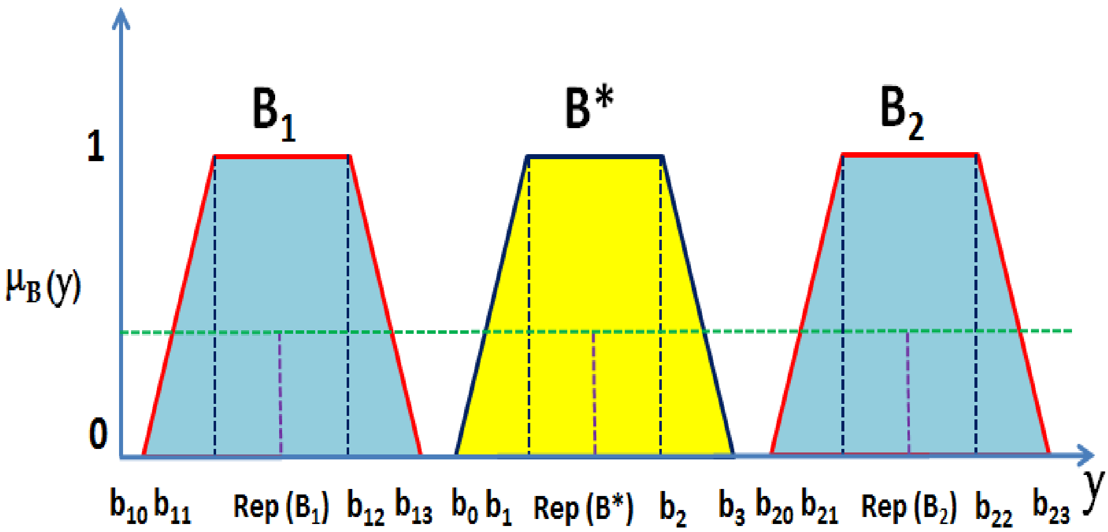
2. Interpolative Reasoning: Now the interpolative reasoning is performed using this new rule, giving the same approximated conclusion as the original rule. In the simplest case, if $A^{*} = A_i$, $i = 1, 2$, then $B^{*} = B_i$. However when A^{*} lies between A_1 and A_2 similarity between A^{*} and $A^{'}$ is considered subject to ensure that the consequent part B^{*} and $B^{'}$ may keep the same similarity and that B^{*} retains normality and convexity. In this case, to obtain the conclusion B^{*} , the Increment and Ratio transformations are used to compute the appropriate operators which will allow the transformation of $B^{'}$ to B^{*} as detailed in the pseudo code.

2.5.12 GA-Based Weight-Learning Interpolation Method (GAWL)

This method is based on the concepts of Normal Points (Left, Right and Composite) as shown in Figure 2.41, Area of fuzzy sets, Weights of antecedent variables, and

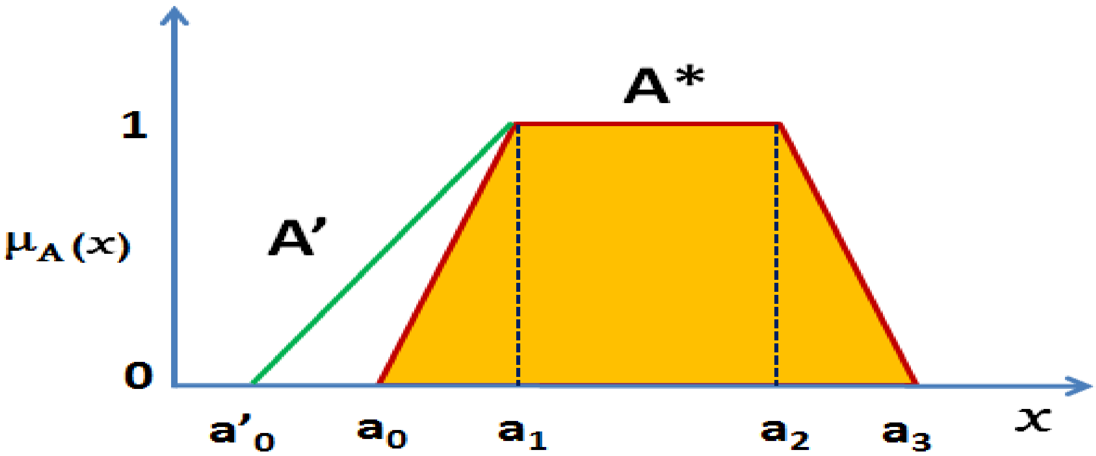


(a) Highest Points and Representation Values for antecedents including observation in IRCT method

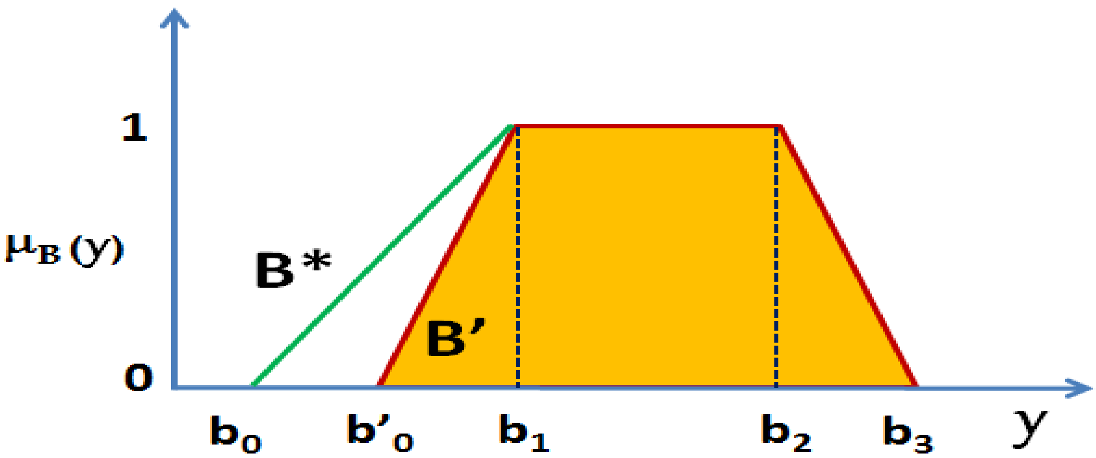


(b) Highest Points and Representation Values for consequents including conclusion in IRCT method

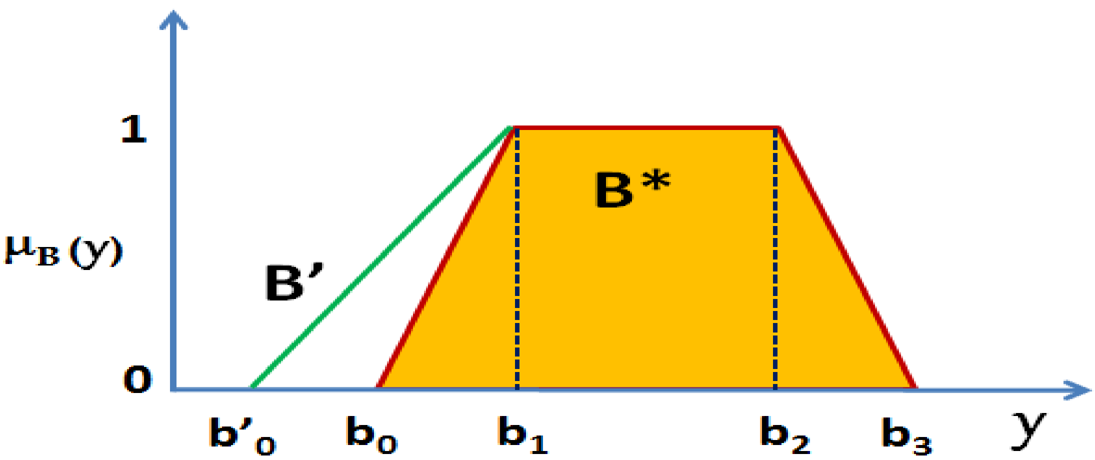
Figure 2.39: IRCT fuzzy interpolation



(a) Comparison of average fuzzy set $A^{'}$ with A^{*}



(b) Increment Transformation for obtaining average fuzzy set $B^{'}$ in IRCT method



(c) Ratio Transformation for obtaining average fuzzy set $B^{'}$ in IRCT method

Figure 2.40: Increment and Ratio transformations

a Genetic Algorithm for automatically learning the weights of antecedent variables [36, 37, 38]. It assumes that fuzzy rule based systems have more than one variable in the antecedents of fuzzy rules and these antecedent variables may have different weights for different degrees of importance. The associated GA-based weight-learning algorithm automatically learns the optimal weights of the antecedent variables of the fuzzy rules. This method can deal with complex fuzzy sets like polygonal, Gussian and bell shapes and counts for the multiple rules in approximating conclusion. While calculating the weights, the involvement of many points and the areas of many parts increases the complexity of the algorithm.

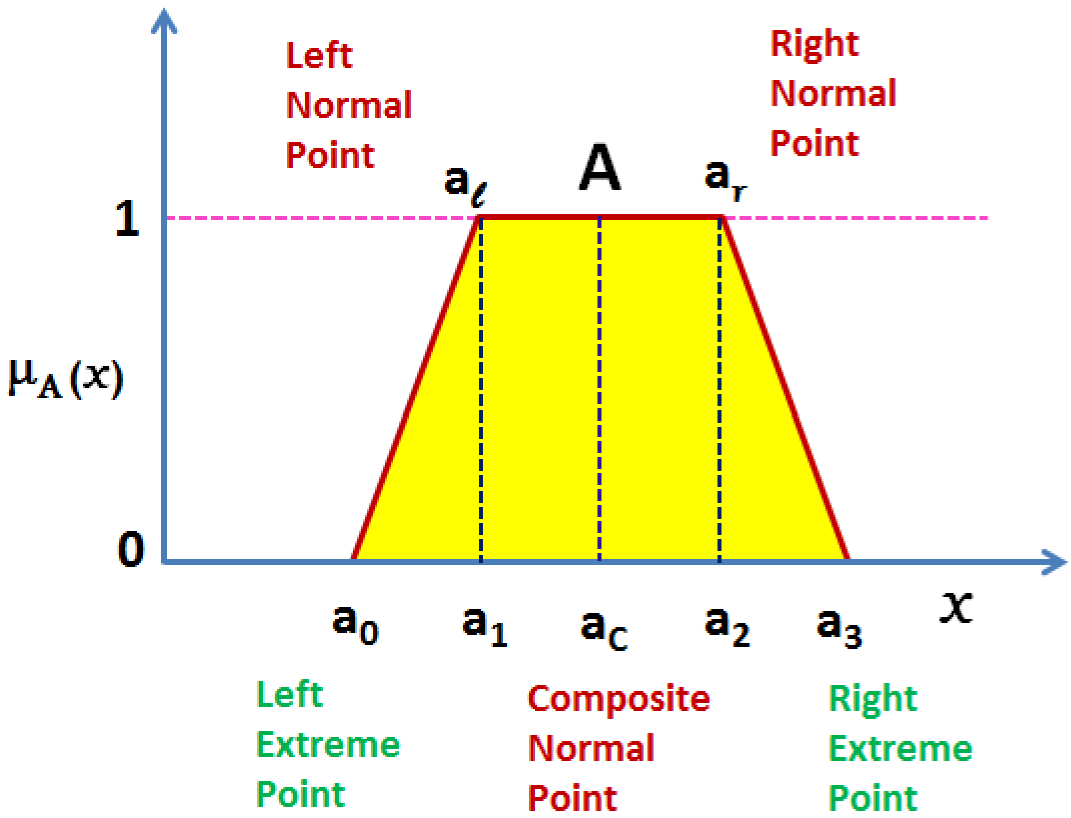


Figure 2.41: Representation of different points in GAWL method

GAWL Algorithm

The method works by integrating the contributions of normal points, areas and weights in computing the conclusion. These points are shown for a trapezoidal fuzzy sets in Figure 2.44. For simplicity, let ψ denote the weight of an antecedent variable and the fuzzy rule base consist of the following:

GAWLInterpolation(X,Y,α,N, M)
X, the set of input universe;
Y, the set of output universe;
A_1,A_2, the antecedent fuzzy sets;
B_1,B_2, the consequent fuzzy sets;
A^*, the observation antecedent fuzzy sets;
B^*, the conclusion consequent fuzzy sets;
μ, the fuzzy membership function;
α, the α-cut level;
a_{ij}, the point of the antecedent fuzzy set A_{ij} ;
n, the total number of points on a fuzzy set;
t, the mid value $(n-1)/2$ of n;
a_l, the left normal point of the antecedent fuzzy set A ;
a_r, the right normal point of the antecedent fuzzy set A ;
a_c, the composite normal point of the antecedent fuzzy set A ;
$a_{ij,c}$, the composite normal point of the antecedent fuzzy set A_{ij} ;
ψ_j, the weight of the antecedent variable X_j ;
w_{ij}, the weight of the antecedent fuzzy set A_{ij} ;
W_i, the weight of the fuzzy rule i ;
N, the number of fuzzy rules;
M, the number of antecedent variables in a fuzzy set;
b_l, the left normal point of the conclusion fuzzy set B^* ;
b_r, the right normal point of the conclusion fuzzy set B^* ;
b_c, the composite normal point of the conclusion fuzzy set B^* ;
S_K, the area of K^{th} part of the fuzzy set ;
S_{L_p}, the area between points a_p and a_{p+1} ;
$S_{R_{n-q-1}}$, the area between points a_{q-1} and a_q ;

Figure 2.42: The GAWLInterpolation Pseudocode-1.

$$Rule-1: IF X_1\ is\ A_{11}(\psi_1)\ and\ X_2\ is\ A_{12}(\psi_2)\ and\ ...X_M\ is\ A_{1M}(\psi_M)\quad THEN\ Y\ is\ B_1$$

$$Rule-2: IF X_1\ is\ A_{21}(\psi_1)\ and\ X_2\ is\ A_{22}(\psi_2)\ and\ ...X_M\ is\ A_{2M}(\psi_M)\quad THEN\ Y\ is\ B_2$$

....

$$Rule-N: IF X_1\ is\ A_{N1}(\psi_1)\ and\ X_2\ is\ A_{N2}(\psi_2)\ and\ ...X_M\ is\ A_{NM}(\psi_M)\quad THEN\ Y\ is\ B_N$$

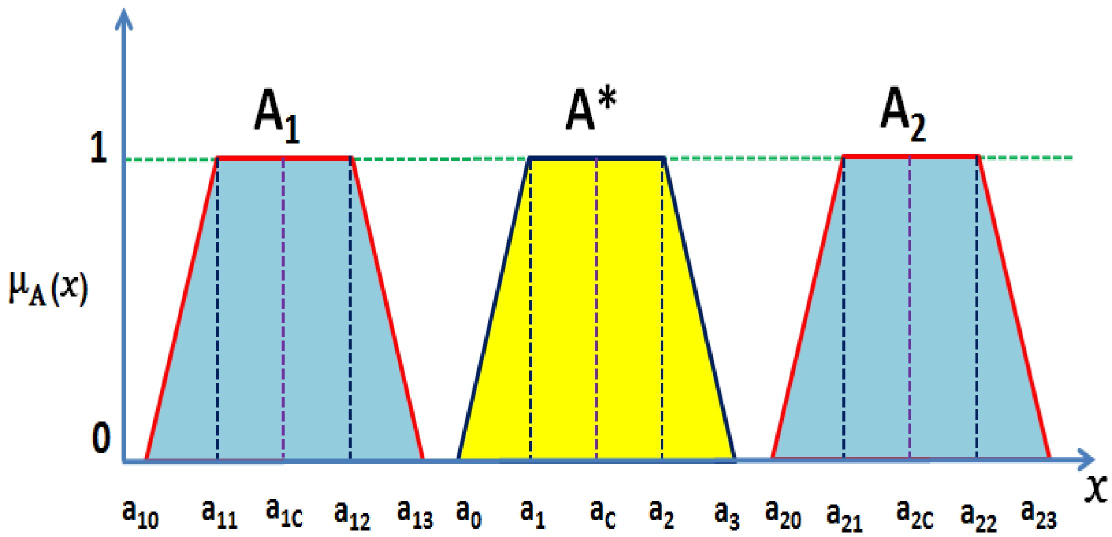
$$Observation: IF X_1\ is\ A_1^*\ and\ X_2\ is\ A_2^*\ and\ ...X_M\ is\ A_M^*$$

———————————————————————————————————

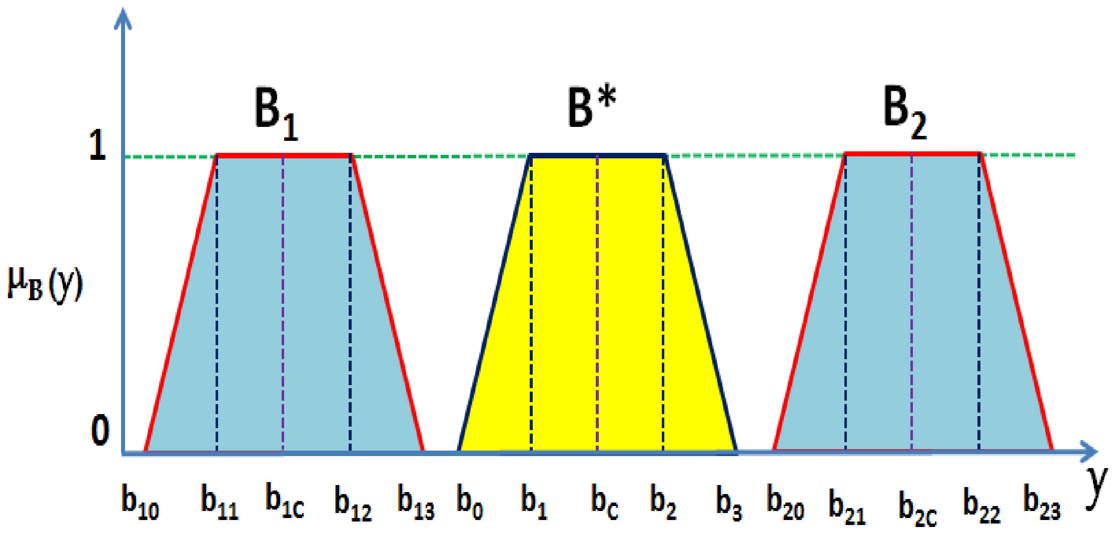
$$Conclusion: Y\ is\ B^*$$

(1) $A_1, A_2 \in X, B_1, B_2 \in Y$
(2) $A_1 \prec A^* \prec A_2, B_1 \prec B_2$
(3) $\forall \alpha \in (0,1]$
(4) $a_c \leftarrow (a_l + a_r)/2,$
(5) $d(A_j^*, A_{ij}) \leftarrow |a_{j,c}^* - a_{ij,c}|$
(6) $d(A_{kj}, A_{hj}) \leftarrow |a_{kj,c} - a_{hj,c}|$
(7) $w_{ij} \leftarrow \psi_j \times \left(1 - \dfrac{d(A_j^*, A_{ij})}{\max\limits_{1 \le k,h \le 1} d(A_{kj}, A_{hj})}\right)$
(8) $W_i \leftarrow \dfrac{\min\limits_{j=1,\ldots,M} w_{ij}}{\sum\limits_{i=1}^{n} \min\limits_{j=1,\ldots,M} w_{ij}}$
(9) $b_c^* \leftarrow \sum\limits_{i=1}^{N} W_i b_{i,c}$
(10) $\pi(A_{ij}) \leftarrow |a_{ij,l} - a_{ij,r}|$
(11) $\pi(B_i) \leftarrow |b_{i,l} - b_{i,r}|$
(12) $\pi(A_j^*) \leftarrow |a_{j,l}^* - a_{j,r}^*|$
(13) $\pi(B^*) \leftarrow \begin{cases} \left(\sum\limits_{j=1}^{M} \psi_j \times \pi(A_j^*)\right) \times \left(\sum\limits_{\substack{i=1 \\ \exists ij, \pi(A_{ij})>0}}^{N} W_i \times \dfrac{\pi(B_i)}{\sum\limits_{j=1}^{M} \psi_j \times \pi(A_{ij})}\right), \\ if\ \exists\, ij, \pi(A_{ij}) > 0 \\ \sum\limits_{i=1}^{N} W_i \times \pi(B_i), \\ if\ \forall\, ij, \pi(A_{ij}) = 0 \end{cases}$
(14) $b_l^* \leftarrow b_c^* - \pi(B^*)/2$
(15) $b_r^* \leftarrow b_c^* + \pi(B^*)/2$
(16) $S_K(B^*) \leftarrow \begin{cases} \left(\sum\limits_{j=1}^{M} \psi_j \times S_K(A_j^*)\right) \times \left(\sum\limits_{\substack{i=1 \\ \exists j, S_K(A_{ij})>0}}^{N} W_i \times \dfrac{S_K(B_i)}{\sum\limits_{j=1}^{M} \psi_j \times S_K(A_{ij})}\right), \\ if\ \exists\, ij, S_K(A_{ij}) > 0 \\ \sum\limits_{j=1}^{M} \psi_j \times S_K(A_j^*), \\ if\ \forall\, ij, S_K(A_{ij}) = 0 \end{cases}$
(17) $b_p^* \leftarrow b_l^* - \sum\limits_{k=p}^{t-1} \dfrac{2S_{L_k}(B^*)}{\alpha_k + \alpha_{k+1}}$
(18) $b_q^* \leftarrow b_r^* + \sum\limits_{k=p}^{t-1} \dfrac{2S_{R_{n-q-1}}(B^*)}{\alpha_{n-k-1} + \alpha_{n-k}}$
(19) $B^* \leftarrow (b_0^*, b_1^*, \ldots, b_l^*, b_r^*, \ldots, b_{n-2}^*, b_{n-1}^*)$

Figure 2.43: The GAWLInterpolation Pseudocode-2.

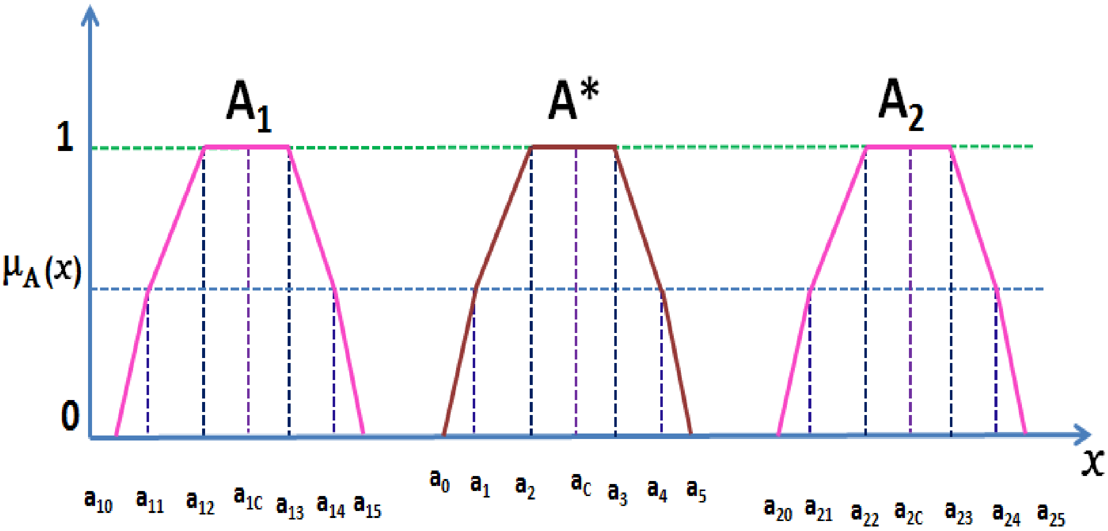


(a) Different points for antecedents including observation in GAWL method

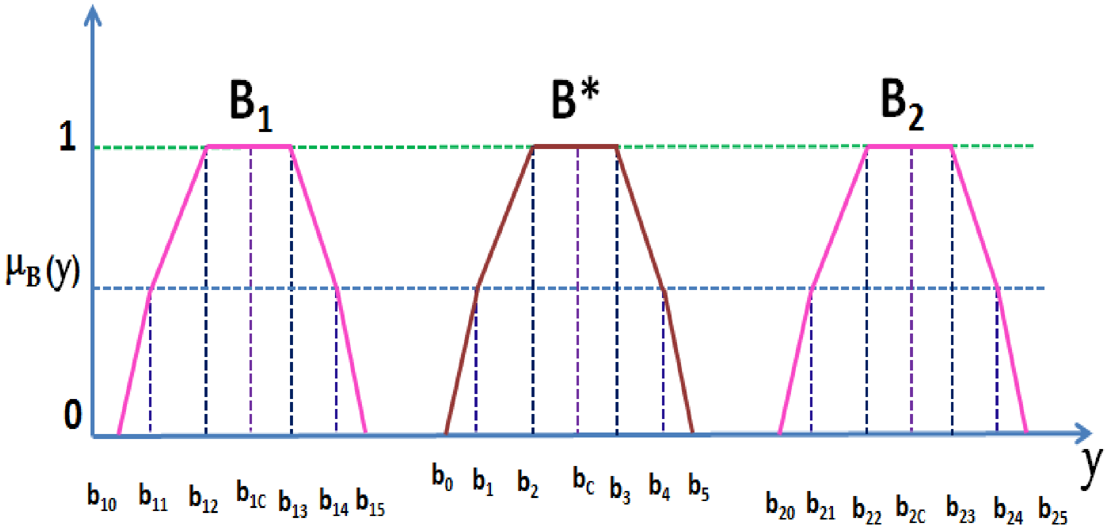


(b) Different points for consequents including conclusion in GAWL method

Figure 2.44: GAWL fuzzy interpolation with trapezoidal



(a) Hexagonal membership function for antecedents including observation in GAWL method



(b) Hexagonal membership function for consequents including conclusion in GAWL method

Figure 2.45: GAWL fuzzy interpolation with hexagonal

The GAWL algorithm is implemented in the following steps, which is outlined in the pseudo code of Figures 2.42 and 2.43:
1. Calculate the composite normal point.
2. Calculate the weight w of each antecedent fuzzy set with respect to the corresponding observation and the weight W of the rule.
3. Calculate the left normal point b_l^* and right normal point b_r^* of the emerging conclusion.
4. Divide the membership function of each polygonal fuzzy set appearing in the fuzzy rules and observations into the left and right areas on the basis of points.
5. Calculate the areas of all parts of emerging conclusion B^*.
6. Find out all the points by using all normal points and all areas to obtain the final conclusion B^*.

2.6 Evaluation Criteria for Fuzzy Rule Interpolation (FRI) Methods

While fuzzy Rule Interpolation (FRI) offers a flexible solution for the problem of sparse fuzzy rule-based inference, there are many aims that require careful consideration in devising such systems. In particular, it should be ensured that these methods should: 1) produce the normal conclusion, 2) maintain the piece-wise linearity, 3) be applicable to different kinds of fuzzy sets, 4) be able to handle multidimensional environments, 5) minimise computational complexity [19, 26, 88, 91, 92, 95, 117, 141, 204].

On the basis of reviewing a wide range of fuzzy interpolation methods, a set of important performance evaluation criteria are identified and generalized. Although it is not necessary that all such criteria are fulfilled in developing and applying the above methods. It is expected however that most of the criteria should be satisfied by a useful fuzzy rule interpolation technique, with other problem-specific parameters to fulfil given certain application.

For simplicity, unless stated otherwise, the following discussion assumes that only two adjacent rules and one observation are considered. Also, the two-rule antecedents and consequents are represented by two-triangular fuzzy sets A_1, A_2 and B_1, B_2, respectively, in a manner that $A_1, A_2 \in X$ and $B_1, B_2 \in Y$, where X and Y are the input and output universes of discourse and A and B are the generic fuzzy sets of X and Y such that:

$$A = \{x, \mu_A(x) | \mu_A(x) \in [0,1], x \in X\} \tag{2.7}$$

$$B = \{y, \mu_B(y) | \mu_B(y) \in [0,1], y \in Y\} \tag{2.8}$$

where $\mu_A(x)$ is the membership function for a generic fuzzy set A of X, whereas $\mu_{A_1}(x), \mu_{A^*}(x), \mu_{A_2}(x)$ are the membership functions for three fuzzy sets A_1, A^*, A_2 respectively which also include the observation A* (considered to be triangular). Similarly the $\mu_B(y)$ is the membership function for a generic fuzzy set B of Y, whereas $\mu_{B_1}(x), \mu_{B^*}(x), \mu_{B_2}(x)$ are the membership functions for three fuzzy sets B_1, B^*, B_2 respectively which also include the conclusion B* (shown expected triangular).

2.6.1 Avoidance of Abnormal Conclusion (AAC)

A fuzzy rule interpolation method should produce valid conclusion fuzzy sets. This means that the resultant membership value must be in the range of [0, 1] only and does not produce more than one membership function value for one input [19, 26, 88, 92, 95, 117, 141, 204].

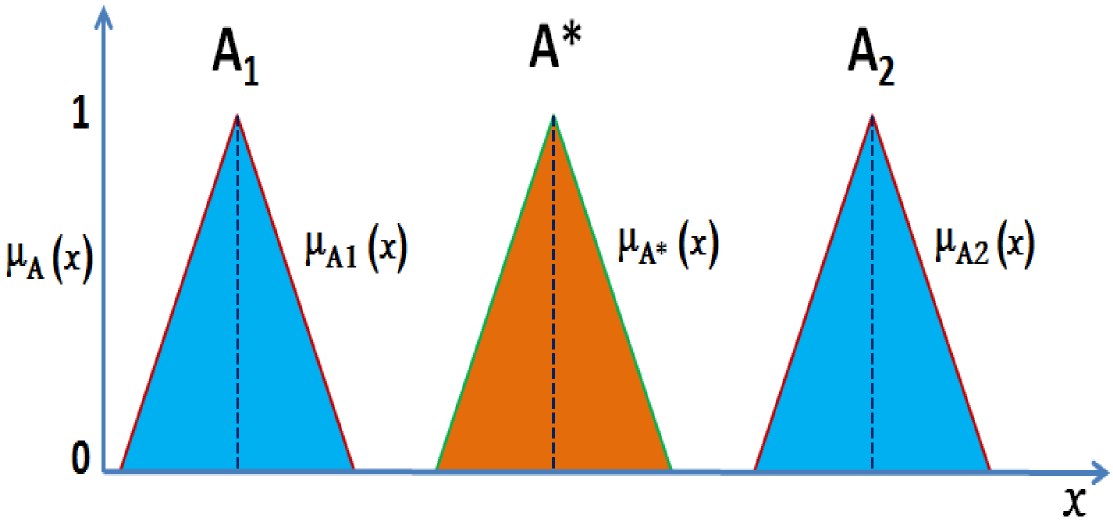
2.6.2 Preservation of Piece-Wise Linearity (PPWL)

The fuzzy rule interpolation method should maintain the piece-wise linearity of an interpolated result. This means that a piece-wise linear conclusion should be inferred from piece-wise linear rules and observations [19, 26, 88, 92, 95, 117, 141, 204]. Strictly speaking there must not be any further interpolation other than computing with the odd points only [83, 84].

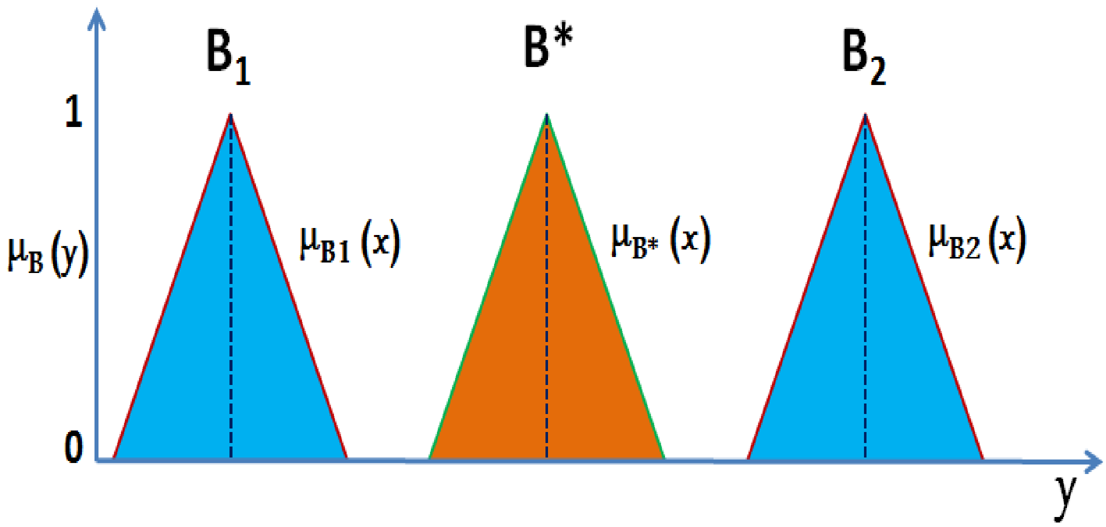
2.6.3 Preservation of Convexity and Normality (PCNF)

A fuzzy rule interpolation method should maintain the normality and convexity for any interpolative results. This means that if an observation is normal and convex then the interpolated conclusion should also be normal and convex [19, 26, 88, 92, 95, 117, 141, 204]. The normality condition is given below which shows that at least one element membership function value must be equal to 1:

$$\mu_A(x) = 1, \ \exists\, x \in X \tag{2.9}$$



(a) 3-Rule antecedents are represented by 3-triangular fuzzy sets A_1, A^*, A_2



(b) 3-Rule consequents are represented by 3-triangular fuzzy sets B_1, B^*, B_2

Figure 2.46: Rule antecedents and consequents representation using triangular fuzzy sets

The convexity condition is given below which dictates that membership function values must be increased or decreased monotonically on either sides of the maximum point:

$$\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq min(\mu_A(x_1); \mu_A(x_2)) \tag{2.10}$$

where $\lambda \in [0,1], x_1, x_2 \in X$.

2.6.4 Applicability to Arbitrary Fuzzy Sets (AAFS)

A fuzzy rule interpolation method should ideally be applicable to any type of fuzzy set or membership functions. This means that it should not be just applicable to linear shaped fuzzy sets [19, 26, 88, 92, 95, 117, 141, 204]. It should be applicable to all shapes like triangular, trapezoidal, gaussian, generalised bell, sigmoidal, S-shaped, Z-shaped or $\prod$-shaped fuzzy sets.

2.6.5 Preservation of Neighbouring Quality (PNQ)

A fuzzy rule interpolation method should maintain the neighbouring quality of the interpolated result. This means that if the observation is surrounded by the antecedent sets of two neighbouring rules then inferred conclusion must be surrounded by the consequent sets of those rules [19, 26, 88, 92, 95, 117, 141, 204]. If the antecedents of the two given rules are A_1 and A_2 and their consequents are B_1 and B_2, the observed rule antecedent A* should lie between A_1 and A_2 such that the inferred conclusion by interpolation method should fall between the two rules consequents B_1 and B_2. That is,

$$If\ A_1 \prec A^*(Observation) \prec A_2 \quad Then\ B_1 \prec B^*(Conclusion) \prec B_2$$

where $\prec$ is a partial order operator.

2.6.6 Mapping Similarity (MS)

A fuzzy rule interpolation method should be able to maintain the similarity between the antecedent fuzzy sets and that between the consequent fuzzy sets. This means the similar observations must lead to similar results [19, 26, 88, 92, 95, 117, 141, 204].

2.6.7 Multiple Rules for Support (MRS)

A fuzzy rule interpolation method should be able to handle fuzzy interpolative reasoning with unlimited multiple fuzzy rules. This means that there should not be any restriction on the number of rules in the sparse rule base system although it has a limited number of scattered rules.

2.6.8 Multiple Membership Functions for Support (MMFS)

A fuzzy rule interpolation method should be able to deal with different kinds of membership function with different rules. Simply it means that the method should work when the fuzzy sets of the antecedents and the consequents of the different fuzzy rules have different kinds of membership function.

2.6.9 Multiple Antecedent Variables for Support (MAVS)

A fuzzy rule interpolation method should support problem domain of multidimensional antecedents. Simply it means that the interpolation method produces the correct inferred conclusion when more than one input variable are given in the antecedent part of the rule [19, 26, 88, 92, 95, 117, 141, 204].

2.6.10 Rule-Base Preservation (RBP) / Modus Ponens Validity (MPV)

A fuzzy rule interpolation method should maintain the compatibility with the sparse rule base. This means that if the observation matches with the antecedent part of a rule, the inferred conclusion must match with the consequent part of that rule [19, 26, 88, 92, 95, 117, 141, 204]. In the rule base, if the antecedent part is A and its consequent part is B, so when the input observation matches with this antecedent part inferred conclusion must be matched with the consequent part of the rule.

2.6.11 Approximation Capability (AC)

A fuzzy rule interpolation method should have a strong approximation capability to match in the rule universe. This means that the inferred result must approximate with the highest degree the relation between the antecedent and consequent universes. The result must meet to the approximated function independently from the measurement however the measured points tend to infinite [90].

2.6.12 Fuzziness of Inferred Result (FIR)

A fuzzy rule interpolation method should be consistent in terms of the fuzziness of the inferred result. This means that the fuzziness of the rule base must be followed by the inferred conclusion except singleton set condition [19, 26, 88, 92, 95, 117, 141, 204]. A singleton set is a fuzzy set with a membership function equal to unity at a single point on the universe of discourse and zero otherwise. A crisp inferred conclusion is obtained if the observation is a singleton itself or all considered consequents of the rule base are singleton shaped.

2.6.13 Overlapping Antecedent Rules for Support (OARS)

A fuzzy rule interpolation method should be able to support rules where antecedents overlap with each other. This means that the method is operable on such problem where two adjacent fuzzy rules have some common members or their intersection are empty.

2.6.14 Computational Complexity (CC)

Irrespective of all desirable features of fuzzy rule interpolation techniques the success of a particular method depends on its computational complexity, which should be minimised.

2.7 Critical Analysis for Fuzzy Rule Interpolation Methods

In the previous sections typical fuzzy rule interpolation methods are outlined and a number of evaluation criteria for the fuzzy rule interpolation methods are provided. This section evaluates all the previously mentioned fuzzy rule interpolation methods with regard to the identified criteria.

2.7.1 Analysis of the KH Method

This is one of the simplest and earliest fuzzy interpolation techniques based on the classical linear interpolation using the α-cut concept. It is fully in accordance with the semantic interpretation of rules as proposed by [54]. This semantic interpretation is the extended version of the analogical inference [183], reflecting the revision principle [146]. Semantic interpretation states that a more similar conclusion must be concluded to the corresponding consequent when the observation and an antecedent are more similar.

This technique has many advantages, for example it behaves approximately linearly between the α levels. Its computational complexity is light because it calculates the α-cut set for the conclusion and therefore, it may be suitable for real time application. It is more suitable for triangle and trapezoidal shaped fuzzy sets because these can be easily described by few characteristics points that the α-cut process. It is initially developed for Single Input Single Output (SISO) fuzzy systems, but it can be extended for the case of Multiple Input Single Output (MISO) fuzzy systems using Minkowski distances. It preserves the original rule base and supports multiple rules.

As indicated previously, this method is restricted to convex and normal fuzzy (CNF) sets. It is shown to rely on implicative rules, viewed as constraints. It does not always provide the normal conclusion and does not maintain the piece-wise linearity of the resultant conclusion. Theoretically, an infinite number of α-cuts are required for a correct result. However, it considers only two α levels which may cause error in the result. Sometimes the bounds of the results are not in the expected order, because the interpolation weights for left-hand sides and for the right-hand sides are not related to each other. It does not always produce the convex and normal conclusion even if the given fuzzy sets are convex and normal [204, 205] as shown in the Figures 2.47 and 2.48.

2.7.2 Analysis of the CP Method

This is also one of the simplest and earliest fuzzy interpolation techniques, based on the classical linear interpolation using imprecise data points. This method applies the extension principle [53] to the fuzzy points defined by the Cartesian products of the left-hand side and the right-hand side of each rule. It has several advantages, for example it may be used with arbitrary fuzzy sets because there is no condition such as CNF imposed. It preserves the rule base and supports multiple conjunctive antecedent variables and multiple rules.

However, it sometimes produces a subnormal conclusion [88] so again the normal conclusion and piece-wise linearity are potential problems associated with this

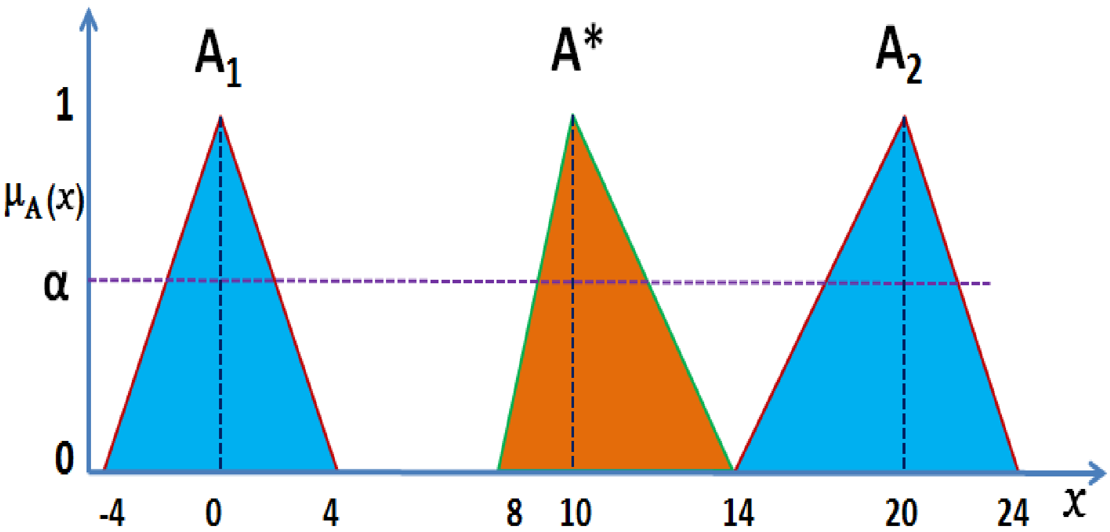


Figure 2.47: Given normal and convex antecedent fuzzy sets in the KH method [205]

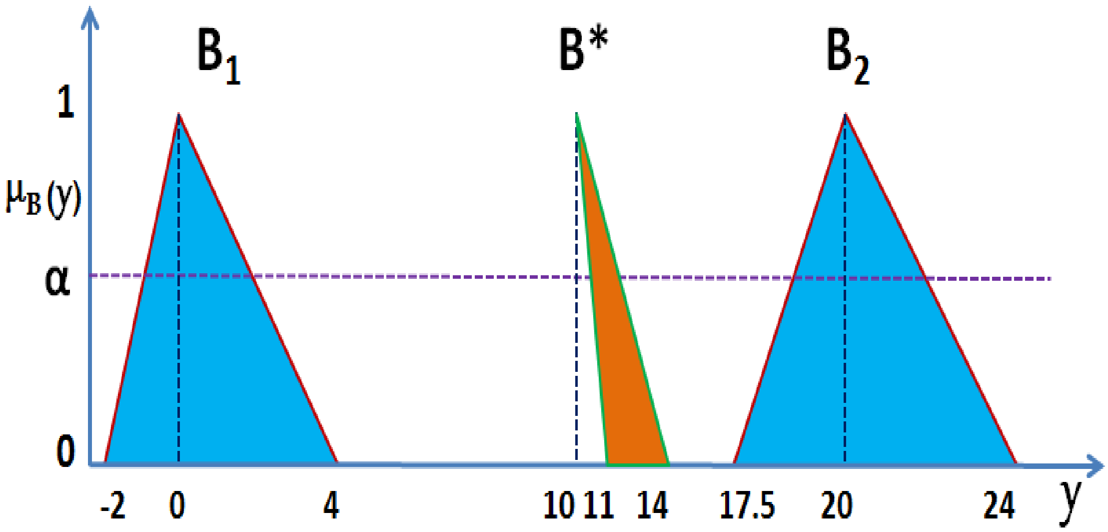


Figure 2.48: Produced nonconvex consequent fuzzy sets in the KH method [205]

method. It does not always generate a convex and normal conclusion even if the given observation and antecedent fuzzy sets are all convex and normal. The computational complexity is slightly higher than that of the KH because both the required relation construction and solving such relations involve more computation [88]. The method is only applicable to crisp observations, and its generalization to fuzzy observations gives an imprecise conclusion which can be situated between the external limits of the conclusions of the rules [28]. Actually, this linear interpolation method from imprecise points can yield very imprecise results even if the input is precise, say $X = x$, because it is clear that the result is the interval on the Y-axis obtained by cutting the shaded area by the straight line $X = x$. Thus with imprecise inputs, the output becomes even more imprecise [57].

2.7.3 Analysis of the FIVE Method

This approximate fuzzy reasoning method is basically developed as an alternative to the compositional rule of inference (CRI). The crisp conclusion obtained by this method has a number of advantages over that achieved using classical min-max CRI. In particular, the control function of this method always fits the points of the fuzzy rules, while the control function of the CRI does not usually fit these points. This method can be used either for dense rule-bases or for sparse rule-bases. A crisp conclusion is directly fetched from the vague conclusion without any defuzzification [119].

The accuracy of this method depends on how accurately the approximate scaling function that it employs is determined. This function is critical for each problem dimension as it explains the vague environment of a given problem [92, 95]. Once the vague environments for the antecedent and consequent universes are determined any rule can be represented by a single point in the environments and then linear interpolation is used to calculate the conclusion. However, such environments for both the antecedent and the consequent are required in advance [119].

This method produces a crisp value as output. If a fuzzy conclusion is needed, the crisp point has to be converted into a fuzzy value by consequence universe examination. The method is restricted to convex and normal fuzzy (CNF) sets. In general, it does provide a normal conclusion but this may depend on the proper approximate scaling function. However, it does not always produce convex (and normal) conclusion nor ensure piece-wise linearity [92, 95]. The computational complexity is slightly higher than those addressed earlier because of the additional calculation of scaling function and approximate scaling function. It supports multiple antecedent variables and the rules, nevertheless.

2.7.4 Analysis of the MACI Method

This method is actually a modified and improved version of the KH interpolation method, based on the vector representation of fuzzy sets. The general MACI method tailors the conclusion as a finite sum of KH interpolations. There are many benefits of this method, for example it is valid on the intervals between the characteristic points. Thus, it is sufficient to calculate only the characteristic points of the conclusion, which reduces computational complexity. It also provides better linear approximation behaviour between the breakpoints. The method preserves the fuzziness of the given rule-base and supports multiple antecedent variables and multiple rules.

The original version of MACI was confined only to CNF sets, but a modified version can handle non-convex fuzzy sets as well as subnormal fuzzy sets [176]. Non-convex fuzzy sets can be considered as union of convex fuzzy sets. This property is exploited in order to solve the problem of non-convexity. Each peak of a fuzzy set is treated as a local reference point and the connecting flanks can be split into a monotone decreasing and a monotone increasing part. Subnormal fuzzy sets are first normalised; afterwards the conclusion can be determined by means of the normalised fuzzy sets. Finally, a denormalisation procedure is applied depending on the height of the fuzzy sets at hand.

It may provide normal conclusions although this is conditional because it is only possible when conclusion B^* fulfils particular conditions such as $b_i^* \geq b_j^*, \forall\, i \leq j$ where b^* is the characteristic point of the conclusion vector, and when an appropriate transformation method is used to transform fuzzy sets. It does not always maintain the piece-wise linearity of the conclusion, but the deviation of the piece-wise linear conclusion from the accurate one is less as compared to methods developed previously. Nevertheless, the convex and normal conclusion can not always be assured [28, 92, 95, 119]. It keeps computational complexity low when it is applied to CNF sets, but the complexity may increase considerably when the CNF restriction is removed.

2.7.5 Analysis of the IMUL Method

This method integrates the features of the MACI method and the conservation of relative fuzziness (CRF) method [177]. The core of the conclusion is determined by the MACI method and the flanks divided using the CRF method. It is specially designed for multidimensional problems (more than one input variable). The fuzziness of a fuzzy set is described as a difference between the support and the core [115]. The advantage of this fuzzy interpolation technique is not only that it takes the fuzziness of the sets at the input spaces, but also takes the core at the consequents into the calculation. This results in a more accurate conclusion. It has another interesting feature different from other previous methods in that the interpolated results inherit the fuzziness from the input space rather than the output space. In other words, the interpolated conclusion B^* should be similar to observation A^* rather than B_1 and B_2 and no defuzzification is needed when all observations are crisp [195].

It is restricted to convex and normal fuzzy (CNF) sets however, due to the limitation of the underlying MACI and CRF methods. It may be applied to non-convex fuzzy sets under certain conditions, relying on the use of the improved version of the MACI. It provides the normal conclusion because the MACI and CRF methods are both used for approximating the conclusion. Once again, it generates convex and normal conclusions to a great extent but is unable to provide piece-wise linear conclusions. Unlike all other previous techniques the computational complexity of this method is slightly high because of the calculations of multidimensional reference points, cores and flanks. However, it does preserve the fuzziness of the rule base and supports multiple antecedent variables and multiple rules.

2.7.6 Analysis of the SFE Method

This method is also an extension and improvement of the KH interpolation method, by exploiting the notion of the slopes of flanking edges. The slopes of an interpolated conclusion are determined with respect to the slopes of neighbouring consequents relative to the relationship between the slopes of observation and antecedents. The most important advantage of this method is that its conclusion is a valid convex and normal fuzzy set. It preserves the fuzziness of the original rule base and supports multiple antecedent variables and multiple rules.

This method provides normal conclusions but it is restricted to triangular convex and normal fuzzy (CNF) sets only. Another disadvantage is the restriction expressing that the same linear combinations need to exist on both the left and the right side of the relationship holding between slopes of the respective edges of the antecedent sets and the slope of the respective edge of the observation [92, 95]. It does not guarantee piece-wise linearity, convexity and normality in the interpolated conclusion. Due to the calculation of the highest point and the endpoints of the support for all the fuzzy sets involved, its computational complexity is slightly higher than that of the KH method.

2.7.7 Analysis of the ST Method

This method is developed to addressed the issue of normality and convexity of that KH interpolation fails to ensure. It is based on the supposition that there is a common similarity in the antecedent and consequent parts. It guarantees the normality and convexity for the conclusion fuzzy set if fuzzy rules involve convex and normal fuzzy (CNF) sets only [205]. It fulfils the neighbouring structure of rules and mapping similarity as well as supporting multiple rules and multiple antecedent variables. It also maintains the logical interpretation of modus ponens. It is able to approximate the conclusion at the highest degree with respect to the triangular shaped fuzzy sets and maintains the fuzziness of interpolated results. It has the property that the membership function of a conclusion B^* has a shape similar to that of an observation A^* but also has a shape which is a linear combination of B_1 and B_2.

It provides a normal and convex conclusion only when $A^* = \beta A_2 + (1-\beta)A_1$ where β is the interpolation parameter/ratio, while in practice the fuzzy set A^* is quite arbitrary so this condition cannot always be satisfied. However, it is assumed to be the starting point of the method. The piece-wise linearity is achievable for certain cases where the membership functions are continuous. Its computational complexity is relatively high because of calculation required to form the new fuzzy set that is similar to the observation involves computing the midpoint of the core of the underlying fuzzy sets. Unfortunately, it is not a generalised version of the KH interpolation method [205].

2.7.8 Analysis of the SGR Method

This method is specially developed for arbitrary shape fuzzy sets. It is based on the semantic and interrelational features of fuzzy sets. A so called revision function is used to determine the final conclusion B^*. The main advantage of this method is that it always gives a conclusion interpretable as a fuzzy set (i.e., the abnormal shape of the conclusion is precluded) and it can be applied for arbitrary shape fuzzy sets. This method possesses the properties of producing normal and convex interpolated fuzzy sets independent of the fuzzy set representation, so as they are convex and normal [88]. It fulfils the neighbouring structure of rules and mapping similarity as well as supporting multiple rules and multiple antecedent variables. It also maintains the logical interpretation of modus ponens.

The method provides a normal conclusion but the piece-wise linearity is limited to particular where the centres of the sets are ordered, allowing a certain part of the observation to exceed the support of the antecedents [118]. Its computational complexity is slightly high because the calculation required to form the new fuzzy set that is similar to an observation involves the computation of the representative point. However, it is able to approximate the conclusion at the highest degree with respect to the trapezoidal shaped fuzzy sets and maintains the fuzziness of inferred results. The only problematic point of this technique is that the calculation of the revision function, even for the special piece-wise linear case needs considerable time [195].

2.7.9 Analysis of the IRG Method

This method is mainly developed as an interpolation method involving fuzzy sets using generalised parameters of locations and shapes. It is based on the gradual behaviour and analogical approach hypotheses [28]. The main advantage of this method is that it always gives interpretable conclusions when the data is linguistically given on the basis of an analogical scheme of reasoning. This method satisfies the properties of normality and convexity as indicated by the experimental results in [28]. It fulfils the neighbouring structure of rules and mapping similarity as well as supporting multiple rules and multiple antecedent variables. It also maintains the logical interpretation of modus ponens.

The method works with certain assumptions such as the total order holding in fuzzy sets based on their locations and the shape of fuzzy sets is restricted to L-R fuzzy intervals (i.e. left and right hand parts of the membership functions). It is again restricted to convex and normal fuzzy (CNF) sets and provides a normal conclusion, and piece-wise linearity. Its computational complexity is also slightly high due to the calculations of location, shape, kernel, and support parameters. However, it is able to approximate the conclusion at the highest degree with respect to the triangular and trapezoidal shaped fuzzy sets and maintains the fuzziness of inferred results. The major drawback of this method is that it is very difficult to semantically evaluate its generalised parameters such as location-distinguishability and shape-distinguishability precisely.

2.7.10 Analysis of the TFRI Method

This method is well suited for polygonal shaped fuzzy sets as well as multidimensional environments. It uses the proposed scale and move transformations to interpolate the conclusion using similarity measures. The main advantage of this method is that it can handle fuzzy sets whose membership functions involve vertical slopes. This is very different from many other interpolation methods. It satisfies the properties of normality and convexity and provides a normal conclusion as shown in the reported experimental results in [109]. This technique maintains the neighbouring structure of rules and mapping similarity, and supports multiple rules and multiple antecedent variables. It also retains the logical interpretation of modus ponens.

This method is implemented with only two surrounding rules. It does not investigate the possible effect of arranging the rule-base in a certain partial order for rules of complex condition patterns. It is restricted to convex and normal fuzzy (CNF) sets and cases where observation is a singleton set or both surrounding antecedents are singleton sets. It provides the piecewise linear conclusion to a certain extent. The application of operations at each α-level for the sake of conservation of convexity increases its computational complexity [92, 95]. It is able to approximate the conclusion at a high degree with regards to all kinds of shapes, especially the trapezoidals in the case of multiple antecedent variables while maintaining the fuzziness of inferred results. However, this method needs more investigation for interpolation of rules involving multiple antecedent variables and multiple rules [85, 86]. Despite extensive support to many evaluation criteria this method does not always support overlapping antecedent rules [124].

2.7.11 Analysis of the IRCT Method

This method is quite similar to TFRI and uses the initial work of representative values from TFRI. It is therefore applicable to polygonal shaped fuzzy sets as well as multidimensional environments. Unlike TFRI, it proposes different incremental and ratio transformations to obtain the conclusion using the similarity measures. The main advantage of this method is that it can handle more complex fuzzy sets like the TFRI method. It also satisfies the properties of normality and convexity and provides the normal conclusion as shown in the experimental results in [109]. It fulfils the neighbouring structure of rules and mapping similarity as well as supporting multiple rules and multiple antecedent variables. It maintains the logical interpretation of modus ponens.

Although this method is implemented for multiple shapes it is restricted to convex and normal fuzzy (CNF) sets and assumes single antecedent variables in rules. To a great extent it provides the piecewise linear conclusion. Its computational complexity is high concerning the calculation of different parameters and transformations. It is able to approximate the conclusion at the highest degree with respect to a single antecedent and maintains the fuzziness of inferred results. The experimental result in [124] seems to coincide with that of using the IRCT method regarding overlapping antecedent rules. However, it suffers from the same problem of TFRI when dealing with overlapping antecedent rules because of their underlying very similar approaches. It does not provide better results than TFRI and does not have the wider scope like TFRI.

2.7.12 Analysis of the GAWL Method

This method is applicable to the polygonal, Gaussian and bell shaped fuzzy sets as well as multidimensional environments. It is based on the calculation of various points and area of fuzzy sets, a genetic algorithm based weight learning mechanism to automatically learn the optimal weights of the antecedent variables of the fuzzy sets. The main advantage of this technique is that it uses the weights of all antecedent variables to prioritise the fuzzy rules therefore it can produce a better interpolation as compared to the others. It satisfies the properties of normality and convexity and provides a normal conclusion. It fulfils the neighbouring structure of rules and mapping similarity, and supports multiple rules and multiple antecedent variables with their weights. It maintains the logical interpretation of modus ponens. This method may produce normal and convex results when the antecedents and the consequents of the fuzzy rules are of different kinds of membership function [36, 37, 38].

Despite consideration of many complex parameters, in certain cases the result obtained by this method is not piecewise-linear. As with almost all other methods, it is also restricted to convex and normal fuzzy (CNF) sets. Its computational complexity is rather high due to the calculation of weights, involving many points and areas of many parts. This major drawback is due to the calculation of many points and areas of fuzzy sets and the use of genetic algorithms. The method is able to approximate the conclusion at the highest degree and maintains the fuzziness of inferred results. The method may not be very effective in a case where the weights of antecedent variables have no or equal importance.

2.7.13 Summary of Comparison of Typical Interpolation Methods based on Evaluation Criteria

For all the outlined typical interpolation methods, comparisons are summarised based on the identified criteria in Table 2.1. This comparison shows that all the methods satisfy the following key criteria: preservation of neighbouring quality, mapping similarity, multiple rules for support, multiple antecedent variables for support, rule-base preservation/ modus ponens validity, approximation capability, and fuzziness of inferred result. Two criteria: avoidance of abnormal conclusion, and preservation of convexity and normality (CNF) are fulfilled by : SFE, ST, SGR, IRG, TFRI, IRCT and GAWL. No method produces normal and convex results when the antecedents and the consequents of the fuzzy rules are of different kinds of membership function. The KH method has the lowest computational complexity while the method GAWL has the highest complexity. The empty entry in Table 2.1 shows that no exact empirical information is available in the literature.

2.8 Summary

This chapter has presented a systematic review of typical fuzzy rule interpolation (FRI) methods. It has summarised the evaluation criteria for FRI on the basis of identified evaluation criteria. All reviewed methods fulfil most of the evaluation criteria. However, certain initial methods such as KH, CP, FIVE, MACI, IMUL, SFE have limitations concerning the shapes of fuzzy sets involved. The issue of multi-dimensional environment and that of piece-wise linearity. However, these issues are dealt with successfully by their successors. The more recent methods such as ST, SGR, IRG, TFRI, IRCT, and GAWL satisfy many criteria, making them potentially more suitable than the earlier approaches for practical applications. However, the computational complexities of these methods are also higher, mainly because of the presence of the required computing mechanisms such as transformation.

Most of the reviewed FRI methods rely on a pre-defined, static fuzzy rule-base, from which the interpolation results are calculated. These methods do not have a

Table 2.1: Summary of Typical Interpolation Methods with regards to Evaluation Criteria

Methods	1. AAC	2. PPWL	3. PCNF	4. AAFS	5. PNQ	6. MS	7. MRS
KH				CNF Sets	$\surd$	$\surd$	$\surd$
CP				$\surd$	$\surd$	$\surd$	$\surd$
FIVE				CNF Sets	$\surd$	$\surd$	$\surd$
MACI				CNF Sets	$\surd$	$\surd$	$\surd$
IMUL				CNF Sets	$\surd$	$\surd$	$\surd$
SFE	$\surd$		$\surd$	CNF Sets	$\surd$	$\surd$	$\surd$
ST	$\surd$	Partial	$\surd$	CNF Sets	$\surd$	$\surd$	$\surd$
SGR	$\surd$	Partial	$\surd$	CNF Sets	$\surd$	$\surd$	$\surd$
IRG	$\surd$	Partial	$\surd$	CNF Sets	$\surd$	$\surd$	$\surd$
TFRI	$\surd$	Partial	$\surd$	CNF Sets	$\surd$	$\surd$	$\surd$
IRCT	$\surd$	Partial	$\surd$	CNF Sets	$\surd$	$\surd$	$\surd$
GAWL	$\surd$	Partial	$\surd$	CNF Sets	$\surd$	$\surd$	$\surd$

Table 2.1: Summary of Typical Interpolation Methods with regards to Evaluation Criteria

Methods	8. MMFS	9. MAVS	10. RBP	11. AC	12. FIR	13. OARS	14. CC
KH	×	✓	✓	✓	✓		Lowest
CP	×	✓	✓	✓	✓		> KH
FIVE	×	✓	✓	✓	✓		> CP
MACI	×	✓	✓	✓	✓		> CP
IMUL	×	✓	✓	✓	✓		> MACI
SFE	×	✓	✓	✓	✓		> KH
ST	×	✓	✓	✓	✓	Partial	> IMUL
SGR	×	✓	✓	✓	✓	Partial	> IMUL
IRG	×	✓	✓	✓	✓	Partial	> IMUL
TFRI	×	✓	✓	✓	✓	Partial	> IRG
IRCT	×	✓	✓	✓	✓	Partial	= TFRI
GAWL	×	✓	✓	✓	✓	Partial	> IRCT

mechanism to support self-diagnosis [206] or self-modification [148] of the original rules. Nevertheless, there is a need to develop FRI for situations where the environment changes and there is a great deal of uncertainty, while following the most evaluation criteria and maintaining the precision of interpolated results in dynamic (self-adaptive) fuzzy rule base systems. This chapter summarised and compared outlined FRI approaches based on the evaluation criteria and literature, leaving such further development of FRI techniques, and also their real world applications as on-going research.

Chapter 3

K-Means Clustering Based Dynamic Fuzzy Rule Interpolation

FUZZY rule interpolation (FRI) [56, 110, 111, 112, 162] is of particular significance for reasoning in the presence of insufficient knowledge. Given a sparse rule base, if an observation has no overlap with antecedent values, no rule can be invoked in classical fuzzy inference, and therefore no consequence can be derived. FRI techniques can support inference in such cases. Most existing FRI systems, regardless of their underlying theory and implementation, tend to process a large amount of interpolated rules, which are generally discarded once the outcomes in response to the given observations are derived. However, interpolated rules may contain potentially useful information, e.g., covering regions that were not covered by the original rule base. If dynamically and intelligently maintained these rules may help greatly improve the overall interpolative coverage and efficacy. This process can be especially beneficial if the frequently appearing observations are of high similarity, where a dynamically created rule may reduce the overheads of interpolation.

A number of techniques [15, 127, 130, 142, 188, 191, 213] exist in the field of dense fuzzy rule-based systems and adaptive fuzzy control, which support dynamic modifications to a given dense rule base. There are also approaches developed for the automatic generation of fuzzy rule-based models [11, 12, 196], using techniques such as neural network [196], genetic algorithm [11, 12], etc. These techniques learn from the data in order to refine a given rule-based system. They can maintain a concurrent, real time rule base for inference and thus, entail more appropriate reasoning results. Unfortunately, such approaches are not directly applicable to sparse rule-based systems due to their assumption of fully covered rules, as well as the underlying computational differences between the use of compositional rule of inference and rule interpolation.

In this chapter, an initial investigation into the feasibility of a dynamic FRI framework is reported and a prototype for implementing the framework is suggested with components built upon widely available techniques. In particular, the collection of the interpolated rules is first partitioned into hyper-cubes (multi-dimensional blocks), in order to identify regions that have accumulated a desirable number of candidate rules. The *k-means* clustering algorithm and cluster quality measurements are then employed to select subsets of such rules, so that aggregated rules may be promoted to further refine the original rule base.

The rest of the chapter is organised as follows. Section 3.1 explains the T-FRI approach that is used extensively in the current implementation. Section 3.2 briefly discusses the *k-means* clustering method which is used for clustering of interpolated rules. Section 3.3 illustrates the proposed *k-means* based dynamic FRI approach, and details an implementation of the method. Section 3.4 provides experimental results that demonstrates the procedures of the proposed approach, and verifies its correctness and accuracy with comparison to conventional FRI. Finally, Section 3.5 summarises this chapter.

3.1 Transformation-Based Fuzzy Rule Interpolation

This section provides an outline of transformation-based fuzzy rule interpolation (T-FRI), including both the underlying concepts and the interpolation steps. For simplicity, in this work, fuzzy sets are represented using triangular membership functions. Suppose that an original, sparse rule base $\mathbb{R}$ exists, with rules $R_i \in \mathbb{R}$ and an observation O:

R_i: IF x_j is $A_{i,j}, j \in \{1, \cdots, N\}$, THEN y is B_i
O: $A_{\circ,1}, \cdots, A_{\circ,j}, \cdots, A_{\circ,N}$

where $A_{i,j} = (a_0, a_1, a_2)$ is the triangular linguistic term for rule R_i, defined on the domain of the antecedent variable x_j, $j \in \{1, \cdots, N\}$, where N is the total number of antecedents, and B_i is the consequent. The observed fuzzy set of variable x_j is denoted by $A_{\circ,j}$. The representative value $rep(A)$ of a triangular fuzzy set is defined as the mean of the X coordinates of the triangle's three points: the left and right extremities a_0, a_2 (with membership values = 0), and the normal point a_1 (with membership value = 1).

$$rep(A) = (a_0 + a_1 + a_2)/3 \tag{3.1}$$

3.1.1 Determine the Closest Rules for the Observation

The distance between R_i and O is determined by computing the aggregated distance of all antecedent variables:

$$d(R_i, O) = \sqrt{\sum_{j=1}^{N} d_j^2}, \quad d_j = \frac{d(A_{i,j}, A_{\circ,j})}{range_{x_j}} \tag{3.2}$$

where $d(A_{i,j}, A_{\circ,j}) = |rep(A_{i,j}) - rep(A_{\circ,j})|$ is the distance between two fuzzy sets in the j^{th} antecedent dimension, with $range_{x_j} = \max x_j - \min x_j$ over the domain of the variable x_j. $d_j \in [0, 1]$ is therefore the normalised result of the otherwise absolute distance measure, so that distances are compatible with each other across different variable domains. The M, $M \geq 2$ rules which have the least distance measurements, with regard to the observed values $A_{\circ,j}$ and the conclusion $B_\circ$, are then chosen to perform interpolation.

3.1.2 Construct the Intermediate Rule

Let the normalised displacement factor $\omega_{i,j}$, as shown in Eqn. 3.3, denote the weight of the j^{th} antecedent of the i^{th} rule:

$$\omega_{i,j} = \frac{\omega_{i,j}^{\dagger}}{\sum_{i=1}^{M} \omega_{i,j}^{\dagger}} \tag{3.3}$$

As $A_{i,j}$ and $A_{\circ,j}$ may totally coincide with each other, the value of $d(A_{i,j}, A_{\circ,j})$ may equal 0. This will make $\omega_{i,j}^{\dagger}$ to be infinite. So, the following non-increasing function can be used to present the weight:

$$\omega_{i,j}^{\dagger} = \exp^{-d(A_{i,j}, A_{\circ,j})} \tag{3.4}$$

The so-called intermediate fuzzy terms $A_j^{\dagger\dagger}$ are constructed from the antecedents of the M rules.

$$A_j^{\dagger\dagger} = \sum_{i=1}^{M} \omega_{i,j} A_{i,j} \tag{3.5}$$

These are then shifted to $A_j^{\dagger}$ such that they have the same representative values as those of $A_{\circ,j}$:

$$A_j^{\dagger} = A_j^{\dagger\dagger} + \delta_j range_{x_j} \tag{3.6}$$

where δ_j is the bias between $A_{\circ,j}$ and $A_j{}^{\dagger}$ on the j^{th} variable domain:

$$\delta_j = \frac{rep(A_{\circ,j}) - rep(A_j{}^{\dagger})}{range_{x_j}} \tag{3.7}$$

Similar to Eqn. 3.6, the shifted intermediate consequence $B^{\dagger}$ can be computed, with the parameters ω_{B_i} and δ_B being aggregated from the corresponding values of $A_j{}^{\dagger}$, such that:

$$\omega_{B_i} = \frac{1}{N}\sum_{j=1}^{N} \omega_{i,j}, \ \delta_B = \frac{1}{N}\sum_{j=1}^{N} \delta_j \tag{3.8}$$

3.1.3 Scale Transformation

Let $A_j^{\dagger\dagger} = (a_0{}^{\dagger\dagger}, a_1{}^{\dagger\dagger}, a_2{}^{\dagger\dagger})$ denote the fuzzy set generated by the scale transformation in the j^{th} antecedent dimension. By using the scale rate s_j, the current support of $A_j^{\dagger}$, $(a_0{}^{\dagger}, a_2{}^{\dagger})$ is transformed into a new support $(a_0{}^{\dagger\dagger}, a_2{}^{\dagger\dagger})$, such that $a_2{}^{\dagger\dagger} - a_0{}^{\dagger\dagger} = s_j \times (a_2^{\dagger} - a_0^{\dagger})$.

$$\begin{cases} a_0{}^{\dagger\dagger} = \frac{a_0{}^{\dagger}(1+2s_j)+a_1{}^{\dagger}(1-s_j)+a_2{}^{\dagger}(1-s_j)}{3} \\ a_1{}^{\dagger\dagger} = \frac{a_0{}^{\dagger}(1-s_j)+a_1{}^{\dagger}(1+2s_j)+a_2{}^{\dagger}(1-s_j)}{3} \\ a_2{}^{\dagger\dagger} = \frac{a_0{}^{\dagger}(1-s_j)+a_1{}^{\dagger}(1-s_j)+a_2{}^{\dagger}(1+2s_j)}{3} \\ s_j = \frac{a_2{}^{\dagger\dagger}-a_0{}^{\dagger\dagger}}{a_2{}^{\dagger}-a_0{}^{\dagger}} \end{cases} \tag{3.9}$$

From the above and the given observation terms $A_{\circ,j}$, and also the scale and move transformation $T(A_j^{\dagger}, A_{\circ,j})$, the scaling factor s_B for the consequent is calculated using Eqn. 3.10.

$$s_B = \frac{\sum_{j=1}^{N} s_j}{N} \tag{3.10}$$

3.1.4 Move Transformation

$A_j^{\dagger\dagger}$ is subsequently moved using the move rate m_j as given in Eqn. 3.11, so that the final transformed fuzzy set matches the exact shape of the observed value $A_{\circ,j}$.

$$\begin{cases} m_j = \frac{3(a_0 - a_0{}^{\dagger\dagger})}{a_1{}^{\dagger\dagger} - a_0{}^{\dagger\dagger}}, & a_0 \geq a_0{}^{\dagger\dagger} \\ m_j = \frac{3(a_0 - a_0{}^{\dagger\dagger})}{a_3{}^{\dagger\dagger} - a_2{}^{\dagger\dagger}}, & \text{otherwise} \end{cases} \tag{3.11}$$

From this, the move factor m_B for the consequent is calculated such that:

$$m_B = \frac{\sum_{j=1}^{N} m_j}{N} \tag{3.12}$$

The final interpolated result $B_\circ$ can now be estimated by applying the scale and move transformation to $B^\dagger$, using the parameters s_B, and m_B.

3.2 K-Means Clustering

Clustering is a data-mining technique in which records are grouped together based on their locality and connectivity within the n-dimensional space [66]. It is used to place data elements into related groups without advance knowledge of the group definitions. It identifies a finite set of clusters (categories) to describe the data, and related records are grouped together on the basis of having similar values for attributes [157]. Records within a cluster are more similar to each other, and more different from records that are in other clusters. The clusters may be mutually exclusive, hierarchical or overlapping.

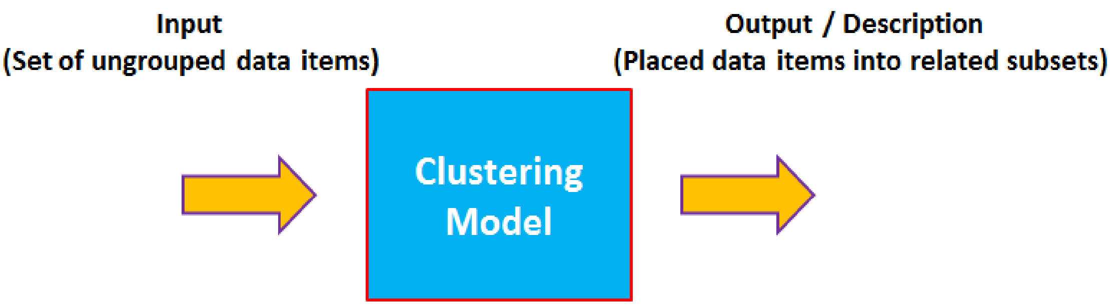


Figure 3.1: Clustering Model

K-means clustering [134] is one of the simplest unsupervised learning algorithms for solving the well-known clustering problem. The procedure classifies in simple terms a given data set through a certain number of clusters (assume k clusters) fixed a priori. k centroids are defined, one for each cluster. These centroids should be placed with care, since different placements yield different results. As a rule of thumb, they should place as far apart as possible. Following placement, each point belonging to a given data set should be associated with the nearest centroid. When no point is pending, the first step is complete and an early grouping has been realised. At this point k new centroids need to be re-calculated as barycenters of the clusters resulting from the previous step. With these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop it may be observed that the k centroids change their location step by step until no more changes are done. In other words, the centroids cease to move [139].

In general, we have n data points $x_i, i = 1...n$ that have to be partitioned in k clusters. The goal is to assign a cluster to each data point. *k-means* is a clustering method that aims to find the positions $c_j, j = 1...k$ of the clusters that minimize the distance from the data points to the cluster. *k-means* clustering can be derived by:

$$\mathbb{C} = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^j - c_j||^2 \tag{3.13}$$

where where x_i^j is the i^{th} data point belonging to the j^{th} cluster and c_j is the centroid of the j^{th} cluster, $d(x_i^j, c_j) = ||x_i^j - c_j||^2$ is a chosen distance measure between a data point x_i^j and the cluster centre c_j is an indicator of the distance of the n data points from their respective cluster centres.

The *k-means* algorithm has the following properties:

- There are always k clusters.
- There is always at least one item in each cluster.
- The clusters are non-hierarchical and they do not overlap.
- Every member of a cluster is closer to its cluster than any other cluster because closeness does not always involve the 'centre' of clusters.

This algorithm consists of the following steps [49]:

- The dataset is partitioned into k clusters and the data points are randomly assigned to the clusters resulting in clusters that have roughly the same number of data points.
- For each data point:
  - Calculate the distance from the data point to each cluster.
  - If the data point is closest to its own cluster, leave it where it is. If the data point is not closest to its own cluster, move it into the closest cluster.
- Repeat the above step until a complete pass through all the data points results in no data point moving from one cluster to another. At this point the clusters are stable and the clustering process ends.

The *k-means* algorithm is popular because it is easy to implement, and its time complexity is $O(n)$, where n is the number of patterns. Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find optimal configuration, corresponding to the global objective function minimum [9]. The algorithm is also significantly sensitive to the randomly-selected cluster centres. The choice of initial partition can greatly affect the final clusters that result, in terms of inter-cluster and intra-cluster distances and cohesion. The *k-means* algorithm can be run multiple times to reduce this effect [87].

3.3 Dynamic Fuzzy Rule Interpolation (D-FRI) Approach

This section provides an outline of the proposed dynamic FRI framework, where the essential formulae are presented. Fig. 3.2 illustrates the overall operation for this framework. Initially, there exists a set of original (sparse) rules $\mathbb{R}$. While running the FRI system, an interpolation mechanism such as T-FRI continuously fills a pool of interpolated rules $\mathbb{R}'$. The antecedent domains of $\mathbb{R}'$ are then partitioned into a

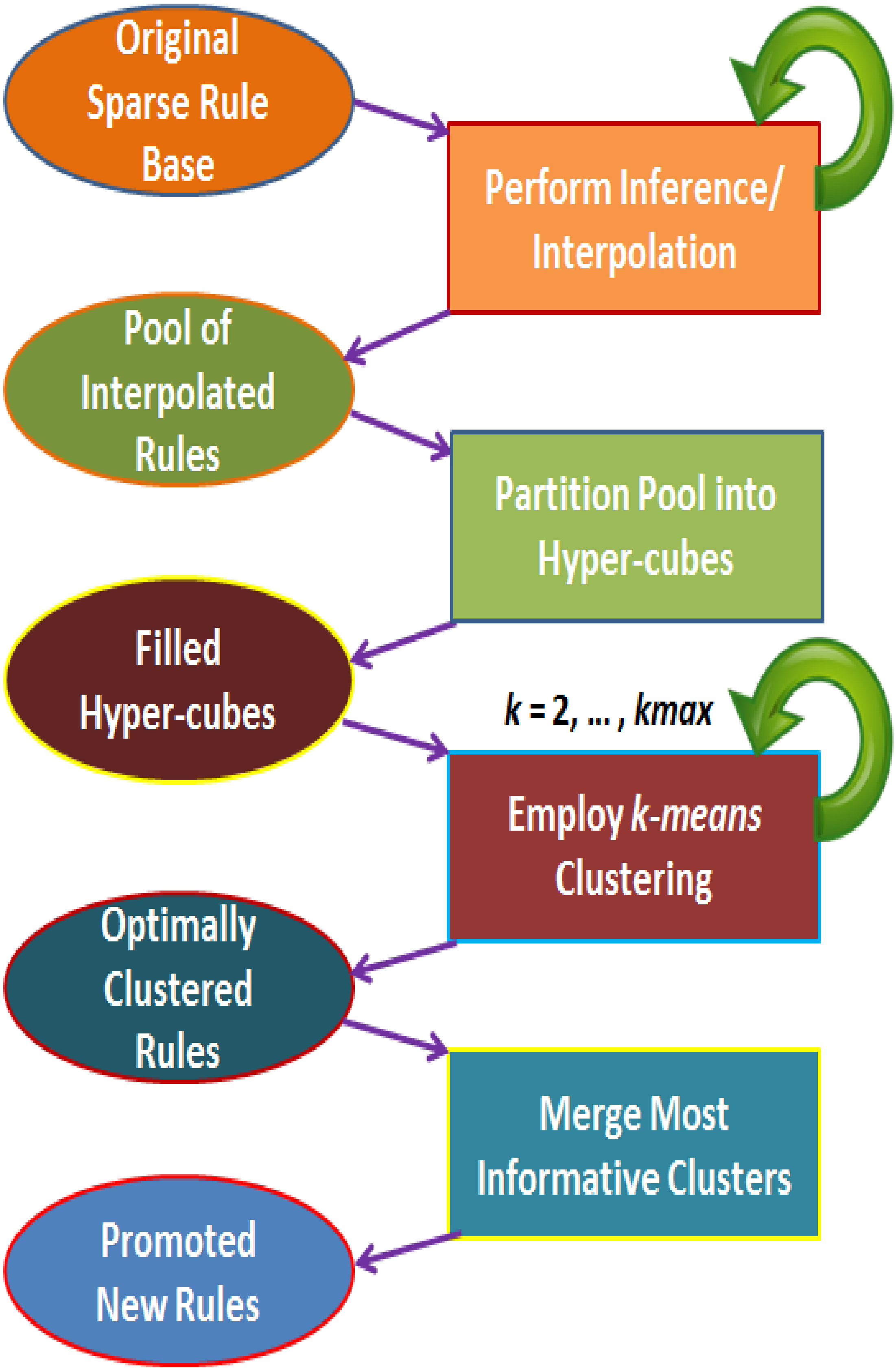


Figure 3.2: Procedure of *k-means* clustering based Dynamic FRI

set of hyper-cubes $\mathbb{H}$, so that certain regions that have accumulated a good number of interpolated rules can be determined. A clustering algorithm is then employed to group the similar rules together, for each "filled" hyper-cubes, whilst the most informative clusters are also identified. Finally, an aggregation process is applied to the selected groups of rules, in order to construct and promote new rules to becoming member of $\mathbb{R}$. This may be an intuitive process that can be applied to any sparse rule-based system, so that the overheads of interpolating similar, commonly observed values could be greatly reduced. If suitable strategies are adopted for the internal components of antecedent partitioning, rule clustering, and aggregation, the accuracy of the resulting dynamic FRI system may be further improved. In the algorithm, the averaged Euclidean distance $d(R_p, R_q)$ between two rules can be defined as:

$$d(R_p, R_q) = \sqrt{\sum_{1}^{N} \frac{(rep(A_{p,i}) - rep(A_{q,i}))^2}{range_{x_i}}} \quad (3.14)$$

3.3.1 Collecting Pool of Interpolated Rules

In the beginning, there exists a sparse rule base R which is developed by the experts or derived from the data. For every given observation, the transformation based FRI (T-FRI) method [85, 86] is run to infer the conclusion. The interpolated conclusion for the given observation is used to perform the appropriate task within the FRI reasoning system. Such interpolated results are also stored to form a pool of interpolated rules $\mathbb{R}'$ for future use. While running the FRI reasoning system, the T-FRI method continuously fills in this pool. After a desired number of interpolated rules in the pool are accumulated, these interpolated results as by-product of previous FRI are then used for further processing to modify the sparse rule base.

3.3.2 Partitioning of Input Space

In order to identify the uncovered regions of $\mathbb{R}$, as well as the most frequently interpolated areas that are covered by $\mathbb{R}'$, a partitioning-based method is employed. By dividing the value ranges of the antecedent variables, the antecedent domain can be partitioned into a set of hyper-cubes $\mathbb{H}$. A given rule R_i (or R'_i) is then assigned to its associated hyper-cubes H_p by checking whether its antecedent values lie within the boundaries of H_p:

$$R_i \in H_p \text{ if } rep(A_{i,j}) \in [\min H_{p,j}, \max H_{p,j}), j \in \{1, \ldots, N\} \quad (3.15)$$

The total number of hyper-cubes and their sizes should be dynamically adjusted according to the current state of the (sparse) rule base. Being a preliminary investigation, as well as for the simplicity of explanation, this work assumes pre-determined partitions, where the input dimensions are evenly divided into η intervals. The total number of hyper-cubes $|\mathbb{H}|$ is therefore η^N. A given hyper-cube H_p is considered "filled" if it has accumulated a number of rules $|H_p|$ exceeding a pre-defined threshold σ, relative to the size of the hyper-cube. Note that alternative measures such as entropy may be employed to determine whether a given hyper-cube should be processed.

3.3.3 Clustering of Interpolated Rules

The standard k-means clustering algorithm [134], given in Algorithm 3.3.1 is used to group the similar rules $R' \in H$ together. In the algorithm, the distance between a given rule R' and the centroid μ_q of a cluster C_q is calculated similar to Eq. 3.14 as follows:

$$d(R', \mu_q) = \sqrt{\sum_{1}^{N}(rep(A'_i) - \mu_{q,i})^2 + (rep(B') - \mu_{q,N+1})^2} \tag{3.16}$$

for $\forall q \in \{1, \dots, k\}$ **do**
  randomly assign $R' \in H$ to C_q
  $\mu_{q,j} = rep(A'_j)$, $j \in \{1, \dots, N\}$
  $\mu_{q,N+1} = rep(B')$
while $\forall q \in \{1, \dots, k\}, \mu'_q \neq \mu_q$ **do**
  for $\forall R' \in H$ **do**
    assign R' to C_q with $\min d(R', \mu_q)$
  for $\forall q \in \{1, \dots, k\}$ **do**
    $\mu'_{q,i} = \frac{\sum_{R' \in C_q} A'_i}{|C_q|}, i \in \{1, \dots, N\}$
    $\mu'_{q,N+1} = \frac{\sum_{R' \in C_q} B'}{|C_q|}$
return $\mathbb{C}$

Algorithm 3.3.1: KMeans(H, k)

This process is carried out iteratively with respect to different values of k, so that the best k, or the most suitable cluster arrangement $\mathbb{C}^*$ may be discovered. The quality is obtained by computing the Dunn Index DI_k [58], which is defined as a ratio of cluster isolation and compactness. A higher value of DI_k indicates a more favourable result:

$$DI(\mathbb{C}_k) = \min_{p,q \in \{1,\ldots,k\}, p \neq q} \{ \frac{m_{pq}}{\max_{r \in \{1,\ldots,k\}} s_r} \} \tag{3.17}$$

where the s_q and m_{pq} are the intra-cluster (compactness) and inter-cluster (isolation) distance measurements, respectively:

$$s_q = \sqrt{\sum_{R' \in C_q} \frac{d(R', \mu_q)^2}{|C_q|}}, \ m_{pq} = d(\mu_p, \mu_q) \tag{3.18}$$

A selection process is then employed to choose one or more suitable groups of rules to be the candidates for potential promotion into the rule base. This may be implemented as picking the largest, most compact (in case of a tie) cluster. The resultant, complete algorithm for the selection of quality clusters is given in Algorithm 3.3.2.

```
for ∀k ∈ {2, ..., |H|/2} do
    ℂ_k = KMeans(H, k)
    if DI(ℂ_k) > DI* then
        DI_* = DI(ℂ_k), ℂ* = ℂ_k
return C_q ∈ ℂ* with max|C_q| and min s_q
```

Algorithm 3.3.2: Iterative Rule Clustering

3.3.4 Rule Promotion

Rule promotion is based on the highest quality cluster arrangement (k) obtained by the Dunn Index [58]. From this best cluster arrangement, only those clusters that satisfy the threshold condition σ are selected for rule promotion process. The total number of newly promoted rules can also be controlled by further selecting the most compact (see Eq. 3.18) clusters. For every selected cluster a single rule is generated where all informative rules $R' \in C_q \subset H$ are taken for further generalisation in an effort to form a new, aggregated rule, which is pure robust and is hereafter referred to as R^*.

This work adopts a weighted combination method, it uses the cluster centroid μ_q to compute the contributions from the individual candidate rules. Similar to the process of constructing intermediate rules suggested in the T-FRI approach [85, 86], a matrix w_{ij} of dimension $|C_q|$, $N+1$ is used. It indicates the weight of A'_{ij} of an interpolated rule $R'_i \in C_q$ to the jth antecedent A^*_j of R^*:

$$w_{i,j} = \frac{1}{d(A'_{i,j}, \mu_{q,j})}, i \in \{1, \ldots, |C_q|\}, j \in \{1, \ldots, N\} \tag{3.19}$$

and that of B'_i to B^*:

$$w_{i,N+1} = \frac{1}{d(B'_i, \mu_{q,N+1})} \tag{3.20}$$

The normalised weights can also be obtained:

$$w'_{i,j} = \frac{w_{i,j}}{\sum_{i=1}^{|C_q|} w_{i,j}} \tag{3.21}$$

From this, the components of the dynamically promoted new rule R^* may be constructed:

$$A^*_j = \sum_{i=1}^{|C_q|} w'_{i,j} A'_{i,j},\ j \in \{1, \ldots, N\}, B^* = \sum_{i=1}^{|C_q|} w'_{i,N+1} B'_i$$

This newly promoted R^* is then added to the original (sparse) rule base: $\mathbb{R} = \mathbb{R} \cup \{R^*\}$, while the rules involved in the aggregation process are removed from the pool of interpolated rules: $\mathbb{R}' = \mathbb{R}' \setminus C_q$. This partitioning-clustering-promotion procedure is applied for any hyper-cubes satisfying $|H_p| \geq \sigma$. The entire dynamic FRI process may repeat until the original rule base reaches a state with sufficient coverage of the problem domain.

3.4 Experimentation and Discussion

A numerical example is employed to demonstrate the process of the proposed approach, as well as validating its accuracy. A function of three crisp input variables, shown in Eq. 3.22 is chosen to populate a sparse rule base $\mathbb{R}$ of size 100. An initial fuzzy rule is generated by fuzzifying the crisp inputs and their associated function output, where a numerical value a is converted to a fuzzy set A with a support length of 1: $A = (a-0.5, a, a+0.5)$, $Rep(A) = a$. This provides a simple non-linear rule base suitable for the purpose of this preliminary investigation. The experiment in this section invokes three different values of H, where the antecedent dimensions are evenly partitioned into $H \in \{2, 3, 4\}$ intervals, as a result, 2^3, 3^3, and 4^3 hyper-cubes can be created, with threshold value σ set to 20, 10, and 5, respectively, corresponding to the different sized hyper-cubes.

$$y = 1 + \sqrt{x_1} + \frac{1}{x_2} + \frac{1}{\sqrt{x_3^3}},\ x_1, x_2, x_3 \in [1, 20] \tag{3.22}$$

3.4.1 K-Means Clustering Results

Table 3.1: Interpolated Rules within a Hyper-Cube

	$A_{i,1}$	$A_{i,2}$	$A_{i,3}$	B_i
R'_1	(2,2.5,3)	(17.1,17.6,18.1)	(10,10.5,11)	(7.2,7.7,8.2)
R'_2	(3.8,4.3,4.8)	(19.4,19.9,20.4)	(10.1,10.6,11.1)	(11.2,11.7,12.2)
R'_3	(6.6,7.1,7.6)	(18.9,19.4,19.9)	(11.4,11.9,12.4)	(14.8,15.3,15.8)
R'_4	(1.8,2.3,2.8)	(19,19.5,20)	(12.4,12.9,13.4)	(6.8,7.3,7.8)
R'_5	(2.9,3.4,3.9)	(16.8,17.3,17.8)	(10.7,11.2,11.7)	(7.7,8.2,8.7)
R'_6	(3.7,4.2,4.7)	(17.3,17.8,18.3)	(10.2,10.7,11.2)	(9.3,9.8,10.3)
R'_7	(3.4,3.9,4.4)	(13.6,14.1,14.6)	(8.4,8.9,9.4)	(11.1,11.6,12.1)
R'_8	(5.8,6.3,6.8)	(15.1,15.6,16.1)	(12.5,13,13.5)	(13.6,14.1,14.6)
R'_9	(1.2,1.7,2.2)	(18.2,18.7,19.2)	(10.9,11.4,11.9)	(6.7,7.2,7.7)
R'_{10}	(4.4,4.9,5.4)	(16.3,16.8,17.3)	(12.5,13,13.5)	(9.3,9.8,10.3)
R'_{11}	(4.1,4.6,5.1)	(15.2,15.7,16.2)	(7.8,8.3,8.8)	(10.5,11,11.5)
R'_{12}	(2.7,3.2,3.7)	(18.6,19.1,19.6)	(7.8,8.3,8.8)	(7.9,8.4,8.9)

Table 3.2: K-Means Clustering Outcomes

$\mathbb{C}$	DI
$\{R'_2,R'_3,R'_6,R'_7,R'_8,R'_{10},R'_{11}\},\ \{R'_1,R'_4,R'_5,R'_9,R'_{12}\}$	0.726
$\{R'_3,R'_8\},\ \{R'_1,R'_4,R'_5,R'_6,R'_7,R'_9,R'_{11},R'_{12}\},\ \{R'_2,R'_{10}\}$	0.489
$\{R'_7,R'_{11}\}, \{R'_1,R'_4,R'_5,R'_6,R'_9,R'_{10},R'_{12}\}, \{R'_3,R'_8\}, \{R'_2\}$	0.875

Table 3.3: Normalised Weights

	$w'_{i,1}$	$w'_{i,2}$	$w'_{i,3}$	$w'_{i,4}$
$w'_{1,j}$	0.042	0.184	0.058	0.026
$w'_{2,j}$	0.035	0.067	0.022	0.016
$w'_{3,j}$	0.138	0.116	0.625	0.126
$w'_{4,j}$	0.028	0.310	0.092	0.011
$w'_{5,j}$	0.019	0.160	0.168	0.014
$w'_{6,j}$	0.017	0.069	0.021	0.011
$w'_{7,j}$	0.721	0.093	0.014	0.795

Table 3.1 details a set of 12 interpolated rules recorded during simulation for a given hyper-cube H. The clustering outcomes with $k \in \{2,3,4\}$ are shown in Table 3.2. The result of the highest quality is with $k = 4$, $DI_4 = 0.875$, where the selected, largest cluster is $\{R'_1, R'_4, R'_5, R'_6, R'_9, R'_{10}, R'_{12}\}$, of size 7. The normalised weights are given in Table 3.3, indicating the contribution of the individual terms. Finally, the aggregated rule R^*: $A^*_1 = (2.7, 3.2, 3.7)$, $A^*_2 = (17.6, 18.1, 18.6)$, $A^*_3 = (10.7, 11.2, 11.7)$, and $B^* = (7.8, 8.3, 8.8)$ with $rep(B^*) = 8.330$ can be obtained. This new rule is promoted and put into the original rule base $\mathbb{R}$, while the 7 rules which are now subsumed by it are removed from $\mathbb{R}'$.

If the antecedent values A^*_1, A^*_2, A^*_3 are treated as a new observation, the consequent $B_\circ = (7.5, 8, 8.5)$ may be obtained using the standard interpolation, with $rep(B_\circ) = 8.013$. A ground truth of $y = 8.243$ may be computed with the de-fuzzified antecedent values. Although showing an $\epsilon = 0.316$ difference to the expected interpolative result, the promoted rule is actually a fair amount closer to the underlying ground truth value (by 0.143). Note that this better result is obtained without performing FRI but through standard fuzzy rule firing, once the rule involved has been promoted into the rule base, thereby saving significant computation effort that FRI would otherwise require.

3.4.2 Sparse Rule Base Fulfilment

In this simulation-based evaluation, the representative values of the consequent of the dynamically promoted rules are recorded. There are then compared against the results of conventional interpolation (ϵ_{dvi}), and against the ground truths calculated using the base function (ϵ_{dvt}). The differences between conventional interpolation and the ground truths (e_{ivt}) are also provided. The percentage error $\epsilon_{\%} = \epsilon / range_y$ is calculated relative to the range of the consequent variable. Since stochastic elements are present in the initial rule generation, as well as within the clustering procedure, the dynamic process is repeated five times for each set of the parameter values. Table 3.4 shows the averaged $\epsilon_{\%}$ and the standard deviations.

According to the simulation, the proposed approach delivers more accurate rules with derived consequent being values closer to the ground truth, when compared to the case of conventional interpolation. It implies that these promoted rules, once added to the rule base, would not only avoid the need of future interpolations of similar observations, but also improve the inference accuracy overall. For this

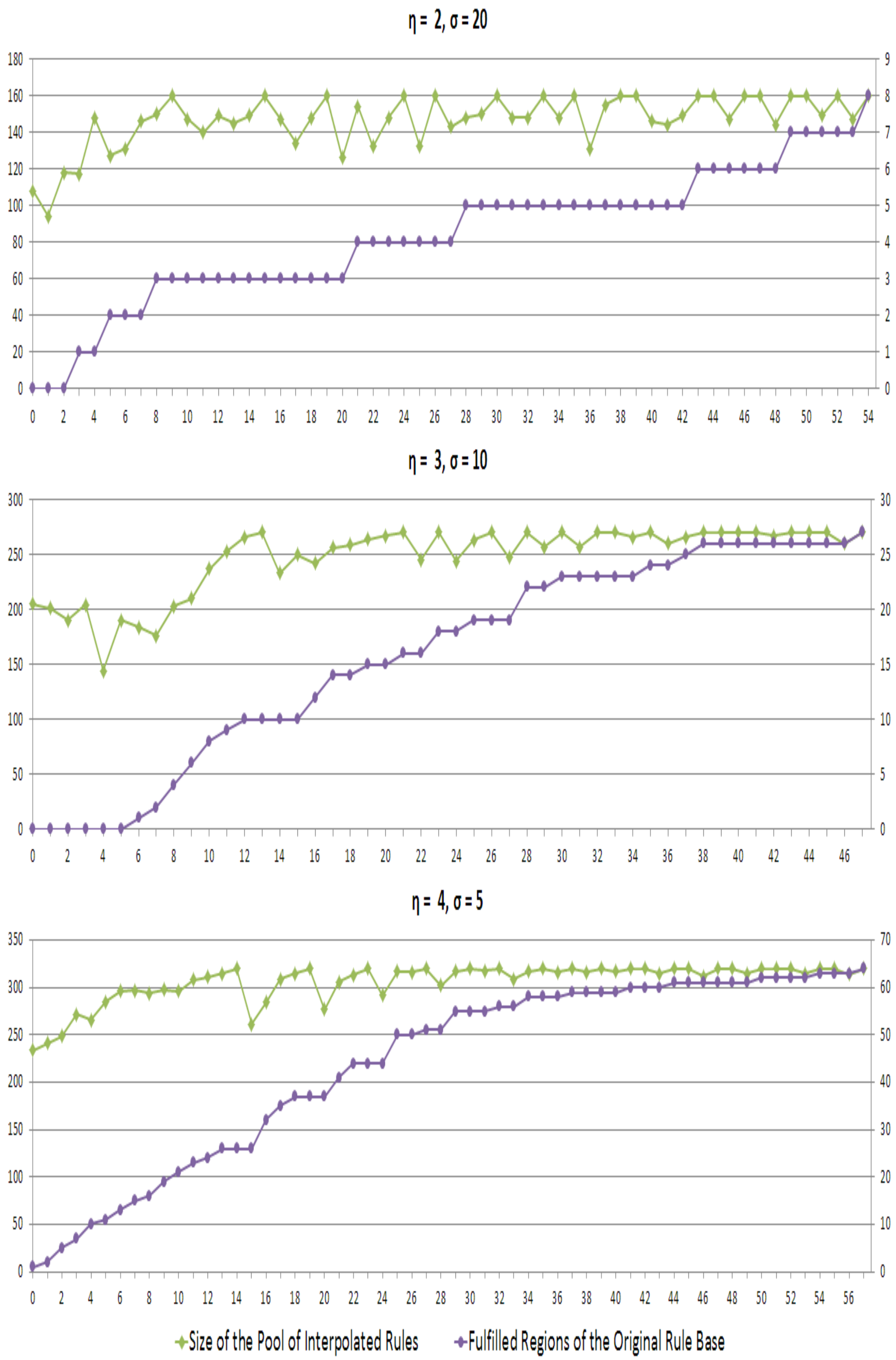


Figure 3.3: Iterative K-Means Clustering Based Dynamic FRI Results

Table 3.4: K-Means Clustering Results for Different Intervals

	$\lvert\mathbb{B}\rvert = 8$			$\lvert\mathbb{B}\rvert = 27$			$\lvert\mathbb{B}\rvert = 64$		
	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$
AVG	3.73	3.97	4.27	2.89	**3.58**	4.09	**2.88**	3.95	4.55
SD	3.34	3.65	4.37	**2.97**	**3.47**	3.99	3.23	4.22	4.51

example problem, the best parameter configuration is $H = 3$, $\sigma = 10$, which produces most accurate and stable rules. For the configuration of $H = 4, \sigma = 5$, the promoted rules are closer to the interpolative outcomes, but further from the ground truth.

Figure 3.3 illustrates graphically the number of fulfilled regions of $\mathbb{R}$, when the proposed dynamic FRI algorithm is performed continuously. Here, the same partitioning process is carried out on the original (sparse) rule base, which acts as a preliminary yet compatible way of measuring rule base coverage. During the simulation, the dynamic process is disabled for regions with sufficient coverage, in order to avoid excessive generation. The plots show the size of the pool of interpolated rules $\mathbb{R}'$, and the number of fulfilled hyper-cubes of $\mathbb{R}$, with regard to the number of iterations. The value of $\lvert\mathbb{R}'\rvert$ varies throughout the whole process as rules involved in promotion are constantly removed, while new interpolated rules are recorded. The coverage improves gradually over time as new rules are promoted and added to $\mathbb{R}$.

3.5 Summary

This chapter has presented an initial attempt towards building an intelligent framework for dynamic fuzzy rule interpolation (D-FRI), where the by-products of performing interpolation: the interpolated rules are analysed, aggregated, and promoted into the original sparse rule base. According to the simulated example, the accuracy of the these promoted rules outperform that of conventional FRI, the cluster-based weighted aggregation also produces more robust rules that are close estimates of the group truth. These potential rules would be very useful in producing an effective dynamic sparse rule base. Eventually, the resultant system may gradually relax the need of FRI while maintaining an efficient yet accurate set of rules.

Although promising, the current approach assumes the availability of an initial partition of the antecedent domains. Clearly, an intelligent way of configuring the initial domain partitioning is an essential part of this dynamic framework. Equally crucial is a strategy of adjusting the size of hyper-cubes at run-time according to the current state of the rule base. For this, techniques developed for link-based [24], grid-based or model-based [75] clustering may prove beneficial. Additionally, several state-of-the-art aggregation methods [24] may further improve the quality of the promoted rules. Although the T-FRI approach is employed in the current implementation to perform interpolation, the flexibility of the proposed framework may also allow the use of more general, similarity-based calculations [27, 63], which would support different choices of similarity measures. Alternative distance metrics such as the Hausdorff distance measure [41] may also be used, which has shown promising results for finding the closest rules in case of FRI [150]. The effect of the dynamically promoted rules on the overall efficacy of practical FRI systems remains active research, especially in terms of the reduction in processing time, as well as the potential improvement in inference accuracy.

Chapter 4

Genetic Algorithm-Aided Dynamic Fuzzy Rule Interpolation

In the previous chapter, a dynamic fuzzy rule interpolation approach [148] has been introduced to better exploit the interpolation results provided by a given FRI method. However, this approach relies heavily on the use of the standard k-means clustering algorithm. Yet, for many application problems, it is difficult to predict the value of k (the number of clusters) [134]. This GA-aided D-FRI improves upon the original approach, by employing a genetic algorithm (GA) based clustering technique [40, 68, 128] in place of k-means clustering. In this work, the collection of the interpolated rules is pre-partitioned into hyper-cubes (multi-dimensional blocks), in order to reduce the search complexity of the GA process. The non-empty hyper-cubes are then identified and used as the input for the GA. After a certain number of generations, the GA identifies a "best" chromosome (cluster arrangement) based on a given fitness function such as the Dunn Index [58]. Here, a chromosome is viewed as a combination of strong and weak clusters, where the weak clusters are merged into the closest strong clusters in order to obtain the final result. In the end, the densest clusters that have accumulated a sufficient number of candidate rules are selected for rule aggregation and promotion.

The remainder of this chapter is organised as follows. Section 4.1 introduces the theoretical underpinnings of GAs that is used in the current implementation of the GA-based dynamic FRI. Section 4.2 illustrates the proposed GA-based dynamic

FRI method. Section 4.3 provides experimentation results that demonstrates the procedures of the proposed approach, and verifies its correctness and accuracy by comparing its outputs to those of conventional FRI. Finally, Section 4.4 summarises chapter.

4.1 Genetic Algorithms

Genetic Algorithms (GAs) are a class of stochastic search and optimization procedures that are inspired by the Darwinian principle of survival of the fittest individuals and natural selection [79]. GAs can find closest optimal solutions in complex search spaces depending on their defined parameters. GAs differ from classical optimisation and search methods in the following ways [71]:

- GAs work with a coding of the parameter set, not the parameters themselves.
- GAs search from a population of points, not a single point.
- GAs use fitness function, not derivatives or other auxiliary knowledge.
- GAs use probabilistic transition rules, not deterministic rules.

Genetic algorithms can be used to encode a possible solution to a given problem using a chromosome-like data structure, applying recombination operators to these structures in order to exploit vital information. Generally, a random population of chromosomes is generated at the beginning of the GA implementation. These chromosomal structures are then evaluated by allocating reproductive opportunities to those chromosomal structures which better fit the solution domain, whist discouraging similar opportunities with poorer solutions. The quality of a solution is naturally defined with respect to the present population [192].

Their operation is dependent on two important operators: crossover and mutation. The population (the set of chromosomes) is initially generated randomly and their members are then selected for reproductive process with respect to their fitness values. The chromosomes with higher fitness values have better chances to reproduce. The reproductive process is repeated until desired conditions are met, such as a desired fitness level, or a maximum number of generations. The generic procedure of GAs is shown in Figure 4.1 and it can also be summarised as follows [40, 128]:

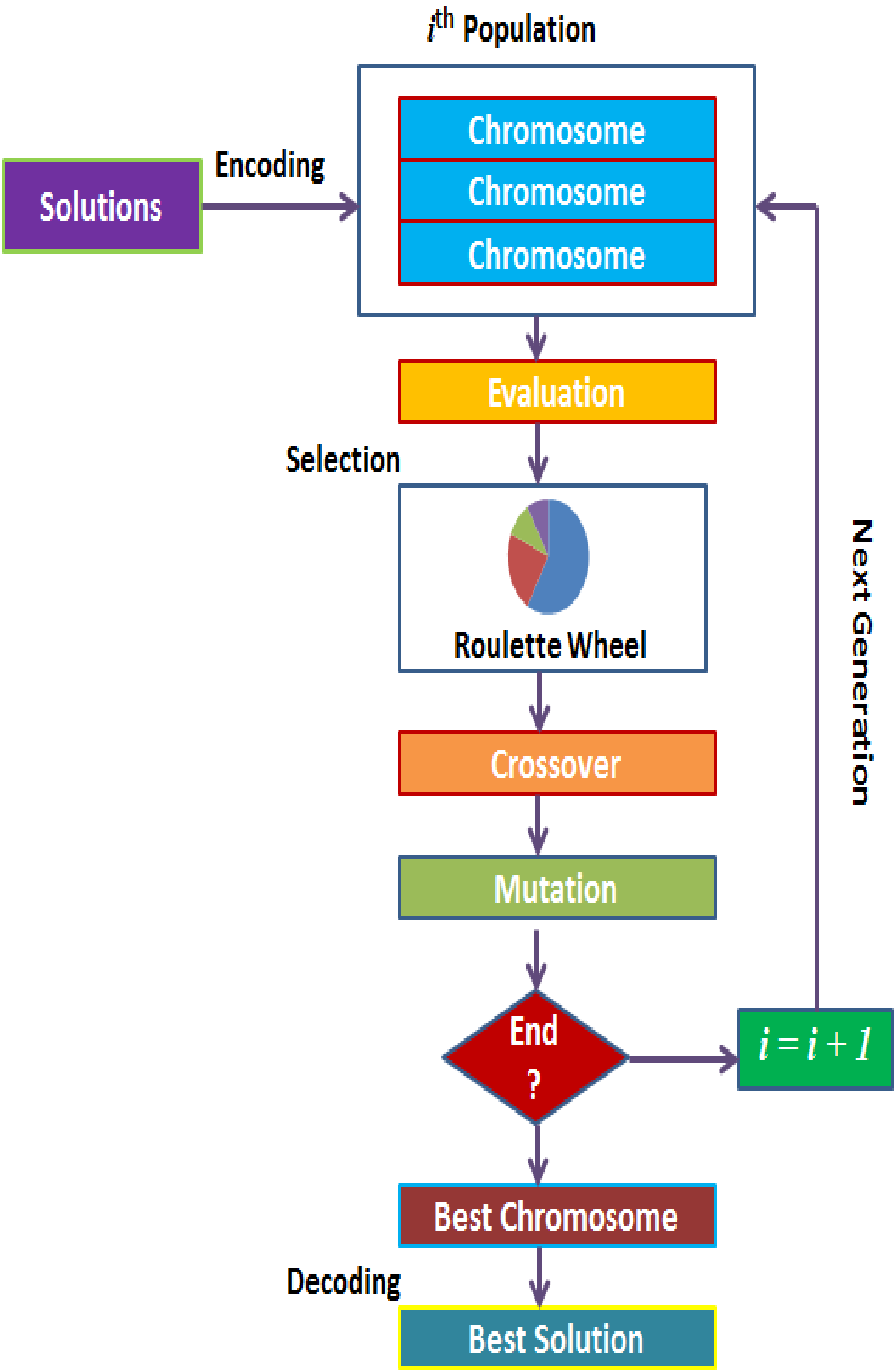


Figure 4.1: Genetic Algorithm flow chart

- *I*nitialisation: Generate random population $\mathbb{P}$ of $|\mathbb{P}|$ chromosomes $[X_1, X_2, ..., X_{|\mathbb{P}|}]$, where each chromosome X_i is an order collection of genes $= [x_1^i, \ldots, x_r^i, x_{r+1}^i, \ldots, x_{|X_i|}^i]$.

- *F*itness Calculation: Evaluate the fitness $f(X_i)$ of each chromosome X_i in the population $\mathbb{P}$, $\forall i \in \{1, ..., |\mathbb{P}|\}$.

- *C*hromosome Selection: Select two parent chromosomes X_p and X_q from a population $\mathbb{P}$ according to their fitness (the better fitness, the bigger chance to be selected).

- *C*rossover: With a crossover rate δ_c, cross over the parents X_p and X_q to form new offsprings ($X_p^{'}$ and $X_q^{'}$). If no crossover was performed, the offsprings are exact copies of their parents.

- *M*utation: With a mutation rate δ_m, mutate the offsprings ($X_p^{'}$ and $X_q^{'}$) at each locus (position in chromosome).

- *A*cceptance: The new offsprings ($X_p^{'}$ and $X_q^{'}$) then together form the new population $\mathbb{P}_{new}$, and are used for the subsequent generation.

- *R*epeat: If the given condition is not satisfied, repeat the process from the step-*F*itness Calculation.

- *T*ermination: If the termination condition is satisfied, stop, and return the best chromosome X^{best} in the final population.

4.2 Dynamic Fuzzy Rule Interpolation (D-FRI) Approach

This section describes the proposed GA-based dynamic FRI, the overall process of its working is presented in Figure 4.2. Generally speaking, there initially exists a set of original (sparse) rules $\mathbb{R}$. While running the FRI system, an interpolation mechanism such as T-FRI gradually fills a pool of interpolated rules $\mathbb{R}'$. The domains of those antecedents appearing in $\mathbb{R}'$ are partitioned into a set of hyper-cubes $\mathbb{H}$. These hyper-cubes are examined to find all non-empty blocks $\mathbb{H}^*$, so that the GA-based clustering algorithm can be employed to find the "best" clustering arrangement leading to a set of strong hyper-cubes $\mathbb{H}^1$ and another of weak hyper-cubes $\mathbb{H}^0$. The strong hyper-cubes are candidate cluster centres in the final clustering outcome. The weak ones are the hyper-cubes that have much less concentration of rules, which are merged into the strong hyper-cubes in order to form the final arrangement. Using GA-based clustering allows the best clusters to be determined without the need to pre-specify the number of clusters k, which is otherwise required by the standard k-means clustering method [134]. After the clustering process, the clusters that have accumulated a sufficient number of interpolated rules (say, more than a certain threshold σ) are selected. Finally, an aggregation process is applied to those selected clusters, in order to construct and promote new rules to become members of the rule base $\mathbb{R}$.

This approach is intuitive and no restriction is imposed over the use of any specific FRI method. The main benefit is to greatly reduce the overheads of interpolating similar, commonly observed values once similar cases have been dealt with, so that only straightforward application of the compositional rule of inference is needed to be carried out.

The following details the key procedures involved in this approach, including antecedent partitioning, interpolated rule clustering and rule promotion. In this work, without losing generality the distance $d(R_p, R_q)$ between two rules R_p ans R_q is defined by:

$$d(R_p, R_q) = \sqrt{\sum_{1}^{N} \frac{(rep(A_{p,i}) - rep(A_{q,i}))^2}{range_{x_i}}} \tag{4.1}$$

4.2.1 Collecting Pool of Interpolated Rules

In the beginning, there exists a sparse rule base R which is developed by the experts or derived from the data. For every given observation, the transformation based FRI (T-FRI) method [85, 86] is run to infer the conclusion. The interpolated conclusion for the given observation is used to perform the appropriate task within the FRI reasoning system. Such interpolated results are also stored to form a pool of interpolated rules $\mathbb{R}'$ for future use. While running the FRI reasoning system, the T-FRI method continuously fills in this pool. After a desired number of interpolated rules in the pool are accumulated, these interpolated results as by-product of previous FRI are then used for further processing to modify the sparse rule base.

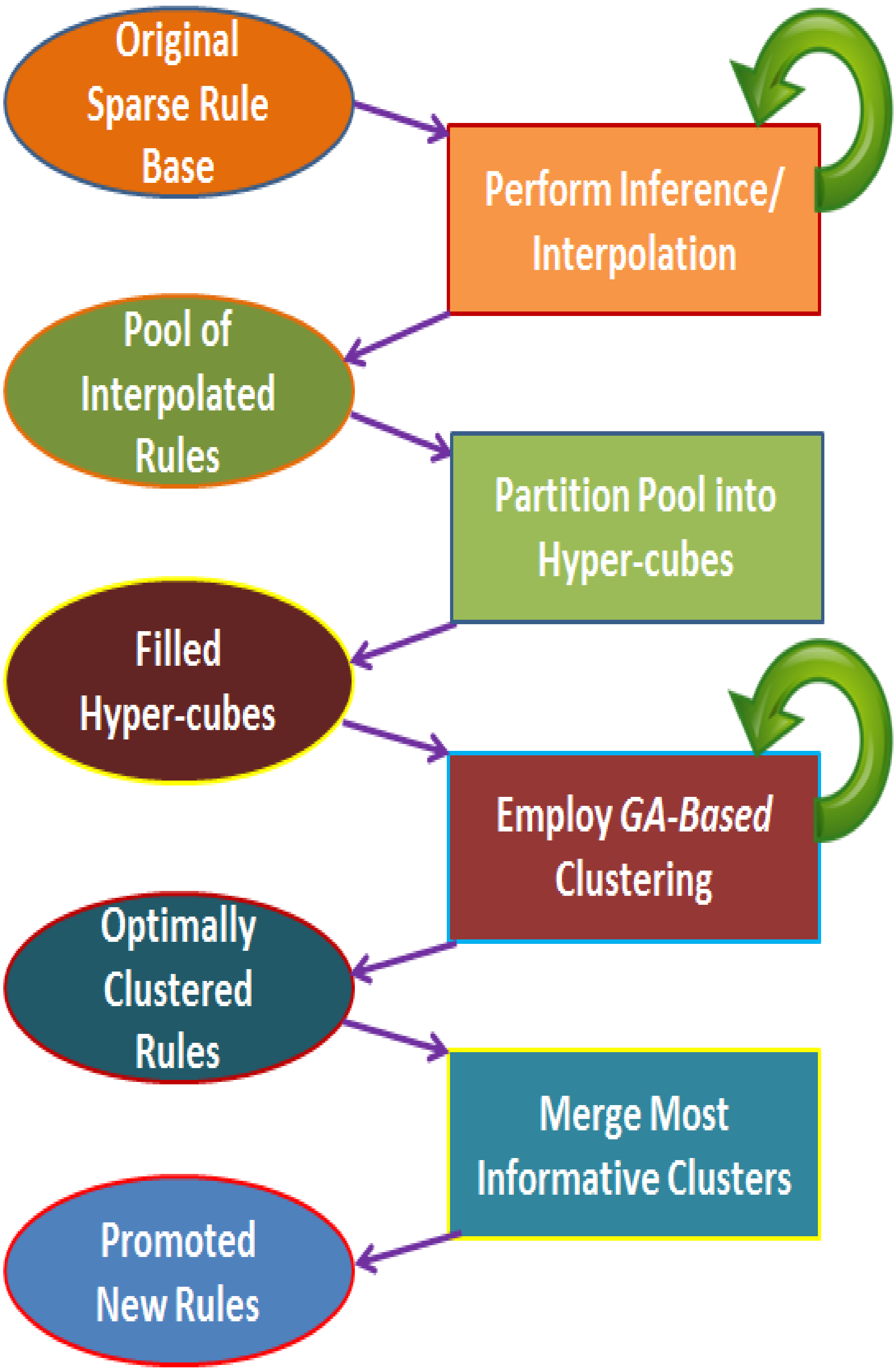


Figure 4.2: Procedure of GA-Aided Dynamic FRI

4.2.2 Partitioning of Input Space

A grid-based partitioning method is used to identify the uncovered regions of $\mathbb{R}$ and the most frequently interpolated areas that are covered by $\mathbb{R}'$. The antecedent domain is partitioned into a set of hyper-cubes $\mathbb{H}$, by dividing the value ranges of the antecedent variables. A given rule R, which may be an original rule R_k or an interpolated rule $R_k^{'}$ is then assigned to the hyper-cube H_p by checking whether its antecedent values lie within the boundaries of H_p:

$$R \in H_p \text{ if } rep(A_{k,j}) \in [\min H_{p,j}, \max H_{p,j}), j \in \{1, \ldots, N\} \tag{4.2}$$

where $A_{k,j}$ is the value of the j^{th} antecedent of the rule R.

Ideally, the total number of hyper-cubes and their sizes should be dynamically adjusted according to the current state of the (increasingly less) sparse rule base. However, for simplicity, the pre-determined partitions are considered in the current implementation, where the input dimensions are evenly divided into η intervals. The total number of hyper-cubes $|\mathbb{H}|$ is therefore η^N. Whilst all the hyper-cubes are checked, only the non-empty hyper-cubes $\mathbb{H}^*$ are to be used for the later clustering process.

$$\mathbb{H}^* \subseteq \mathbb{H}, \forall H \in \mathbb{H}^*, |H| \neq 0 \tag{4.3}$$

4.2.3 Clustering of Interpolated Rules

A modified genetic algorithm as given in Algorithm 4.2.1 is used for clustering, which groups similar interpolated rules $R' \in H$ together, forming the clusters. In this algorithm, each execution of the statement "$if\,(r < \delta_c)\, then$" generates a new random number r, independently of the previous r. In this work, the customization and implementation of the GA is specified as follows:

4.2.3.1 Chromosome and Population Representation

The encoding of the chromosome and its parameters in a GA are dependent on the specific problem. In general, the coding may be of one of the following types: binary encoding, permutation encoding, real-value encoding and tree encoding [71]. In this work the binary coding is adopted for simplicity and each chromosome is represented as a sequence of 0s and 1s as illustrated in Fig. 4.3. Here, a gene of 0 represents a weak cluster at that position, implying a possible absence of a good cluster, and a gene of 1 represents a strong cluster, or a potential presence of a good cluster.

$\mathbb{P}_{new}$, new population
$X_i^{'}$, i^{th} chromosome of $\mathbb{P}_{new}$
$X^{best}, X^{best} \in \mathbb{P}, f(X^{best}) = \max_{\forall X \in \mathbb{P}} f(X)$
$f(X_i)$, fitness value of X_i
δ_c , crossover rate
r , random number, where $0 \leq r \leq 1$
k_{max} , maximum number of generations
while *($k_{max} \neq 0$)* **do**
    $\mathbb{P}_{new} = \emptyset$
    for *($i = 1; i < |\mathbb{P}|; i + 2$)* **do**
        $X_i^{'} = roulettewheelselection(\mathbb{P})$
        $X_{i+1}^{'} = roulettewheelselection(\mathbb{P})$
        if ($r < \delta_c$) **then**
            $X_i^{'} = crossover(X_i^{'}, X_{i+1}^{'}, true)$
            $X_{i+1}^{'} = crossover(X_i^{'}, X_{i+1}^{'}, false)$
        $X_i^{'} = mutate(X_i^{'})$
        $X_{i+1}^{'} = mutate(X_{i+1}^{'})$
        $\mathbb{P}_{new} = \mathbb{P}_{new} + [X_i^{'}, X_{i+1}^{'}]$
    $k_{max} = k_{max} - 1$
    $\mathbb{P} = \mathbb{P}_{new}$
return X^{best}

Algorithm 4.2.1: Genetic Algorithm for Clustering

The above concept of strong and weak clusters helps determine the fitness value of a chromosome, which is computed on the basis of all the possible strong clusters (1s), ignoring all the possible weak clusters (0s) in that particular chromosome. If the group of all possible strong clusters within a chromosome yields a very high fitness value a good cluster arrangement is indicated. This also confirms the validity of possible strong clusters as the real clusters. Following this, the weak clusters (0s) are merged to their relevant closest strong clusters (1s) to obtain the final arrangement of quality clusters.

The length of chromosome $|X|$ is set to the total number of non-empty blocks/ hyper-cubes $|\mathbb{H}^*|$. The intuition behind this setting is to cover all interpolated rules as a part of the blocks in the clustering process so that the best clustering arrangement can be achieved. This means that every non-empty block is represented by one gene within the chromosome which may be one of the possible future clusters. If the block

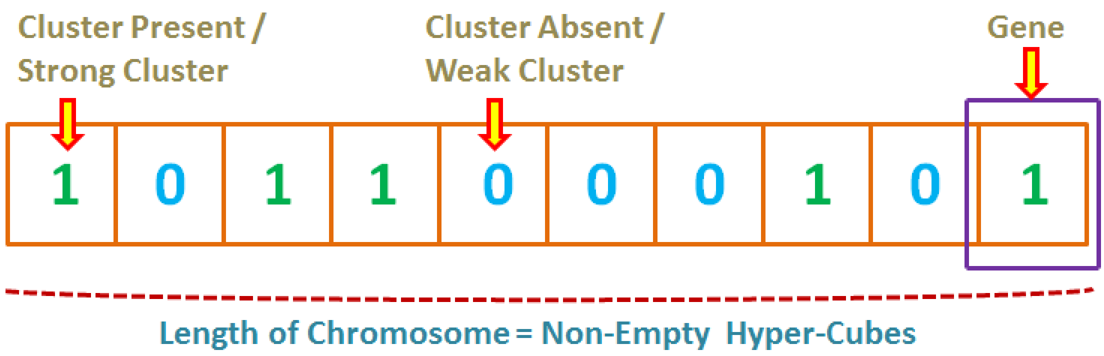


Figure 4.3: Chromosome Representation in GA-based Dynamic FRI

is empty then obviously it has no rules in it and it is quite logical to ignore such blocks in the clustering process in order to avoid unnecessary computation.

The initial population $\mathbb{P}\ [X_1, X_2, ..., X_{|\mathbb{P}|}]$ is generated randomly, to start the GA search process, where the size of the population $|\mathbb{P}|$ is adjusted in relation to the number of non-empty hyper-cubes $\mathbb{H}^*$. In the GA literature, a population between 20 to 30 chromosomes is typically employed in implementation though a larger population may be utilised [71, 207]. Being a preliminary investigation, a fixed population size (20) is adopted herein.

4.2.3.2 Fitness Calculation

The fitness function is a problem-dependent parameter in GAs, which decides on the quality of individual chromosomes. In this work, a chromosome represents a potential cluster arrangement, and the Dunn Index (DI) [58] is utilised to assess its quality on the basis of cluster isolation and compactness. A higher value of DI indicates a more favourable result:

$$f(X_i) = \min_{p,q \in \{1,...,i\}, p \neq q} \{ \frac{m_{pq}}{\max_{r \in \{1,...,i\}} s_r} \} \tag{4.4}$$

where s_r and m_{pq} are the intra-cluster (compactness) and inter-cluster (isolation) distance measurements, respectively:

$$s_r = \sqrt{\sum_{R' \in C_r} \frac{d(R', \mu_r)^2}{|C_r|}}, \ m_{pq} = d(\mu_p, \mu_q) \tag{4.5}$$

In the above, C_r is the r^{th} cluster, the distance between a given interpolated rule R' and the centroid μ_q of a cluster C_q is calculated in a way similar to Eq. 4.1 should that:

$$\forall R'_j, R'_k \in \mathbb{R}', d(R'_j, \mu_q) = d(R'_k, \mu_q) \tag{4.6}$$

where

$$d(R', \mu_q) = \sqrt{\sum_{1}^{N} (rep(A'_i) - \mu_{q,i})^2}, R' \in \mathbb{R}' \tag{4.7}$$

4.2.3.3 Selection, Crossover and Mutation

Based on the fitness values, parent chromosomes are selected to generate offsprings in the next population using the roulette wheel selection algorithm [121], as summarised in Algorithm 4.2.2. In roulette wheel selection, each chromosome is assigned a segment of roulette wheel, with a size proportional to its fitness value. Naturally, the bigger the fitness value is, the larger the segment will be.

$\mathbb{P} = [X_1, ..., X_{|\mathbb{P}|}]$, population
X_i, i^{th} chromosome of population $\mathbb{P}$
$f(X_i)$, fitness value of X_i
r , random number, where $0 \leq r \leq 1$
$threshold = r \times \sum_{i=1}^{|\mathbb{P}|} f(X_i)$
for $\forall i \in \{1, \ldots, |\mathbb{P}|\}$ **do**
    if $(threshold > 0)$ **then**
        $threshold = threshold - f(X_i)$
    else
        return X_i

Algorithm 4.2.2: Roulettewheelselection($\mathbb{P}$)

Crossover and mutation control the generation of offsprings. Crossover process exchanges information between two parent chromosomes while generating the two offsprings and it is outlined in Algorithm 4.2.3. The rate of crossover δ_c is generally high at about $70\% - 80\%$ [207]. The mutation operation tries to avoid premature convergence and explores potential alternative solution regions and it is outlined in Algorithm 4.2.4. However, high mutation rate δ_m has a negative impact on the search ability of the GA and therefore, is set to a very low value [207].

4.2.3.4 Termination

The entire reproductive process is repeated until the maximum number of generations $k_{\max}$ is reached. When the GA terminates, the best chromosome X^{best} of the final population is treated as the search outcome.

$X_i = [x^i_1, \ldots, x^i_r, x^i_{r+1}, \ldots, x^i_{|X|}]$
x^i_r, r^{th} gene of i^{th} chromosome X_i
r , random integer, where $1 \leq r \leq |X|$
if $(left = true)$ **then**
  return $[x^i_1, \ldots, x^i_r] + [x^{i+1}_{r+1}, \ldots, x^{i+1}_{|X'_{i+1}|}]$
else
  return $[x^{i+1}_1, \ldots, x^{i+1}_r] + [x^i_{r+1}, \ldots, x^i_{|X'_i|}]$

Algorithm 4.2.3: Crossover(X'_i,X'_{i+1}, $left$)

r , random number, where $0 \leq r \leq 1$
δ_m , mutation rate
x^i_j, j^{th} gene of i^{th} chromosome X_i
for $\forall j \in \{1, \ldots, |X'_i|\}$ **do**
  if $(r < \delta_m)$ **then**
    $x^i_j = \neg x^i_j$
return X'_i

Algorithm 4.2.4: Mutate(X'_i)

4.2.3.5 Cluster/Hyber-cube Merging and Filtering

As previously explained, the "best" chromosome indicates the best clustering strategy determined by the GA. It shows whether a given hyper-cube is to be assigned as a candidate cluster centre (a strong hyper-cube $H^1 \in \mathbb{H}^1$), with which one or more weak hyper-cubes $H^0 \in \mathbb{H}^0$ may be merged subsequently. This arrangement is awarded with the highest fitness value (as judged by metrics such as the Dunn Index shown in Eq. 4.4) over the search process concerned, thereby forming the final clustering outcome. The process which combines the strong and weak hyper-cubes/clusters is outlined in Algorithm 4.2.5. A selection process is then carried out in order to choose one or more clusters of rules as the candidates for rule promotion. This process may be implemented as picking the clusters which contain more than σ rules, and in case of a tie, the most compact (see Eq. 4.5) clusters will be selected.

4.2.4 Rule Promotion

This merging process provides the highest quality clustering arrangement which is similar to the highest quality cluster arrangement (k) of the *k-means* clustering based D-DRI approach, obtained by the Dunn Index [58]. From this cluster arrangement,

$H_i^0 \in \mathbb{H}^0$, i^{th} weak hyper-cube
$H^1_{H_i^0} \in \mathbb{H}^1$, the closest strong hyper-cube to H_i^0
$\mu_{H_i^0}$, centroid of hyper-cube H_i^0
for $\forall H_i^0 \in \mathbb{H}^0, i \in \{1, ..., |\mathbb{H}^0|\}$ **do**
    find $H^1_{H_i^0} = \arg\min_{H^1 \in \mathbb{H}^1} |\mu_{H^1} - \mu_{H_i^0}|$
    $H^1_{H_i^0} = H^1_{H_i^0} \cup H_i^0$
return $\mathbb{H}^1$

Algorithm 4.2.5: Merge($\mathbb{H}^1$, $\mathbb{H}^0$)

only those clusters that satisfy the threshold condition σ are selected for rule promotion process. Again, the total number of newly promoted rules can also be controlled by further selecting the most compact (see Eq. 4.5) clusters. For every selected cluster a single rule is generated where all the informative rules $R' \in C_q \subseteq H^*$ are taken for further generalisation in an effort to form a new, aggregated rule, which is hereafter referred to as R^*.

This work adopts a weighted combination method, using the cluster centroid μ_q to compute the contributions from the individual candidate rules. Similar to the process of constructing intermediate rules as described in the T-FRI approach [85, 86], a matrix w_{ij} of dimension $(|C_q|\ , N+1)$ is used. It indicates the weight of A'_{ij} of an interpolated rule $R'_i \in C_q$ regarding the jth antecedent A^*_j of R^*:

$$w_{i,j} = \frac{1}{d(A'_{i,j}, \mu_{q,j})}, i \in \{1, \ldots, |C_q|\}, j \in \{1, \ldots, N\} \tag{4.8}$$

and that of B'_i to B^*:

$$w_{i,N+1} = \frac{1}{d(B'_i, \mu_{q,N+1})} \tag{4.9}$$

The normalised weights can also be obtained:

$$w'_{i,j} = \frac{w_{i,j}}{\sum_{i=1}^{|C_q|} w_{i,j}} \tag{4.10}$$

From this, the components of the dynamically promoted new rule R^* is constructed as follows:

$$A^*_j = \sum_{i=1}^{|C_q|} w'_{i,j} A'_{i,j},\ j \in \{1, \ldots, N\}, B^* = \sum_{i=1}^{|C_q|} w'_{i,N+1} B'_i$$

This newly promoted R^* is then added to the original (sparse) rule base such that $\mathbb{R} := \mathbb{R} \cup \{R^*\}$, while the rules involved in the aggregation process are removed from the pool of interpolated rules: $\mathbb{R}' := \mathbb{R}' \setminus C_q$. This partitioning-clustering-promotion procedure is applied for all hyper-cubes satisfying $|H_p^*| \geq \sigma$. The entire dynamic FRI process may repeat until the original rule base reaches a state with sufficient coverage of the problem domain. The resultant, complete algorithm for dynamic interpolation supported by a GA is given in Algorithm 4.2.6.

$\mathbb{R}$, original sparse rule base
$\mathbb{R}'$, interpolated rule base
R^*, dynamically generated new rule
$\mathbb{H}$, all partitioned hyper-cubes
$\mathbb{H}^* = \mathbb{H}^1 \cup \mathbb{H}^0$, set of non-empty hyper-cubes
$\mathbb{H}^1$, set of strong hyper-cubes
$\mathbb{H}^0$, set of weak hyper-cubes
$\mathbb{C}$, set of clusters
C_i, i^{th} cluster of $\mathbb{C}$
σ , threshold for promoting new rules
$\mathbb{H} = partition(\mathbb{R}')$
$\mathbb{H}^* = \{H | H \in \mathbb{H}, |H| \neq 0\}$
$\mathbb{H}^1 = GA(\mathbb{H}^*)$
$\mathbb{C} = merge(\mathbb{H}^1, \mathbb{H}^0)$
for $\forall C_i \in \mathbb{C}$ **do**
  if $|C_i| > \sigma$ **then**
    $R^* = aggregate(C_i)$
    $\mathbb{R} = \mathbb{R} \cup \{R^*\}$
    $\mathbb{R}' = \mathbb{R}' \setminus C_i$

Algorithm 4.2.6: GA-based Dynamic Interpolation($\mathbb{R}$, $\mathbb{R}'$,σ)

4.2.5 Achieving Dynamism

In the FRI reasoning system, for every new observation an interpolated conclusion, after a certain time period, when the desired number of interpolated rules is reached as indicated above, the D-FRI procedure is performed to update the sparse rule base. This is an on-going process and executed as and when required to update the sparse rule base. Where the newly promoted rule is added to the original sparse rule base the least used rules which are close to any of the newly introduced ones can be deleted from the rule base to maintain its concurrency. This is a dynamic process in which new observations are used to achieve interpolated results which are then added to the existing pool of interpolated rules $\mathbb{R}'$ and regrouped into new clusters. The resulting best aggregated clusters are promoted when appropriate into the original sparse rule base. This self-organising process runs iteratively to provide the dynamic sparse rule base and consequently, an adaptive FRI reasoning system.

4.2.6 Complexity Analysis

The proposed dynamic approach can be decomposed into three core parts: rule base partitioning, GA-based clustering, and rule promotion. The complexity of the rule base partitioning procedure is shown in Eq. 4.11, which depends on the number of rules in the interpolated rule base $|\mathbb{R}^{'}|$, the number of rule antecedents N, and the number of partition intervals η:

$$O_{\text{partition}} = O(|\mathbb{R}^{'}|N\eta) \tag{4.11}$$

The complexity of the GA-based clustering operation given in Eq. 4.12 is affected by the maximum number of generations k_{max}, the size of the population $|\mathbb{P}|$, and the complexity of the fitness evaluation O_{fitness}. Additional factors such as the use of genetic operators [207, 121] also play a role, but their impact varies depending on their actual implementations. Thus,

$$O_{\text{ga}} = O(|\mathbb{P}|k_{\text{max}}) \cdot O_{\text{fitness}} \tag{4.12}$$

where

$$O_{\text{fitness}} = O(|\mathbb{H}| + |\mathbb{R}^{'}|^2 + \frac{|\mathbb{R}^{'}|^2}{|\mathbb{H}|^2} + \mathbb{H}^2) \tag{4.13}$$

The fitness complexity O_{fitness} combines the cost of chromosome transformation: $O(|\mathbb{H}|)$, the cost of the hyper-cube merging process: $O(|\mathbb{R}^{'}|^2)$, and finally the complexity of the Dunn Index calculation, which is based on both intra- and inter-cluster distance calculations. The complexity of rule promotion depends on the number of clusters $|\mathbb{C}|$ (derived from the output of GA), the number of interpolated rules $|\mathbb{R}^{'}|$, and the number of antecedent dimensions N, such that

$$O_{\text{promotion}} = O(\frac{|\mathbb{R}^{'}|N}{|\mathbb{C}|}) \tag{4.14}$$

The overall complexity of the proposed GA-based dynamic fuzzy rule interpolation approach is therefore the sum of the three above, i.e., $O_{\text{partition}} + O_{\text{ga}} + O_{\text{promotion}}$.

4.3 Experimentation and Discussion

A numerical example is employed to demonstrate the process of the proposed approach, as well as to evaluate its performance. A function of three crisp input variables, shown in Eq. 4.15 is chosen to populate a sparse rule base $\mathbb{R}$ of size 100.

$$y = 1 + \sqrt{x_1} + \frac{1}{x_2} + \frac{1}{\sqrt{x_3^3}}, \; x_1, x_2, x_3 \in [1, 20] \tag{4.15}$$

An initial fuzzy rule is generated by fuzzifying the crisp inputs and their associated function output, where a numerical value a is converted to a fuzzy set A with a support length of 1: $A = (a - 0.5, a, a + 0.5), Rep(A) = a$. This provides a simple non-linear rule base suitable for the purpose of this preliminary investigation. The experiment in this section invokes three different values of η, where the antecedent dimensions are evenly partitioned into $\eta \in \{4, 5, 6\}$ intervals, as a result, $4^3 = 64$, $5^3 = 125$, and $6^3 = 216$, hyper-cubes can be created. The parameters of the GA are already explained with their optimised values previous. In particular, they are set to the following optimised values: crossover rate $\delta_c = 0.7$, mutation rate $\delta_m = 0.05$, population size $|\mathbb{P}| = 20$, and maximum generation $k_{\max} = 100$.

4.3.1 GA-Based Clustering Results

The GA-based clustering algorithm is performed over 500 interpolated rules, where 90, 132 and 167 new rules have been promoted for intervals 4, 5, and 6, respectively. The representative values of the consequent of the dynamically promoted rules are recorded. They are then compared to the results of conventional interpolation ($\epsilon_{\%dvi}$), and to the ground truths calculated using the base function ($\epsilon_{\%dvt}$). The differences between conventional interpolation and the ground truths ($e_{\%ivt}$) are also provided. Here the percentage error $\epsilon_{\%} = \epsilon / range_y$ is calculated relative to the range of the consequent variable. Since stochastic elements are present in the initial rule generation, as well as within the clustering procedure, the GA dynamic process is repeated 50 times for each set of the parameter values. Table 4.1 shows the averaged value $\epsilon_{\%}$ and the standard deviations of $\epsilon_{\%}$.

According to the simulation results, for $\eta = 5$ and 6, the implemented algorithm promotes more accurate rules, with derived consequent values closer to the ground truth, than those obtainable using conventional interpolation. For this example

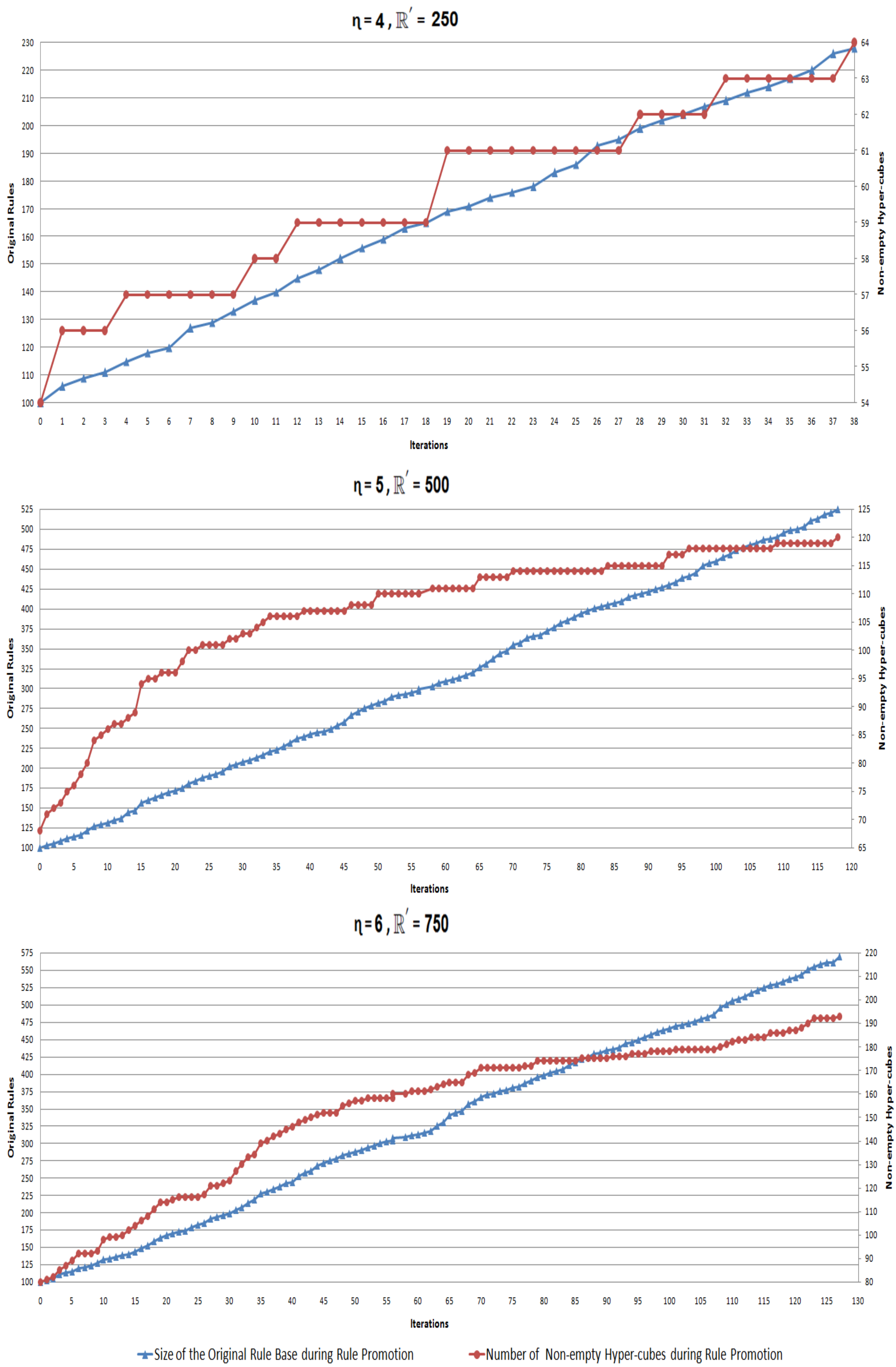


Figure 4.4: Iterative GA Based Dynamic FRI Results

Table 4.1: GA-Based Clustering Results

	$\eta = 4$			$\eta = 5$			$\eta = 6$		
	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$
AVG	2.68	2.24	2.07	2.38	**1.24**	2.45	3.47	**2.06**	3.74
SD	2.77	2.01	3.35	2.70	**1.25**	2.63	3.03	**1.97**	3.73

problem, the best parameter configuration is $\eta = 5$, which produces both accurate and stable rules. For the configuration of $\eta = 6$, the promoted rules are also closer to the ground truth than the outcomes obtained by the conventional T-FRI. These results imply that the rules promoted using intervals $\eta = 5$ and $\eta = 6$, once added to the rule base, will not only avoid the need of future interpolations of similar observations, but also improve the inference accuracy (i.e., the quality of the rule base) overall. Note that large intervals ($\eta = 4$) do not yield good quality rules for this experimental scenario. This is as can be expected, because the size of the individual hyper-cubes are too large to form any meaningful clustering arrangement. The use of the GA also greatly relaxes the needs to specify decent starting conditions, since its stochastic mechanisms are insensitive to the initial states.

4.3.2 Sparse Rule Base Fulfilment

An extended dynamic rule promotion process is also performed for the same intervals $\eta \in \{4, 5, 6\}$ but with a different number of interpolation rules 250, 500, and 750, respectively. The aim is to observe the level of fulfilment of the sparse regions in the rule base, assuming the proposed dynamic process is in its normal operation (i.e., performs interpolation consecutively). Fig. 4.4 illustrates graphically the number of fulfilled regions $\mathbb{H}^*$ in $\mathbb{R}$, and the number of rules $|\mathbb{R}|$ in relative to the number of iterations carried out. Here, the same partitioning process is carried out on the original (sparse) rule base, which acts as a preliminary yet compatible way of measuring rule base coverage. The graphs show the values of $|\mathbb{R}|$ and $|\mathbb{H}^*|$ varying throughout the whole process as rules having been promoted may be subsequently removed as new interpolated rules are recorded in the following iterations.

The coverage improves gradually over time as new rules are promoted and added to $\mathbb{R}$. For the case of $\eta = 4$, all original sparse rule-base regions are filled in 38 iterations with 238 original rules as the final size of the rule base. However, when $\eta = 5$ and $\eta = 6$, the rule base can not fully cover the problem space but is closer to fulfilling all regions. In case of the interval $\eta = 5$, 120 hyper-cubes are filled in 125 hyper-cubes through 118 iterations with 525 as the final size of the rule base. Similarly, in case of the interval $\eta = 6$, 193 hyper-cubes are filled in 216 hyper-cubes through 127 iterations with 570 original rules in the final rule base.

Both sets of experiments (accuracy and fulfilment), once analysed together, help to reach the conclusion that the initial, sparse rule base is gradually refined into a denser rule base. The overall accuracy of the resultant rule base is also improved.

4.3.3 GA-Aided D-FRI vs. *k-means* Clustering Based D-FRI

The GA-aided D-FRI approach is an improvement over *k-means* clustering based D-FRI. The key improvement is to determine the number of clusters automatically which is not possible in *k-means* clustering. Also, in *k-means* clustering, it is difficult to compare the quality of the clusters produced with regard to different criteria (e.g., for different initially set partitions or values of the k) [64, 73, 144]. However, the GA-based clustering is not affected by the initially fixed value of k. Additionally, *k-means* clustering has strong sensitivity to outlier and noise data points because a small number of such data points can affect the mean value significantly [4, 34, 169]; however, GA generally provides a more stable optimization under the noisy and dynamic environments [7, 14, 20, 62, 145, 151].

4.4 Summary

This chapter has presented a GA-based, dynamic fuzzy rule interpolation (D-FRI) approach. The proposed D-FRI employs a GA-based new clustering approach to replace the standard k-means clustering algorithm. This new GA-based clustering approach overcomes the problem of the prediction of the value of k (the number of clusters) in the standard k-means clustering algorithm. While running the dynamic FRI process, the interpolated rules are analysed, selected, aggregated, and promoted when appropriate into the original sparse rule base. According to experimental simulations, the accuracy of using the rule base containing both the original and promoted rules outperform that of using just the original when the conventional T-FRI is employed. Thus, this approach enhances the original sparse rule base dynamically and develops a more effective interpolative reasoning system. It is interesting to note that the resultant system may gradually relax the need of FRI while maintaining an efficient yet accurate reasoning system. This is because the rule base is enriched gradually such that it is no longer sparse and the compositional rule of inference can be applied directly.

An intelligent method for configuring the rule base partitioning remains a vital part of future development. Additionally, the use of state-of-the-art aggregation methods [25, 172] may further improve the quality of the promoted rules. Ideas developed for dynamic rule learning [11, 12, 196] and nature-inspired clustering algorithms [40, 68, 128] may also provide useful insights. While the T-FRI approach is employed in the current implementation to perform interpolation, the flexibility of the proposed approach may allow the use of more general, similarity-based calculations [27, 63], which would support different choices of similarity measures. Although the current focus of the work is on rule promotion (addition), it is also necessary to examine the scenario of dynamic rule base consolidation including the removal of redundant and inconsistent rules, which is an integral component of a truly intelligent and dynamic approach.

Chapter 5

Integration of Inference and Interpolation

SPARSE fuzzy rule-based systems are compact and effective systems. Fuzzy rule interpolation offers a useful mechanism for approximating a conclusion in problems involving a sparse rule base when there is no matched rule available for a given observation. However, all available FRI methods interpolate results at the expense of significant computational overheads [206]. Fortunately, in many cases, the need of interpolation and its complexity may be avoided if a rule exists in the rule base that matches or even partially matches the given observation. This can be implemented by an appropriate, conventional inference mechanism such as CRI. If the given observation does not match any rules then an interpolation method is applied, otherwise it is straightforward to infer the conclusion by the use of conventional rule firing method.

If CRI is not applicable for the given scenario then interpolation or extrapolation can be used depending on the locations of the closest rules. Therefore, fuzzy interpolation or extrapolation requires few (normally, only two) closest rules to infer a result. If these closest rules are not reasonable then interpolated or extrapolated results will be affected adversely. This presents an important constraint over the choice of the distance metric that determines the closeness between an observation and the rule antecedents. This is because elements considered here are represented by membership functions rather than simple n-dimensional points. Many interpolation or extrapolation methods use the common distance metric based on the Centre of Gravity (COG), e.g., [85, 86, 206]. Unfortunately, the centres of gravity may be the same for two extremely different fuzzy sets, as shown in Figures 5.1 and 5.2. It is therefore desirable to investigate the use of a different distance metric. Having noted this, the Hausdorff Distance (HD) and Earth Mover's Distance (EMD) metrics are employed here in order to identify the closest rules for interpolation. Hausdorff distance measures the distance between two membership functions rather than two points. Whereas, EMD is an intuitive and natural way to find the distance in a multi-dimensional environment.

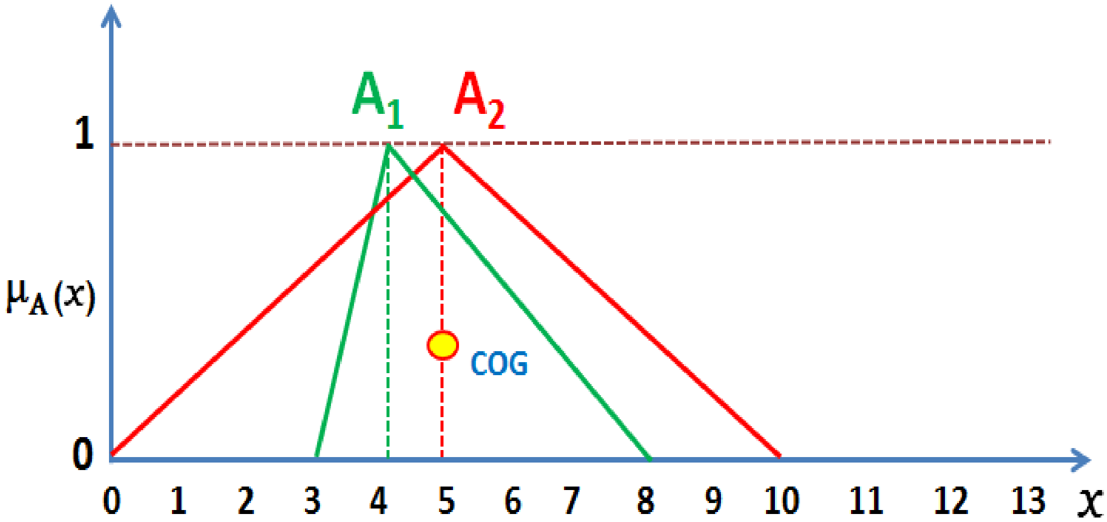


Figure 5.1: Same centre of gravity of two different triangular fuzzy sets

This chapter proposes an approach which integrates fuzzy interpolation (FRI) and inference (CRI) effectively. This approach uses the concept of α-cut overlapping between fuzzy sets to decide an exact or partial matching between rules and observation so CRI would be utilised for reasoning. Otherwise, the closest rules may be selected for FRI by exploiting either the COG, HD or EMD metrics depending on the requirement of the accuracy or complexity of the given application.

The remainder of this chapter is organised as follows. Section 5.1 presents the concept of α-cut overlapping that is used to verify the need of CRI. Section 5.2 illustrates the different distance metrics: COG, HD and EMD. Section 5.3 introduces the proposed integrated approach for CRI and FRI. Section 5.4 provides experimental results that demonstrate how the matching rule may be selected for CRI based on α-cut, and compares the three distance metrics: COG, HD and EMD. Finally, Section 5.5 summarises this chapter.

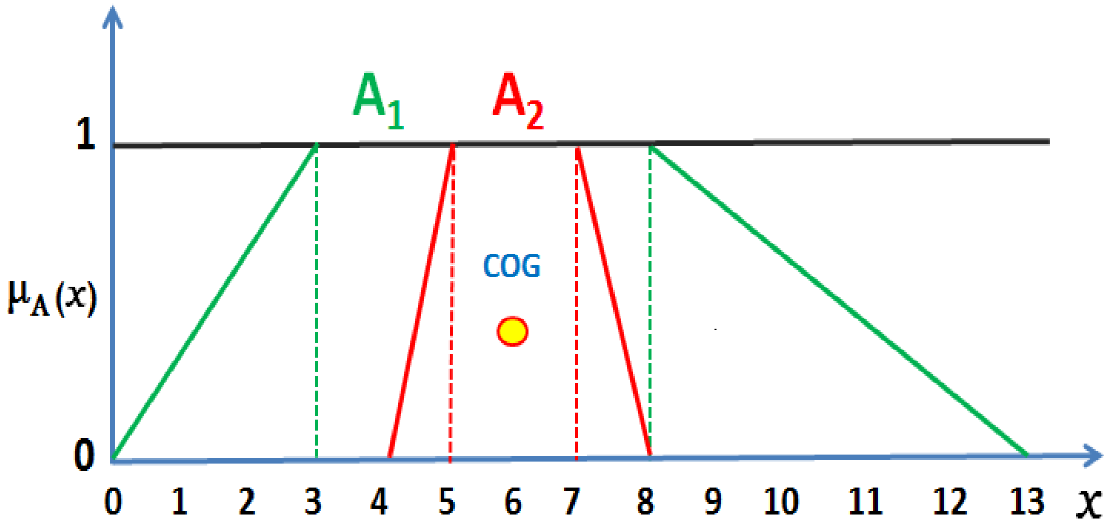


Figure 5.2: Same centre of gravity of two different trapezoidal fuzzy sets

5.1 Alpha-cut Overlapping

An α-cut converts a fuzzy set into a crisp set with respect to a given confidence level α. Formally, let A be a fuzzy set in the universe of discourse X, that is, $A \in F(X)$, $\mu_A(x)$ is the membership function of A and $\alpha \in [0,1]$. Then α-cut of A is [110]:

$$A_\alpha = \{x \in X | \ \mu_A(x) \geq \alpha\} \quad (5.1)$$

For simplicity trapezoidal fuzzy sets are considered in the present work. Also, each rule is having multiple antecedents variable. The concept of α-cuts is shown in Figure 5.3 for two trapezoidal rule antecedent fuzzy sets A_1 and A_2 and one observed trapezoidal fuzzy set $A_\circ$, where inf and sup stand for $infimum$ and $supremum$ operator, respectively. For any given rule R_i given an α-cut threshold α, and the antecedent fuzzy set $A_{i,j} = (a_{i,j,0}, a_{i,j,1}, a_{i,j,2}, a_{i,j,3})$ and an observation $A_{0,j} = (a_{o,j,0}, a_{o,j,1}, a_{o,j,2}, a_{o,j,3})$, where i stands for the index of the rule and j for that of the antecedent appearing in the rule as shown in Figure 5.4, the α-cut based inference method proceeds as follows:

First of all, an observation is compared with the possible antecedent values for α-cut matching based on the given α level. The left extreme point $a_{i,j,0}$ and right extreme point $a_{i,j,3}$ of an antecedent value $A_{i,j}$, $i = 1,2,...,n$, and $j = 1,2,...,N$, is compared with the α-cut infimum point and supremum point of the corresponding observation $A_{\circ,j}$, $j = 1,2,...,N$, to check whether there is any α-cut matching or not. This includes the checking for the special case of containment $((a_{o,j,0} < a_{i,j,0}) \wedge (a_{o,j,3} > a_{i,j,3}))$.

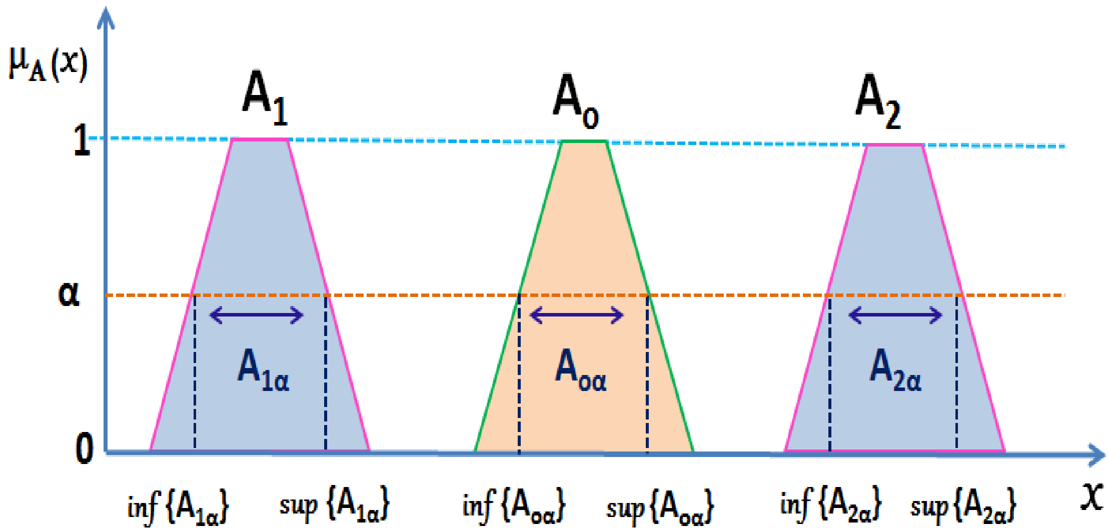


Figure 5.3: α-cut concept in trapezoidal fuzzy sets

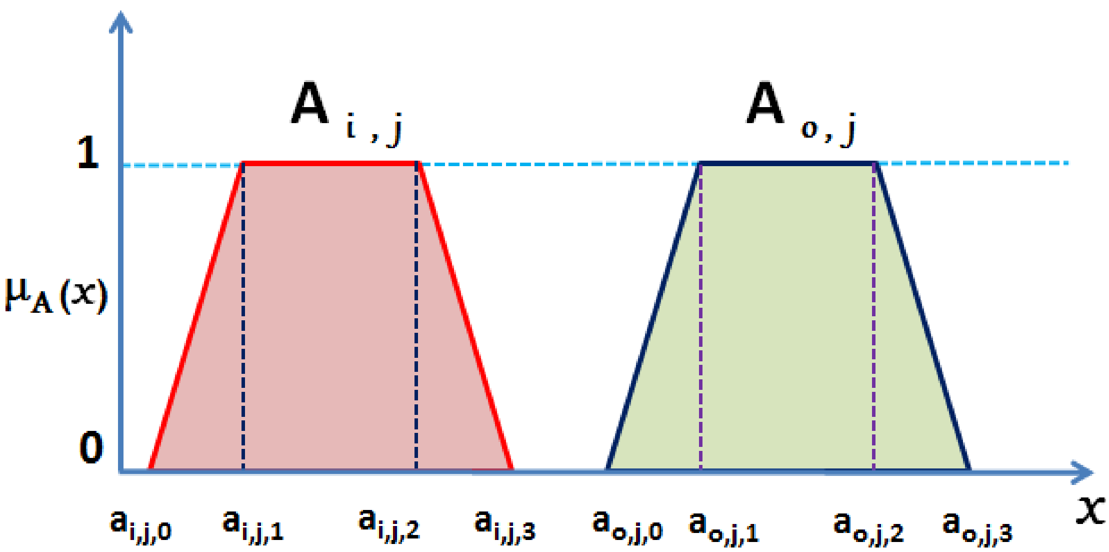


Figure 5.4: Two trapezoidal fuzzy sets' points $a_{i,j}$ and $a_{\circ,j}$

Since the detection of possible overlap rules is only necessary to be run above the α-level, much calculation for rules which overlap with the observation below the α-level is saved. If only one rule is matched with all antecedents above the α-level then the conclusion is inferred on the basis of the matched rule. However, if more than one rule are matched with all antecedents above the threshold then the rule that is of the highest matching degree is selected. The matching degree is computed by finding the sum of all areas of polygons which are formed by the overlap of two partially matched fuzzy sets of the rule and the observation. The rule with the highest matching value is subsequently used to derive the conclusion via CRI.

In the process of integration, the alpha-cut overlapping finds the best matching rule above the alpha threshold for the given observation in the existing rule base. If this partial or exact matching rule is found then inference result should be based on this rule, namely, there is no need for FRI and conventional fuzzy inference can be performed directly. The use of composition rule of inference typically involves just a single best matched rule, with a standard reasoning mechanism well-developed in the literature. Although in general multiple rules may also be used, such multiple rules based CRI unnecessarily increases the overall complexity of the system and therefore, is not adopted here.

5.2 Distance Metrics for Selection of Closet Rules

If none of the rules in the rule base overlaps with the observation above the α level, the closest rules for the given observation are selected for interpolation or extrapolation. A proper distance metric should be defined in order to measure the closeness of a rule and the observation. The most commonly used metric is perhaps the Centre of Gravity (COG). It works well in many cases, but sometimes it does not produce reasonable measurements as argued previously. However, an incorrect measurement of relative distances affects the selection of the closest rules. Here, the Hausdorff Distance (HD) metric and Earth Mover's Distance (EMD) are also adopted instead in order to measure the distance between rule and observation. The underlying trapezoidal fuzzy sets $A_{i,j}$ and $A_{\circ,j}$ of the rule and the observation are illustrated in Figure 5.4 with the left extreme point a_0, two extreme points a_1 and a_2 of the nucleus(where a full membership value is reached), and a right extreme point a_3 of each fuzzy set indicated.

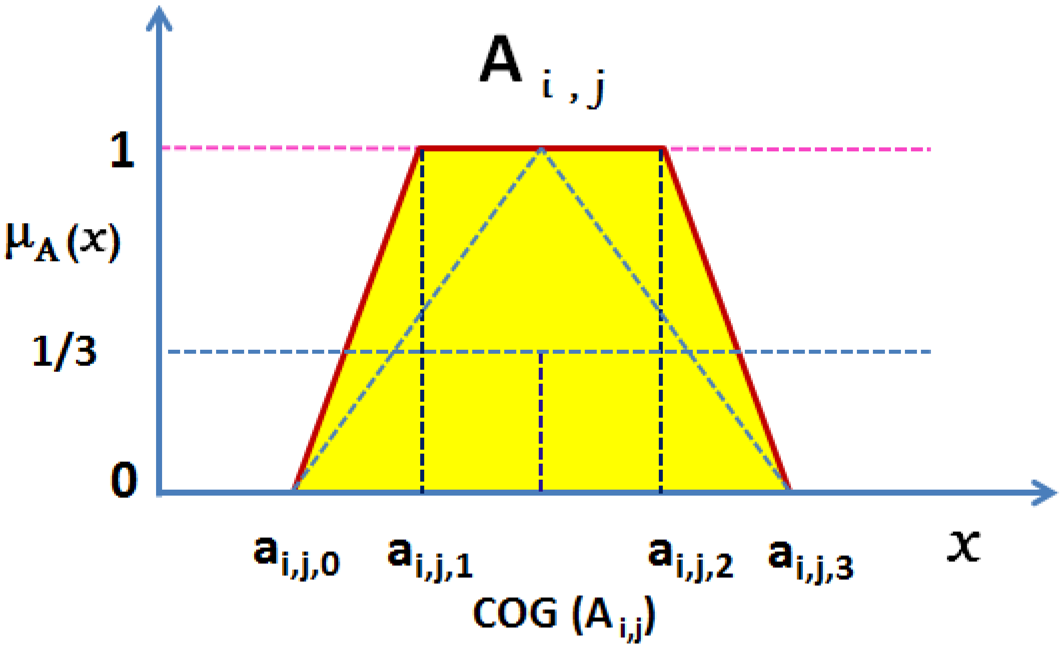


Figure 5.5: Centre of Gravity (COG) calculation for trapezoidal fuzzy set

5.2.1 Centre of Gravity (COG)

The concept of the Centre of Gravity (COG) is that of an average of the masses factored by their distances from a reference point. The COG is an important property since it reflects both the location and the shape of a fuzzy set [82]. The COG metric is used to determine the closeness of two sets based on their COG reference points. For a trapezoidal fuzzy set $A_{i,j} = (a_{i,j,0}, a_{i,j,1}, a_{i,j,2}, a_{i,j,3})$ as shown in Figure 5.5, the COG is calculated as follows [85]:

$$COG(A_{i,j}) = \frac{1}{3}(a_{i,j,0} + \frac{a_{i,j,1} + a_{i,j,2}}{2} + a_{i,j,3}) \tag{5.2}$$

The COG distance between the two fuzzy sets $A_{\circ,j}$ (observation) and $A_{i,j}$ (antecedent) can be calculated as follows [85]:

$$d(A_{\circ,j}, A_{i,j}) = d(COG(A_{\circ,j}), COG(A_{i,j})) \tag{5.3}$$

where $d(A_{\circ,j}, A_{i,j})$ is any conventional distance metric. Finally, for an observation O: $(A_{\circ,1}, A_{\circ,2}, A_{\circ,3}, A_{\circ,4})$ and an i^{th} rule R_i: $(A_{i,1}, A_{i,2}, A_{i,3}, A_{i,4})$ as shown in Figure 5.6, both represented by a trapezoidal fuzzy membership function, the COG distance between them is defined by:

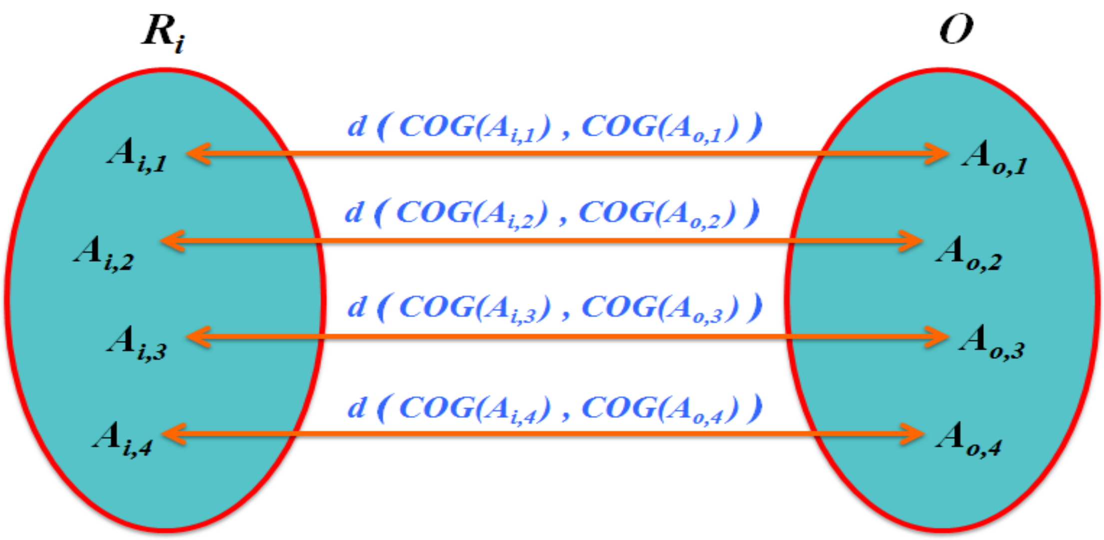


Figure 5.6: Centre of Gravity (COG) calculation between rule R_i and observation O

$$COG(R_i, O) = \sum_{j=1}^{N} \frac{d(COG(A_{\circ,j}), COG(A_{i,j}))}{range_{x_j}} \tag{5.4}$$

where $COG(A_{\circ,j})$ and $COG(A_{i,j})$ are the COGs of sets $A_{\circ,j}$ and $A_{i,j}$ respectively, as shown in Figure 5.4, $range_{x_j} = \max x_j - \min x_j$ over the domain of the variable x_j.

5.2.2 Hausdorff Distance (HD)

The Hausdorff Distance (HD) metric is typically used to determine the closeness of two sets of points that are subsets of a metric space. It captures the concept of the maximum distance of set $A_\circ$ to the nearest point in the other set A_i [41], as shown in Figure 5.7. For the current implimentation, given an observation O: $(A_{\circ,1}, A_{\circ,2}, A_{\circ,3}, A_{\circ,4})$ and an i^{th} rule R_i: $(A_{i,1}, A_{i,2}, A_{i,3}, A_{i,4})$ as shown in Figure 5.8, both represented by a trapezoidal fuzzy membership function, the HD metric is defined by:

$$HD(R_i, O) = \sum_{j=1}^{N} \frac{\max\limits_{a_{\circ,j,l} \in A_{\circ,j}} \left\{ \min\limits_{a_{i,j,k} \in A_{i,j}} \{d_{j,kl}(a_{\circ,j,l}, a_{i,j,k})\} \right\}}{range_{x_j}} \tag{5.5}$$

where $a_{\circ,j,l}$ and $a_{i,j,k}$ are the points of sets $A_{\circ,j}$ and $A_{i,j}$ respectively, as shown in Figure 5.4, $range_{x_j} = \max x_j - \min x_j$ over the domain of the variable x_j, and $d(a_{\circ,j,l}, a_{i,j,k})$ is any conventional distance metric between these points.

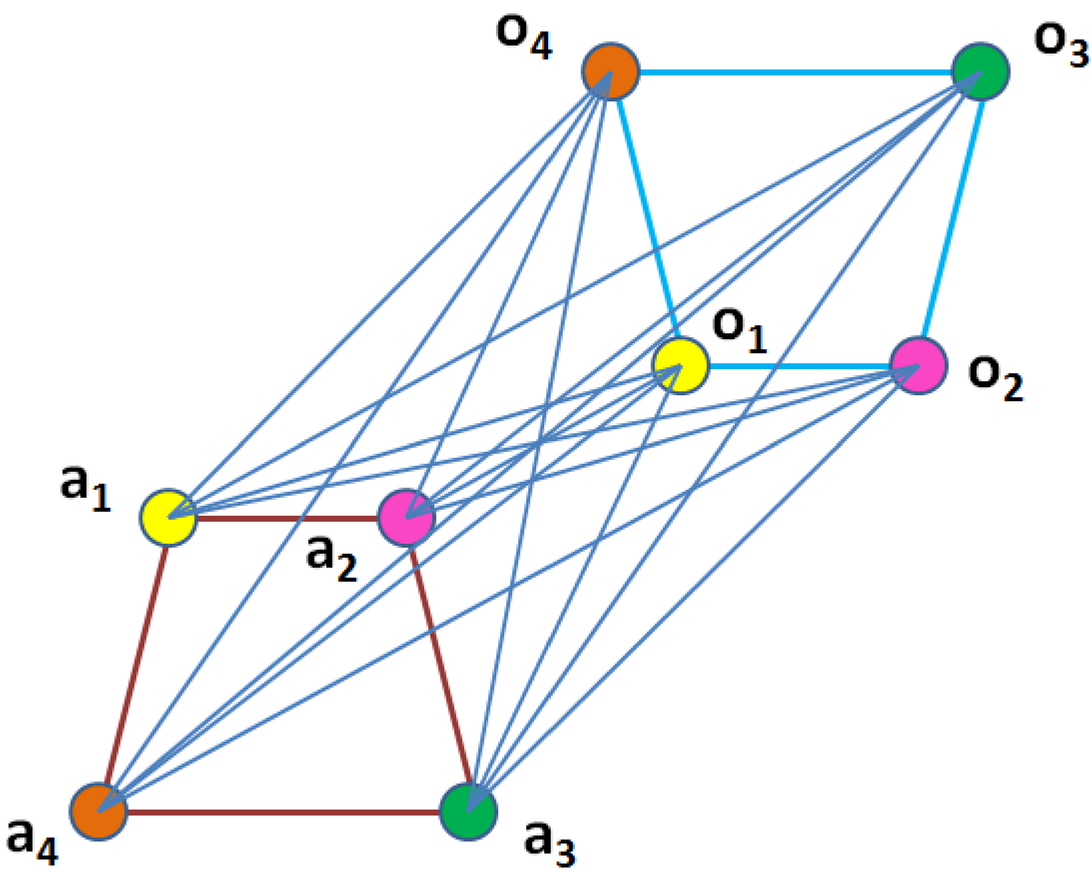


Figure 5.7: Hausdorff Distance (HD) calculation between two fuzzy sets

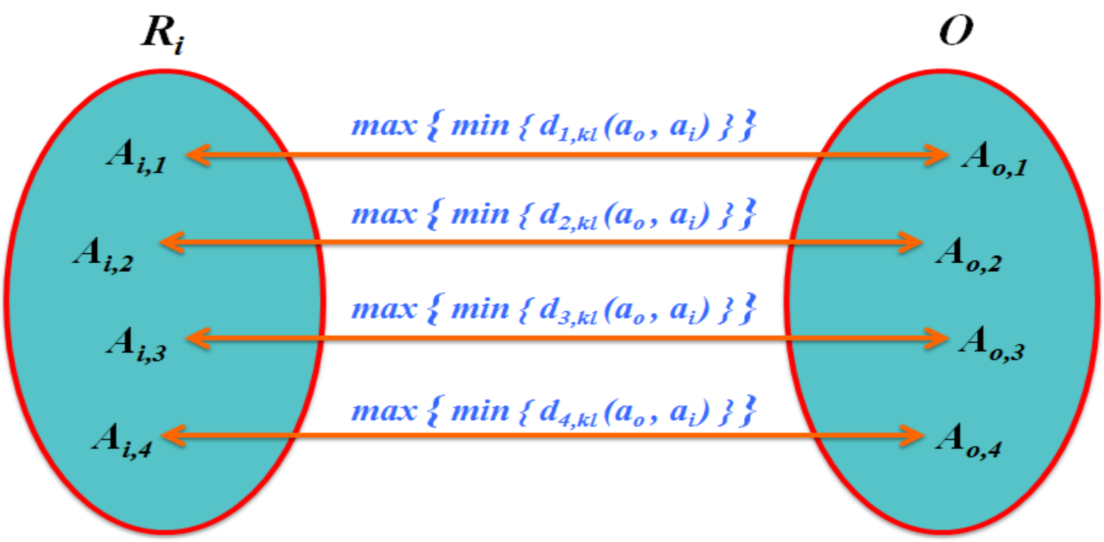


Figure 5.8: Hausdorff Distance (HD) calculation between rule R_i and observation O

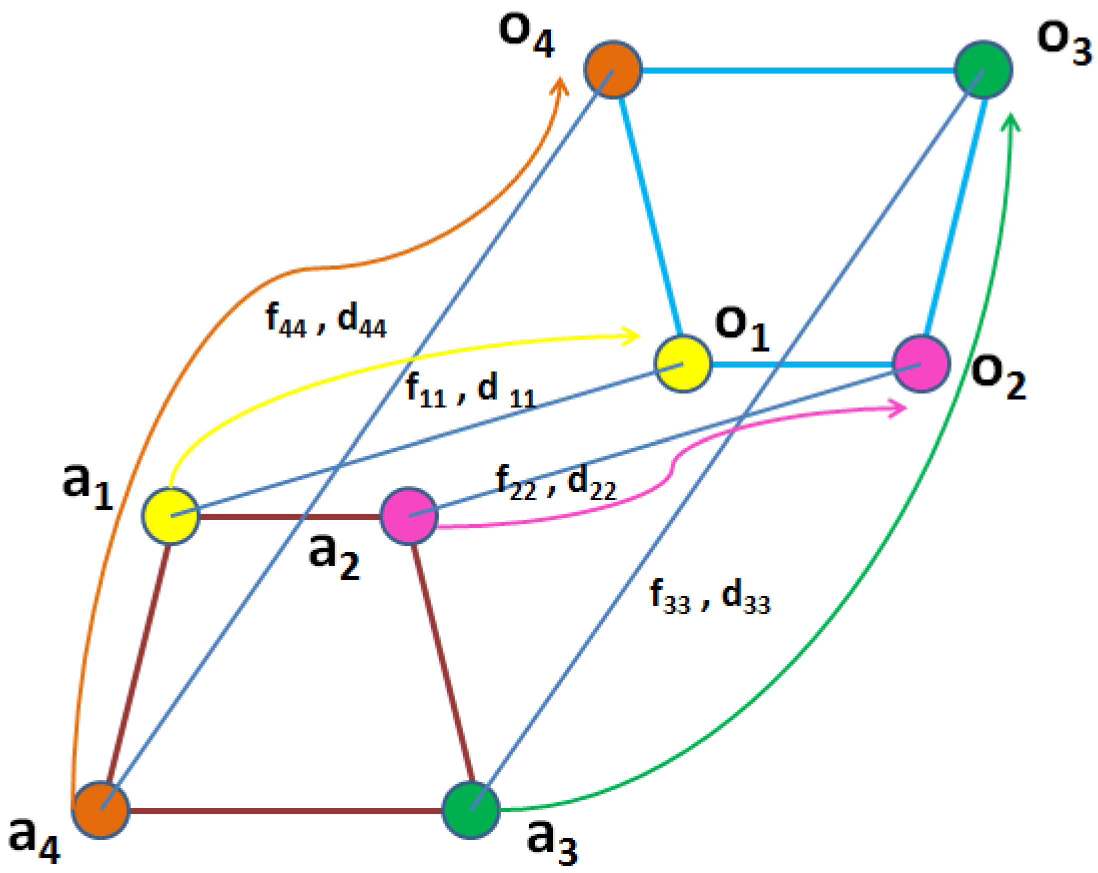


Figure 5.9: Earth Mover's Distance (EMD) calculation between two fuzzy sets

5.2.3 Earth Mover's Distance (EMD)

The Earth Mover's Distance (EMD) is an intuitive and natural distance metric to compare two multi-dimensional distributions in a certain feature space where a distance measure between individual features, which we call the ground distance is given. The distance measurement is done by measuring the least amount of work needed to transform set A_i to set $A_\circ$ as shown in Figure 5.9. Here, a unit of work corresponds to a unit of ground distance [160]. For the current implementation, given an observation $A_\circ$ and an i^{th} rule antecedent A_i, both represented by a trapezoidal fuzzy membership function, the EMD metric is defined by:

$$EMD(R_i, O) = \sum_{j=1}^{N} \frac{\min_{F_j} \frac{\sum_{k=1}^{m} \sum_{l=1}^{n} f_{j,kl} d_{j,kl}}{\sum_{k=1}^{m} \sum_{l=1}^{n} f_{j,kl}}}{range_{x_j}} \quad (5.6)$$

where $f_{j,kl}$ is the amount of mass transported from $a_{i,j,k}$ to $a_{\circ,j,l}$ for morphing $A_{i,j}$ into $A_{\circ,j}$, and $F_j = \{f_{j,kl}\}$ is an admissible flow from $\{a_{i,j,k}\}$ to $\{a_{\circ,j,l}\}$ as shown in

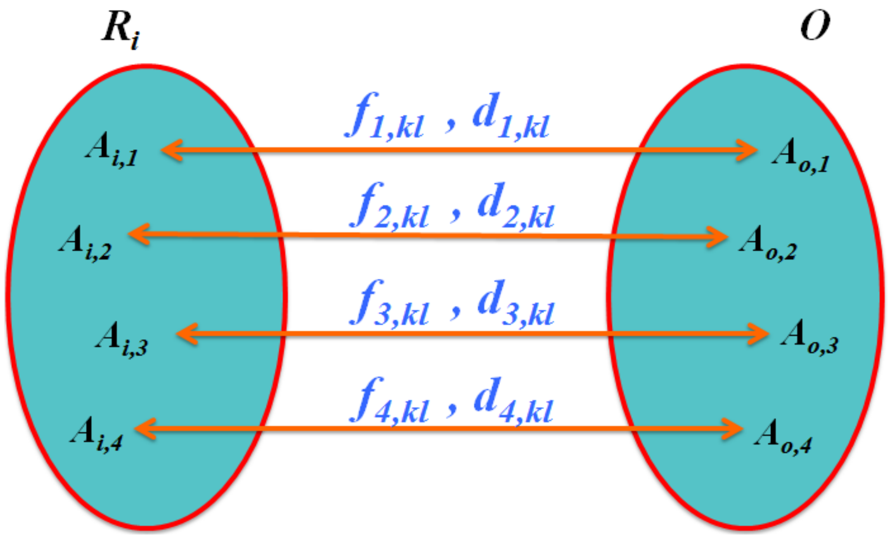


Figure 5.10: Earth Mover's Distance (EMD) calculation between rule R_i and observation O

Figure 5.10, $range_{x_j} = \max x_j - \min x_j$ over the domain of the variable x_j, and $d(a_{i,j,k}, a_{\circ,j,l})$ is any conventional distance metric between $a_{i,j,k}$ and $a_{\circ,j,l}$.

In this integrated D-FRI implementation, two closest rules are considered for interpolation or extrapolation due to the reason that for computational simplicity, most existing FRI methods prefer the use of a minimum number of closest rules, which is two, to determine the interpolation result. Also, the two closest rules based FRI computation is sufficient to generate the interpolation results (which are approximate solutions to a given problem in the first place). However, the selected transformation based FRI method [77,78] also works with more than two closest rules. Thus, the general formulism is provided in this work. Moreover, the main aim of this D-FRI approach is to utilise the FRI results to modify the sparse rule base rather than the detailed analysis of FRI method itself. Putting these considerations together, the two rules based FRI implementation is adopted to simplify and perform the integrated D-FRI implementation.

5.3 Integrated System

Depending on the nature of the rule base either fuzzy inference (CRI) or interpolation (FRI) may be employed to draw the conclusion. CRI methods rely on a dense rule base in which any observation can find at least a complete or partial matching rule. In many real-world problems, obtaining such a complete rule base is costly or even impractical. Interpolation is more robust when working on sparse rule bases. However, the resulting interpolated conclusions may be not as accurate as their inferred counterparts if partial matching between a given observation and the rule-base can be established. To compensate for the drawbacks of these two techniques, an integrated reasoning system is proposed, where both inference and interpolation methods can work together to produce the conclusion for an observation given a sparse rule base.

IntegratedSystem(R, O, α)
$R = \{R_i = rule(A_{i,1}, \cdots, A_{i,j}, \cdots, A_{i,N}, B_i)\}$,
$O = observation(O, \cdots, A_{\circ,j}, \cdots, A_{\circ,N})$,
α, α-cut threshold.

$R_{overlap} = \alpha$-CutOverlapping(R, O, α);
if $R_{overlap} \neq NULL$ **then**
  $B_o =$ CRI$(R_{overlap}, O)$;
else
  $R_{close1} =$ COG/HD/EMD_Closest(R, O);
  $R_{close2} =$ COG/HD/EMD_Closest$(R - R_{close1}, O)$;
  if $R_{close1} < O < R_{close2}$ **or** $R_{close2} < O < R_{close1}$ **then**
    $B_o =$ Interpolation$(R_{close1}, O, R_{close2})$;
  else
    $B_o =$ Extrapolation$(R_{close1}, R_{close2}, O)$;
  end if
end if

Algorithm 5.3.1: Integration of Interpolation and Inference

The proposed integrated interpolation and inference technique involves two sub-methods: α-cut based inference and selection of closest rules based on either COG, HD or EMD. The system initially tests the usability of the conventional compositional rule of inference by detecting the matching degree between the observation and the rules. If one or more partial matchings are found above a given confidence level, the rule with the highest degree of matching is fired via CRI in order to derive the conclusion. Otherwise, two (or more if preferred) closest rules to the observation are selected to compute the conclusion by interpolation or extrapolation (depending on the positions of those selected rules).

The overall operation of the integrated system is depicted by flow-chart in Figure 5.11. For efficiency, the system starts by matching the α-cut sets of the rule antecedents with the observation. If certain rules match the observation with a given

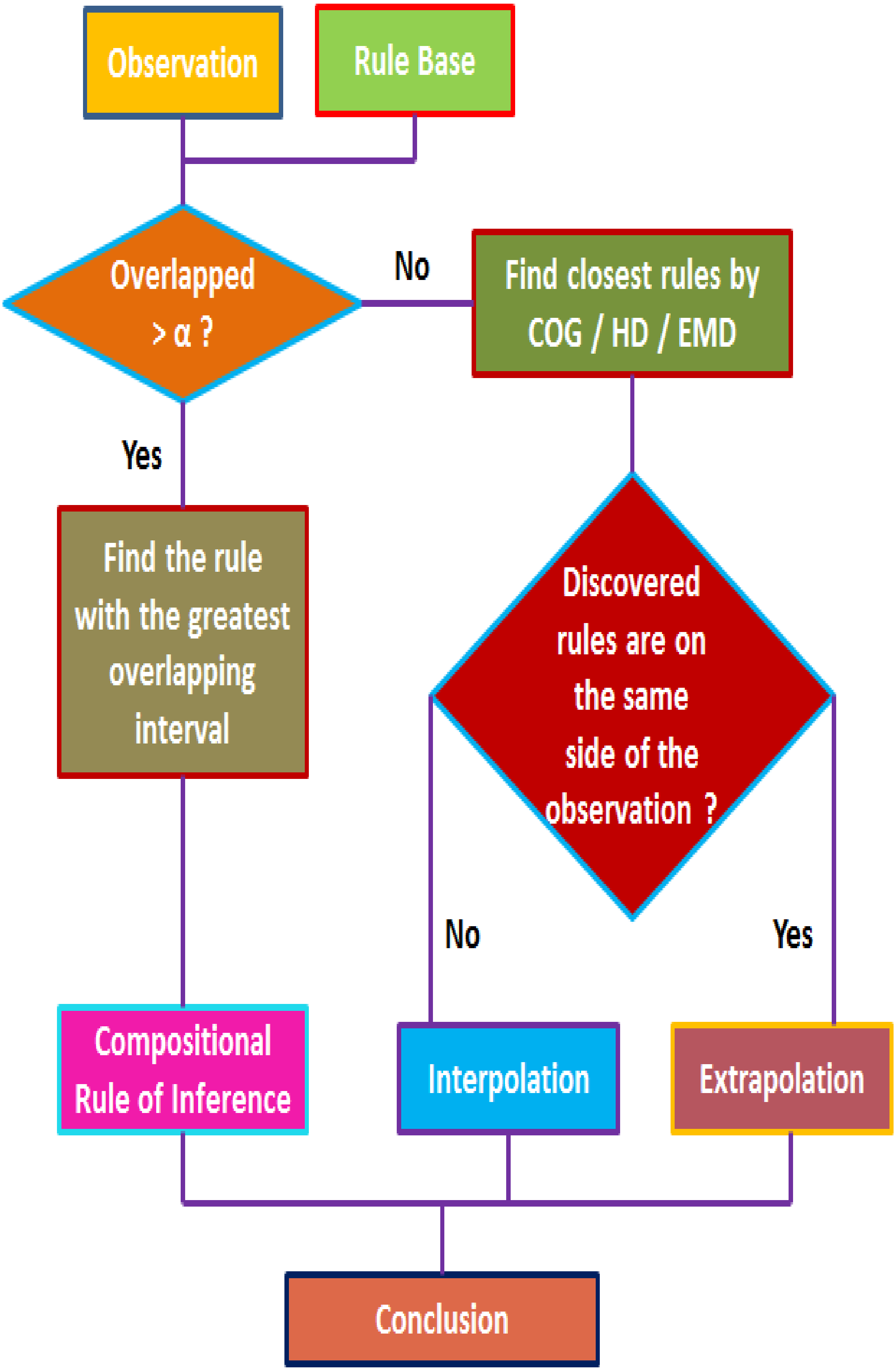


Figure 5.11: Integrated interpolation and inference system

confidence level α then determine the rule whose antecedent overlaps the most. From that, compositional rule of inference is performed using the highest overlapped rule. If no such match is found between the observation and any rule, then it employs a COG, HD or EMD metric, to identify rules that are the closest to the observation in order to perform interpolation, or extrapolation. If these closest rules are on the same side of the observation, then extrapolation is used, otherwise interpolation is used. In this research work, to infer the conclusion, the scale and move transformation interpolation (T-FRI) method [85, 86] is utilised (although any other fuzzy interpolation mechanism may act as an alternative). Here, it is assumed that only two rules are needed in order to perform interpolation or extrapolation. However, it is not difficult to extend this if more than two rules are performed for interpolation. The algorithm for the integrated system is shown in Algorithm 5.3.1-5.3.3.

$\alpha-$CutOverlapping(R, O, α)
A_{iL}, A_{iR}, left and right extreme points of A_i,
inf{$A_{\circ\alpha}$}, infimum value of crisp set $A_{\circ\alpha}$,
sup{$A_{\circ\alpha}$}, supremum value of crisp set $A_{\circ\alpha}$.

$maxArea = 0, maxIndex = -1$;
for each R_i in R **do**
  if $A_{iL} < \inf\{A_{\circ\alpha}\} < A_{iR}$ **or** $A_{iL} < \sup\{A_{\circ\alpha}\} < A_{iR}$ **or** $A_{oL} < A_{iL} < A_{iR} < A_{oR}$ **then**
    $overlap$ = overlapping area of $A_\circ$ and A_i above α;
    if $overlap > maxArea$ **then**
      $maxArea = overlap$;
      $maxIndex = i$;
    end if
  end if
end for
if $maxIndex == -1$ **then**
  return NULL;
else
  return $R_{maxIndex}$;
end if

Algorithm 5.3.2: $\alpha-$cut Overlapping between Observation and Rule Antecedents

5.4 Experimentation and Discussion

In this section, the example problems based on triangular and trapezoidal fuzzy sets are given to demonstrate how inference and interpolation can be integrated to make reasoning system more effective and accurate. The triangular fuzzy set example is

```
COG/HD/EMD_Closest(R, O)
closeDist = Max_Value, closeIndex = −1;
for each R_i in R do
    dist = COG/HD/EMD(R_i, O);
    if dist < closeDist then
        closeDist = dist;
        closeIndex = i;
    end if
end for
return R_closeIndex;
```

Algorithm 5.3.3: COG/HD/EMD based Identification of Closet Rules

used to simplify the concept of integration of two methods. It depicts the various cases of overlapping, interpolation, extrapolation and distance metric comparison diagrammatically. This will help to understand simple yet more effective notion of integrated fuzzy reasoning. The second trapezoidal fuzzy set example will provide the in-depth study of proposed integrated approach and its effectiveness.

5.4.1 Results for Integrated System involving Triangular Fuzzy Sets

This subsection demonstrates the use of the proposed method involving only triangular fuzzy sets. Here, two illustrative examples are shown for α-cut overlapping operation and comparison among COG, HD and EMD metrics respectively. These simple examples are used to explain the procedure of the proposed integrated system.

Table 5.1: Sparse Rule-Base

ID	Sparse Rule-Base	
	Antecedent (A)	*Consequent* (B)
$R1$	$(1,2,3)$	$(2.5,3,5.5)$
$R2$	$(4,5,6)$	$(4.5,5.5,6)$
$R3$	$(9.9,12.5,13)$	$(10.12,14.32,15.11)$
$R4$	$(12.5,14,14.5)$	$(14.12,15.32,18.11)$

5.4.1.1 α-cut Overlapping Operation

In this example, a sparse rule base consisting of four rules, with a confidence level $\alpha = 0.6$, is considered. The rule base is given in Table 5.1, and the antecedents in these rules are shown in Figure 5.12(a). Three typical observations are considered for

Table 5.2: Results for the Three Examples

ID	Observation	Result	Operation	Closest Rules
	A^*	B^*	Method	Antecedent (A)
1	$(2.7, 5.1, 5.4)$	$(4.5, 5.5, 6)$	Inference	(4, 5, 6)
2	$(2.5, 3.6, 4.4)$	$(3.48, 4.41, 5.61)$	Interpolation	(1, 2, 3), (4, 5, 6)
3	$(9, 10.6, 11)$	$(10.19, 12.35, 13.58)$	Extrapolation	(9.9, 12.5, 13),(12.5, 14, 14.5)

Table 5.3: Comparison among HD , COG and EMD Metrics

Observations	Two Closest Rule Antecedents		Closest by HD	Closest by COG	Closest by EMD
A^*	A_1	A_2	$A(HD)$	$A(COG)$	$A(EMD)$
$(6, 7, 9)$	$(1, 2, 3)$	$(10, 13.5, 15)$	$(10, 13.5, 15)$	$(1, 2, 3)$	$(1, 2, 3)$
$(3, 6, 10)$	$(1, 3, 6)$	$(8, 9, 10.5)$	$(1, 3, 6)$	$(8, 9, 10.5)$	$(8, 9, 10.5)$
$(5, 6, 6)$	$(1, 2, 2)$	$(9, 10.6, 10.6)$	$(9, 10.6, 10.6)$	$(1, 2, 2)$	$(1, 2, 2)$
$(3, 4, 6)$	$(-2, -1, 2)$	$(8, 8.8, 10)$	$(-2, -1, 2)$	$(8, 8.8, 10)$	$(8, 8.8, 10)$

three reasoning conditions: inference, interpolation and extrapolation respectively. The results with respect to each of the three observations are listed in Table 5.2. For the first case, which is shown in Figure 5.12(b), the observation A^* does not overlap with A_1, A_3, A_4 above the α-cut level, but it partially matches A_2. Thus, rule R2 is the only rule which has an overlap area above α and is selected to compute the conclusion by running the compositional rule of inference. In the second case, which is shown in Figure 5.12(c), A^* does not match any of the rules. Therefore, the COG, HD or EMD between A^* and the antecedent of each rule are calculated and compared. This leads to that rule R1 and R2 are selected to perform interpolation (as A_1 and A_2 are the closest to A^*). The third case is similar to the second, which is shown in Figure 5.12(d). However, since the closest rules of A^*, i.e. rule R3 and R4 are both on the right hand side of A^*, extrapolation is employed in order to infer the conclusion. Overall, this example explains all the three reasoning conditions and when they are arisen.

5.4.1.2 Comparison among COG, HD and EMD Metrics

In this example, four special cases are considered to compare the use of the COG, HD and EMD distance metrics. Figure 5.13 shows all of these cases and Table 5.3 lists the results. In case-1, the observation $A^* = (6,7,9)$ lies between the two rule antecedents $A_1 = (1,2,3)$ and $A_2 = (10,13.5,15)$ without any overlap. The HD metric selects the rule with antecedent $A_2 = (10,13.5,15)$ as the closest, whereas the COG and EMD metrics select that with antecedent $A_1 = (1,2,3)$. In case-2, the observation $A^* = (3,6,10)$ overlaps with the two antecedent fuzzy sets $A_1 = (1,3,6)$ and $A_2 = (8,9,10.5)$. The HD metric selects the rule with antecedent $A_1 = (1,3,6)$ as the closest whereas the COG and EMD metrics select $A_2 = (8,9,10.5)$. In case-3, the observation $A^* = (5,6,6)$ lies between the two rule antecedent values $A_1 = (1,1,2)$ and $A_2 = (9,10.6,10.6)$. The HD metric selects the rule with antecedent $A_2 = (9,10.6,10.6)$ as the closest whereas the COG and EMD metrics select $A_1 = (1,1,2)$. In case-4, the observation $A^* = (3,4,6)$ lies between two rule antecedent fuzzy sets $A_1 = (-2,-1,2)$ and $A_2 = (8,8.8,10)$. The HD metric selects the rule with antecedent $A_1 = (-2,-1,2)$ as the closest whereas the COG and EMD metrics select $A_2 = (8,8.8,10)$.

All four cases are different in nature. Interestingly, the results in all these cases are also different. The HD metric outcomes are completely different from the COG and EMD metrics. This indicates that the closest rules selected for interpolation or

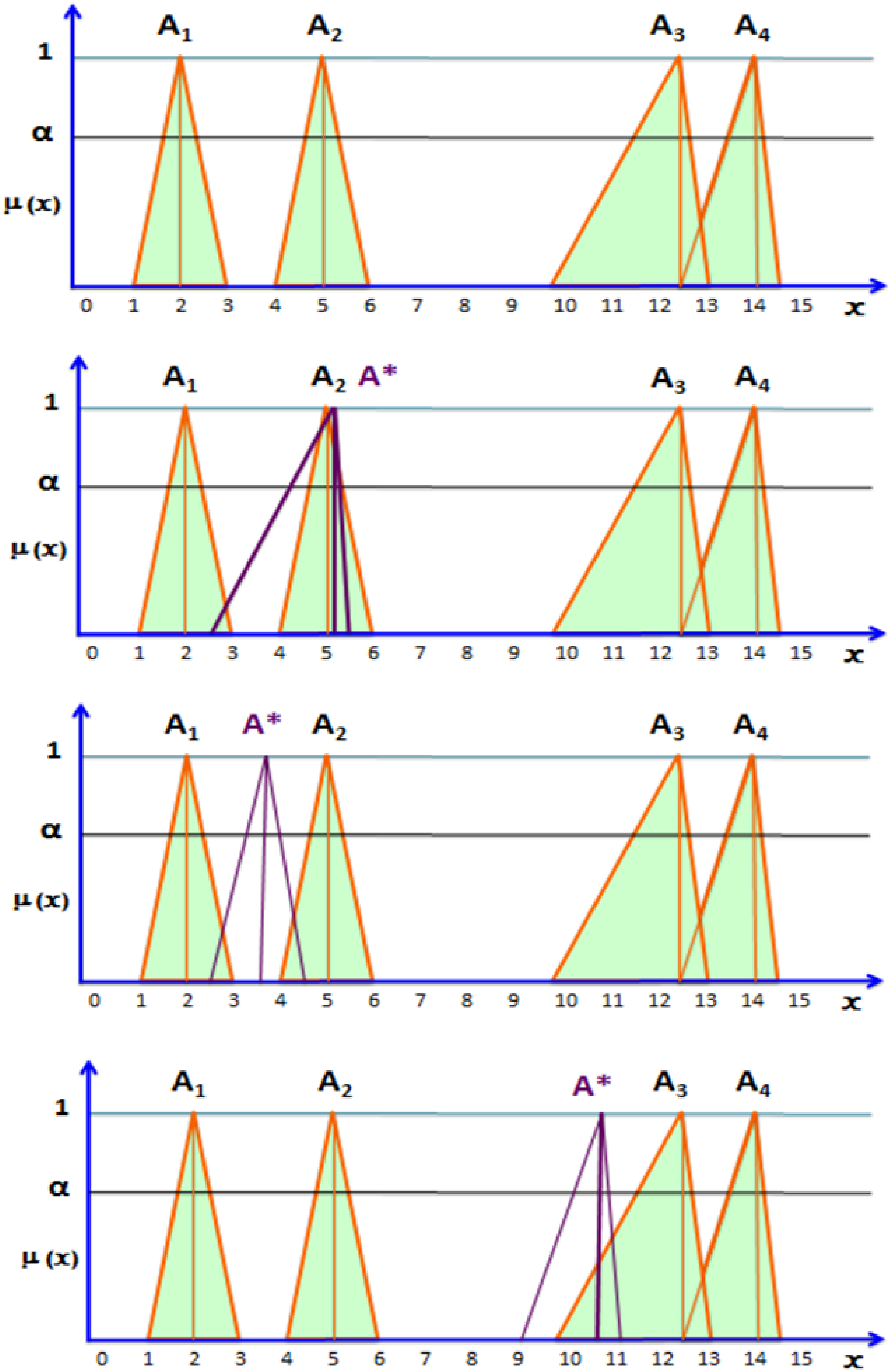


Figure 5.12: α-cut overlapping example: (a) Sparse rule base, (b) Inference condition, (c) Interpolation condition, and (d) Extrapolation condition

extrapolation are heavily dependent on the choice of the distance metric. Thus, the distance metric could greatly affect the inference results generated by fuzzy rule interpolation methods. It leads to the selection of the most accurate distance metric. However, the closest rules selected by the HD metric have antecedent values whose membership functions are of a shape that more closely resemble the shape of the observation in most cases. This may help to maintain the interpretability of the integrated inference system. In this particular example, another fascinating fact from all the four cases is that the COG and EMD are quite similar and selecting similar rules. Nevertheless, this example only finds one closest rule and therefore demonstrates the initial findings for the COG, HD and EMD metrics.

5.4.2 Results for Integrated System involving Trapezoidal Fuzzy Sets

This subsection demonstrates the use of the proposed method involving trapezoidal fuzzy sets. Here, two illustrative examples are shown using a sparse rule base consisting of eight typically selected rules to cover variety of cases for analysis. An each rule R_i is having four antecedents $A_i = \left\{A_{i,1}, A_{i,2}, A_{i,3}, A_{i,4}\right\}$, a consequent $B_i = \frac{1}{N}\sum_{j=1}^{N} A_{i,j}$, with a confidence level $\alpha = 0.5$. The sparse rule base given in Table 5.4, is used for both $\alpha-$cut overlapping operation to determine the use of CRI and comparison of three distance metrics: COG, HD and EMD by finding the closest rules for interpolation or extrapolation and subsequently their inference results.

5.4.2.1 α-cut Overlapping Operation

The first example demonstrates the effectiveness and results for $\alpha-$cut overlapping operation to perform the compositional rule of inference (CRI) in case of sparse rule base rather interpolation or extrapolation. It is shown in the Table 5.5 where five observations are considered in such a way that they overlap with many existing rules of the given sparse rule. The first observation $O_1 : A_{\circ,1} = (11.1, 12.8, 14.1, 15.2), A_{\circ,2} = (13.1, 14.8, 16.1, 17.2), A_{\circ,3} = (15.1, 16.8, 18.1, 19.2), A_{\circ,4} = (18.1, 19.8, 21.1, 22.2)$ overlaps with rule R5 to rule R8 above the $\alpha-$level. Here R5 has the lowest overlap area and R7 has the highest overlap area with the observation thus R7 is selected to compute the conclusion by running the compositional rule of inference. The second observation $O_2 : A_{\circ,1} = (2.3, 3.4, 4.4, 5.5), A_{\circ,2} = (4.3, 5.4, 6.4, 7.5), A_{\circ,3} = (7.3, 8.4, 9.4, 10.5), A_{\circ,4} = (9.3, 10.4, 11.4, 12.5)$ overlaps with rule R1 to rule R4

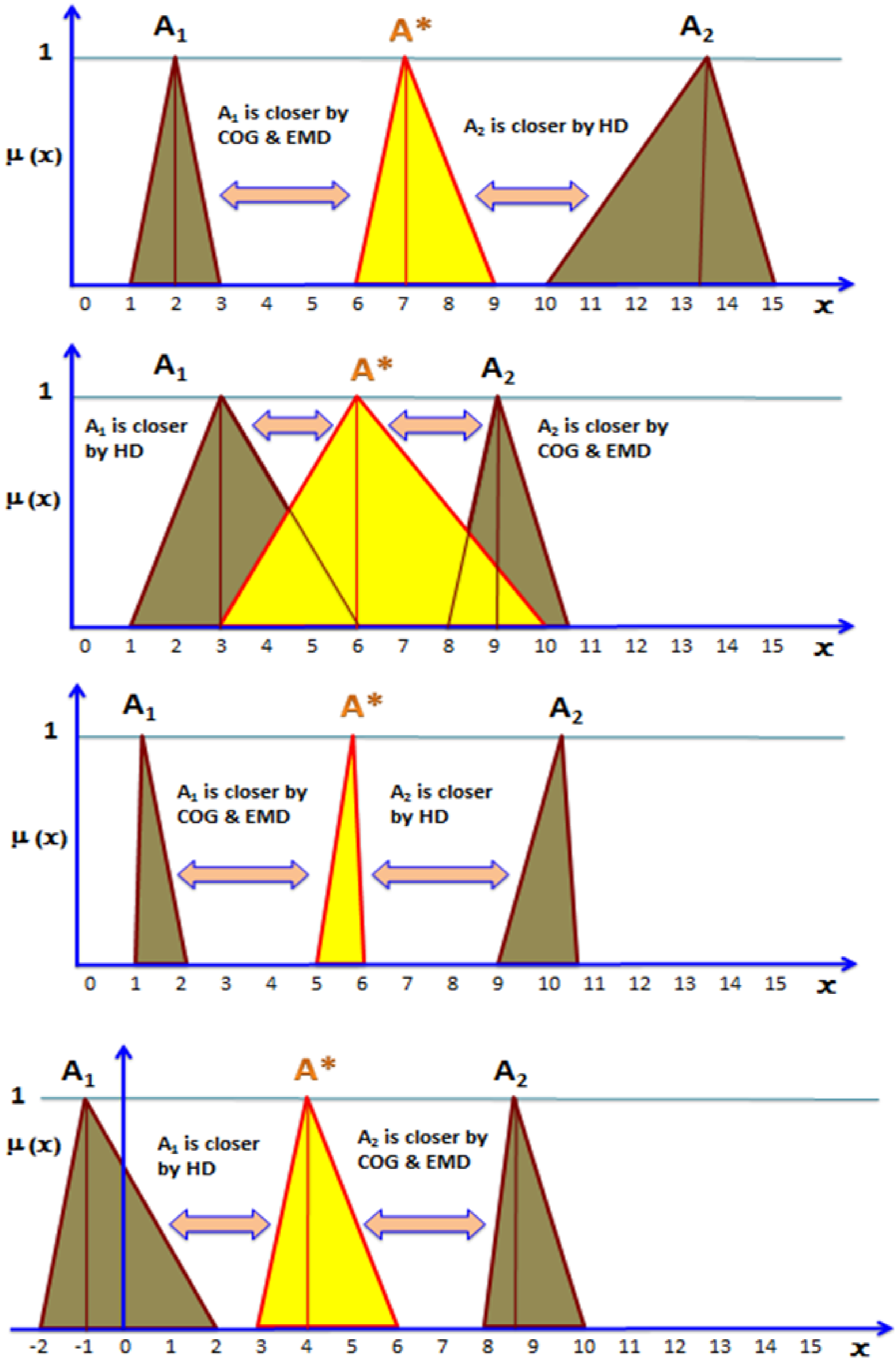


Figure 5.13: Comparison among COG, HD and EMD metrics

Table 5.4: Sparse Rule-Base

Rule	Sparse Rule-Base				
ID	Antecedents				*Consequent*
	$A_{i,1}$	$A_{i,2}$	$A_{i,3}$	$A_{i,4}$	B_i
$R1$	$(0.9, 2.2, 3.1, 5.6)$	$(2.9, 4.2, 5.1, 7.6)$	$(5.9, 7.2, 8.1, 10.6)$	$(7.9, 9.2, 10.1, 12.6)$	$(4.4, 5.7, 6.6, 9.1)$
$R2$	$(1.6, 2.2, 3.3, 5.5)$	$(3.6, 4.2, 5.3, 7.5)$	$(6.6, 7.2, 8.3, 10.5)$	$(8.6, 9.2, 10.3, 12.5)$	$(5.1, 5.7, 6.8, 9.0)$
$R3$	$(2.8, 3.1, 4.3, 5.3)$	$(4.8, 5.1, 6.3, 7.3)$	$(7.8, 8.1, 9.3, 10.3)$	$(9.8, 10.1, 11.3, 12.3)$	$(6.3, 6.6, 7.8, 8.8)$
$R4$	$(2.1, 3.3, 4.6, 5.4)$	$(4.1, 5.3, 6.6, 7.4)$	$(7.1, 8.3, 9.6, 10.4)$	$(9.1, 10.3, 11.6, 12.4)$	$(5.6, 6.8, 8.1, 8.9)$
$R5$	$(10.4, 11.0, 11.9, 12.5)$	$(12.4, 13.0, 13.9, 14.5)$	$(15.4, 16.0, 16.9, 17.5)$	$(17.4, 18.0, 18.9, 19.5)$	$(13.9, 14.5, 15.4, 16)$
$R6$	$(10.3, 11.7, 12.6, 13.3)$	$(12.3, 13.7, 14.6, 15.3)$	$(15.3, 16.7, 17.6, 18.3)$	$(17.3, 18.7, 19.6, 20.3)$	$(13.8, 15.2, 16.1, 16.8)$
$R7$	$(10.0, 12.9, 13.7, 14.5)$	$(12.0, 14.9, 15.7, 16.5)$	$(15.0, 17.9, 18.7, 19.5)$	$(17.0, 18.9, 20.7, 21.5)$	$(13.8, 16.2, 17.2, 18)$
$R8$	$(10.2, 11.6, 12.9, 13.8)$	$(12.2, 13.6, 14.9, 15.8)$	$(15.2, 16.6, 17.9, 18.8)$	$(17.2, 18.6, 19.9, 20.8)$	$(13.7, 15.1, 16.4, 17.3)$

above the α—level. Here R1 has the lowest overlap area and R4 has the highest overlap area with the observation thus R4 is selected for CRI. The third observation $O_3 : A_{\circ,1} = (9.7, 10.8, 11.7, 12.2), A_{\circ,2} = (11.7, 12.8, 13.7, 14.2), A_{\circ,3} = (14.7, 15.8, 16.7, 17.2), A_{\circ,4} = (16.7, 17.8, 18.7, 19.2)$ overlaps with rule R5 to rule R8 above the α—level. Here R7 has the lowest overlap area and R5 has the highest overlap area with the observation thus R5 is selected for CRI. The fourth observation $O_4 : A_{\circ,1} = (1.0, 2.3, 3.2, 5.8), A_{\circ,2} = (3.0, 4.3, 5.2, 7.8), A_{\circ,3} = (6.0, 7.3, 8.2, 10.8), A_{\circ,4} = (8.0, 9.3, 10.2, 12.8)$ overlaps with rule R1 to rule R4 above the α—level. Here R3 has the lowest overlap area and R1 has the highest overlap area with the observation thus R1 is selected for CRI. The final fifth observation $O_5 : A_{\circ,1} = (10.1, 11.6, 13.0, 13.9), A_{\circ,2} = (12.1, 13.6, 15.0, 15.9), A_{\circ,3} = (15.1, 16.6, 18.0, 18.9), A_{\circ,4} = (17.1, 18.6, 20.0, 20.9)$ overlaps with rule R5 to rule R8 above the α—level. Here R5 has the lowest overlap area and R8 has the highest overlap area with the observation thus R8 is selected for CRI.

In all the above cases, all observations overlapped with more than one rules above the α—level in the sparse rule base. This is clearly the case of inference in the sparse rule base and therefore we can easily avoid the costly interpolation or extrapolation operations. Finally, the most overlapped rule is selected for CRI among all the overlapping rules with observation based on the greatest overlapping area.

5.4.2.2 Comparison among COG, HD and EMD Metrics

The second example demonstrates the comparative evaluation of three distance metrics: COG, HD and EMD and inference results based on these metrics. Comparative results are shown in Table 5.6 and 5.7 where eleven typical observations are considered to compare the three distance metrics by finding the two closest rules for interpolation or extrapolation. Interestingly, the results in all these cases are quite different for COG, HD and EMD. Fox example, surprisingly, for the first observation $O_1 : A_{\circ,1} = (5.6, 6.9, 8.3, 10.1), A_{\circ,2} = (7.6, 8.9, 10.3, 12.1), A_{\circ,3} = (10.6, 11.9, 13.3, 15.1), A_{\circ,4} = (12.6, 13.9, 15.3, 17.1)$, the first closest rule (i.e. R7) selected by the HD ($d_{HD}(R_7, O_1) = 4.40$) and EMD ($d_{EMD}(R_7, O_1) = 2.89$) is the least closest rule (at the last position) given by COG ($d_{COG}(R_7, O_1) = 5.05$) as shown in Table 5.6. Similarly, for all the other observations, different closest rules are selected by different distance metrics as shown in Table 5.7.

For simplicity, for all the observations, the two closest rules are selected by COG, HD and EMD metrics and shown in the Table 5.7. However, more than two closest

Table 5.5: Results for $\alpha-$Cut Matching Operation

No.	Observation $O\{A_{\circ,1}, A_{\circ,2}, A_{\circ,3}, A_{\circ,4}\}$	Most Overlapped Rule
1	(11.1, 12.8, 14.1, 15.2), (13.1, 14.8, 16.1, 17.2) (15.1, 16.8, 18.1, 19.2), (18.1, 19.8, 21.1, 22.2)	$R7$
2	(2.3, 3.4, 4.4, 5.5), (4.3, 5.4, 6.4, 7.5) (7.3, 8.4, 9.4, 10.5), (9.3, 10.4, 11.4, 12.5)	$R4$
3	(9.7, 10.8, 11.7, 12.2), (11.7, 12.8, 13.7, 14.2) (14.7, 15.8, 16.7, 17.2), (16.7, 17.8, 18.7, 19.2)	$R5$
4	(1.0, 2.3, 3.2, 5.8), (3.0, 4.3, 5.2, 7.8) (6.0, 7.3, 8.2, 10.8), (8.0, 9.3, 10.2, 12.8)	$R1$
5	(10.1, 11.6, 13.0, 13.9), (12.1, 13.6, 15.0, 15.9) (15.1, 16.6, 18.0, 18.9), (17.1, 18.6, 20.0, 20.9)	$R8$

Table 5.6: Order of Rules based on Distance Calculation Between Rule & Observation by COG , HD and EMD Metrics

Observation	Order of Rules by COG		Order of Rules by HD		Order of Rules by EMD	
O	*Rules*	$d_{COG}(R_i, O)$	*Rules*	$d_{HD}(R_i, O)$	*Rules*	$d_{EMD}(R_i, O)$
(5.6, 6.9, 8.3, 10.1), (7.6, 8.9, 10.3, 12.1), (10.6, 11.9, 13.3, 15.1), (12.6, 13.9, 15.3, 17.1)	R1	4.78	R1	4.50	R1	3.51
	R2	4.57	R2	4.60	R2	3.34
	R3	3.85	R3	4.80	R3	3.58
	R4	3.88	R4	4.69	R4	3.21
	R5	3.72	R5	4.80	R5	3.40
	R6	4.25	R6	4.70	R6	3.33
	R7	**5.05**	**R7**	**4.40**	**R7**	**2.89**
	R8	4.40	R8	4.60	R8	3.82

Table 5.7: Two Closest Rules Given by COG , HD and EMD Metrics

No.	Observation $O\{A_{\circ,1}, A_{\circ,2}, A_{\circ,3}, A_{\circ,4}\}$	Closest Rules by COG	Closest Rules by HD	Closest Rules by EMD
O_1	(5.6, 6.9, 8.3, 10.1), (7.6, 8.9, 10.3, 12.1),	$R5$	$R7$	$R7$
	(10.6, 11.9, 13.3, 15.1), (12.6, 13.9, 15.3, 17.1)	$R3$	$R1$	$R4$
O_2	(5.6, 6.3, 7.1, 8.0), (7.6, 8.3, 9.1, 10.0),	$R3$	$R1$	$R4$
	(10.6, 11.3, 12.1, 13.0), (12.6, 13.3, 14.1, 15.0)	$R4$	$R2$	$R2$
O_3	(6.2, 7.1, 8.1, 9.7), (8.2, 9.1, 10.1, 11.7),	$R5$	$R7$	$R7$
	(11.2, 12.1, 13.1, 14.7), (13.2, 14.1, 15.1, 16.7)	$R3$	$R8$	$R4$
O_4	(8.2, 9.1, 9.8, 10.5), (10.2, 11.1, 11.8, 12.5),	$R6$	$R7$	$R5$
	(13.2, 14.1, 14.8, 15.5), (15.2, 16.1, 16.8, 17.5)	$R8$	$R8$	$R6$
O_5	(6.9, 7.4, 8.3, 8.8), (8.9, 9.4, 10.3, 10.8),	$R5$	$R7$	$R2$
	(11.9, 12.4, 13.3, 13.8), (13.9, 14.4, 15.3, 15.8)	$R3$	$R1$	$R7$
O_6	(6.5, 7.0, 7.5, 8.2), (8.5, 9.0, 9.5, 10.2),	$R3$	$R1$	$R4$
	(11.5, 12.0, 12.5, 13.2), (13.5, 14.0, 14.5, 15.2)	$R4$	$R2$	$R7$
O_7	(6.0, 7.0, 7.5, 8.8), (8.0, 9.0, 9.5, 10.8),	$R3$	$R1$	$R4$
	(11.0, 12.0, 12.5, 13.8), (13.0, 14.0, 14.5, 15.8)	$R4$	$R2$	$R3$
O_8	(6.6, 7.3, 8.1, 8.8), (8.6, 9.3, 10.1, 10.8),	$R5$	$R1$	$R2$
	(11.6, 12.3, 13.1, 13.8), (13.6, 14.3, 15.1, 15.8)	$R3$	$R2$	$R1$
O_9	(7.3, 7.8, 8.6, 9.0), (9.3, 9.8, 10.6, 11.0),	$R5$	$R7$	$R7$
	(12.3, 12.8, 13.6, 14.0), (14.3, 14.8, 15.6, 16.0)	$R6$	$R8$	$R2$
O_{10}	(7.1, 7.5, 8.0, 9.1), (9.1, 9.5, 10.0, 11.1),	$R5$	$R7$	$R7$
	(12.1, 12.5, 13.0, 14.1), (14.1, 14.5, 15.0, 16.1)	$R3$	$R8$	$R6$
O_{11}	(5.3, 6.0, 6.5, 7.0), (7.3, 8.0, 8.5, 9.0),	$R3$	$R1$	$R2$
	(10.3, 11.0, 11.5, 12.0), (12.3, 13.0, 13.5, 14.0)	$R4$	$R2$	$R3$

rules can also be selected depending on the method of interpolation or extrapolation. Noticeably, there is no single common result for any of these observations by all three distance metrics. In each case, the selected closest rules are quite different with each other and due to the consequence of these closest rules, the selected inference mechanism is also different by different distance metrics. Although, every distance metric has their unique pattern of rule selection and selected quite common rules for many observations. For example, COG has consistently selected rules R3, R4 and R5, HD has consistently selected rules R1, R2 and R7 and EMD has consistently selected rules R2, R4 and R7. However, EMD looks more sensitive while selecting the closest rules because it changes the closest rules with slight variation of value in observations.

As previously mentioned, the selected rules decide the inference mechanism based on their locations. In many cases, if one distance metric suggests the interpolation operation based on their closest rules for any observation, at the same time another distance metric suggests the extrapolation operation based on their closest rules for the same observation as shown in Table 5.8. In spite of different closest rules, the similar inference mechanism is selected by all three distance metrics for more than half of the observations. The COG and EMD seem quite similar while suggesting the inference mechanism because both have suggested 5 times interpolations and 6 times extrapolations. However, HD is completely different from both and it has suggested only 2 times interpolations and 9 times extrapolations for the same observations.

Finally, based on these closest rules by the COG, HD and EMD, reasoning is performed using T-FRI method [85, 86] to evaluate the impact of different distance metrics on interpolation and extrapolation. The reasoning results are shown in Table 5.9. Subsequently, these three different inference results based on the COG, HD and EMD are then compared to the ground truth ($\epsilon_{\%COG}, \epsilon_{\%HD}, \epsilon_{\%EMD}$) respectively, as shown in Table 5.10. This shows that the inference results based on the closest rules selected by HD and EMD metrics are slightly improved as compared to the inference results based on the closest rules selected by COG. However, HD has more better inference results than EMD. Therefore, from all the examples, it is very clear that HD metric is performing better than both COG and EMD metics. Also, the closest rules selected by the HD metric have antecedent values whose membership functions are of a shape that more closely resemble the shape of the observation in most cases. This may help to maintain the interpretability of the integrated inference system.

Table 5.8: Inference Mechanism by COG , HD and EMD Metrics (Based on the Locations of Closest Rules)

Observations O_i	Inference Decision Based on COG	Inference Decision Based on HD	Inference Decision Based on EMD
O_1	*Interpolation*	*Interpolation*	*Interpolation*
O_2	*Extrapolation*	*Extrapolation*	*Extrapolation*
O_3	*Interpolation*	*Extrapolation*	*Interpolation*
O_4	*Extrapolation*	*Extrapolation*	*Extrapolation*
O_5	*Interpolation*	*Interpolation*	*Interpolation*
O_6	*Extrapolation*	*Extrapolation*	*Interpolation*
O_7	*Extrapolation*	*Extrapolation*	*Extrapolation*
O_8	*Interpolation*	*Extrapolation*	*Extrapolation*
O_9	*Extrapolation*	*Extrapolation*	*Interpolation*
O_{10}	*Interpolation*	*Extrapolation*	*Extrapolation*
O_{11}	*Extrapolation*	*Extrapolation*	*Extrapolation*

However, EMD metric is quite sensitive to the change in values and COG is the least complex metric.

5.5 Summary

This chapter has proposed an integrated approach to interpolation and inference. It applies α-cut to efficiently check whether direct inference can be performed using compositional rule of inference (CRI) in spite of the sparsity in the rule base. If more than one α-cut match between a given observation and any rule antecedent are found, then the rule which has the largest overlap is fired to derive the conclusion, using CRI. As such, it employs an interpolation method only when interpolation is essential, that is when there is no matching between the observation and any rule in the rule-base. This helps expedite the operation of the integrated system whilst improving the accuracy of the overall inference mechanism.

Table 5.9: Interpolation/Extrapolation Results by T-FRI (Based on Closest Rules by COG , HD and EMD Metrics)

Obs. O_i	Result Based on COG $Consequent_{COG}(B_{\circ,i})$	Result Based on HD $Consequent_{HD}(B_{\circ,i})$	Result Based on EMD $Consequent_{EMD}(B_{\circ,i})$
O_1	(9.16, 11.31, 10.66, 13.66)	(9.23, 10.47, 11.81, 13.59)	(9.23, 10.49, 11.79, 13.57)
O_2	(9.12, 10.28, 10.04, 11.52)	(9.1, 9.8, 10.6, 11.5)	(9.12, 10.17, 10.16, 11.52)
O_3	(9.77, 11.25, 10.65, 13.27)	(9.86, 11.07, 11.03, 13.24)	(9.86, 11.08, 11.02, 13.24)
O_4	(11.68, 13.01, 12.95, 13.98)	(11.79, 12.67, 13.31, 14.02)	(11.68, 13.05, 12.92, 13.98)
O_5	(10.4, 10.9, 11.8, 12.3)	(10.49, 10.96, 11.83, 12.33)	(10.49, 10.96, 11.83, 12.33)
O_6	(10.02, 10.83, 10.59, 11.72)	(10, 10.5, 11, 11.7)	(10.09, 10.78, 10.74, 11.73)
O_7	(9.53, 11.052, 10.34, 12.33)	(9.51, 10.8, 10.64, 12.31)	(9.53, 11.05, 10.34, 12.33)
O_8	(10.1, 11.3, 11.1, 12.3)	(10.1, 10.8, 11.6, 12.3)	(10.1, 10.8, 11.6, 12.3)
O_9	(10.8, 11.3, 12.1, 12.5)	(10.88, 11.36, 12.13, 12.52)	(10.9, 11.36, 12.13, 12.53)
O_{10}	(10.67, 11.31, 10.90, 12.67)	(10.73, 11.25, 11.18, 12.66)	(10.73, 11.24, 11.21, 12.65)
O_{11}	(8.78, 9.91, 9.67, 10.48)	(8.8, 9.5, 10.0, 10.5)	(8.78, 9.85, 9.72, 10.48)

Table 5.10: Comparison of Interpolation/Extrapolation Results

Metrics	Ground Truth vs. Result Based on COG $\epsilon_{\%COG}$	Ground Truth vs. Result Based on HD $\epsilon_{\%HD}$	Ground Truth vs. Result Based on EMD $\epsilon_{\%EMD}$
AVG	5.49	1.85	3.64
SD	3.53	2.25	2.37

The chapter has also investigated the use of three distance metrics: COG, HD and EMD to determine the closest rules for interpolation or extrapolation. HD metric calculates the relative distance between two fuzzy sets. It is therefore potentially more appropriate for finding the correct closest fuzzy sets as opposed to the conventional COG-based metric. Detailed experimental results compare and support this observation. EMD metric is also effective in multidimensional environment and calculates the distance by morphing one object into another. Nevertheless, COG is the traditional way of finding closest rules due to its less complex nature.

Chapter 6

Application of D-FRI for an Intrusion Detection System: D-FRI-Snort

In today's networked world, network security has become a major area of concern for organizations seeking to maintain the integrity of their IT infrastructure and the confidentiality of their data resources. Organizations increasingly implement and/or deploy various systems that monitor IT security breaches: firewalls, security/cryptographic systems, anti-virus products, network logging systems, vulnerability assessment tools, and so on [105]. Intrusion constitutes a major breach in computer security, and the concept of intrusion detection was introduced into network security discourse by J. P. Anderson in 1980 [10]. Intrusion is defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of system resources [2]. Firewalls are the most common security mechanism: however, they merely filter external unwanted packets, they do not thwart all intrusions, especially internal ones. For this reason, another layer of security is required, namely an intrusion detection system (IDS) [165].

An intrusion detection system (IDS) examines all inbound and outbound network activity and identifies malicious or suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system [10]. Intruders may be from outside the network or legitimate users of the network. The intrusion detection techniques can be classified into two main categories: anomaly detection and misuse detection [22]. Anomaly detection involves the recognition of changes in the patterns of utilization or behavior of the system, and can be used to detect known and unknown attacks. Misuse detection involves the discovery of intrusions that follow well-defined intrusion patterns; it is very useful in the detection of attacks that follow documented patterns of aggression.

Port scanning is one of the patterns of attack that an intrusion detection system seeks to detect [165]. Port scanning is a technique for identifying open ports and services available on a network host. To initiate a scan, a would-be intruder sends a series of messages in order to learn what network services, each associated with a well-known port number, a computer system provides. Hackers typically utilize port scanning because it is an easy way in which they can quickly discover services that they can compromise. In some cases, hackers are even able to open the ports themselves in order to access the targeted computer [6]. Hackers also use port scanners to conduct tests for open ports on personal computers that are connected to the web.

Snort, created by Martin Roesch in 1998, is the world's most powerful open source network intrusion detection system (NIDS). It is a packet sniffer that monitors network traffic in real time, scrutinizing each packet closely to detect a dangerous payload or suspicious anomalies [120]. It detects a port scanning attack by combining and analyzing various traffic parameters. Snort is easy to install and use and is sufficiently flexible to allow users to define their own IDS security rules. In 2009, Snort was inducted into InfoWorld's open source hall of fame [52].

Fuzzy logic lends itself to addressing the IDS problem in two ways [30]. Firstly, many quantitative features are involved in intrusion detection. The Stanford Research Institute's (SRI) next-generation intrusion detection expert system (NIDES) categorizes security-related statistical measurements into four types: ordinal, categorical, binary categorical and linear categorical. Both ordinal and linear categorical measurements are quantitative features that can potentially be viewed as fuzzy variables. Two examples of ordinal measurements are CPU usage time and connection duration [76]. An example of a linear categorical measurement is the number of different TCP/UDP (Transmission Control Protocol/ User Datagram Protocol) services initiated by the same source host.

Secondly, security itself is an inherently fuzzy domain. Given a quantitative measurement, an interval may be used to denote a normal value. Then, any values falling outside the interval will be considered anomalous to the same degree regardless of their distance to the interval. The same applies to values inside the interval, i.e., all will be viewed as normal to the same degree. The use of fuzziness in representing these quantitative features helps to smooth the abrupt separation of normality and abnormality and provides a measure of the degree of normality or abnormality of a particular measure [153].

This chapter presents and proposes an D-FRI-Snort IDS. This is the first and innovative application of D-FRI techniques to the problem domain of IDS. D-FRI enhances the functionality of port scanning detection and delivers another level of security to the IDS with higher levels of accuracy in alert predictions. The D-FRI facilitates the requirement of the most up-to-date dynamic fuzzy rule base. With the introduction of a D-FRI technique, it is possible to develop an IDS based on a sparse fuzzy rule base rather than on a dense fuzzy rule base as is currently the case with most IDS systems. The advantages are two-fold: firstly, it is very difficult to design a perfect dense rule base for IDS because intrusion detection is a network-based, online, and dynamic activity, and so some kind of incremental learning is required for each rule base modification; secondly, dynamic fuzzy interpolation techniques provide an appropriate result when no rule is matched with the input data.

The remainder of this chapter is organised as follows; section 6.1 gives a brief description and typology of intrusion detection systems; section 6.2 gives an account of port scanning activity in networking; section 6.3 discusses the world's most powerful IDS software, namely Snort; section 6.4 gives a description of the network analysis tools which are used for various experiments within this application; section 6.5 introduces the proposed dynamic integrated system (D-FRI and fuzzy inference system) based IDS (D-FRI-Snort); section 6.6 provides experimental results that demonstrate how the dynamic integrated system can be applied to Snort to enhance the efficiency of IDS; and finally, section 6.7 summarises this chapter.

6.1 Intrusion Detection System (IDS)

An intrusion detection system (IDS) is designed to: monitor network traffic, audit data, and examine traffic data for protocol anomalies, packet payload signatures that represent potential attacks, worms, and unusual/ suspicious activities; and alerts the system or network administrator [46]. It is a defense system to detect and possibly prevent hostile activities that may compromise system security. Certain IDS are capable of distinguishing between insider attacks originating from inside the organization (coming from employees or customers), and external ones (attacks and threats launched by hackers) [105]. IDS may have up to five components or modules for its operation, as shown in the Figure 6.1. Some popular IDS are Snort, OSSEC, OSSIM, Suricata, Bro, Fragroute, BASE, Kismet, and Sguil [181].

6.1.1 IDS Efficiency

An IDS is essentially a burglar alarm for the network: when intrusive activity occurs or appears to be occurring, the IDS generates an alarm to signal to system administration personnel that the network is possibly under attack. Like regular burglar alarms, however, the IDS can generate false positives (in which normal patterns are marked as an intrusion) or false negatives (in which abnormal patterns are not marked as an intrusion) [31]. The effectiveness of an IDS could be measured on the basis of detection rate and false alarm rate. The detection rate shows the percentage of true actions that have been successfully detected, and the false alarm rate is the sum of normal actions incorrectly parsed as an intrusion divided by the total number of normal actions [45]. A good IDS should provide a high detection rate together with a low false alarm rate; the extent to which it does so depends on the techniques and algorithms used in the IDS [197]. IDS results can be classified into the following four categories:

- *True Positive*: A true positive occurs when the IDS generates an alarm in response to abnormal or intrusive user activity. A true positive demonstrates that the IDS successfully detects a particular attack having occurred [16].
- *True Negative*: A true negative occurs when the IDS does not generate an alarm in response to normal user activity. A true negative demonstrates that the IDS successfully detects normal user activity [16].
- *False Positive*: A false positive occurs when the IDS generates an alarm in response to normal user activity. If the IDS generates too many false positives, then the IDS fails to protect the network. If a burglar alarm regularly sounds incorrectly, the police will become conditioned to the fact that the alarmed locale is prone to false alarms. During an actual break-in, the police may respond sluggishly or not at all, judging the alarm to be false [31]. Therefore,

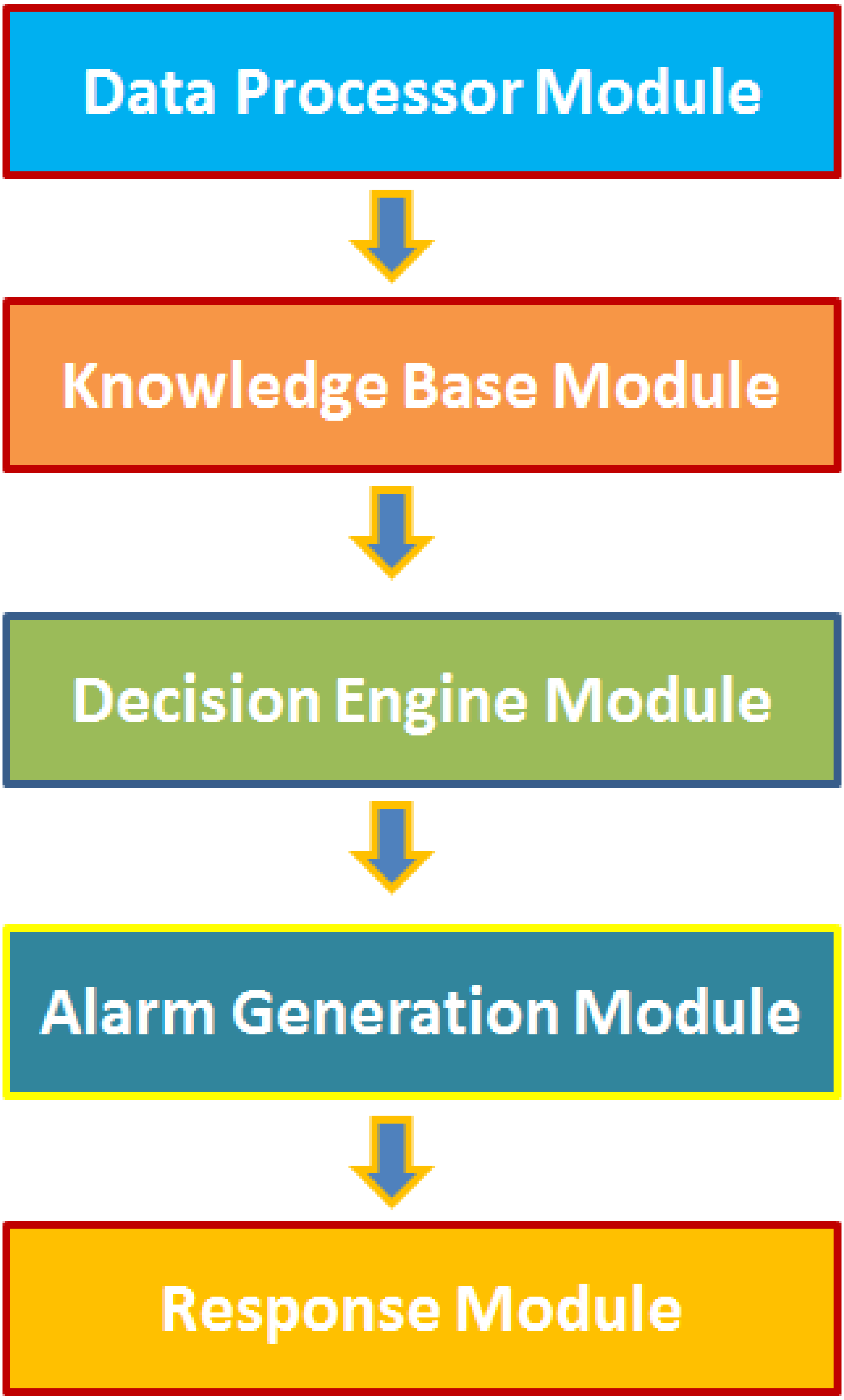


Figure 6.1: Intrusion Detection System (IDS)

it is crucial that the IDS be configured to minimize the number of false positives that it generates.

- *False Negative*: A false negative occurs when the IDS does not generate an alarm in response to abnormal or intrusive user activity. In this situation, an attack occurs against the network and the IDS fails to alarm even though it is designed to detect such an attack. The IDS should almost never generate false negatives. In fact, it is preferable for your IDS to actually generate false positives rather than generate any false negatives [31].

6.1.2 Types of IDS

The classification of IDS depends on their various modes of intrusion detection. Some look for specific signatures of known threats - similar to the way in which antivirus software typically detects and protects against malware - and others compare traffic patterns against a baseline and seek to identify anomalies. In general, intrusion detection systems can be classified into two main categories: anomaly-based, and misuse/signature-based [22, 165, 174].

- *Anomaly-Based Intrusion Detection*: In anomaly-based systems, the system administrator defines the normal or baseline state of the network. Then the IDS monitors network traffic and compares it against stored patterns of normal behavior. The normal behavior patterns codify and encapsulate what is normal for that network - what sort of bandwidth is generally used, what typical packet size is used, what protocols are used, what ports and devices generally connect to each other- and alerts the administrator or user when traffic is detected which is anomalous, or significantly different from the normal behavior.

- *Misuse/Signature-Based Intrusion Detection*: In misuse-based systems, the system administrator maintains a large database of attack signatures (specific attacks that have already been documented). The IDS monitors network traffic and compares it against stored patterns of abnormal or intrusive behavior (signatures or attributes from known malicious threats). This is similar to the way most antivirus software detects viruses and malware. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to the IDS. During that interim period the IDS would be unable to detect the new threat.

An IDS can also be either host- or network-based. A host-based system looks for intrusions on that particular host. Most of these programs rely on secure auditing systems built into the operating system. Network-based systems monitor a network for the tell-tale signs of a break-in on another computer. Most of these systems are essentially sophisticated network monitoring systems that use Ethernet interfaces as packet sniffers [8].

6.2 Port Scanning

Whenever a criminal targets a property for a burglary, usually the first part of the operation is to check if there is an open window or door through which access can be gained. A port scanning is analogous to burglary; the only difference is that the windows and doors are the ports of computers or other digital devices. However, hackers may not opt to break in immediately, they will have determined if easy access is available [147]. This explorative phase uses a port scan to sends client requests to a range of server port addresses on a host, with the goal of finding an active port and exploiting a known vulnerability of that service. Port scanning is an exploration phase and is considered the first stage of a computer attack.

The main aim of the port scanning is to find open ports on a system. Open ports give rise to opportunities for potential loss of data, drive by virus infection, and at times, even complete system compromise [161]. New security risks arise on a daily basis therefore it is essential for the user to protect their virtual files. The use of computer protection by making use of anti-virus, intrusion detections systems and intrusion protection systems should be considered as a top priority for those using computers in both the home, commercial and governmental environments [193].

6.2.1 Computer Ports

In computing, the term port refers either to physical connection or to a virtual point of connection at the software level. In computer hardware terminology, a port is usually a specific physical location on the device for connecting it to various other devices. Generally this involves a socket and plug. Whereas, a port at the software level is used in network communications: it is a logical, rather than physical, connection [143]. The devices and computers connected to the Internet use a protocol called transmission control protocol/ internet protocol (TCP/IP) to communicate with each

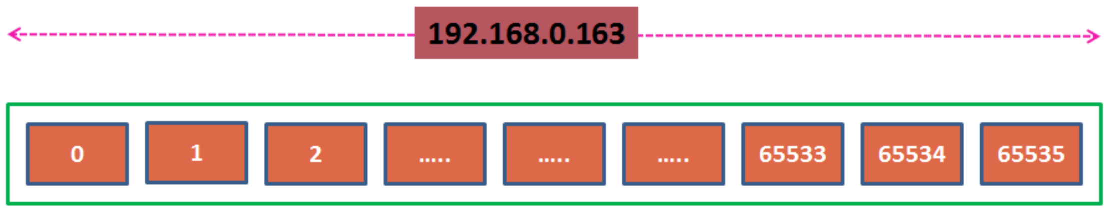


Figure 6.2: IP address with ports

other. In TCP/IP, designating a port is the way a client program addresses a particular server program on a computer in a network.

The IP address identifies a particular computer (network interface) out of the millions of other computers connected to the Internet. This IP address has several thousand potential ports through which it may receive or send data from or to other hosts (see Figure 6.2). When a program on a computer host sends data over the Internet it sends that data to an IP address and a specific port, and it receives data usually via random port on one of its own host's network interfaces. If it uses the TCP protocol to send and receive the data then it will connect and bind itself to a TCP port. If it uses UDP (user datagram protocol) to send and receive data, it will use a UDP port. Note that once an application binds itself to a particular port, that port cannot be used by any other application. It is a first come, first served system [1].

There are 65,536 standard ports on a computer. It can use a total of 65,535 TCP Ports and another 65,535 UDP ports. The Internet Assigned Numbers Authority (IANA) has broken down these port numbers into three categories (i) well known (0 - 1023), (ii) registered (1024 - 49151), and (iii) dynamic and/or private Ports (49152 - 65535) [1]. Port scanning usually targets TCP and UDP ports in the first range. Some common ports and their associated services are shown in the Table 6.1.

6.2.2 Types of Port Scanning

In a TCP connection, a 3-way handshake occurs to make a connection as shown in Figure 6.3. When a client wants to connect with a server, it first sends a TCP packet with the SYN (Synchronize Sequence Number) flag set. The server then sends back a TCP packet with the SYN and ACK (Acknowledge) flags set if the port is open on the server. A RST (Reset) packet is sent to the client if the port is closed. If the port is open and the server sends back the SYN|ACK packet, the client computer then

Table 6.1: Commonly exploited ports and services

Port ID	Service
7	Echo
19	Chargen
$20-21$	FTP (File Transfer Protocol)
23	Telnet (Remote Login)
25	SMTP (Simple Mail Transfer Protocol)
43	Whois
53	DNS (Domain Name System)
69	TFTP (Trivial File Transfer Protocol)
79	Finger
80	HTTP-low (Hyper Text Transfer Protocol)
107	Rtelnet
110	POP3 (Post Office Protocol)
111/2049	SunRPC (Remote Procedure Calls)
$135-139$	NBT (Net BIOS over TCP/IP)
$161, 162$	SNMP (Simple Network Management Protocol)
512	Exec
513	Login
80	HTTP-low (Hyper Text Transfer Protocol)
514	Shell
$6000-xxxx$	X-Windows
8000	HTTP (Hyper Text Transfer Protocol)
8080	HTTP (Hyper Text Transfer Protocol)
514	Shell
31337	Back office

sends an ACK back to the server [47]. The simplest TCP port scanning is done by the method TCP connect() and it will succeed if the port being scanned is listening, otherwise it will fail. There are various types of port scanning and some of them are given here and and shown in Figure 6.4.

- *Stealth Scan*: Such a scan is designed to go undetected by auditing tools. It sends TCP packets to the destination host with stealth flags. Some of the flags are SYN (synchronize sequence number), FIN (finish) and NULL [47].
- *SOCKS Port Probe*: A SOCKS port allows sharing of Internet connections on multiple machines. Attackers scan these ports because a large percentage of users often misconfigure SOCKS ports, potentially permitting arbitrarily chosen

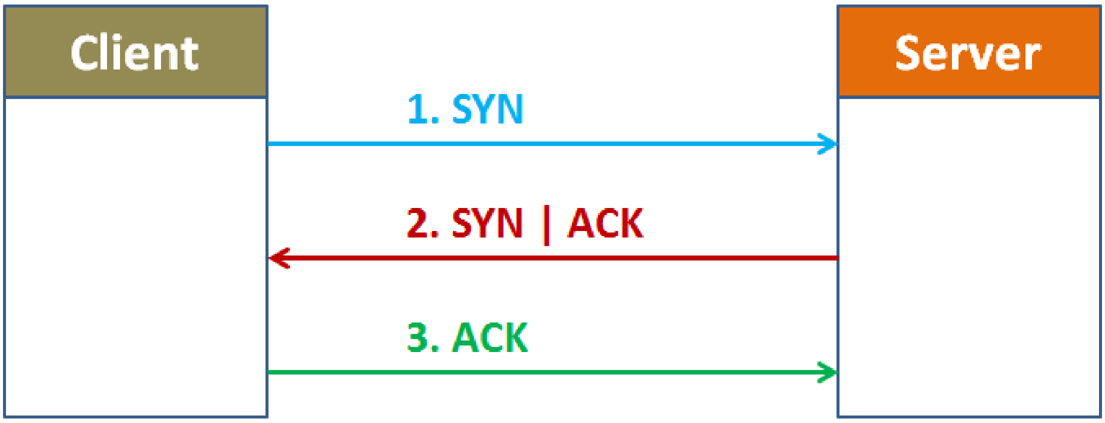


Figure 6.3: TCP 3-way handshake for establishing connection

sources and destinations to communicate. A SOCKS port on a system may allow the attacker to access other Internet hosts while hiding his or her true location [21].

- *Bounce Scan*: A bounce scan takes advantage of a vulnerability of the FTP (file transfer protocol). Some applications that potentially allow bounce scans are email servers and HTTP (hyper text transfer protocol) proxies [21].

- *TCP Scanning*: A TCP connection is never fully established during this type of scanning. If an attacker knows that a remote port is accepting connections, he or she can launch an attack immediately [21]. This is much more difficult for network defenders to detect since connection attempts of this kind are not logged by the server's logging system. Some TCP scans are TCP Connect(), reverse identification, IP header dump scan, SYN, FIN, ACK, XMAS, NULL and TCP fragment [47].

- *UDP Scanning*: UDP scanning attempts to find open UDP ports. However, UDP is a connectionless protocol and therefore it is not often targeted by attackers since it may be easily blocked [21].

6.3 Snort

Snort is an open source network intrusion detection and prevention system capable of performing real-time traffic analysis and packet-logging on IP networks [80]. Snort is based on libpcap (library packet capture), a tool that is widely used in TCP/IP

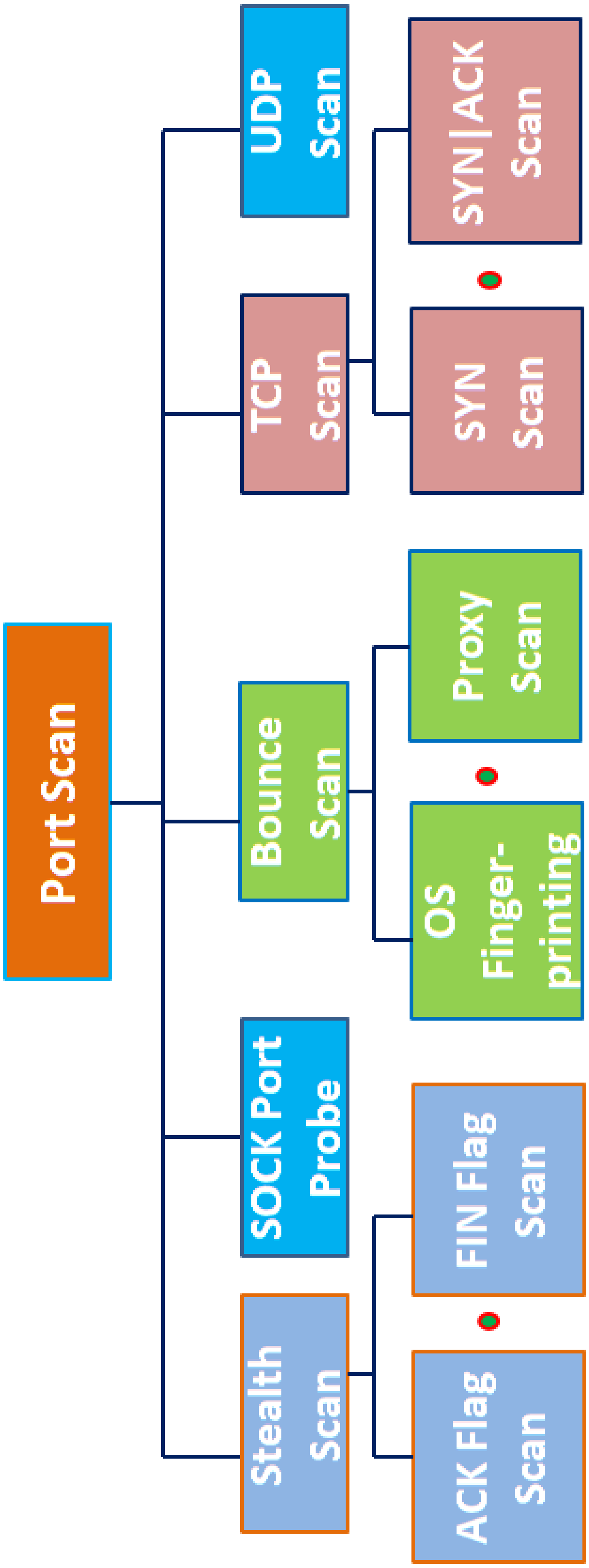


Figure 6.4: Common types of port scanning

traffic sniffers and analyzers [159]. Through protocol analysis and content searching and matching, and various pre-processors, Snort detects attack methods, including denials of service (DoS), buffer overflows, CGI attacks, stealth port scans, SMB probes, worms, and OS fingerprinting attempts [156]. When suspicious behavior is detected, Snort sends a real-time alert to syslog, a separate alerts file, or to a pop-up window. Snort uses a flexible rule-based language to describe traffic that it should collect or pass, and a modular detection engine [187]. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user-specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient [122].

NSS (National Security Systems) Group, a European network security testing organization, tested Snort along with IDS products from 15 major vendors including Cisco, Computer Associates, and Symantec. According to NSS, Snort, which was the sole open source product tested, clearly out-performed the proprietary products [159]. At the time of writing, Snort licenses are free, so any company or user can install and use it for their own network security purposes.

6.3.1 Snort Installation

Snort installation is a potentially lengthy process and requires a good understanding of the Snort documentation. In the current implementation, Snort is installed on a Windows machine, but it may also be installed on Linux or Mac machine. Installation and configuration involved the following steps:

1. The WinPcap interface, Snort's principal dependency, was installed from the www.winpcap.org website (see Figure 6.5). In other contexts, this may be present in a system where a network analysis tool (such as Wireshark or Nmap) has already been installed.

2. The most recent version of Snort (Snort_2_9_6_2_Installer.exe) was downloaded from www.snort.org (see Figure 6.6). Once installed, its root directory can be found as shown in Figure 6.7.

3. In order to use Snort as an IDS, a set of rules were downloaded (snortrules-snapshot-xxxx.tar) from www.snort.org (see Figure 6.8). The rules were decompressed (Figure 6.9) and three folders (preproc_rules, rules, and so_rules) were copied to Snort's root directory (see Figure 6.7).

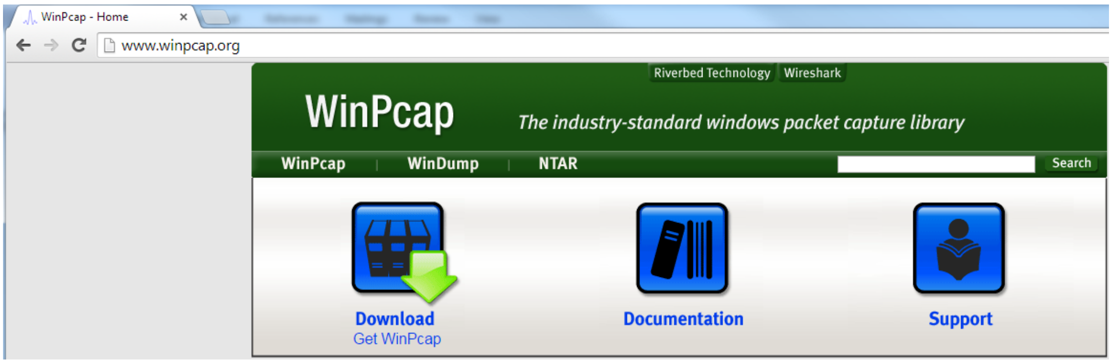


Figure 6.5: Download the latest version of WinPcap from winpcap.org

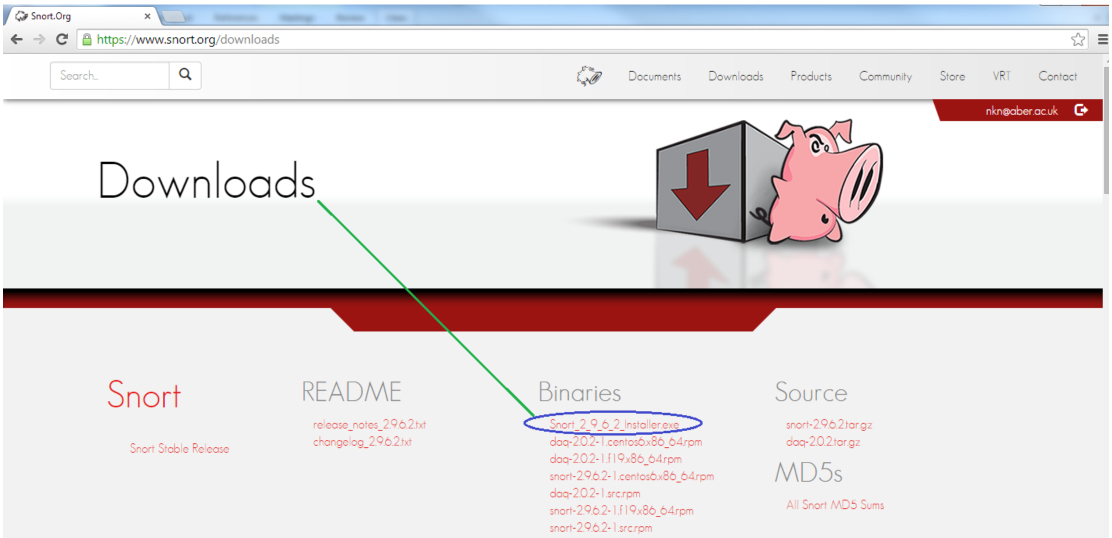


Figure 6.6: Download the latest version of Snort installer from snort.org

4. Finally, etc/snort.conf was edited according to network requirements (see Figure 6.10).

After modifying the etc/snort.conf file, the Snort installation was tested (see Figure 6.11). If etc/snort.conf is correctly modified then the above command will report the test as successful as shown in Figure 6.12.

6.3.2 Snort Working Modes

Snort can be operated in three modes. It can be used as a straight packet sniffer like tcpdump, a packet logger (useful for network traffic debugging), or as a full-blown network intrusion detection/prevention system. Snort sniffer mode (shown in Figure 6.13) is the simplest mode in which Snort displays TCP/IP/UDP/ICMP headers, application data and data link layer headers.

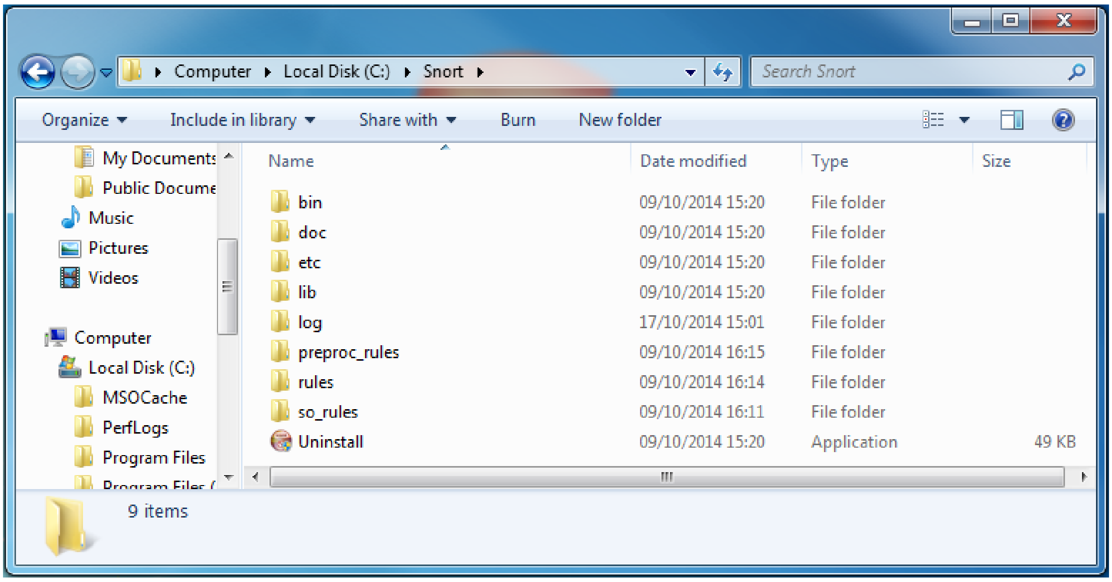


Figure 6.7: Different folders of Snort root directory

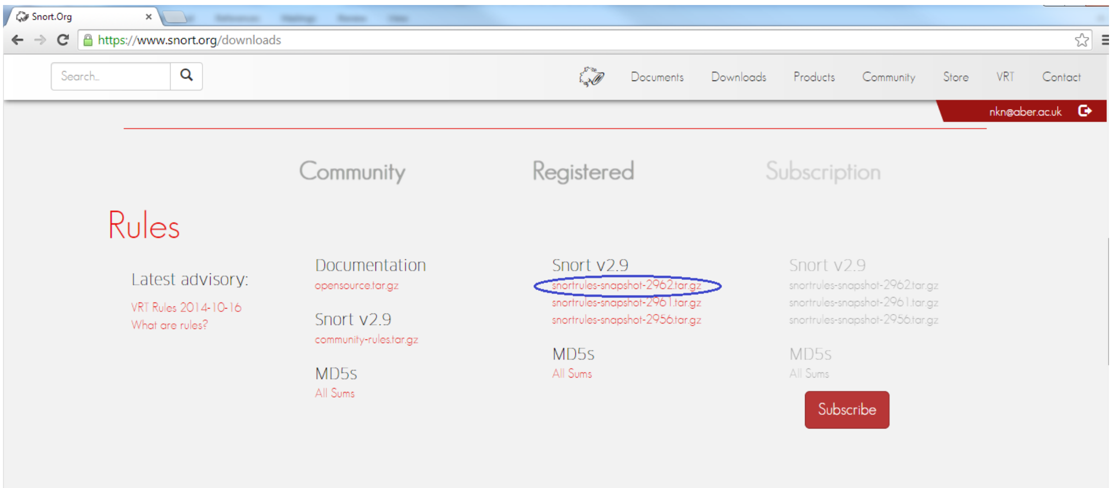


Figure 6.8: Download the latest version of Snort rules from snort.org

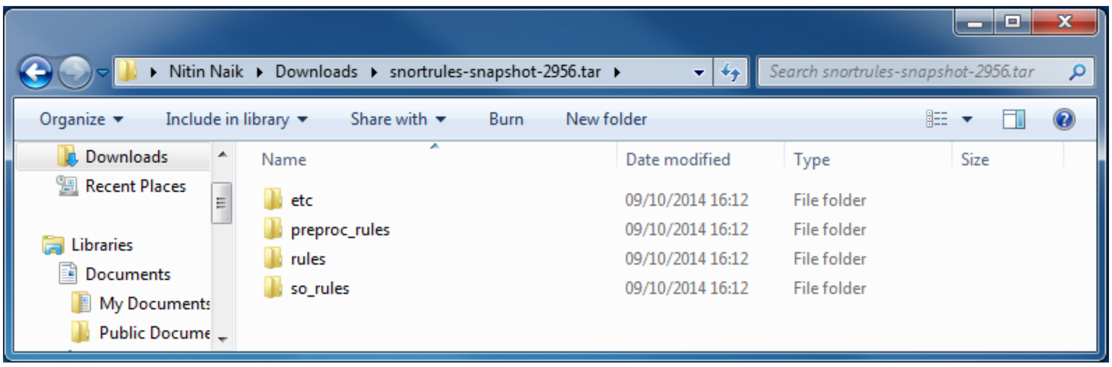


Figure 6.9: Unzip Snort rules

```
#--------------------------------------------------
#   VRT Rule Packages Snort.conf
#
#   For more information visit us at:
#     http://www.snort.org                   Snort Website
#     http://vrt-blog.snort.org/    Sourcefire VRT Blog
#
#     Mailing list Contact:      snort-sigs@lists.sourceforge.net
#     False Positive reports:    fp@sourcefire.com
#     Snort bugs:                bugs@snort.org
#
#     Compatible with Snort Versions:
#     VERSIONS : 2.9.6.2
#
#     Snort build options:
#     OPTIONS : --enable-gre --enable-mpls --enable-targetbased
--enable-ppm --enable-perfprofiling --enable-zlib --enable-
active-response --enable-normalizer --enable-reload --enable-
react --enable-flexresp3
#
#     Additional information:
#     This configuration file enables active response, to run
snort in
#     test mode -T you are required to supply an interface -i
<interface>
#     or test mode will fail to fully validate the configuration
and
#     exit with a FATAL error
#--------------------------------------------------

###################################################
# This file contains a sample snort configuration.
# You should take the following steps to create your own custom
configuration:
#
#  1) Set the network variables.
#  2) Configure the decoder
#  3) Configure the base detection engine
#  4) Configure dynamic loaded libraries
#  5) Configure preprocessors
#  6) Configure output plugins
#  7) Customize your rule set
#  8) Customize preprocessor and decoder rule set
#  9) Customize shared object rule set
###################################################

###################################################
# Step #1: Set the network variables.  For more information, see
README.variables
###################################################
```

Figure 6.10: Snort.conf file

Figure 6.11: Testing command for snort.conf file

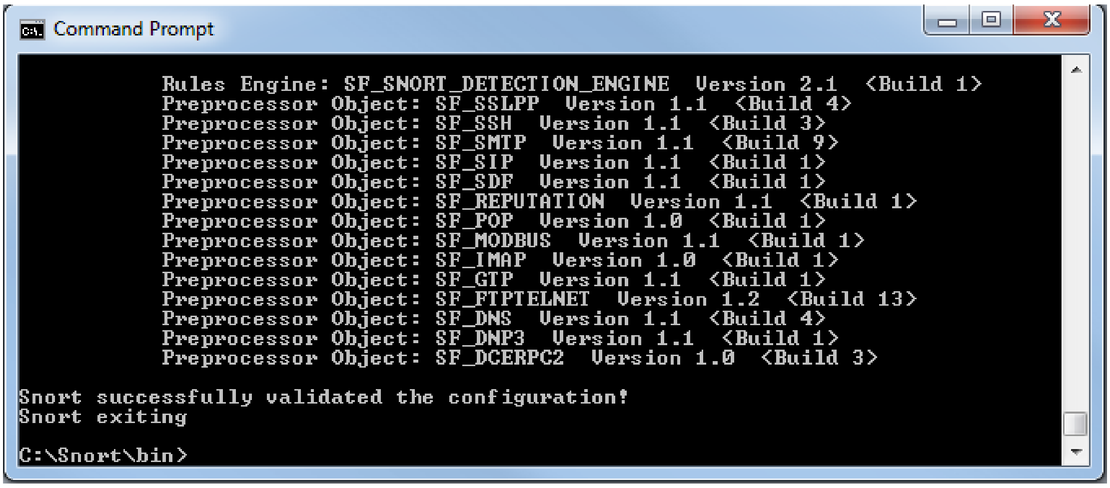


Figure 6.12: Successful testing of snort configuration

Figure 6.13: Snort in sniffer mode

Figure 6.14: Snort in packet logger mode

Figure 6.15: Snort in NIDS mode

Snort packet logger mode (shown in Figure 6.14) logs the sniffer mode data to c:\Snort\log\snort.log.xxxx (all alerts and packet logs are written to c:\Snort\log).

Snort NIDS mode (shown in Figure 6.15) is the most complex and configurable mode which performs detection and analysis on network traffic. This is the preferred mode for the current implementation.

6.4 Network Analysis Tools used in D-FRI-Snort

In the present implementation of D-FRI-Snort (see section 6.5), a number of other network analysis tools (such as Wireshark, NMAP, Winpcap, Advanced Port Scanner and Basic Analysis and Security Engine (BASE)) are used for the purpose of experimentation. D-FRI-Snort is already installed on Windows platforms and all other networking tools used in this application will also be installed on a Windows machine.

6.4.1 Wireshark

Wireshark is a network protocol analyser formerly known as Ethereal (shown in Figure 6.16). It captures packets in real time and displays them in human-readable format. Wireshark includes filters, color-coding and other features that allow an administrator to delve into network traffic and inspect individual packets [39]. It is the industry standard. It is the continuation of a project that started in 1998. Hundreds of developers around the world have contributed to it, and it is still under active development [135].

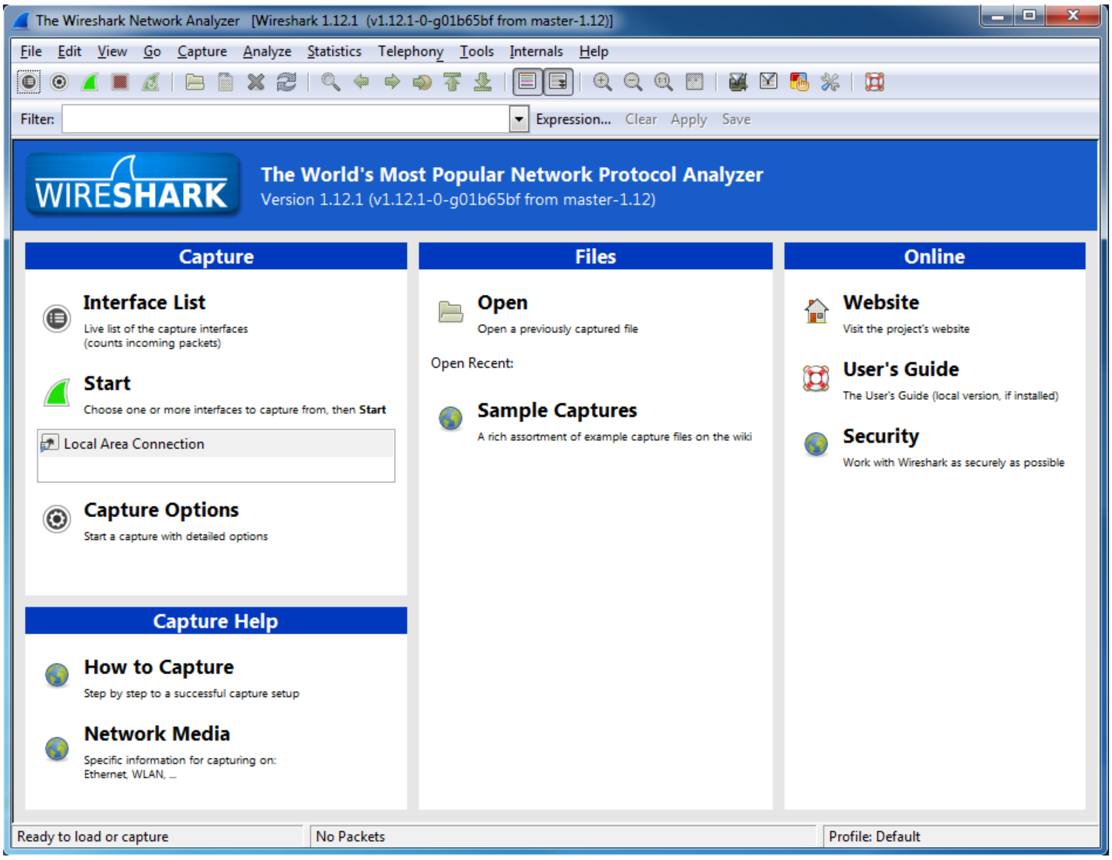


Figure 6.16: Wireshark network analysis tool

Wireshark can read or write data in many different capture file formats: tcpdump (libpcap), Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, NAI Sniffer (compressed and uncompressed), Sniffer Pro, NetXray, Network Instruments Observer, Novell LANalyzer, RADCOM WAN or LAN Analyzer, Shomiti or Finisar Surveyor, Tektronix K12xx, Visual Networks Visual UpTime, and WildPackets EtherPeek, TokenPeek, or AiroPeek [23]. In this implementation, Wireshark is used to analyse Snort log files and network data for various network parameters (as shown in Figure 6.17) that are later used in the design of fuzzy rules and to predict alert levels .

6.4.2 NMAP

Network Mapper (Nmap) devised by Gordon Lyon, shown in Figure 6.18, is a free and open source utility for network discovery and security auditing [132]. The discovery of hosts and services on a computer network, thus creating a map of the network has been used frequently by network administrators and other technical personnel for

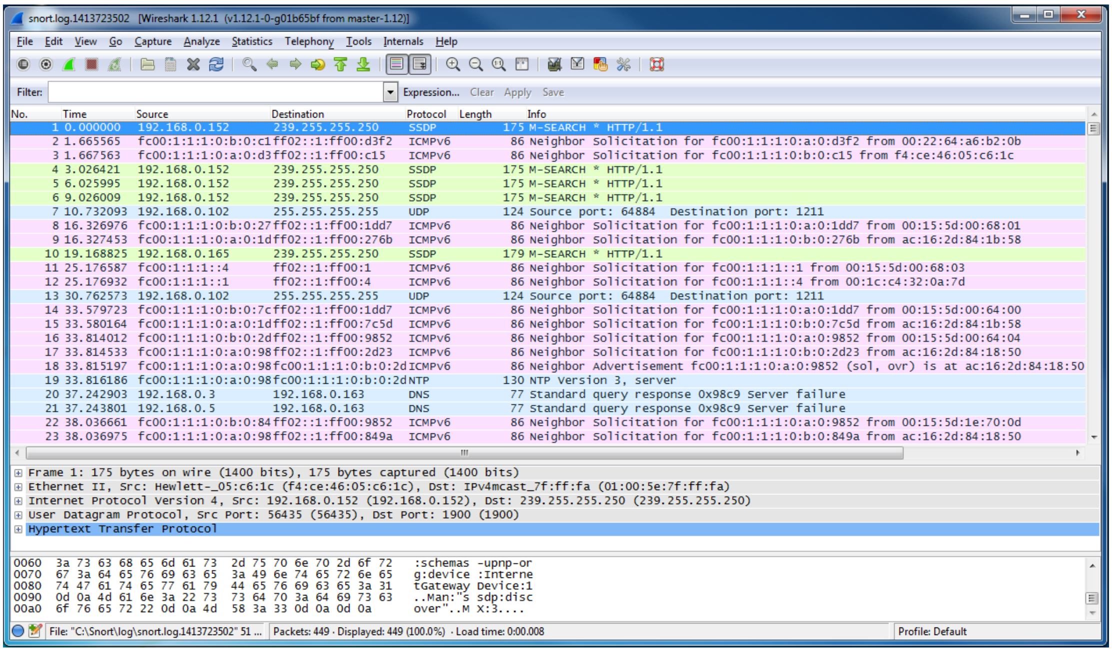


Figure 6.17: Capturing/analysing network traffic data using Wireshark

many years. Indeed, it is also used for many other tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime [133]. Nmap uses raw IP packets to determine the following [152]:

- What are the hosts available on the network;
- What are the application services and their versions offered by the hosts;
- What are the operating systems and their versions on the hosts;
- What types of packet filter or firewall are in use;
- What are the reverse DNS (Domain Name System) names;
- What are the MAC (Media Access Control) addresses.

There are many other characteristics that can also be recorded by this software, which was designed to rapidly scan large networks and single hosts [152].

Nmap is designed to run on all major computer operating systems, with official binary packages available for Linux, Windows, and Mac OS X. Nmap provides two ways of operation, command line and a Graphical User Interface (GUI). In addition

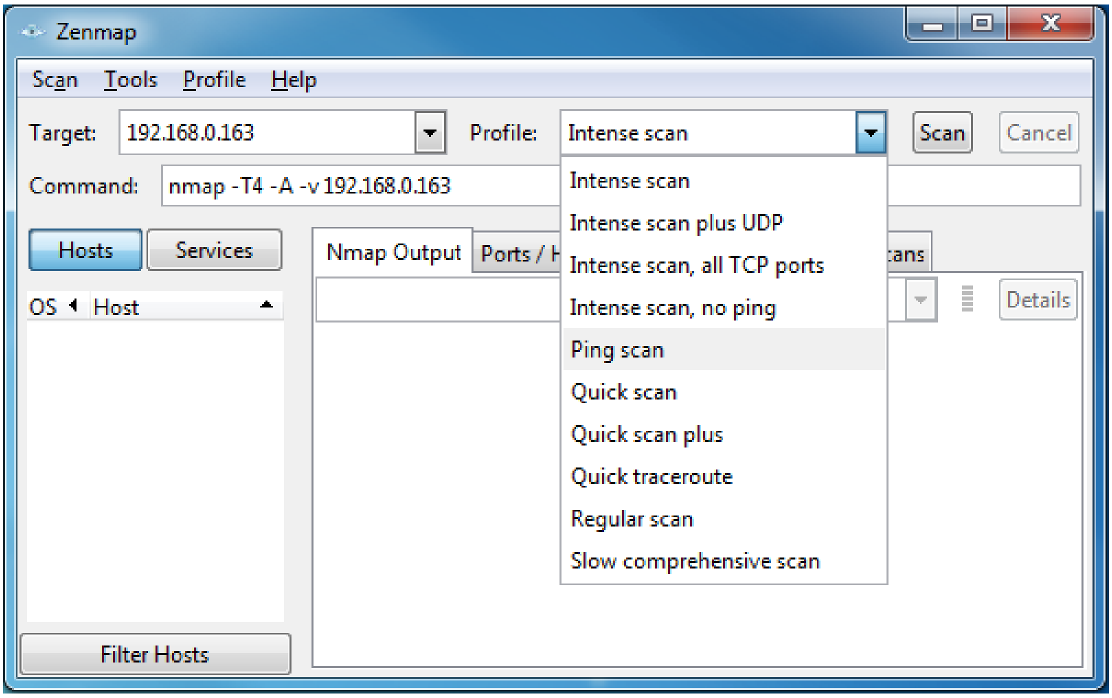


Figure 6.18: NMAP's GUI utility Zenmap and its possible scanning methods

there is a results viewer (Zenmap) that is shown in Figure 6.18, a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping) [132]. In this implementation, Nmap is used to execute a port scan attack on computer.

6.4.3 WinPcap

For Windows environments specifically, there is an industry standard tool called WinPcap, providing link-layer network access. This tool allows applications to capture and transmit network packets bypassing the protocol stack. It is also useful as a kernel-level packet filtering, a network statistics engine and support for remote packet capture [185]. It consists of two important components: (1) a driver that supports the operating system to provide low-level network access, and (2) a library that is used to access the low-level network layers. This library also allows WinPcap to access a Unix API.

Many commercial network tools make use of WinPcap as their base, for example Wireshark, Nmap and Snort. This is not an exhaustive list, but does illustrate its popularity and widespread use in other commercial software [100]. These analysers are well regarded within the industry and used in many implementations of by a number of software manufacturers. In this implementation, WinPcap is the most vital interface software which is used with all the above tools including Snort.

6.5 Dynamic Fuzzy Rule Interpolation Snort (D-FRI-Snort)

This section presents the innovative fuzzy network intrusion detection system D-FRI-Snort. It is a practical application of the current research work on dynamic fuzzy rule interpolation. It is a variant of Snort based on the proposed dynamic integrated system (fuzzy inference system and D-FRI) that enhances Snort by adding further levels of security alert and reducing false positive and false negative results.

6.5.1 D-FRI-Snort Architecture

D-FRI-Snort architecture has three main components: Snort, a fuzzy inference system, and a D-FRI system as shown in Figure 6.19. Snort is a powerful port scanner that is used here in NIDS mode. Snort collects network traffic and performs monitoring and analysis of traffic data. Then the values of some selected network parameters such as average packet time, number of packets sent and received are passed to the fuzzy inference system. The fuzzy inference system assists Snort in deciding the level of port scan attack using these parameter values, something which is not covered by the original Snort [60]. The fuzzy inference system uses a sparse fuzzy rule base and inference engine for its reasoning [178]. The design of the sparse fuzzy rule base is based on three selected networking parameters derived from extensive investigations and experiments carried out during the port scan attack testing. If no suitable match is found in the sparse fuzzy rule base then D-FRI generates interpolated results and predicts the alert level. It also accumulates interpolated results to generate new fuzzy rules and update the sparse fuzzy rule base. This is a process already developed and tested successfully in this research. In this investigation, the same transformation-based interpolation technique is used.

6.5.2 Data Collection

The data collection task was performed by six computers. Five computers were used to orchestrate a port scan attack on a host computer. The port scan attack was executed using the NMAP network scanning tool. Network traffic data was collected

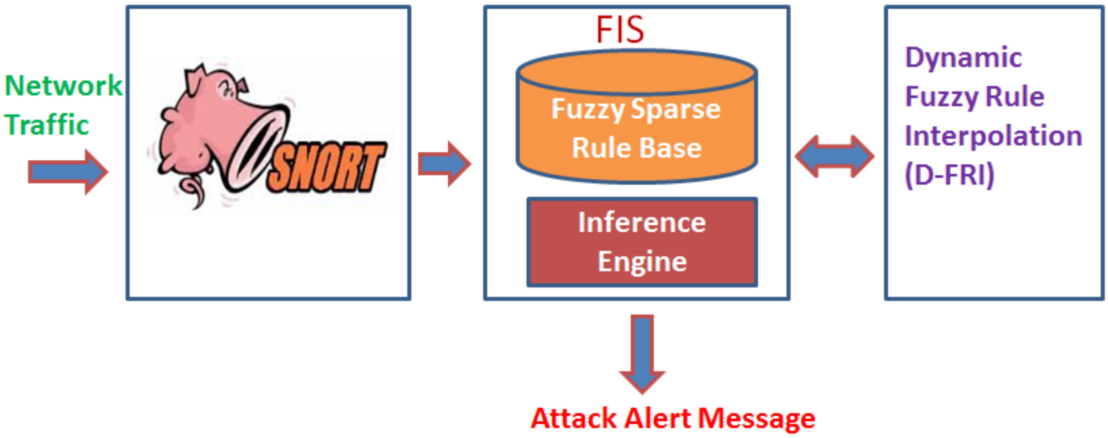


Figure 6.19: D-FRI-Snort

by Snort to be analysed for further processing. Snort captures many important parameters such as: IP addresses, protocol details, average packet time, number of received packets, number of sent packets, packet size, packet format, and dropped packets.

Five rounds of port scanning activities were carried out on a host with a varying number of attacking computers. In the first round, the port scan attack was executed on the host computer by only one computer. In the second round, the attack was carried out by two computers, and in the next round by three computers, and so on until the fifth round. Every round went through ten iterations to obtain the range of values. The time interval of each iteration observation was 5 minutes and 20 seconds, although this time was chosen somewhat arbitrarily. The rounds were named as follows:

1. VL (very low) port scan attack: one computer against a single host.
2. L (low) port scan attack: two computers against a single host.
3. M (medium) port scan attack: three computers against a single host.
4. H (high) port scan attack: four computers against a single host.
5. VH (very high) port scan attack: five computers against a single host.

Following careful consideration, three parameters were selected as fuzzy input variables from all the parameters collected by Snort [50, 51, 61, 67, 72, 106, 164, 166, 170]. These were: (1) the average time for a packet received by the destination/victim in milliseconds (ATP); (2) the number of packets sent by the source (NPS); and (3) the number of packets received by the destination/victim (NPR) as shown in Figure 6.20. Later, these parameters were utilized in the design of the fuzzy rule base and D-FRI system. These parameters were used in the fuzzy inference system to calculate the level of attack, which was an additional enhancement for Snort. For the most part, web traffic was ignored for this scope of the investigation.

6.5.3 Data Analysis

Data analysis was accomplished by a combination of the Wireshark networking tool and Snort. Ranges of values for only three selected parameters (ATP, NPS, and NPR) were collected and analysed for each round. Time is the most effective parameter in the investigation of port scanning attacks [61]and for this reason, average packet time was chosen as one of the deciding criteria in the current investigation. It clearly differentiated the level of attack during the analysis of the range of values for ATP. In the first round, it showed a very high value when very low port scanning was used, and then it gradually decreased with the lowest value in the fifth round when five computers were used for the port scan attack. The second parameter was NPS, again a decisive factor for determining the level of attack as it increased in proportion to the number of attackers. In the first round, it showed the lowest value and then it gradually increased with the highest value in fifth round when five computers were used for the port scan attack. The third parameter was NPR, also a significant element for determining attack level, and like NPS, it increased in proportion to the number of attackers. In the first round, it showed the lowest value and then it gradually increased with the highest value in the fifth round when five computers were used for the port scan attack.

6.5.4 Fuzzy Sets and Rules Generation

The above-mentioned three parameters were used as three input fuzzy variables. For each variable, five linguistic terms/fuzzy sets - very low (VL), low (L), medium (M), high (H) and very high (VH) - were used to represent their ranges. These fuzzy sets were represented by a triangular membership function as D-FRI is also implemented using triangular membership function. These three input variables were chosen to determine the level of port scan attack (PSA), the output fuzzy variable. The output fuzzy variable was also divided into five similar fuzzy sets, such as input variables. These input and output variables and their fuzzy sets are shown in Figure 6.21.

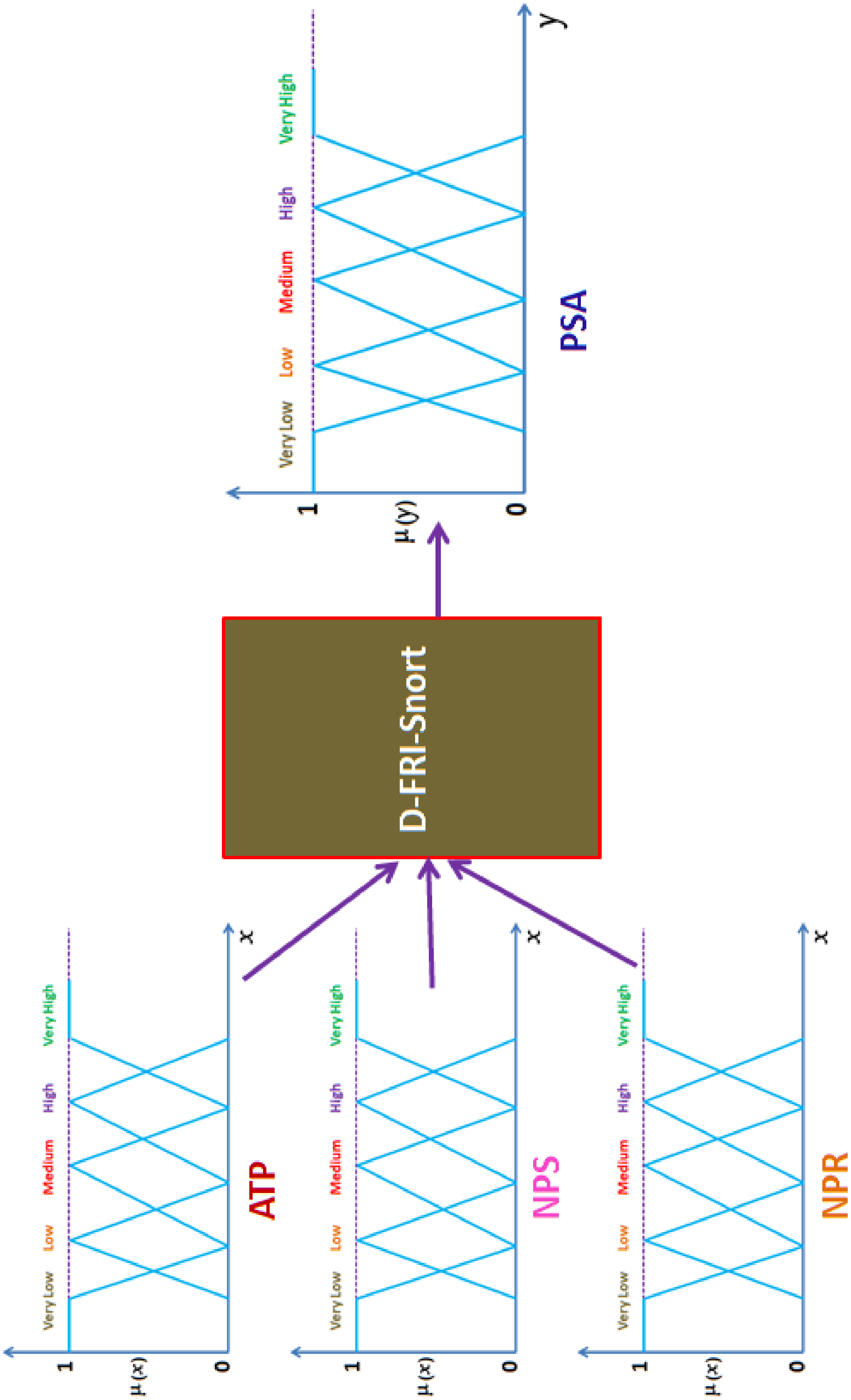


Figure 6.20: D-FRI-Snort fuzzy input and output variables

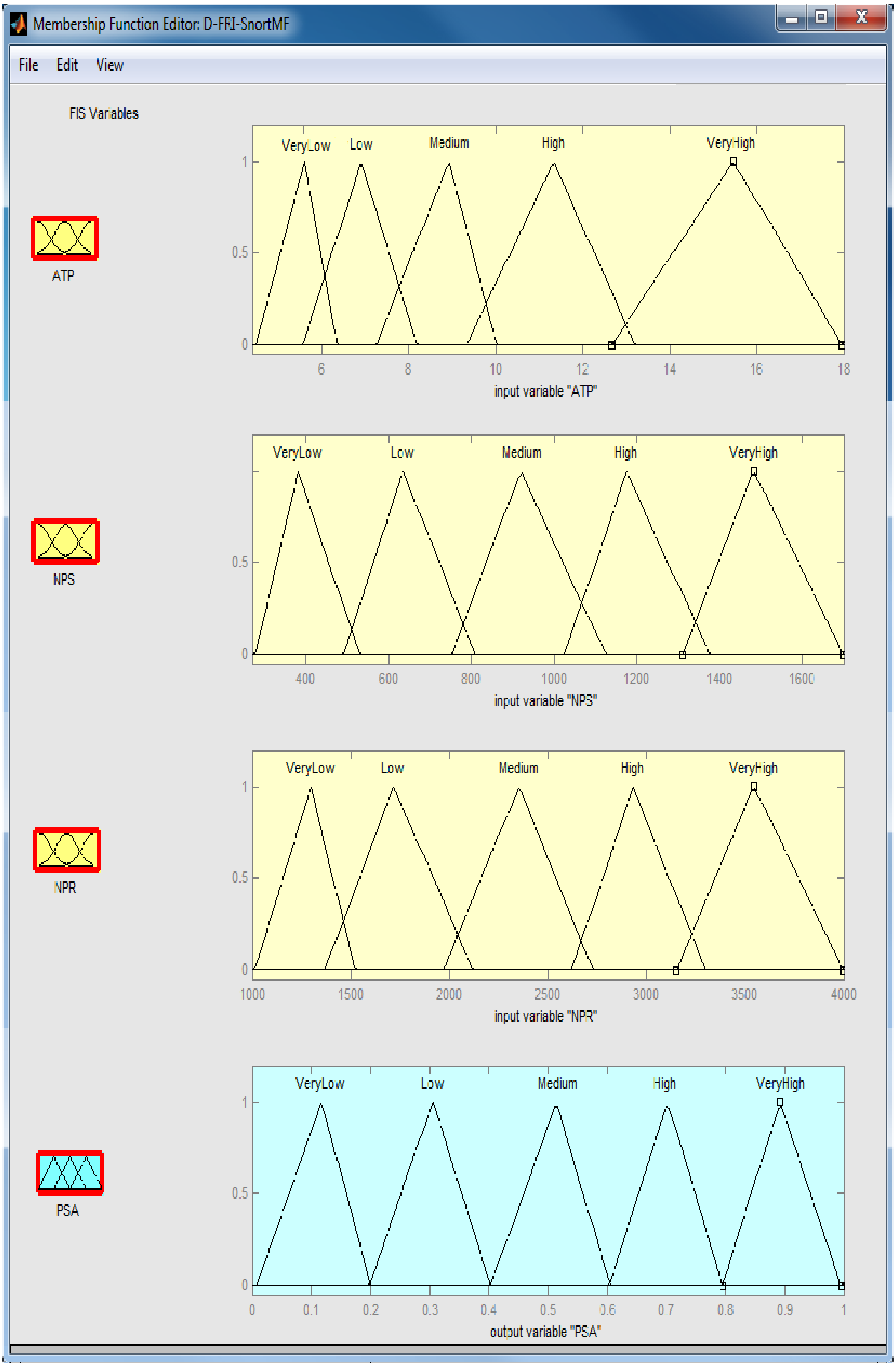


Figure 6.21: D-FRI-Snort fuzzy sets and their membership functions

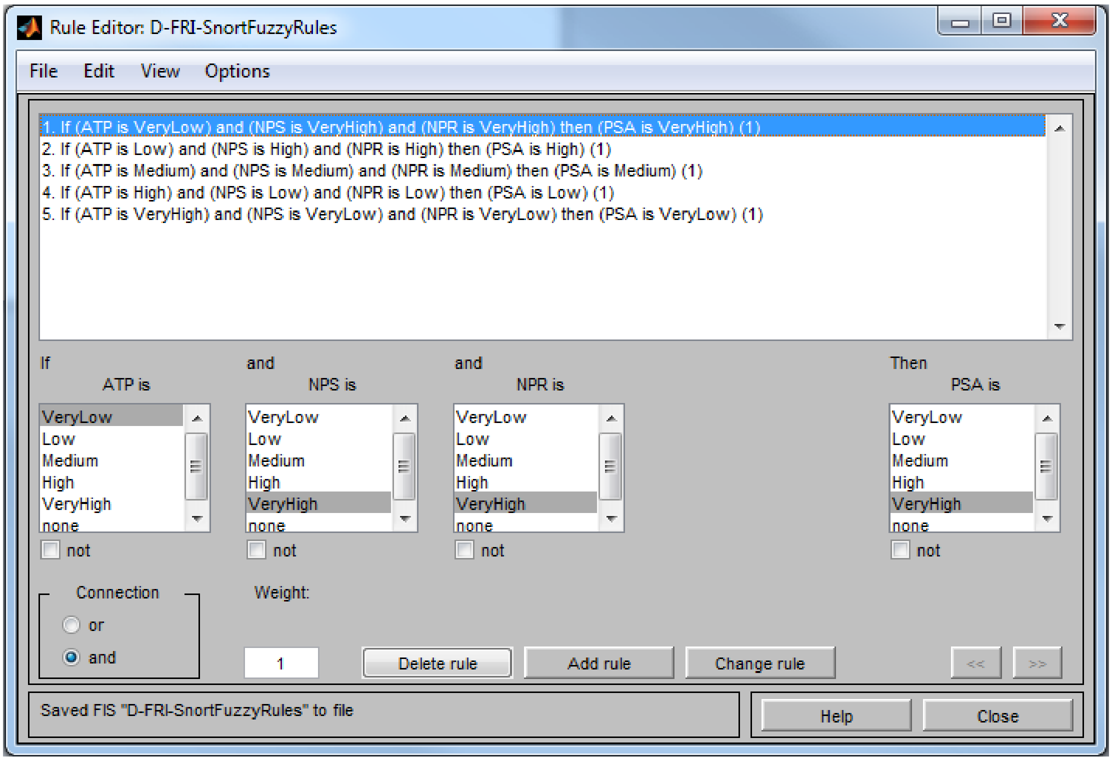


Figure 6.22: D-FRI-Snort fuzzy rules

The fuzzy rules were designed and tested based on the input and output fuzzy variables and their corresponding fuzzy sets [43]. These rules were written depending on: the obtained value ranges; expert knowledge of detecting the port scanning attack; and the relationship between the parameters used to detect that attack in all the five rounds. Here, the fuzzy logic toolbox in Matlab is used to design fuzzy rules and fuzzy rule base. Amongst the designed fuzzy rules, some examples of the selected fuzzy rules to detect the port scan attack are shwon in Figure 6.22.

Finally, the initial sparse fuzzy rule base of 30 rules was achieved (as shown in Figure 6.23). It was developed based on the long and extensive investigations of network traffic data through various networking tools.

6.5.5 D-FRI Operation

This application assumes that it is difficult to obtain all possible rules to cover all the input conditions from the captured traffic data. The fuzzy rule base given above is a sparse fuzzy rule base and cannot cover all possible input observations. In this situation, Snort would not be able to give correct results for those observations which

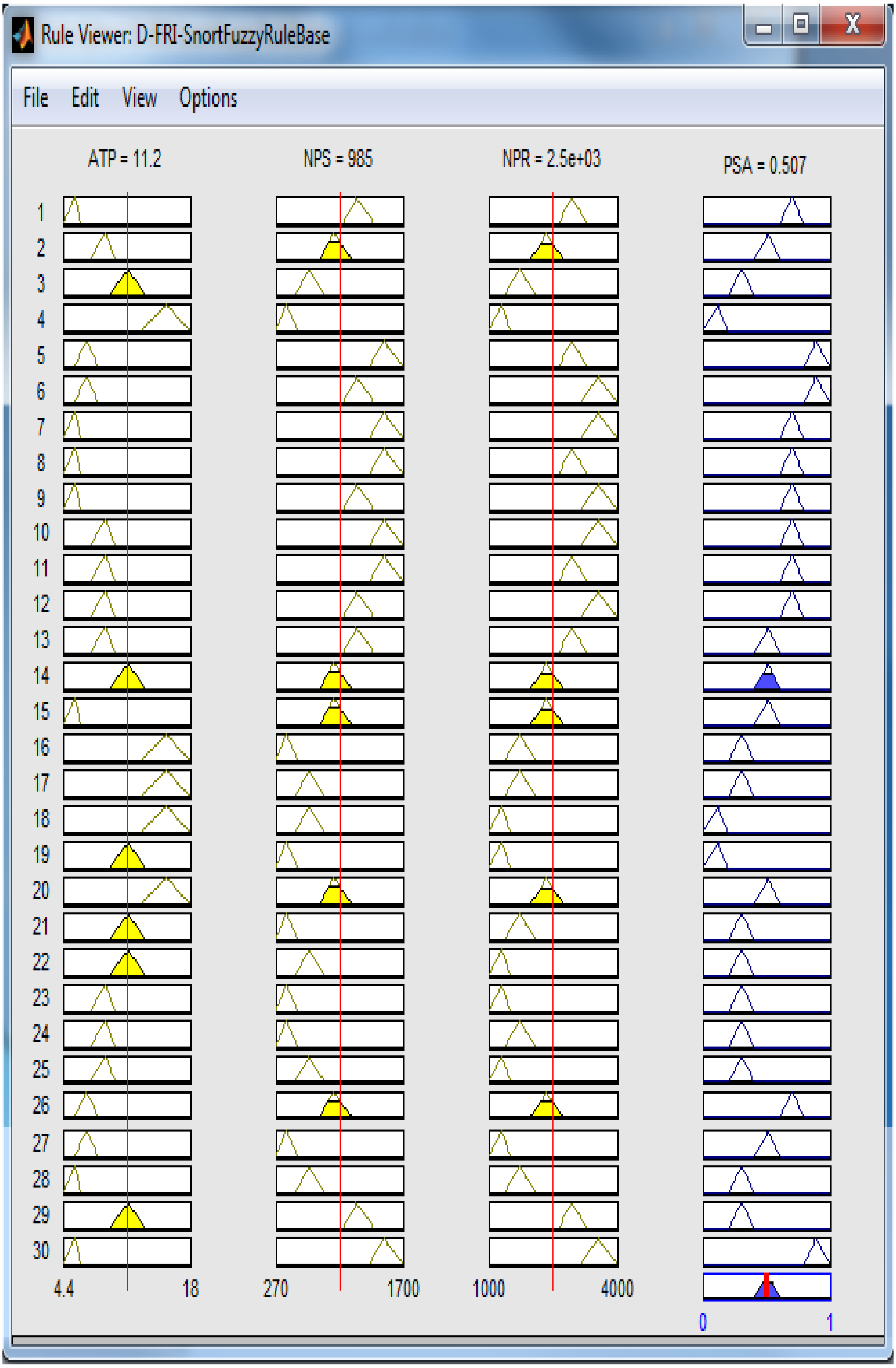


Figure 6.23: D-FRI-Snort sparse fuzzy rule base

are not covered by the sparse fuzzy rule base. This may be a common problem of any fuzzy intrusion detection systems as the design of the fuzzy rule base depends on the network parameters collected from network traffic. Therefore, many fuzzy rule based IDS often generate false results in the absence of sufficient knowledge. However, in this situation, D-FRI would generate the interpolated results in cases when no fuzzy rule is matched with the given observation. Subsequently, it will accumulate the interpolated results of frequently observed attack conditions. Eventually, it will also perform rule based modification dynamically and provide a most updated fuzzy rule base for Snort IDS. Thus it will improve the results of Snort and manage real-time rule base for detection.

This D-FRI-Snort is a new IDS concept which is quite different from the other traditional available IDS in the sense that it uses the fuzzy sparse rule based reasoning system and D-FRI, and supplies a real-time dynamic rule base for better accuracy.

6.6 Experimentation and Discussion

This section exposits the experimental results for D-FRI-Snort in three different situations: the first experiment tests for the D-FRI-Snort alert prediction and compares with Snort; the second tests for the dynamic rule promotion in D-FRI-Snort; and the third tests for D-FRI-Snort alert prediction after the dynamic rule promotion.

6.6.1 Snort vs. D-FRI-Snort Alert Generation

The first experiment was carried out to show the difference between Snort and D-FRI-Snort and their alert results. Table 6.2 shows the experimental results in seven different attack conditions to cover all alerts levels. Snort does not generate alerts in all the attack conditions while D-FRI-Snort generates alerts because of its fuzzy nature. Snort alert generation mechanism heavily depends on many parameters of snort.config file and rules. Sometime Snort can not detect slow port scan or modified signature attack due to snort.config file parameters or nature of the rules [155]. However, Snort generates alerts in most of the port scan attack conditions. From the D-FRI-Snort results, it is very obvious that D-FRI-Snort generates detailed alerts for each level of attack. The D-FRI-Snort results are also easily understandable to end users. The latter can report to the network administrator, or prevent these attacks depending on their severity and the incident response strategy of the organisation.

Table 6.2: Comparison between Snort and D-FRI-Snort Alert Outputs

Observations	Input Parameters			Output- Attack Alerts	
	ATP	NPS	NPR	$Snort-PSA$	$D-FRI-Snort-PSA$
1	17.82	275	1156	no attack alert	very low attack alert
2	17.78	283	1167	no attack alert	very low attack alert
3	15.99	322	1206	attack alert only	very low attack alert
4	10.48	565	1918	attack alert only	low attack alert
5	7.84	1032	2458	attack alert only	medium attack alert
6	6.57	1317	3068	attack alert only	high attack alert
7	5.28	1642	3657	attack alert only	very high attack alert

6.6.2 Dynamic Rule Promotion for D-FRI-Snort

In this experiment, the GA-based D-FRI technique is applied with their default parameter values and settings. No change is made except the input and output fuzzy variables adopted from experiments. Where the three antecedent variables are ATP (average time for a packet received by destination/victim in milliseconds), NPS (number of packets sent by source), NPR (number of packets received by destination/victim) and consequent variable is PSA (port scan attack). The ATP, NPS and NPR antecedent dimensions are evenly partitioned into $\eta \in \{4,5,6\}$ intervals, as a result, $4^3 = 64$, $5^3 = 125$, and $6^3 = 216$, hyper-cubes can be created. The parameters of the GA are set to the following values: crossover rate $\delta_c = 0.7$, mutation rate $\delta_m = 0.05$, population size $|\mathbb{P}| = 20$, and maximum generation $k_{\max} = 100$.

The GA-based clustering algorithm is performed over 500 interpolated rules, where 82, 124 and 158 new rules have been promoted for intervals 4, 5, and 6, respectively. The representative values of the consequent of the dynamically promoted rules are recorded. They are then compared to the results of conventional interpolation ($\epsilon_{\%dvi}$), and to the ground truths calculated using the base function ($\epsilon_{\%dvt}$). The differences between conventional interpolation and the ground truths

($e_{\%ivt}$) are also provided. Here the percentage error $\epsilon_{\%} = \epsilon / range_y$ is calculated relative to the range of the consequent variable. Since stochastic elements are present in the initial rule generation, as well as within the clustering procedure, the GA dynamic process is repeated 50 times for each set of the parameter values. Table 6.3 shows the averaged value $\epsilon_{\%}$ and the standard deviations of $\epsilon_{\%}$.

Table 6.3: D-FRI-Snort Dynamic Rule Promotion Accuracy

	$\eta = 4$			$\eta = 5$			$\eta = 6$		
	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$
AVG	2.71	2.12	2.10	2.40	**1.31**	2.56	3.42	**2.05**	3.81
SD	2.83	2.05	3.27	2.72	**1.32**	2.68	3.05	**2.03**	3.80

According to the simulation results, for $\eta = 5$ and 6, the implemented algorithm promotes more accurate rules, with derived consequent values closer to the ground truth, than those obtainable using conventional interpolation. For this D-FRI-Snort application scenario, the best parameter configuration is $\eta = 5$, which produces both accurate and stable rules. The reason is very obvious that the antecedent ranges are quite similar to the antecedent ranges of application experiment scenario. Similarly, for the configuration of $\eta = 6$, the promoted rules are also closer to the ground truth than the outcomes obtained by the conventional T-FRI. These results imply that the rules promoted using intervals $\eta = 5$ and $\eta = 6$, once added to the Snort fuzzy rule base, will not only avoid the need of future interpolations of similar observations, but also improve the inference accuracy (i.e., the quality of the rule base) overall.

6.6.3 D-FRI-Snort Alert Generation after Dynamic Rule Promotion

As it is clear form the above experiment that for the case where $\eta = 5$ most accurate rules can be generated for the current application. In this experiment, 10 new rules were dynamically promoted to the D-FRI-Snort fuzzy rule base for the case of $\eta = 5$ as shown in Figure 6.24. Afterward, D-FRI-Snort was examined for some new attack conditions. It was tested for the five new alert conditions as given in Table 6.4. It outperformed for these five attack conditions. D-FRI-Snort can also generate reasonable alert warning in the absence of rule match using interpolation technique and over the time it enhances its rule base using these result without any human intervention. Eventually, it can also develop its rules base as a dense

Table 6.4: Attack Alert Outputs of D-FRI-Snort after Dynamic Rule Promotion

Observations	Input Parameters			Output- Attack Alerts
	ATP	NPS	NPR	$D-FRI-Snort-PSA$
1	6.95	1267	2385	high attack alert
2	5.23	643	1875	low attack alert
3	4.61	996	3010	high attack alert
4	7.91	1005	2805	medium attack alert
5	15.64	310	2266	low attack alert

rule base depending on the requirement of the security applications. Therefore, D-FRI-Snort may be an effective security tool for those networks where insufficient traffic knowledge is available for reasoning.

6.7 Summary

This chapter has introduced an operational fuzzy IDS based on the application of the current research work on dynamic fuzzy rule interpolation. In this implementation, a powerful IDS Snort was chosen for enhancement based on the D-FRI, and finally D-FRI-Snort was developed as an integration of both systems. D-FRI-Snort is based on the analysis of the three network traffic parameters: ATP, NPS and NPR. The metrics monitored by the fuzzy system at this phase in the IDS development are clearly sufficient for detecting many types of scanning activity and denial of service attacks. This D-FRI-Snort was also tested against the original Snort and predicted alert results with an additional indication of the level of attack. The use of fuzzy logic can also reduce the number of false alarms and improve the accuracy of alert prediction.

However, the current application was the first and initial attempt towards the use of D-FRI. It is therefore necessary to carry out more extensive and complex experiments of the D-FRI-Snort to make it a generalized IDS for robust network security. While D-FRI-Snort is very effective in the detection of port scanning, this

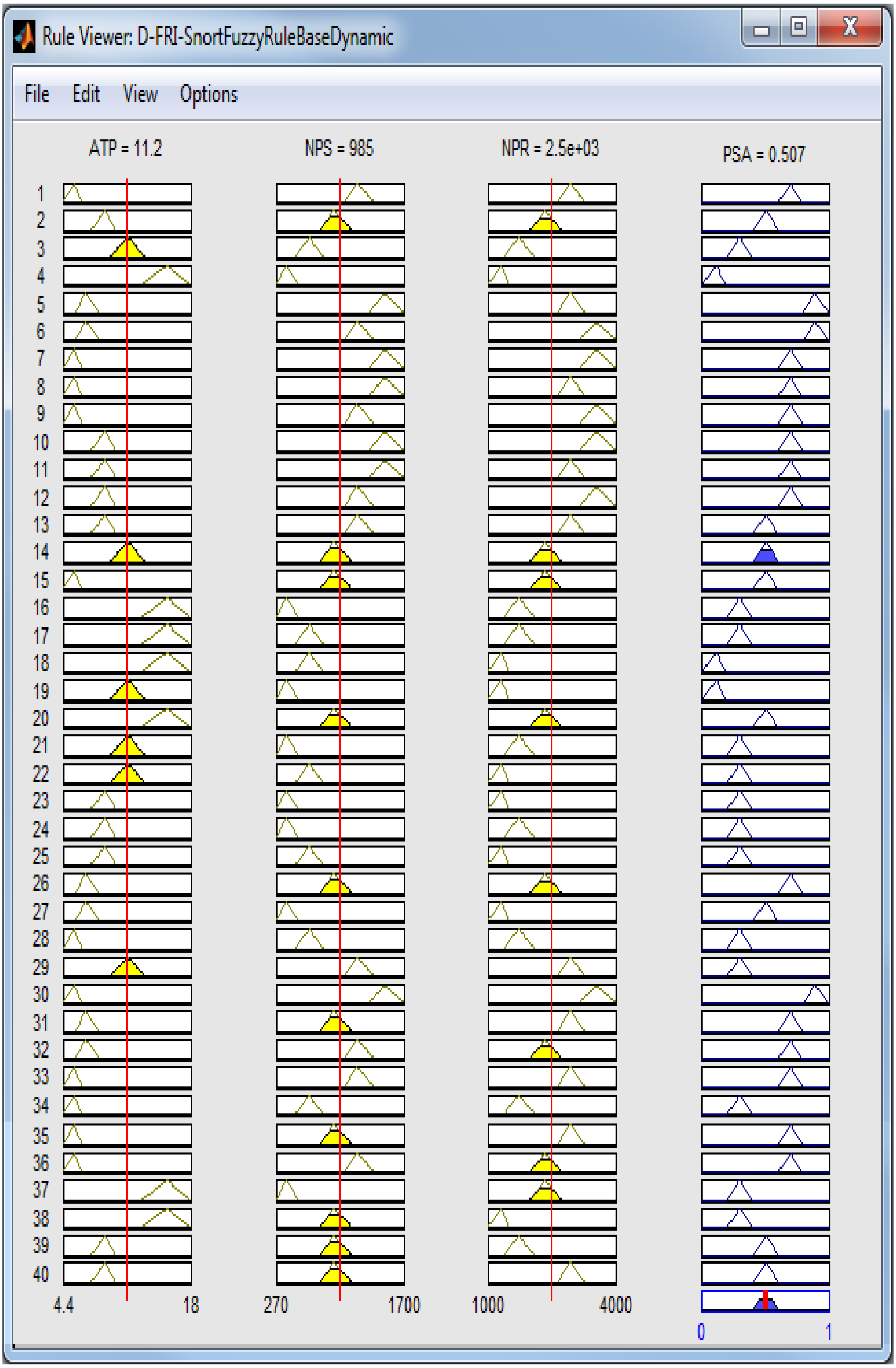


Figure 6.24: D-FRI-Snort extended sparse fuzzy rule base

application does not solve the problem of finding all types of network-based intrusions. Furthermore, some specialized agents and other metrics will be necessary to identify port scanning attacks of greater strength and sophistication. Most importantly, this research application has laid a solid groundwork for D-FRI based intrusion detection systems and revealed promising areas of continued exploration.

Chapter 7

Conclusion

THIS chapter presents a summary of the research as detailed in the preceding chapters. Having reviewed and compared a number of FRI techniques in the literature, the thesis has demonstrated that the developed dynamic fuzzy rule interpolation (D-FRI) approach has effectively leveraged FRI with a dynamic rule base and applying it to security application. The proposed dynamic framework further enhances the efficacy of the overall FRI mechanism. The proposed approach renders the reasoning system more accurate and faster by exploiting both CRI and FRI. A number of theoretical areas have also been identified that exploit the dynamic rule base in a FRI system. The capabilities and potential of the developed applications have been experimentally validated, and compared with conventional FRI work. The conclusion also presents a number of initial thoughts about the directions for future research.

7.1 Summary of Thesis

A detailed literature review was presented in Chapter 2 that covers the fuzzy inference systems (FIS), compositional rules of inference (CRI) and fuzzy rule interpolation (FRI). CRI is a classical inference approach in systems using dense rule bases and important CRI methods: Mamdani inference, TSK inference and Tsukamoto inference have been outlined. However, in most real life applications, rule bases are sparse and FRI is a quite effective approach for reasoning with sparse rule bases. For this reason, a survey of twelve different FRI approaches was made in which the key common notions and mechanisms of the reviewed algorithms were extracted, and a unified style of notation adopted, with pseudocode included.

While conducting the review, evaluation criteria for FRI approaches were identified and all the reviewed approaches were analysed according to these criteria. Most of the reviewed FRI methods rely on a pre-defined, static fuzzy rule base, from which the interpolation or extrapolation results are calculated. Consequently, the accuracy of the interpolation or extrapolation results is heavily dependent on the rules of this static rule base. Unfortunately, these methods do not have a mechanism to support self-diagnosis or self-modification of the original rules. Nevertheless, there is a need to develop FRI for situations where the environment changes and there is a great deal of uncertainty. This is one of the main findings of the FRI review process that leads to the development of a dynamic or self-adaptive rule base that supports dynamic fuzzy rule interpolation (D-FRI).

In Chapter 3, an initial development of the dynamic fuzzy rule interpolation (D-FRI) approach was introduced that facilitates a dynamic rule base for, and enhances the efficacy of, FRI. This approach is mainly based on the applications of *k-means* clustering, and the T-FRI technique and weighted aggregation method. In this dynamic approach, T-FRI performs regular interpolation in a sparse fuzzy rule base and provides the inference results. A by-products of the interpolation, interpolated results are accumulated for further processing, which are analysed, aggregated, and promoted into the original sparse rule base. This process leads to a dynamic sparse rule base and improves the performance of the FRI approach by providing a real-time concurrent rule base.

The performance of the proposed *k-means* D-FRI is verified by simulated application examples and shown to have the potential for developing an effective adaptive FRI system. This initial D-FRI approach can be further improved by applying an intelligent partitioning approach to decide the best partitioning solution automatically; an intuitive clustering approach to find a better cluster arrangement rather then defining a fixed number of clusters k at the outset. For this, a GA-based D-FRI was introduced in Chapter 4. In this improved D-FRI, in order to reduce the search complexity of the GA process, the collection of the interpolated rules is pre-partitioned into hyper-cubes (multi-dimensional blocks) and then GA-based clustering is applied on the blocks rather than on every rule. The remaining D-FRI process uses the same approach as the original version of D-FRI. Experimental results again show the efficiency and accuracy of the proposed GA-based D-FRI when it is compared with conventional FRI.

In Chapter 5, a new approach to integrated fuzzy reasoning systems is presented. This facilitates the integration of two popular reasoning mechanisms: inference (CRI) and interpolation (FRI). This is a useful concept for fuzzy reasoning in order to benefit from the advantages of both types of reasoning when dealing with sparse rule bases. Interpolation is computationally more complex than inference, while inference is only possible in the dense rule base. However, it is not always necessary to perform interpolation in the sparse rule base if the exact or partial match is available for the input observation. This proposed integrated approach applies a pre-interpolation operation to determine the possibility of inference. The work also presents different distance metrics, including Hausdorff Distance (HD) and Earth Mover's Distance (EMD), in order to identify the closest rule to perform interpolation or extrapolation more accurately. Experimental results show that this integration can save a good deal of the computational overheads of interpolation. Results also reveal that the two suggested distance metrics, HD and EMD, are better than the classical COG for more precise interpolation and extrapolation results.

Finally, Chapter 6 has presented an application of the proposed D-FRI framework to the problem domain of intrusion detection systems (IDS). A powerful open-source IDS suite - *Snort* - is chosen as the foundational building block of the application, and the D-FRI framework is embedded within the IDS in order to realise an implementation of *D-FRI-Snort*. D-FRI-Snort enhances the functionality of port scanning detection and delivers another level of security to the IDS with higher levels of accuracy in alert predictions. It is based on the analysis and proper utilization of three network traffic parameters (ATP, NPS and NPR), and it is this that endows Snort with fuzzy functionality. The investigation of *D-FRI-Snort* was carried out using other network analysis tools, such as Wireshark, NMAP, Winpcap, Advanced Port Scanner and Basic Analysis and Security Engine (BASE). *D-FRI-Snort's* inference module predicts alerts with an additional indication of the level of attack which was beyond the capabilities of the original Snort. *D-FRI-Snort's* interpolation module performs the reasoning for those input conditions when no matched rules are found in the rule base. After a certain period, it accumulates interpolated rules and promotes new rules to the sparse rule base. Thus, it updates the sparse rule base dynamically and provides the contemporary rule base. The use of fuzzy logic can also reduce the number of false alarms and improve the accuracy of alert prediction.

In summary, this research has proposed a novel dynamic fuzzy reasoning approach and an integration of inference and interpolation, in an effort to make fuzzy reasoning systems more efficient and accurate. Experimentation has verified the potential of this work. However, further research is needed to enhance the proposed system. Also, it is important to apply this proposed approach to some more real life applications in order to test the practical usability of the approach.

7.2 Future Work

Although promising, much can be done to further improve the work presented in this thesis.The following addresses a number of interesting issues whose successful solution will go towards establishing the current research on a more robust foundation.

7.2.1 Short Term Tasks

This section discusses extensions, enhancements or ongoing tasks that could be readily implemented if additional time were available.

7.2.1.1 D-FRI with a Rule Management Module

The proposed D-FRI approach only creates new rules and promotes them to the sparse rule base based on certain preliminary conditions. This new module will be a sophisticated rule manager [42, 101, 189] for dynamic sparse fuzzy rule base systems. The most important part is to define effective administrative policies required for the management. For example in rule addition, if this approach does not find a rule match within the current rule base for the newly created rule then the new rule may be added to the sparse rule base. For rule modification, if it finds that the two rules are very close or certain different variables are found for existing rules then these rules may also be modified. The rule manager can delete any rule from the rule base, including situations such as: if it is no longer used for a specific time; if two rules are almost identical or close; if the rule is a subset of another rule; or if the rule is already merged with another rule. However, it is difficult to define common administrative policies because these administrative policies depend on the specific application. These policies may change drastically from one application to other. For an intelligent D-FRI approach, this module is a highly necessary component.

7.2.1.2 D-FRI with Alternative FRI Methods

In this research work, for the purpose of preliminary investigation and experimentation, the T-FRI approach [85, 86] is employed. To further generalize D-FRI, is interesting to investigate the use of some different FRI techniques [17, 18, 28, 37, 205]. A comparative study may be helpful in order to determine the most effective and accurate D-FRI system using a certain FRI method. This work has been partially done in the literature review of FRI approaches where the pseudo codes are given for twelve FRI approaches. This is of course a systematic approach that requires a lot of additional experimentation.

7.2.1.3 D-FRI with the Use of Alternative Aggregations

The proposed D-FRI approach creates new rules by applying many operations such as partitioning, clustering and aggregation, to the interpolated results. This is implemented with the use of the given weighted aggregation method. It would be useful to test the application of other aggregation operators such as ordered weighted average (OWA) [199], generalised ordered weighted average (GOWA)[200] and induced ordered weighted average (IOWA) [201]. If better rules can be obtained using such alternative aggregation operators then this will help improve the accuracy of the created rules and thus, the accuracy of the corresponding D-FRI system.

7.2.2 Longer Term Developments

This section proposes several future directions of research.

7.2.2.1 Dynamic Integrated Inference and Interpolation Systems

A dynamic integrated inference and interpolation system is foreseen as one of the most important further developments of this research. Once a genuine D-FRI system including a rule management module (as a short-term goal) is established, it can be merged with the proposed integrated system to make it a dynamic integrated one. This would lead to a new fuzzy reasoning system that might enjoy the benefits of both CRI and FRI approaches when given a dynamic sparse rule base.

7.2.2.2 D-FRI with Dynamic Partitioning

There is one important assumption throughout this research: an initial fixed partitioning-level for a given rule base. For the current implementation, this is sufficient to evaluate the potential of D-FRI. However, fixed partitioning limits the generalised concept of this approach and also affects the accuracy of the proposed framework. This is because it decides the number of partitions for every dimension of the rule base at an early stage irrespective of all the later operations used in D-FRI. This may permanently direct the later operations and affect their outcomes. It is therefore important to be able to obtain the best partitioning during the reasoning process in order to find better quality clusters and, eventually, more precise new rules for reasoning.

There are many ways to implement the dynamic partitioning to make this D-FRI system more adaptive. One is to apply a suitable multidimensional clustering method. There are many grid-based clustering approaches (e.g., STING (STatistical INformation Grid) [190], CLIQUE (CLustering In QUEst) [5], MAFIA (Merging of Adaptive Intervals Approach) [70], WaveCluster [167], O-CLUSTER (Orthogonal partitioning CLUSTERing) [140], ASGC (Axis Shifted Grid Clustering) [35]), and density-based clustering approaches (e.g., DBSCAN (Density Based Spatial Clustering of Application with Noise) [13], DBCLASD (Distribution-Based Clustering of LArge Spatial Databases) [198], OPTICS (Ordering Points to Identify the Clustering Structure) [215], DENCLUE (DENsity based CLUstEring) [78]), FDC (Fast Density-Based Clustering) [214], VDBSCAN (Varied Density Based Spatial Clustering of Applications with Noise) [131], DVBSCAN (Density Varied Based Spatial Clustering of Applications with Noise) [154], GDBSCAN (Generalized Density-Based Spatial Clustering of Applications With Noise) [163], available for multidimensional environments. However, it is important to choose one that may minimise the complexity of the resultant D-FRI. Another method is to apply a feature selection approach for partitioning [77, 98]. Feature selection may be very useful to determine the appropriate initial partitions. However, applying another feature selection approach involves extensive research work and it would require a significant investment in time to implement this approach. Apart from these methods, there might be other effective ways to perform dynamic partitioning, so it would be useful to investigate the best partitioning approach before embarking on this research work in future.

Appendix A

Publications Arising from the Thesis

A few publications have been generated from the research carried out within the PhD project. Below lists the resultant publications that are in close relevance to the thesis, including all the papers already published and articles under review.

1. N. Naik, R. Diao, and Q. Shen, Genetic algorithm-aided dynamic fuzzy rule interpolation, [149] in IEEE World Congress on Computational Intelligence, 2014.

2. N. Naik, R. Diao, C. Quek, and Q. Shen, Towards dynamic fuzzy rule interpolation, [148] in IEEE International Conference on Fuzzy Systems, 2013.

3. N. Naik, P. Su, and Q. Shen, Integration of interpolation and inference, [150] in UK Workshop on Computational Intelligence, 2012.

4. N. Naik, and Q. Shen, Evaluation Criteria and Comparative Analysis of Fuzzy Rule Interpolation, 54 pp., under review.

Appendix B

List of Acronyms

AAC	Avoidance of Abnormal Conclusion
AAFS	Applicability to Arbitrary Fuzzy Sets
AC	Approximation Capability
ACK	Acknowledge
API	Application Programming Interface
ASGC	Axis Shifted Grid Clustering
ATP	Average Time for Packet
BASE	Basic Analysis and Security Engine
B-FRI	Backward Fuzzy Rule Interpolation
CC	Computational Complexity
CF	Conservation of Fuzziness
CGI	Common Gateway Interface
CLIQUE	CLustering In QUEst
CNF	Convex and Normal Fuzzy
CRF	Conservation of Relative Fuzziness
CP	Cartesian Product
CPU	Central Processing Unit

CRI Compositional Rule of Inference

DBSCAN Density Based Spatial Clustering of Application with Noise

DBCLASD Distribution-Based Clustering of LArge Spatial Databases

DENCLUE DENsity based CLUstEring

D-FRI Dynamic Fuzzy Rule Interpolation

D-FRI-Snort Dynamic Fuzzy Rule Interpolation Snort

DI Dunn Index

DNS Domain Name System

DoS Denial of Service

DVBSCAN Density Varied Based Spatial Clustering of Applications with Noise

EMD Earth Mover's Distance

FATI First Aggregate - Then Infer

FCD Fast Density-Based Clustering

FIN Finish

FIR Fuzziness of Inferred Result

FIS Fuzzy Inference System

FITA First Infer - Then Aggregate

FIVE Fuzzy Interpolation based on Vague Environment

FL Fuzzy Logic

FRBS Fuzzy Rule-Based System

FRI Fuzzy Rule Interpolation

FTP File Transfer Protocol

GA Genetic Algorithm

GAWL GA-based Weight-Learning Interpolation Method

GDBSCAN Generalized Density-Based Spatial Clustering of Applications With Noise

GMP Generalised Modus Ponens

GUI Graphical User Interface

H High

HD Hausdorff Distance

HTTP Hyper Text Transport Protocol

IANA Internet Assigned Numbers Authority

ICMP Internet Control Message Protocol

IDS Intrusion Detection System

IP Internet Protocol

IMUL Improved Multidimensional Modified Alhpa-cut Based Interpolation Method

IRG Interpolative Reasoning based on Graduality

KH Koczy-Hirota

L Low

LAN Local Area Network

M Medium

MAC Media Access Control

MACI Modified Alpha-cut based Interpolation Method

MAFIA Merging of Adaptive Intervals Approach

MAVS Multiple Antecedent Variables for Support

MF Membership Function

MISO Multiple Input Single Output

MIMO Multiple Input Multiple Output

MMFS Multiple Membership Functions for Support

MPV Modus Ponens Validity

MS Mapping Similarity

MRS Multiple Rules for Support

NAI Network Associates

NIDES Next-generation Intrusion Detection Expert System

NIDS Network Intrusion Detection System

NMAP Network Mapper

NPR Number of Packet Received

NPS Number of Packet Sent

NSS National Security Systems

OARS Overlapping Antecedent Rules for Support

O-CLUSTER Orthogonal partitioning CLUSTERing

OPTICS Ordering Points to Identify the Clustering Structure

OS Operating System

OSSEC Open Source SECurity

OSSIM Open Source Security Information Management

PCNF Preservation of Convexity and Normality

PNQ Preservation of Neighbouring Quality

POP Post Office Protocol

PPWL Preservation of Piece-Wise Linearity

RBP Rule-Base Preservation

RST Reset

RV Representative Value

SFE Slopes of Flaking Edges

SGR Spatial Geometric Representation

SISO Single Input Single Output

SMTP Simple Mail Transfer Protocol

SNMP Simple Network Management Protocol

SOCKS Socket Secure

SRI Stanford Research Institute

ST Similarity Transfer

STING STatistical INformation Grid

SYN Synchronize Sequence Number

TCP Transmission Control Protocol

T-FRI Transformation-based Fuzzy Rule Interpolation

TFTP Trivial File Transfer Protocol

TSK Takagi-Sugeno-Kang

UDP User Datagram Protocol

VDBSCAN Varied Density Based Spatial Clustering of Applications with Noise

VH Very High

VL Very Low

WAN Wide Area Network

XMAS Christmas

Bibliography

[1] L. Abrams. (2004) TCP and UDP ports explained. [Online]. Available: http://www.bleepingcomputer.com/tutorials/tcp-and-udp-ports-explained/.

[2] A. Adetunmbi, S. Zhiwei, S. Zhongzhi, and O. Adewale, "Network anomalous intrusion detection using fuzzy-bayes," *International Federation for Information Processing, Intelligent Information Processing*, vol. 228, no. 3, pp. 525–530, 2006.

[3] V. Adriaenssens, B. D. Baets, P. L. M. Goethals, and N. D. Pauw, "Fuzzy rule-based models for decision support in ecosystem management," *The Science of the Total Environment*, pp. 116–128, 2004.

[4] C. C. Aggarwal, *Outlier Analysis*, 1st ed. New York, USA: Springer, 2013.

[5] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *ACM SIGMOD international conference on Management of data*, 1998, pp. 94–105.

[6] T. A. Ahanger, "Port Scan - A Security Concern," *International Journal of Engineering and Innovative Technology*, vol. 3, no. 10, pp. 241–246, 2014.

[7] A. N. Aizawa and B. W. Wah, "Scheduling of genetic algorithms in a noisy environment," *Evolutionary Computation*, vol. 2, no. 2, pp. 97–122, 1994.

[8] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner, "State of the practice of intrusion detection technologies," Software Engineering Institute, Carnegie Mellon University, Tech. Rep., 2000.

[9] M. S. Anbarasi and K. M. Mehata, "Enhanced k-means clustering for patient reported outcome," *Recent Advances in Computer Engineering and Applications*, pp. 19–26, 2012.

[10] J. P. Anderson, "Computer security threat monitoring and surveillance," James P Anderson Co., Fort Washington, Pennsylvania, Tech. Rep., 1980.

[11] P. Angelov, "An evolutionary approach to fuzzy rule-based model synthesis using rules indices," *Fuzzy Sets and Systems*, vol. 137, no. 3, pp. 325–338, 2003.

[12] P. Angelov and R. Buswell, "Automatic generation of fuzzy rule-based models from data by genetic algorithms," *Information Sciences*, vol. 150, no. 1/2, pp. 17–31, 2003.

[13] M. Ankerst, M. M. Breunig, K. H. P., and J. Sander, "Optics: Ordering points to identify clustering structure," in *ACM SIGMOD international conference on Management of data*, 1999, pp. 49–60.

[14] D. V. Arnold, *Noisy Optimization with Evolution Strategies, Genetic Algorithms and Evolutionary*, 1st ed. Boston, Dordrecht, UK: Kluwer Academic Publishers, 2002.

[15] K. J. Astrom and B. Wittenmark, *Adaptive Control*, 2nd ed. Addison-Wesley, 1995.

[16] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," *ACM Transaction on Information and System Security*, vol. 3, no. 3, pp. 186–205, 2003.

[17] P. Baranyi, T. D. Gedeon, and L. T. Koczy, "A general method for fuzzy rule interpolation: Specialized for crisp, triangular, and trapezoidal rules," in *3rd European Congress on Fuzzy and Intelligent Techniques (EUFIT'95)*, 1995, pp. 99–103.

[18] ——, "Rule interpolation by spatial geometric representation," in *Information Processing and Management of Uncertainty in Knowledge-based Systems*, 1996, pp. 483–488.

[19] P. Baranyi, L. T. Koczy, and T. D. Gedeon, "A generalized concept for fuzzy rule interpolation," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 6, pp. 820–837, 2004.

[20] H. Ben-Romdhane, E. Alba, and S. Krichen, "Best practices in measuring algorithm performance for dynamic optimization problems," *Soft Computing*, vol. 17, no. 6, pp. 1005–1017, 2013.

[21] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Surveying port scans and their detection methodologies," *The Computer Journal*, vol. 54, no. 10, pp. 1565–1581, 2011.

[22] M. Biswanath, L. H. Todd, and N. Karl, "Network intrusion detection," *IEEE Network*, vol. 8, no. 3, pp. 26–41, 1994.

[23] J. Bone. (2011) Wireshark. [Online]. Available: http://download.cnet.com/Wireshark/3000-2085_4-10668290.html.

[24] T. Boongoen and Q. Shen, "Semi-supervised OWA aggregation for link-based similarity evaluation and alias detection," in *IEEE International Conference on Fuzzy Systems*, 2009, pp. 288–293.

[25] ——, "Nearest-neighbor guided evaluation of data reliability and its applications," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 6, pp. 1622–1633, 2010.

[26] B. Bouchon-Meunier, D. Dubois, C. Marsala, H. Prade, and L. Ughetto, "A comparative view of interpolation methods between sparse fuzzy rules," in *9th World Congress of the International Fuzzy Systems Association (IFSA'01)*, 2001, pp. 483–487.

[27] B. Bouchon-Meunier, F. Esteva, L. Godo, M. Rifqi, and S. Sandri, "A principled approach to fuzzy rule based interpolation using similarity relations," in *Conference of the European Society for Fuzzy Logic and Technology*, 2005, pp. 757–763.

[28] B. Bouchon-Meunier, C. Marsala, and M. Rifqi, "Interpolative reasoning based on graduality," in *IEEE International Conference on Fuzzy Systems*, 2000, pp. 483–487.

[29] B. Bouchon-Meunier, R. Mesiar, C. Marsala, R. Mesiar, and M. Rifqi, "Compositional rule of inference as an analogical scheme," *Fuzzy Sets Systems*, vol. 138, pp. 53–65, 2003.

[30] S. M. Bridges and R. B. Vaughan, "Intrusion detection via fuzzy data mining," in *12th Annual Canadian Information Technology Security Symposium*, 2000, pp. 19–23.

[31] E. Carter. (2002) Intrusion detection system. [Online]. Available: http://www.ciscopress.com/articles/article.asp?p=25334.

[32] O. Castillo and P. Melin, "Fuzzy logic," in *Soft Computing for Control of Non-Linear Dynamical Systems*, O. Castillo and P. Melin, Eds. New York: Physica-Verlag HD, 2001, pp. 5–27.

[33] ——, *Soft Computing and Fractal Theory for Intelligent Manufacturing*. Physica-Verlag Heidelberg, New York, 2003.

[34] S. Chakraborty, N. K. Nagwani, and L. Dey, "Performance comparison of incremental k-means and incremental dbscan algorithms," *International Journal of Computer Applications*, vol. 27, no. 11, pp. 14–18, 2011.

[35] C. I. Chang, N. P. Lin, and N. Y. Jan, "An axis-shifted grid-clustering algorithm," *Tamkang Journal of Science and Engineering*, vol. 12, no. 2, pp. 183–192, 2009.

[36] Y. C. Chang and S. M. Chen, "A new method for multiple fuzzy rules interpolation with weighted antecedent variables," in *IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 76–81.

[37] ——, "Weighted fuzzy rule interpolation based on ga-based weight-learning techniques," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 4, pp. 729–744, 2011.

[38] Y. C. Chang, S. M. Chen, and C. Liau, "Fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on the areas of fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 5, pp. 1285–1301, 2008.

[39] M. Chapel. (2008) Wireshark tutorial: How to sniff network traffic. [Online]. Available: http://searchsecurity.techtarget.com/tip/Wireshark-tutorial-How-to-sniff-network-traffic.

[40] B. B. Chaudhuri and G. Garai, "Grid clustering with genetic algorithm and tabu search process," *Journal Of Pattern Recognition Research*, vol. 4, no. 1, pp. 152–168, 2009.

[41] B. B. Chaudhuri and A. Rosenfeld, "A modified Hausdorff Distance between fuzzy sets," *Information Sciences*, vol. 118, pp. 159–171, 1999.

[42] J. M. Chen, S. H. Wu, and H. M. Lee, "The study of automatic insertion and deletion of fuzzy rules in fuzzy neural network models," in *Fuzzy Systems Symposium,*, 1996, pp. 248–253.

[43] S. M. Chen and H. L. Lee, "Generating weighted fuzzy rules for handling classification problems," *International Journal of Electronic Business Management*, vol. 3, no. 2, pp. 116–128, 2005.

[44] S. Chen and Y. Ko, "Fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on α-cuts and transformations techniques," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1626–1648, 2008.

[45] W. Chimphlee, A. H. Abdullah, M. N. M. Sap, S. Chimphlee, and S. Srinoy, "A rough-fuzzy hybrid algorithm for computer intrusion detection," *The International Arab Journal of Information Technology*, vol. 4, no. 3, pp. 247–253, 2007.

[46] Cisco. (2014) Chapter 6 - Network security using Cisco IOS IPS. [Online]. Available: https://learningnetwork.cisco.com/servlet/JiveServlet/download/6651-1-8391/Ch%206_Network_Security_Using_Cisco_IOS_IPS.pdf.

[47] B. Claypool. (2002) Stealth port scanning methods. [Online]. Available: http://www.giac.org/paper/gsec/1985/stealth-port-scanning-methods/103446.

[48] V. Cross and T. Sudkamp, "Fuzzy implication and compatibility modification," in *IEEE International Conference on Fuzzy Systems*, 1993, pp. 219–224.

[49] DEIB. (2014) A tutorial on clustering algorithms. [Online]. Available: http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html.

[50] J. E. Dickerson and J. A. Dickerson, "Fuzzy network profiling for intrusion detection," in *19th International Conference of the North American Fuzzy Information Processing Society*, 2000, pp. 301–306.

[51] J. E. Dickerson, J. Juslin, Q. Koukousoula, and J. A. Dickerson, "Fuzzy intrusion detection," in *20th International Conference of the North American Fuzzy Information Processing Society*, 2001, pp. 1506–1510.

[52] D. Dineley. (2009) The greatest open source software of all time. [Online]. Available: http://www.infoworld.com/article/2631146/open-source-software/the-greatest-open-source-software-of-all-time.html.

[53] D. Dubois and H. Prade, *Fuzzy Sets and Systems*. Academic Press, New York, 1988.

[54] ——, "Gradual rules in approximate reasoning," *Information Sciences*, vol. 61, pp. 103–122, 1992.

[55] ——, "On fuzzy interpolation," in *3rd International Conference on Fuzzy Logic and Neural Networks*, 1994, pp. 353–354.

[56] ——, "On fuzzy interpolation," *International Journal Of General System*, vol. 28, no. 2-3, pp. 103–114, 1999.

[57] D. Dubois, H. Prade, and L. Ughetto, "A new perspective on reasoning with fuzzy rules," *International Journal Of Intelligent Systems*, vol. 18, no. 5, pp. 541–567, 2003.

[58] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[59] R. Eberhart, P. Simpson, and R. Dobbins, *Computational Intelligence PC Tools*. Academic Press, Orlando, FL, 1996.

[60] W. El-Hajj, F. Aloul, Z. Trabelsi, and N. Zaki, "On detecting port scanning using fuzzy based intrusion detection system," in *International Conference on Wireless Communications and Mobile Computing*, 2008, pp. 105–110.

[61] W. El-Hajj, H. Hajj, Z. Trabelsi, and F. Aloul, "Updating snort with a customized controller to thwart port scanning," *Security and Communication Networks*, vol. 4, no. 8, pp. 807–814, 2010.

[62] A. Elshamli, H. Abdullah, and S. Areibi, "Genetic algorithm for dynamic path planning," in *Canadian Conference on Electrical and Computer Engineering*, 2004, pp. 677–680.

[63] F. Esteva, M. Rifqi, B. Bouchon-Meunier, and M. Detyniecki, "Similarity-based fuzzy interpolation method," in *Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2004, pp. 1443–1449.

[64] A. Fahim, A. M. Salem, F. A. Torkey, M. A. Ramadan, and G. Saake, "An efficient k-means with good initial starting points," *Georgian Electronic Scientific Journal: Computer Science and Telecommunications*, vol. 19, no. 2, pp. 47–57, 2009.

[65] A. Fernandez and F. Herrera, "Linguistic fuzzy rules in data mining:follow-up mamdani fuzzy modeling principle," in *Combining Experimentation and Theory*, E. Trillas, P. P. Bonissone, L. Magdalena, and J. Kacprzyk, Eds. Berlin: Springer-Verlag, 2012, pp. 103–122.

[66] E. Forgy, "Cluster analysis of multivariate data: efficiency vs. interpretability of classification," *Biometrics*, vol. 21, no. 3, pp. 768–780, 1965.

[67] W. Fuertes, P. Zambrano, M. Sanchez, and P. Gamboa, "Alternative engine to detect and block port scan attacks using virtual network environments," *International Journal of Computer Science and Network Security*, vol. 11, no. 11, pp. 14–23, 2011.

[68] B. Gabrys and A. Bargiela, "General fuzzy min-max neural network for clustering and classification," *IEEE Trans. on Neural Network*, vol. 11, no. 3, pp. 769–783, 2000.

[69] T. D. Gedeon and L. T. Koczy, "Conservation of fuzziness in rule interpolation," in *Symposium on New Trends in Control of Large Scale Systems*, 1996, pp. 13–19.

[70] S. Goil, H. Nagesh, and A. Choudhary, "Mafia: Efficient and scalable clustering for very large data sets," Center for Parallel and distributed Computing, Tech. Rep. CPDC-TR-9906-010, 1999.

[71] D. Goldberg, *Genetic algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[72] K. Goztepe, "Designing a fuzzy rule based expert system for cyber security," *International Journal of Information Security Science*, vol. 1, no. 1, pp. 13–19, 2012.

[73] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2/3, pp. 107–145, 2001.

[74] A. Haman and N. D. Geogranas, "Comparison of mamdani and sugeno fuzzy inference systems for evaluating the quality of experience of hapto-audio-visual applications," in *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, 2008.

[75] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, 2nd ed. Elsevier, 2006.

[76] Harshna and N. Kaur, "Fuzzy data mining based intrusion detection system using genetic algorithm," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 1, pp. 5021–5028, 2014.

[77] D. Higham, G. Kalna, and J. Vass, "Multidimensional partitioning and bi-partitioning: analysis and application to gene expression datasets," *International Journal of Computer Mathematics*, vol. 85, no. 3/4, pp. 475–485, 2008.

[78] A. Hinneburg and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *American Association for Artificial Intelligence*, 1998.

[79] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Univ Michigan Press, 1975.

[80] N. Horn. (2014) Snort (2.9.5.6). [Online]. Available: http://slackbuilds.org/repository/14.1/network/snort/.

[81] W. Hsiao, S. Chen, and C. Lee, "A new interpolative reasoning method in sparse rule-based systems," *Fuzzy Sets and Systems*, vol. 93, no. 1, pp. 17–22, 1998.

[82] Z. Huang and Q. Shen, "A new fuzzy interpolative reasoning method based on centre of gravity," in *IEEE International Conference on Fuzzy Systems*, 2003, pp. 25–30.

[83] ——, "Fuzzy interpolation with generalized representative values," in *UK Workshop on Computational Intelligence*, 2004, pp. 161–171.

[84] ——, "Scale and move transformation-based fuzzy interpolative reasoning: a revisit," in *IEEE International Conference on Fuzzy Systems*, 2004, pp. 623–628.

[85] ——, "Fuzzy interpolative reasoning via scale and move transformations," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 2, pp. 340–359, 2006.

[86] ——, "Fuzzy interpolation and extrapolation: A practical approach," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 13–28, 2008.

[87] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[88] S. Jenei, "Interpolation and extrapolation fuzzy quantities revisited - (i) an axiomatic approach," *Soft Computing*, vol. 5, pp. 179–193, 2001.

[89] S. Jenei, E. P. Klement, and R. Konzel, "Interpolation and extrapolation of fuzzy quantities - (II) the multiple-dimensional case," *Soft Computing*, vol. 6, pp. 258–270, 2002.

[90] R. Jensen and Q. Shen, *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*. IEEE Wiley-IEEE Press, New Jersey, 2008.

[91] Z. C. Johanyak and A. Berecz, "Survey on practical applications of fuzzy rule interpolation," in *1st International Scientific and Expert Conference TEAM*, 2009, pp. 205–213.

[92] Z. C. Johanyak and S. Kovacs, "A brief survey and comparison on various interpolation based fuzzy reasoning methods," *Acta Polytechnica Hungarica*, vol. 3, no. 1, pp. 91–105, 2006.

[93] ——, "Fuzzy rule interpolation based on polar cuts," *Computational Intelligence, Theory and Applications*, vol. 4, pp. 499–511, 2006.

[94] ——, "Fuzzy rule interpolation by the least squares method," in *International Symposium of Hungarian Researchers on Computational Intelligence*, 2006, pp. 495–506.

[95] ——, "Survey on various interpolation based fuzzy reasoning methods," *Production Systems and Information Engineering*, vol. 3, no. 1, pp. 39–56, 2006.

[96] ——, "Fuzzy rule interpolation by the least squares method," in *5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics (SAMI 2007)*, 2007, pp. 189–200.

[97] Z. C. Johanyak, D. Tikk, S. Kovacs, and K. W. Wong, "Fuzzy rule interpolation matlab toolbox - FRI toolbox," in *IEEE International Conference on Fuzzy Systems*, 2006, pp. 351–357.

[98] N. Jrad, R. Phlypo, and M. Congedo, "SVM feature selection for multidimensional EEG data," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011)*, 2011, pp. 781–784.

[99] O. Kaleva, "Interpolation of fuzzy data," *Fuzzy Sets and Systems*, vol. 61, no. 1, pp. 63–70, 1994.

[100] G. J. Kallarackal. (2014) Winpcap - Link-layer network access in windows. [Online]. Available: http://techgurulive.com/winpcap-link-layer-network-access-in-windows/.

[101] N. Karami and J. Iijima, "Modeling dynamic business rules using a dynamic knowledge approach," *Industrial Engineering and Management Systems*, vol. 6, no. 1, pp. 72–82, 2007.

[102] A. Kaur and A. Kaur, "Comparison of mamdani-type and sugeno-type fuzzy inference systems for air conditioning system," *International Journal of Soft Computing and Engineering*, vol. 2, no. 2, pp. 323–325, 2012.

[103] S. Kaur and G. Bharti, "Two inputs two output fuzzy controller system design using MATLAB," *International Journal of Advances in Engineering, Science and Technology*, vol. 2, no. 3, pp. 209–218, 2012.

[104] M. F. Kawaguchi and M. Miyakoshi, "A fuzzy rule interpolation technique based on bi-spline in multiple input systems," in *IEEE International Conference on Fuzzy Systems*, 2000, pp. 488–492.

[105] P. Kazienko and P. Dorosz. (2003) Intrusion detection. [Online]. Available: http://www.windowsecurity.com/articles/ids-part2-classification-methods-techniques.

[106] J. Kim and J. H. Lee, "A slow port scan attack detection mechanism based on fuzzy logic and a stepwise policy," in *International Conference on Intelligent Environments*, 2008, pp. 1–5.

[107] F. Klawonn, "Fuzzy sets and vague environments," *Fuzzy Sets and Systems*, vol. 66, pp. 207–221, 1994.

[108] ——, "The connection between interpolation in vague environments and fuzzy control," *Tatra Mountains Mathematical Publications*, vol. 6, pp. 63–74, 1995.

[109] Y. K. Ko and S. M. Chen, "Fuzzy interpolative reasoning via cutting and transformations techniques," in *International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems*, 2007, pp. 238–249.

[110] L. T. Koczy and K. Hirota, "Rule interpolation by α-level sets in fuzzy approximate reasoning." *Bulletin for Studies and Exchanges on Fuzziness and its Applications (BUSEFAL)*, vol. 46, pp. 115–123, 1991.

[111] ——, "Approximate reasoning by linear rule interpolation and general approximation," *International Journal of Approximate Reasoning*, vol. 9, no. 3, pp. 197–225, 1993.

[112] ——, "Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases," *Information Sciences*, vol. 71, no. 1-2, pp. 169–201, 1993.

[113] ——, "Ordering distance and closeness of fuzzy sets," *Fuzzy Sets and Systems*, vol. 59, pp. 281–293, 1993.

[114] ——, "Size reduction by interpolation in fuzzy rule bases," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, no. 1, pp. 14–25, 1997.

[115] L. T. Koczy, K. Hirota, and T. D. Gedeon, "Fuzzy rule interpolation by the conservation of relative fuzziness," Tokyo Inst. of Techn., Yokohama, Tech. Rep., 1997.

[116] L. T. Koczy, K. Hirota, and L. Muresan, "Interpolation in hierarchical fuzzy rule bases," in *IEEE International Conference on Fuzzy Systems*, 2000, pp. 471–477.

[117] L. T. Koczy and S. Kovacs, "The convexity and piecewise linearity of the fuzzy conclusion generated by linear fuzzy rule interpolation," *Bulletin for Studies and Exchanges on Fuzziness and its Applications (BUSEFAL)*, vol. 62, pp. 23–29, 1994.

[118] L. T. Koczy and D. Tikk, "A survey of fuzzy interpolation techniques1," in *International Symposium of Hungarian Researchers on Computational Intelligence*, 2000, pp. 5–13.

[119] S. Kovacs and L. T. Koczy, "Approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rulebase as a practical alternative of the classical CRI," in *7th International Fuzzy Systems Association World Congress*, 1997, pp. 144–149.

[120] M. Krishnamurthy, E. S. Seagren, R. Alder, A. W. Bayles, J. Burke, S. Carter, and E. Faskha, *How to Cheat at Securing Linux*. Syngress Publishing, Inc., 2008.

[121] R. Kumar and Jyotishree, "Blending roulette wheel selection and rank selection in genetic algorithms," *International Journal of Machine Learning and Computing*, vol. 2, no. 4, pp. 365–370, 2006.

[122] C. B. Lahti and R. Peterson, *Sarbanes-Oxley Compliance Using COBIT and Open Source Tools*, 3rd ed. O'Reilly & Associates, Inc., 2005.

[123] P. M. Larsen, "Industrial applications of fuzzy logic control," *International Journal of Man-Machine Studies*, vol. 12, no. 1, pp. 3–10, 1980.

[124] L. W. Lee and S. M. Chen, "A new fuzzy interpolative reasoning method for sparse fuzzy rule-based systems," *Lecture Notes in Computer Science LNCS*, pp. 745–755, 2007.

[125] ——, "Fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on the ranking values of fuzzy sets," *Expert Systems with Applications*, vol. 35, no. 3, pp. 850–886, 2008.

[126] ——, "Fuzzy interpolative reasoning using interval type-2 fuzzy sets," *New Frontiers in Applied Artificial Intelligence*, pp. 92–101, 2008.

[127] S. Lee, X. Zeng, and H. Wang, "Generating automatic fuzzy system from relational database system for estimating null values," *Cybernetics and Systems*, vol. 40, no. 6, pp. 528–548, 2009.

[128] H. L. Lin, F. W. Yang, and Y. T. Kao, "An efficient ga-based clustering technique," *Tamkang Journal of Science and Engineering*, vol. 8, no. 2, pp. 113–122, 2005.

[129] J. Y. Lin, C. T. Cheng, Y. G. Sun, and K. Chau, "Long-term prediction of discharges in manwan hydropower using adaptive-network-based fuzzy inference systems models," in *Lecture Notes in Computer Science - Advances in Natural Computation*, L. Wang, K. Chen, and Y. S. Ong, Eds. Berlin: Springer-Verlag, 2005, pp. 1152–1161.

[130] B. D. Liu, C. Y. Chen, and J. Y. Tsao, "Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, no. 1, pp. 32–53, 2001.

[131] P. Liu, D. Zhou, and N. Wu, "VDBSCAN: Varied density based spatial clustering of applications with noise," in *International Conference on Service Systems and Service Management*, 2007, pp. 1–4.

[132] G. Lyon. (1997) Nmap. [Online]. Available: http://nmap.org/.

[133] ——. (1997) NMAP(1) - linux man page. [Online]. Available: http://linux.die.net/man/1/nmap.

[134] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[135] Macupdate. (2014) Wireshark: Network protocol analyzer. [Online]. Available: http://www.macupdate.com/app/mac/23765/wireshark.

[136] E. H. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers," *International Journal of Man-Machine Studies*, vol. 8, no. 1, pp. 669–678, 1976.

[137] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[138] D. McNeill and P. Freiberger, *Fogic Logic*. Simon and Schuster, New York, 1993.

[139] D. Michie, D. Spiegelhalter, and C. Taylor. (1994) Machine learning, neural and statistical classification. [Online]. Available: http://www1.maths.leeds.ac.uk/~charles/statlog/whole.pdf.

[140] B. L. Milenova and M. M. Campos, "O-Cluster: Scalable clustering of large high dimensional data sets," in *IEEE Conference on Data Mining*, 2002, pp. 290–297.

[141] S. Mizik, "Fuzzy rule interpolation techniques in comparison," *MFT. Periodika 2001-04, Hungarian Society of IFSA*, 2001.

[142] S. Mohan and S. Bhanot, "Comparative study of some adaptive fuzzy algorithms for manipulator control," *International Journal of Computational Intelligence*, vol. 3, pp. 303–311, 2006.

[143] R. Moody. (2001) Ports and port scanning: An introduction. [Online]. Available: http://www.isaca.org/Journal/Past-Issues/2001/Volume-5/Pages/Ports-and-Port-Scanning-An-Introduction.aspx.

[144] E. Mooi and M. Sarstedt, "Chapter 9: Cluster analysis," in *A Concise Guide to Market Research: The Process, Data, and Methods Using IBM SPSS Statistics*, E. Mooi and M. Sarstedt, Eds. Berlin: Springer-Verlag, 2011, pp. 237–284.

[145] N. Mori and H. Kita, "Genetic algorithms for adaptation to dynamic environments - a survey," in *26th Annual Conference of the IEEE Industrial Electronics Society*, 2000, pp. 2947–2952.

[146] M. Mukaidono, L. Ding, and Z. Shen, "Approximate reasoning based on revision principle," in *5th Berkeley Symposium on Mathematical Statistics and Probability*, 1990, pp. 94–97.

[147] MyIP. (2014) Facts about port scanning. [Online]. Available: http://whatismyipaddress.com/port-scan.

[148] N. Naik, R. Diao, C. Quek, and Q. Shen, "Towards dynamic fuzzy rule interpolation," in *IEEE International Conference on Fuzzy Systems*, 2013, pp. 1–7.

[149] N. Naik, R. Diao, and Q. Shen, "Genetic algorithm-aided dynamic fuzzy rule interpolation," in *IEEE World Congress on Computational Intelligence*, 2014, pp. 2198–2205.

[150] N. Naik, P. Su, and Q. Shen, "Integration of interpolation and inference," in *UK Workshop on Computational Intelligence*, 2012, pp. 1–7.

[151] V. Nissen and J. Propach, "Optimization with noisy function evaluations," in *Lecture Notes in Computer Science - Parallel Problem Solving from Nature*, A. E. Eiben, T. Back, M. Schoenauer, and H. P. Schwefel, Eds. Berlin: Springer-Verlag, 2006, vol. 1498, pp. 159–168.

[152] Nmap. (2014) Chapter 15: Nmap Reference Guide. [Online]. Available: http://nmap.org/book/man.html.

[153] A. Patcha, "Network anomaly detection with incomplete audit data," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2006.

[154] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar, "A density based algorithm for discovering density varied clusters in large spatial databases," *International Journal of Computer Applications*, vol. 3, no. 6, pp. 1–4, 2010.

[155] M. Richard. (2001) Intrusion detection FAQ: Are there limitations of intrusion signatures? [Online]. Available: http://www.sans.org/security-resources/idfaq/limitations.php.

[156] M. Roesch. (2014) Snort. [Online]. Available: https://www.snort.org/.

[157] L. Rokach and O. Maimon. (2014) Chapter 15: Clustering Methods. [Online]. Available: http://www.ise.bgu.ac.il/faculty/liorr/hbchap15.pdf.

[158] T. J. Ross, *Fuzzy Logic with Engineering Applications*, 3rd ed. Hoboken, New Jersey, USA: Wiley-Blackwell, 2010.

[159] M. Rouse. (2014) Snort. [Online]. Available: http://searchmidmarketsecurity.techtarget.com/definition/Snort.

[160] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *IEEE International Conference on Computer Vision*, 1998, pp. 59–66.

[161] G. Sadowsky, J. X. Dempsey, A. Greenberg, B. J. Mack, and A. Schwartz, *Information Technology Security Handbook*, 2nd ed. O'Reilly & Associates, Inc., 2003.

[162] S. Saga, H. Makino, and J. I. Sasaki, "A method for modelling freehand curves- The fuzzy spline interpolation," *Systems and Computers in Japan*, vol. 26, pp. 77–87, 1995.

[163] J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 169–194, 1998.

[164] S. Sangeetha and V. Vaidehi, "Fuzzy aided application layer semantic intrusion detection system - fasids," *International Journal of Network Security and Its Applications*, vol. 2, no. 2, pp. 39–56, 2010.

[165] K. Scarfone and P. Mell, *Guide to Intrusion Detection and Prevention Systems (IDPS)*. National Institute of Standards and Technology, 2007.

[166] R. Shanmugavadivu and N. Nagarajan, "Network intrusion detection system using fuzzy logic," *Indian Journal of Computer Science and Engineering*, vol. 2, no. 1, pp. 101–111, 2011.

[167] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wavecluster: A multi-resolution clustering approach for very large spatial databases," in *24th Conference on Very Large Data Bases (VLDB)*, 1998, pp. 428–439.

[168] D. Shepard, "A two dimensional interpolation function for irregularly spaced data," in *23rd ACM International Conference*, 1968, pp. 517–524.

[169] K. Singh, D. Malik, and N. Sharma, "Evolving limitations in k-means algorithm in data mining and their removal," *International Journal of Computational Engineering & Management (IJCEM)*, vol. 12, pp. 105–109, 2011.

[170] A. Siraj, S. M. Bridges, and R. B. Vaughn, "Fuzzy cognitive maps for decision support in an intelligent intrusion detection system," in *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 2001, pp. 2165–2170.

[171] S. N. Sivanandam, S. Sumathi, and S. N. Deepa, *Introduction to Fuzzy Logic using MATLAB*. Springer-Verlag, Berlin Heidelberg, 2007.

[172] P. Su, C. Shang, and Q. Shen, "Owa aggregation of fuzzy similarity relations for journal ranking," in *IEEE International Conference on Fuzzy Systems*, 2013, pp. 1–7.

[173] M. Sugeno and G. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, no. 1, pp. 15–33, 1988.

[174] A. Sundaram, "An introduction to intrusion detection," *The ACM Magazine for Student*, vol. 2, no. 4, pp. 3–7, 1996.

[175] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 1, pp. 116–132, 1985.

[176] D. Tikk and P. Baranyi, "Comprehensive analysis of a new fuzzy rule interpolation method," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 3, pp. 281–296, 2000.

[177] D. Tikk, P. Baranyi, Y. Yam, L. T. Koczy, and L. Nadai, "A new method for avoiding abnormal conclusion for α-cut based rule interpolation," in *IEEE International Conference on Fuzzy Systems*, 1999, pp. 383–388.

[178] D. Tikk, I. Joó, L. T. Kóczy, P. Várlaki, B. Moser, and T. D. Gedeon, "Stability of interpolative fuzzy kh-controllers," *Fuzzy Sets and Systems*, vol. 125, no. 1, pp. 105–119, 2002.

[179] Y. Tsukamoto, "An approach to fuzzy reasoning method," in *Advances in Fuzzy Sets Theory and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds. Amsterdam: North-Holland, 1979, pp. 137–149.

[180] ——, "Fuzzy logic based on lukasiewicz logic and its application to diagnosis and control," Doctoral Dissertation, Tokyo Institute of Technology, 1979.

[181] B. Tullis. (2013) Conducting defensive information warfare on open platforms. [Online]. Available: http://events.linuxfoundation.org/sites/events/files/slides/Defensive%20Information%20Warfare%20on%20Open%20Platforms_0.pdf.

[182] I. B. Turksen and Y. Tian, "How to select combination operators for fuzzy expert systems using cri," *Johnson Space Center, North American Fuzzy Logic Processing Society (NAFIPS 1992)*, vol. 1, pp. 29–38, 1992.

[183] I. B. Turksen and Z. Zhong, "An approximate analogical reasoning approach based on similarity measures," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, no. 6, pp. 1049–1056, 1988.

[184] M. Van Pelt, *Fuzzy Logic Applied to Daily Life*. Champaign, Illinois, USA: University of Illinois Press, 2008.

[185] G. Varenni. (2013) Winpcap. [Online]. Available: http://www.winpcap.org/.

[186] G. Vass, L. Kalmoar, and L. T. Koczy, "Extension of the fuzzy rule interpolation method," in *International Conference on Fuzzy Sets Theory Applications*, 1992, pp. 1–6.

[187] J. Wack, M. Tracy, and M. Souppaya, "Guideline on network security testing," Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Tech. Rep., 2003.

[188] D. Wang, X. Zeng, and J. A. Keane, "An evolving construction scheme for fuzzy systems," *IEEE Transaction on Fuzzy Systems*, vol. 18, no. 4, pp. 755–770, 2010.

[189] H. Wang, S. Kwong, Y. Jin, W. Wei, and K. F. Man, "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction," *Fuzzy Sets and Systems*, vol. 93, no. 149, pp. 149–186, 2005.

[190] W. Wang, J. Yang, and R. Muntz, "Sting : A statistical grid appraoch to spatial data mining," in *23rd Conference on Very Large Data Bases (VLDB)*, 1997, pp. 186–195.

[191] Y. Wang, H. Deng, and Z. Chen, "Adaptive fuzzy logic controller with rule-based changeable universe of discourse for a nonlinear MIMO system," in *Intelligent Systems Design and Applications*, 2005, pp. 8–13.

[192] D. Whitley. (1993) A Genetic Algorithm Tutorial. [Online]. Available: http://www.cs.colostate.edu/~genitor/MiscPubs/tutorial.pdf.

[193] J. Williams, "The ACPO Good practice guide for managers of e-Crime investigation," Metropolitan Police Service, UK, Tech. Rep., 2013.

[194] K. W. Wong, T. D. Gedeon, and D. Tikk, "An improved multidimensional α-cut based fuzzy interpolation technique," in *International Conference on Artificial Intelligence in Science and Technology*, 2000, pp. 29–32.

[195] ——, "Fuzzy rule interpolation for multidimensional input spaces with applications: A case study," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 809–819, 2005.

[196] S. Wu and M. Joo, "A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 578–594, 2001.

[197] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A Review," Department of Computer Science, Memorial University of Newfoundland, Tech. Rep., 2008.

[198] X. Xu, M. Ester, H. P. Kriegel, and J. Sander, "A distribution-based clustering algorithm for mining in large spatial databases," in *14th International Conference on Data Engineering*, 1998, pp. 324–331.

[199] R. R. Yager, "On ordered weighted averaging aggregation operators in multi-criteria decision making," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, pp. 183–190, 1988.

[200] ——, "Generalized owa aggregation operators," *Fuzzy Optimization and Decision Making*, vol. 3, pp. 93–107, 2004.

[201] R. R. Yager and P. Filev, "Induced ordered weighted averaging operators," *IEEE Trans. Syst., Man, Cybern.*, vol. 29, pp. 141–150, 1999.

[202] Y. Yam and L. Kóczy, "Representing membership functions as points in high-dimensional spaces for fuzzy interpolation and extrapolation," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 6, pp. 761–772, 2000.

[203] Y. Yam, M. Wong, and P. Baranyi, "Interpolation with function space representation of membership functions," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 3, pp. 398–411, 2006.

[204] S. Yan, M. Mizumoto, and W. Z. Qiao, "Reasoning conditions on koczy's interpolative reasoning method in sparse fuzzy rule bases," *Fuzzy Sets and Systems*, vol. 75, pp. 63–71, 1995.

[205] ——, "An improvement to Koczy and Hirota's interpolative reasoning in sparse fuzzy rule bases," *International Journal of Approximate Reasoning*, vol. 15, pp. 185–201, 1996.

[206] L. Yang and Q. Shen, "Adaptive fuzzy interpolation," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 6, pp. 1107–1126, 2011.

[207] M. Younes, M. Rahli, and L. A. Koridak, "Economic power dispatch using evolutionary algorithm," *Journal of Electrical Engineering*, vol. 57, no. 4, pp. 211–217, 2006.

[208] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[209] ——, "Outline of a new approach to the analysis of complex systems and decision process," *IEEE Trans. Syst., Man, Cybern.*, vol. 3, pp. 28–44, 1973.

[210] ——, "The concept of a linguistic variable and its applications to approximate reasoning. part i," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[211] ——, "The concept of a linguistic variable and its applications to approximate reasoning. part ii," *Information Sciences*, vol. 8, no. 4, pp. 319–357, 1975.

[212] ——, "The concept of a linguistic variable and its applications to approximate reasoning. part iii," *Information Sciences*, vol. 9, no. 1, pp. 43–80, 1975.

[213] H. Zhang and Z. Bien, "Adaptive fuzzy control of MIMO nonlinear systems," *Fuzzy Sets and Systems*, vol. 115, no. 1, pp. 191–204, 2000.

[214] B. Zhou, D. W. Cheung, and B. Kao, "A fast algorithm for density-based clustering in large database," in *Methodologies for Knowledge Discovery and Data Mining*, N. Zhong and L. Zhou, Eds. Berlin Heidelberg: Springer-Verlag, 1999, pp. 338–349.

[215] Y. F. Zhou, Q. B. Liu, S. Deng, and Q. Yang, "An incremental outlier factor based clustering algorithm," in *1st International Conference on Machine Learning and Cybernetics*, 2002, pp. 1358–1361.

[216] H. Zimmermann, *Fuzzy set theory and its applications*, 4th ed. Boston, Dordrecht, UK: Kluwer Academic Publishers, 2001.