University of Alberta

A Fuzzy Rulebases Similarity Measure Based on Linguistic Gradients

by

Hui Li ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science

Department of Electrical and Computer Engineering

Edmonton, Alberta

Fall, 2004

Your file Votre référence
ISBN: 0-612-95795-0
Our file Notre référence
ISBN: 0-612-95795-0

Index of Tables

Index of Figures

Chapter 1 INTRODUCTION

The prime characteristics of traditional hard computing are precision, certainty, and rigor. A computer can do a job involving routine computation more efficiently than a human being. But hard computing paradigm is seldom suitable for many real life problems. For example, a computer is far behind human beings on tasks like driving. By contrast, soft computing (SC) exploits the tolerance for imprecision, uncertainty, approximate reasoning and partial truth for obtaining low cost solutions. It simulates biological mechanisms responsible for generating natural intelligence and, at the same time, aims at a formalization of the cognitive processes humans employ so effectively in daily tasks [132].

SC consists of several computing paradigms including neural network (NN), fuzzy set theory, genetic computing such as genetic algorithms (GA), and probabilistic reasoning such as chaotic systems [113].

In general, SC does not perform much symbolic manipulation, in this sense it complements conventional AI approaches [37]. Conventional AI manipulates symbols on the assumption that human intelligence behavior can be stored in symbolically structured knowledge bases: this is known as: "the physical symbol system hypothesis". The knowledge-based system (or expert system) is an example of the most successful conventional AI product. Knowledge acquisition and representation has limited the application of AI theories (shortcoming of symbolicism). SC has become a part of "modern AI". Researchers have directed their attention toward biologically inspired methodologies such as brain modeling, evolutionary algorithm and immune modeling.

These new paradigms simulate chemico-biological mechanisms responsible for natural intelligence generation. There are two important aspects of SC [37]. First, SC is a consortium of methodologies that aim at exploiting the tolerance for imprecision, uncertainty, and partial truth to achieve tractablility, robustness, and low solution cost. Second, the constituent methodologies of SC are complementary and synergistic rather than competitive, so that in many applications it is advantageous to employ these methodologies in combination.

Within SC, the main contribution of fuzzy logic is a mechanism for dealing with imprecision and partial truth – a mechanism in which the principal tools are the calculus of fuzzy if-then rules. The premises and conclusions of fuzzy if-then rules are linguistic variables, the meanings of which are given by associating the atomic terms with fuzzy membership functions via semantic rules.

One important application of fuzzy logic is fuzzy systems in the field of control. Fig 1.1 gives the block diagram of a fuzzy system. The key components of a fuzzy system are the rulebase and the inference engine. The rulebase uses fuzzy set theory to express a set of if-then rules in natural language, while the inference engine executes an approximate reasoning strategy based on these rules. Through the use of linguistic rules, the fuzzy system becomes intuitively understandable for humans. In any actual control system, all inputs and outputs are numeric value, but the inference engine takes linguistic inputs only. Therefore, a fuzzifier is used to convert numeric value to linguistic value. Likewise, if the inference engine outputs linguistic values, a defuzzifier is needed to convert linguistic value to numeric value.

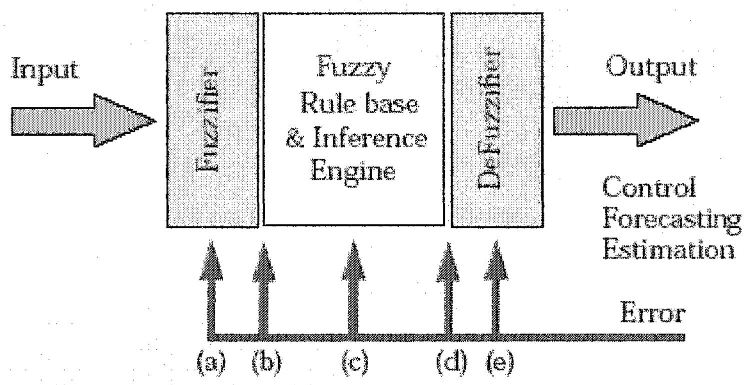


Figure 1.1 Block diagram of a general fuzzy inference system: The error value from a given performance measure is fed back and used to adapt all or one of the following: (a) Membership function shapes and cardinality, (b) & (d) And/Or aggregation operators, (c) The rule base, and (e) The defuzzification technique.

One of the current areas of concentration in soft computing research is the area of "computing with words" (CW), which refers to a set of techniques for using words from natural language as the basic units of information in a computer [38]. Using words as a basic unit of information instead of numbers will hopefully allow computers to mimic the tolerance for imprecision and ambiguity found in human thought processes. CW poses certain problems and opens up new directions of research. One important problem is how we can analyze a fuzzy system using CW methods. Fuzzy sets and rulebases can be defined by human experts [37] or automatically generated by employing various methods [2, 39-44]. Even for the same system, different persons or different methods will generate different antecedents, consequents and rulebases. Each of these different fuzzy systems is an approximate description of the underlying system, and it would be very useful to determine how similar these different interpretations are. The algorithm we propose is a method for measuring similarity between rulebases within the CW framework.

There are many similarity measures for fuzzy sets and linguistic variables [45-48]. However, there is no such measure for fuzzy rulebases in the literature. One traditional measure for comparing fuzzy inference systems is the root-mean-square (RMS) difference. This approach has its drawbacks. RMS is computed over a discretized input space $\mathbf{Z}^n$. The step size (sampling rate) must be set to a very small value that greatly affects performance. Larger step sizes improve performance, but may miss significant local behaviours. Moreover, the output of a Mamdani system [51] is the union of some linguistic values, which need to be defuzzified to get crisp value. The defuzzification accuracy also affects the performance.

The field of content-based image retrieval (CBIR) has also faced similar problems, and has found mechanisms to overcome them. One possibility is including structural information about an image (or a rulebase; we will discuss the common ground between these domains later.) In this paper, we defined a new similarity measure for fuzzy rulebases called Similarity Confidence Level (SCL). This similarity operator is inspired by a CBIR algorithm proposed by J. Rose, *et al.* [36], which compares the projection vectors of gradient maps of 2 images. The SCL algorithm provides a measure of similarity based directly on the structural characteristics of a fuzzy rulebase, rather than determining a numeric distance from a defuzzified reasoning surface. The final SCL is a value in the range [0, 1], with 0 representing no similarity and 1 representing identity.

The remainder of this thesis is organized as follows: Chapter 2 reviews the fuzzy systems, SC, and granular computing theory. Chapter 3 presents the major contribution of this thesis, the definition of the fuzzy systems Similarity Confidence Level based on linguistic gradient. Chapter 4 gives the experimental result and analysis. A summary of the result and direction of future research in Chapter 5 concludes this thesis.

Chapter 2 BACKGROUND

This chapter, we first review the history of fuzzy logic, the fuzzy sets theory, and fuzzy systems. Second, we describe in details the linguistic mathematic and linguistic gradient operator, which is the foundation of our research. Then, we present the new development in a sub-area of fuzzy theory, granular computing. At the end of this chapter, we introduce other methodologies of soft computing (SC), the artificial neural network and genetic algorithm.

2.1 Historic Notes

Classical prepositional logic and predicate logic handle binary values, i.e. 0 and 1, or true and false. They have achieved great successes in many areas. But the binary values limit their ability to represent imprecision and uncertainty. Thus, people invented some ways to represent partially truth or probability, using gray scale between 0 and 1 or values between true and false.

Theory of fuzzy sets was founded by Lotfi Zadeh [7] in 1965. In his paper, Zadeh presented a new set whose boundaries were not well defined, although most of the though in this idea was envisioned by Max Black [49] in 1937. In fact, Zadeh was influenced by multi-valued logic. Along with another paper by Zadeh in 1973 [50], this paper forms the theoretical basis for the field of fuzzy systems. The first axiomatic treatment of fuzzy set operations was presented by Bellman and Giertz [14] in 1973. A thorough investigation of properties of the max and min operators was done by Voxman and Goetschel [15]. However, the axiomatic skeletons for fuzzy operations based on *triangular norms* (or *t-norms*) and *triangular conorms* (or *t-conorms*) can be dated back to the early 1960's by

Schweizer and Sklar [16-18]. The *extension principle* was introduced by Zadeh [19] that provided a general procedure for extending crisp domains of mathematical expression to fuzzy domains. A further elaboration of the principle was presented by Yager [20]. The development of fuzzy system applications began with a paper by Mamdani in 1974 [51] and continued with papers by Takagi and Sugeno in 1985 [52], and Sugeno and Kang in 1986 [53]. Fuzzy techniques were used to implement an expert system called Linkman [21], by Blue Circle Cement and SIRA in Denmark. The system incorporates the experience of operators in a cement production facility and has been in operation since 1982, making it the first major industrial fuzzy system application. The success of the Sendai City subway in 1987, the first large-scale project utilizing fuzzy control, spurred the commercial applications and research in Japan and United States [4]. The field of fuzzy systems is currently the principal application of fuzzy set theory. This field is essentially split into two camps: Mamdani fuzzy systems [51] and Sugeno fuzzy systems [52, 53]. These two types of fuzzy systems both rely on fuzzy rules for their functionality. However, their implementations of these rules are different. Sugeno fuzzy systems are more amenable to classical analytical techniques [3, 54-56, 58, 59], whereas Mamdani fuzzy systems are more intuitive.

Zadeh distinguished two main directions in fuzzy logic [22]. Fuzzy logic in the broad sense is older, better known, heavily applied but not asking deep logical questions, serving mainly as apparatus for fuzzy control, analysis of vagueness in natural language and several other application domains. Being one of the techniques of soft computing, it is a method tolerant to sub-optimality and impreciseness and giving quick, simple and sufficiently good solutions. Novak [23], Zimmermann [24], Klir *et al.* [25], and Nguyen [26] address developments in this direction.

Fuzzy logic in the narrow sense is symbolic logic with a comparative notion of truth developed fully in the scope of classical logic, including syntax, semantics, axiomatization, truth-preserving deduction, completeness, etc.; both propositional and predicate logic. It is a branch of multi-valued logic based on the paradigm of inference under vagueness. This fuzzy logic is a relatively young discipline, both serving as a foundation for the fuzzy logic in a broad sense and of independent logical interest. A basic monograph is by Hajek [27], further monographs are by Turunen [28], Novak *et al.* [29].

2.2 Fuzzy set theory

2.2.1 Fuzzy Sets

The most widely used tool for passing information is natural language. Natural language is intuitive, at the same time vague and imprecise. However, it is still the most powerful tool for human communication. The basic function of fuzzy sets theory is to use linguistic variables instead of numeric variables to represent imprecise concepts. Linguistic variables can take values from a collection of atomic terms that are given meaning by associating with fuzzy sets. For example, a numeric variable "age" can take a real value greater than 0; a linguistic variable "age" can take a linguistic value "young", "mid-age", or "old", where "young", "mid-age", and "old" are fuzzy sets. Therefore, our review begins with fuzzy set theory.

A classical set is a set with crisp boundary, i.e. an element belongs to the set or otherwise does not belong to the set. It does not reflect the nature of abstract and imprecise human concepts and thoughts. For example, we define a classical set of "tall person", of which the boundary is that the height is over 6 ft or any other precise value. This is unnatural and inadequate to represent our usual concept of a "tall person". It makes no sense that we classify a person of 6 ft as tall and a person of 5.99ft as not tall. The flaw comes from the sharp transition between inclusion and exclusion in a set. Fig 2.1 (a) shows an example of crisp set.

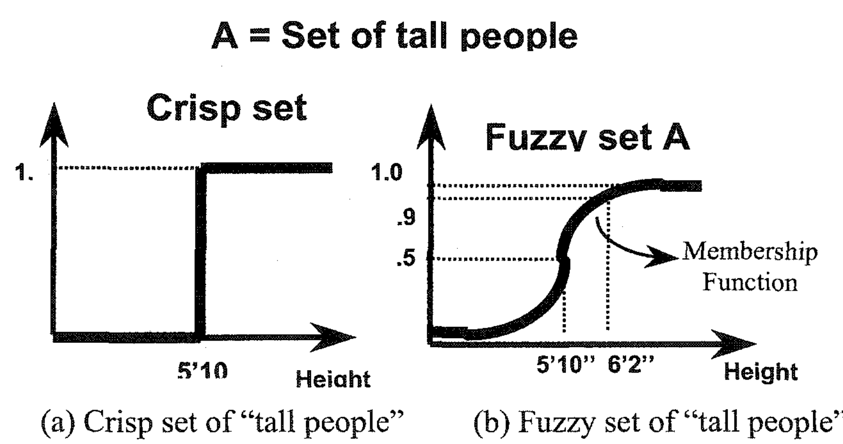


Figure 2.1 Crisp set and fuzzy set definition of "tall people"

A fuzzy set is a set without a crisp boundary, i.e. the transition from membership in a set to non-membership is gradual and characterized by membership functions that give fuzzy sets flexibility in modeling commonly used linguistic expressions.

Definition 2.1

X is a collection of objects denoted by x, then a fuzzy set A in X is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) \mid x \in X, \mu_A \in [0,1]\}, \qquad (2.1)$$

where $\mu_A(x)$ is the Membership Function (MF) for the fuzzy set A.

The MF maps each element x of X to a membership value in the interval [0, 1] representing the grade of membership of x in A. The universe of discourse, X, can be discrete nominal, discrete ordinal or continuous. Fig 2.1 (b) gives an example of the definition of a fuzzy set "tall people".

For simplicity of notation, a fuzzy set A can be denoted as follows:

$$A = \begin{cases} \sum_{x_i \in X} \mu_A(x_i)/x_i & \text{if } X \text{ is a discrete space.} \\ \int_X \mu_A(x)/x, & \text{if } X \text{ is a continuous space.} \end{cases} \qquad (2.2)$$

The MF provides a mathematically rigorous representation of uncertainty [4]. The classical set (crisp set) is a special case of fuzzy sets, in which the MF can take values from $\{0, 1\}$.

Here we provide a review of some properties and operations of fuzzy set that will be important in the remainder of this thesis.

Definition 2.2

The support of a fuzzy set A is the set of all points x in X such that $\mu_A(x)>0$:

$$\text{support}(A) = \{x \mid \mu_A(x)>0\}. \qquad (2.3)$$

Definition 2.3

The core of a fuzzy set is the crisp set of elements whose membership in the fuzzy set equals 1.

$$core(A) = \{x \mid \mu_A(x) = 1\} \tag{2.4}$$

Definition 2.4

A crossover point of a fuzzy set is a point $x \in X$ at which $\mu_A(x) = 0.5$:

$$crossover(A) = \{x \mid \mu_A(x) = 0.5\} \tag{2.5}$$

Definition 2.5

The α -cut or α -level set of a fuzzy set A is a crisp set defined by

$$A_\alpha = \{x \mid \mu_A(x) >= \alpha\} \tag{2.6}$$

The strong α -cut or strong α -level set is defined by:

$$A'_\alpha = \{x \mid \mu_A(x) > \alpha\} \tag{2.7}$$

The α -cut provides a means of transforming a fuzzy set into a collection of crisp sets.

Definition 2.6

A fuzzy set A is convex iff for any $x_1, x_2 \in X$ and any $\lambda \in [0, 1]$,

$$\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min\{\mu_A(x_1), \mu_A(x_2)\}, \tag{2.8}$$

Convexity is important in the definition of a fuzzy number, and in the interpretation of fuzzy sets; convex fuzzy sets tend to have a clear intuitive meaning, while non-convex fuzzy sets do not.

Definition 2.7

A fuzzy set A is normal if its core is nonempty. In other words, A fuzzy set is normal if $\exists x \in X$ such that $\mu(x) = 1$

Normality is a standard requirement of fuzzy sets used in fuzzy inferential systems, as it simplifies the mathematics involved.

Definition 2.8

A fuzzy number A is a fuzzy set in the real line $\boldsymbol{R}$ that satisfies the conditions for normality and convexity.

Because fuzzy sets are a generalization of the classical set theory, the embedding of conventional models into a larger setting endows fuzzy models with greater flexibility to capture various aspects of incompleteness or imperfection in a real process. The flexibility of fuzzy set theory is associated with the elasticity property of the concept of its membership function.

The specification of membership function is subjective, which means that the membership functions specified for the same concept by different persons may vary considerably. It is convenient to employ standardized functions with adjustable parameters, such as Triangular MFs and Trapezoidal MFs, which are piecewise differentiable. Fig 2.2 gives an example of a triangular fuzzy set "mf2" and its core, supports and crossover points. There are other types of everywhere-differentiable MFs such as Gaussian MFs, Generalized bell MFs and Sigmoidal MFs. However, besides being defined by human experts, other methods for assigning MFs to fuzzy sets were proposed to reduce subjectivity. Hadipriono and Sun [30] introduced angular fuzzy set. Takagi and Hayashi [31] employed neural network to create MFs. Karr and Gentry [32] proposed using genetic algorithm (GA) to compute MFs. De Luca and Termini [33] defined non-probabilistic entropy to assign MFs.

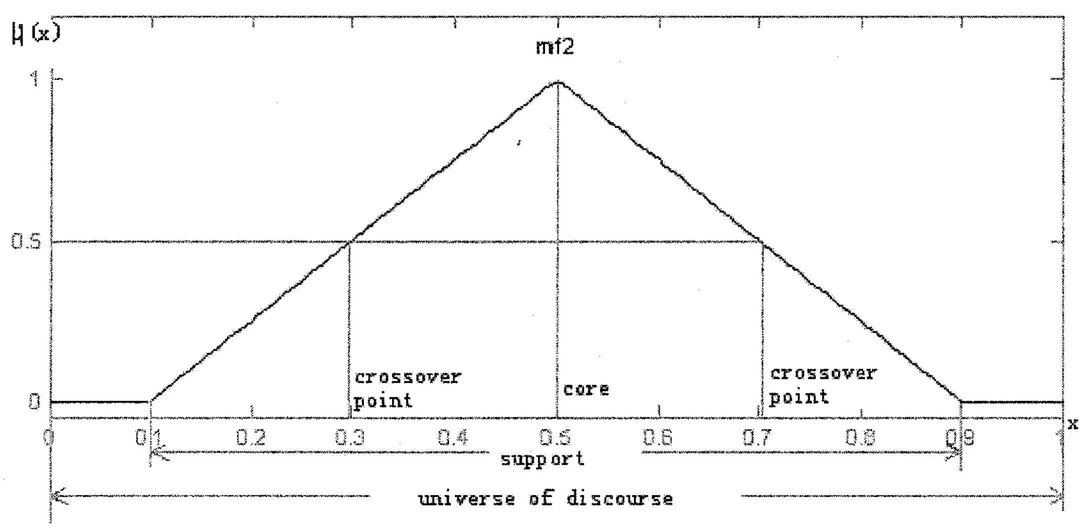


Figure 2.2 Core, supports and crossover points of a triangular fuzzy set "mf2"

Union, intersection, and complement are the most basic operations on classical sets. Corresponding to the ordinary set operations of union, intersection, and complement, Zadeh defined similar operations for fuzzy sets [7]:

Definition 2.9

The union of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cup B$ or $C = A$ OR B, whose MF is given by

$$\mu_C(x) = \mu_A(x) \vee \mu_B(x) \qquad (2.9)$$

where $\bigvee$ denotes max.

Definition 2.10

The intersection of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cap B$ or $C = A$ AND B, whose MF is given by

$$\mu_C(x) = \mu_A(x) \wedge \mu_B(x) \qquad (2.10)$$

where $\bigwedge$ denotes min.

Definition 2.11

The complement of fuzzy set A, denoted by $\overline{A}$ or NOT A, is defined as

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x) \qquad (2.11)$$

There are many other viable ways to define operations on fuzzy sets. The only restriction is that these functions must obey certain axioms for each operation.

A fuzzy complement operator is a continuous functions N: [0, 1] -> [0, 1] which meets the following axiomatic requirements:

- $N(0) = 1$ and $N(1) = 0$ (boundary)
- $N(a) >= N(b)$ if $a <= b$ (monotonicity)

Another optional requirement is

- $N(N(a)) = a$ (involution)

Obviously, these axiomatic requirements for fuzzy complements do not determine N(.) uniquely. Other possible fuzzy complement definitions include Sugeno's complement [8] and Yager's complement [9].

The intersection of two fuzzy sets A and B is specified in general by a function T:[0, 1] x [0, 1] -> [0, 1], This class of operators satisfies:

- $T(0, 0) = 0$, $T(a, 1) = T(1, a) = a$ (boundary)
- $T(a, b) <= T(c, d)$, if $a <= c$ and $b <= d$ (monotonicity)
- $T(a, b) = T(b, a)$ (commutativity)
- $T(a, T(b, c)) = T(T(a, b), c)$ (associativity)

Besides min, other most frequently used intersection operators are: algebraic product, bounded product, and drastic product [37].

The fuzzy union operator is specified in general by function S: [0, 1] x [0, 1] -> [0, 1]. This class of fuzzy union operations satisfies:

- $S(1, 1) = 1$, $S(0, a) = S(a, 0) = a$ (boundary)
- $S(a, b) <= S(c, d)$ if $a<=c$ and $b <= d$ (monotonicity)

- $S(a, b) = S(b, a)$ (commutativity)
- $S(a, S(b, c)) = S(S(a, b), c)$ (associativity)

Some union definitions other than max are algebraic sum, bounded sum, and drastic sum [37].

These essential requirements for these operators cannot uniquely determine the classical fuzzy intersection and union. Intriguingly, the axioms for the fuzzy intersection operator are identically the axioms satisfied by a class of functions called t-norms. Likewise, the class of fuzzy unions is identical to the class t-conorms.

Basic identities of classical sets may not be true for fuzzy sets, *e.g.* law of contradiction and low of excluded middle do not hold for Zadeh's union and intersection definitions. For other definitions of t-norm and t-conorm, distributions laws, law of contradiction, law of the excluded middle, and absorptions laws need to be checked [10].

2.2.2 Fuzzy Relations

We start with crisp relations and then generalize to fuzzy relations. If X and Y are two sets, then $X \times Y$ is their Cartesian product. A crisp relation R between X and Y is a subset of $X \times Y$, i.e. $R \subseteq X \times Y$. Crisp relation R can be represented by its characteristic function

$$R(x, y) = \begin{cases} 1, (x, y) \subset R \\ 0, (x, y) \not\subset R \end{cases} \tag{2.12}$$

where $x \subset X$ and $y \subset Y$. $R(x, y) = 1$ means that x and y are related through relation R and $R(x, y) = 0$ means that they are not related. The inverse of R, written R^{-1}, is defined by $R^{-1}(x, y) = R(y, x)$.

If R is a crisp relation between X and Y, and S is a crisp relation between Y and Z, then the composition $R \circ S = T$ creates a new crisp relation between X and Z. The definition of T is $T(x, z) = 1$ iff there is a $y \in Y$ so that $R(x, y) = S(y, z) = 1$. We may write T as follows:

$$T(x,z) = \max_y \{\min(R(x,y), S(y,z))\} \tag{2.13}$$

the min can be replaced by any t-norm. All t-norms will give the same result.

A fuzzy relation R is just fuzzy subset of $X \times Y$. So now $R(x, y)$ can be any number in the interval [0, 1]. $R(x, y)$ gives the strength of the relationship between x and y. Now let R be a fuzzy relation on $X \times Y$ and S a fuzzy relation on $Y \times Z$. Then $T = R \circ S$ is defined as

$$T(x,z) = \max_y \{t - norm(R(x,y), S(y,z)\} \tag{2.14}$$

In general, t-conorm and t-norm can be used to interpret OR and AND operations, respectively, in composition operations. Most commonly used are max-min composition and max-product composition. The max-min composition of R and S is a fuzzy set defined by

$$\mu_T(x,z) = \max_y \min[\mu_R(x,y), \mu_S(y,z)] \tag{2.15}$$

The max-product composition is defined as

$$\mu_T(x,z) = \max_y [\mu_R(x,y)\mu_S(y,z)] \tag{2.16}$$

A generalization to n-array relations is straightforward.

2.2.3 Linguistic Variables

One of the key concepts in fuzzy systems theory is the idea of linguistic variable (LV). LV was proposed by Zadeh [19, 50] as an alternative approach to modeling human thinking. In an approximate manner, this approach serves to summarize information and express it in terms of fuzzy sets instead of crisp numbers.

Definition 2.12

A linguistic variable is characterized by a quintuple $(x, T(x), X, G, M)$ in which x is the name of the variable; $T(x)$ is the term set of x, i.e., the set of its linguistic values or linguistic terms; X is the universe of discourse; G is a syntactic rule which generates the terms in $T(x)$ through the application of linguistic hedges (in general, a context-free grammar); and M is a semantic rule which associates with each linguistic value A its meaning $M(A)$, where $M(A)$ denotes a fuzzy set in X.

The term set of a LV consists of several atomic terms modified by negation (not) and/or hedges (very, more or less, quite, etc.), and then linked by connectives (and, or, either and neither). The connectives, hedges, and negation are treated as operators that change the meaning of their operands in a specified, context-independent fashion. Hedge is a word from natural language that alters the meaning of a term. In fuzzy systems, a hedge is taken to be an operator that modifies the fuzzy set associated with a term by the semantic rule M. In [60], Zadeh defined a language as a fuzzy relation from a set of terms T to a universe of discourse U. In this paper, Zadeh defines the "meaning" of a term t to be a fuzzy set on *U*. The language *L* defined by the mapping from a given set of terms *T* to a given *U*., μ*L*(*t*, *y*), is presented as a context-free grammar. The meaning of a composite term is determined by constructing a parse tree for that term, substituting the fuzzy set associated with the atomic portion of the term for the atomic entry in the parse tree, and then applying hedge functions in the order defined by the parse tree. Several hedges are defined by the use of the standard fuzzy set operators, as well as the operators concentration (CON), dilation (DIL), and contrast intensification (INT):

$$\mu_{CON(A)} = (\mu_A(x))^2 \tag{2.17}$$

$$\mu_{DIL(A)} = (\mu_A(x))^{0.5} \tag{2.18}$$

$$\mu_{INT(A)} = \begin{cases} 2\cdot(\mu_A(x))^2, if\, 0 \le \mu_A(x) \le 0.5 \\ 1-2[1-\mu_A(x)]^2, if\, 0.5 \le \mu_A(x) \le 1 \end{cases} \tag{2.19}$$

Using these operators, Zadeh derived expressions for the hedges *very, plus or minus, highly, much, more or less, slightly* and *sort of*. These fall into a class of hedges called Type I, which do not require knowledge concerning the actions of the hedges on different elements of the fuzzy set.

2.2.4 Fuzzy if-then rules

Definition 2.13

A fuzzy if-then rule assumes the form

If *x* is *A* then *y* is *B*,

where *A* and *B* are linguistic values defined by fuzzy sets on universes of discourse *X* and *Y*, respectively. "*x* is *A*" is called the antecedent or premise, while "*y* is *B*" called the consequence or conclusion. This expression sometimes abbreviated as $A \to B$.

The meaning of if-then rule has to be formalized before we employ it to model and analyze a system. In essence, a fuzzy if-then rule is defined as a binary fuzzy relation R on the product space $X \times Y$; There are two ways to interpret the fuzzy rule $A \rightarrow B$,

as A coupled with B:

$$R = A \rightarrow B = A \times B = \int_{X \times Y} t - norm(\mu_A(x), \mu_B(y))/(x, y) \qquad (2.20)$$

or A entails B:

$$R = A \rightarrow B = \overline{A} \cup B \qquad (2.21)$$

Based on these two interpretations and various t-norm and t-conorm cooperators, a number of qualified methods can be formulated to calculate the fuzzy relation [61].

2.2.5 Fuzzy Reasoning

Fuzzy reasoning is an inference procedure that derives conclusions from a set of fuzzy if-then rules and known facts. Before introducing fuzzy reasoning, we shall discuss the compositional rule of inference, which plays a key role in fuzzy reasoning. The compositional rule of inference is proposed by Zadeh [50].

Assume that F is a fuzzy relation on $X \times Y$ and A is a fuzzy set of X, B is the resulting fuzzy set to be found of Y. Let μ_A, μ_B, and μ_F be the MFs of A, B and F, respectively. The process of inference in fuzzy systems theory is based on the Compositional Law of Inference:

$$\mu_B(y) = \max_x \min[\mu_A(x), \mu_F(x, y)] \qquad (2.22)$$

Conventionally, B is represented as $B = A \circ F$. Using the compositional rule of inference, we can formalize an inference procedure on a set of fuzzy if-then rules.

A well-known rule of inference in traditional two-valued logic is *modus ponens*, according to which we can infer the truth of a proposition B from the truth of A and the implication $A \rightarrow B$:

Premise1 (fact):	x is A,
Premise2 (rule):	if x is A then y is B
Consequence(conclusion):	y is B

However, in much of human reasoning, *modus ponens* is employed in an approximate manner:

Premise 1 (fact):	x is A'
Premise 2 (rule):	if x is A then y is B
Consequence (conclusion):	y is B'

where A' is close to A and B' is close to B. When A' and B' are fuzzy sets of a appropriate universe, this procedure is called approximate reasoning or fuzzy reasoning; it is also called generalized *modus ponens*. In a formal manner, approximate reasoning can be defined as follow:

Definition 2.14

Let A, A', and B be fuzzy sets of X, X, and Y, respectively. Assume that the fuzzy implication $A \to B$ is expressed as a fuzzy relation R on $X \times Y$. Then the fuzzy set B induced by "x is A" and the fuzzy rule "if x is A then y is B" is defined by

$$B' = A' \circ R = A' \circ (A \to B) \tag{2.23}$$

Now the inference procedure of fuzzy reasoning can be used to derive conclusions, provided that the fuzzy implication R is defined as an appropriate binary fuzzy relation.

2.3 Fuzzy Systems

The fuzzy inference system is a popular computing framework based on the concepts of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning. A general fuzzy inference system consists of three parts (see Fig. 1.1 in page 3). A crisp input is fuzzified by input membership functions and processed by a fuzzy logic interpretation of a set of fuzzy rules. This is followed by the defuzzification stage resulting in a crisp output. The rule base is typically crafted by an expert; though self organizing procedures have been suggested [57, 62-70].

There are a number of different ways to implement the fuzzy inference engine. Among the very first such proposed techniques is that due to Mamdani [51], which describes the inference engine in terms of a fuzzy relation matrix and uses the compositional rule of inference to arrive at the output fuzzy set for a given input fuzzy set. The output fuzzy set is subsequently defuzzified to arrive at a crisp control action. Other techniques include sum-product and threshold inferencing.

The basic fuzzy inference system can take either fuzzy or crisp inputs, but the outputs are always fuzzy sets. If it needs a crisp output, defuzzification is employed to extract a crisp value that best represents a fuzzy set. In the case of crisp input and output, a fuzzy inference system implements a nonlinear mapping from its input space to output space. Furthermore, the outputs of a fuzzy system are independent values. Hence, any n-input, m-output fuzzy system can be decomposed into m separate n-input, single-output fuzzy systems [3]. Therefore, without loss of generality, we will restrict our research to multiple-input, single-output fuzzy systems [3, 4].

The most widely used fuzzy system models are Mamdani fuzzy systems and Sugeno fuzzy systems.

Rules in Mamdani fuzzy models [51] are of this form:

if x is A and y is B then z is C.

Given the input (x, y) then the goal is to determine the output "z is C". First step is to map the input x_i to fuzzy set A_i, y_i to fuzzy set B_i. The next step is to evaluate the firing strength for the premise of each rule, and then apply the result to the conclusion part of each rule using the fuzzy implication. The next step is to find the output C_i of each rule. In the aggregation step, all fuzzy subsets assigned to each output variable are combined together to form a single fuzzy subset C' for each output variable. Using min and max for t-norm and t-conorm, the result can be expressed as:

$$
\begin{aligned}
C' &= \underbrace{(A'*B')}_{premise1} \circ \underbrace{(A*B \rightarrow C)}_{premise2} \\
\mu_{C'}(z) &= \underbrace{\left\{\vee_x [\mu_{A'}(x) \wedge \mu_A(x)]\right\}}_{w_1} \wedge \underbrace{\left\{\vee_y [\mu_{B'}(y) \wedge \mu_B(y)]\right\}}_{w2} \wedge \mu_C(z) \\
&= (\mathrm{w}_1 \wedge \mathrm{w}_2) \wedge \mu_C(z)
\end{aligned}
\tag{2.24}
$$

A defuzzifier is used to convert a fuzzy set to a crisp value. In general, there are five methods for defuzzification: centroid of area, bisector of area, mean of maximum, smallest of maximum, Largest of maximum. The most commonly used defuzzification method is the centroid of area (COA):

$$COA = \frac{\int x\mu(x)dx}{\int \mu(x)dx} \tag{2.25}$$

where $\mu(x)$ is the membership function of x in the fuzzy set to be defuzzified. This equation is for finding the centre of gravity of a 2-dimentional shape [4].

In consideration of computation efficiency or mathematical tractability, a fuzzy inference system in practice may have a certain reasoning mechanism that does not follow the strict definition of the compositional rule of inference. For instance, product may be used for computing firing strengths.

The goal of Sugeno fuzzy models [52, 53] is to generate fuzzy rules for a given input-output data set. A typical fuzzy rule in a Sugeno fuzzy model has the form:

if x is A and y is B then $z = f(x, y)$

Where A and B are fuzzy sets in the antecedent, while $z = f(x, y)$ is a crisp function in the consequent. $f(.,.)$ is very often a polynomial function. If $f(.,.)$ is a first order polynomial, then the resulting fuzzy inference is called a first order Sugeno fuzzy model. If $f(.,.)$ is a constant then it is a zero-order Sugeno fuzzy model (special case of Mamdani model, in which each rule's consequent is specified by a fuzzy singleton).

The output of a zero-order Sugeno model is a smooth function of its input variables as long as the neighbouring MFs in the antecedent have enough overlap. In other words, the overlap of MFs in the consequent of a Mamdani model does not have a decisive effect on the smoothness; it is the overlap of the antecedent MFs that determines the smoothness of the resulting input-output behaviour.

Since each rule has a crisp output, the overall output is obtained via weighted average, thus no defuzzification required.

Mamdani and Sugeno model are similar in that both rely on fuzzy rules for their functionality, but their implementations of these rules are different [35]. Sugeno fuzzy systems are more amenable to classical analytical techniques [54-56, 58, 59, 72], where as Mamdani fuzzy systems are more intuitive. Our research will exclusively focus on Mamdani fuzzy systems.

Turksen and Zhong [73] proposed similarity-based fuzzy reasoning, which does not require the construction of a fuzzy relation. It is based on the computation of the degree of similarity between the fact and the antecedent of a rule. Then based on the similarity value, the membership value of each element of the consequent fuzzy set of the rule is modified to obtain a conclusion. Raha *et al.* [74] improved this method by proposing a set of axioms to compute a reasonable measure of similarity between two imprecise concepts represented as fuzzy sets.

2.4 Granular Computing

2.4.1 Introduction

Granular computing (GC) is an emerging conceptual and computing paradigm of information processing [2]. It concerns processing of information granules, which are collections of entities that are arranged together due to their similarity, functional adjacency, indistinguishability, coherency or alike [2]. Information granules as an abstraction of the reality are aimed at building efficient and user-centered models of the external world and supporting our perception of the physical and virtual world. In this sense, fuzzy sets are also information granules. Some concepts and ideas in granular computing play important roles in our research, therefore it is appropriate to introduce granular computing here.

The research agenda of granular computing includes a series of key and well-defined methodological and algorithmic issues [2]:

1) Construction of information granules. There are a number of formal frameworks for building information granules. They are set theory and interval analysis, fuzzy sets, rough sets, shadowed sets, probabilistic sets, probability-based granular constructs, and higher-level granular constructs. These frameworks are thoroughly investigated and have a vast array of applications. The process of granulation and the nature of information granules imply a certain formalism that is most suitable to capture the problem at hand.
2) Characterization of dimension (granularity) of information granules. One of the common characteristics for different frameworks is granularity, which describes how specific the granule is and how many details it embraces. One commonly used notation is cardinality, which can be computed by counting the number of elements in the information granule. The higher the cardinality, the higher the abstraction of the granule and the lower its granularity. We use information granules to perceive and describe the problem as well as plan some interaction with the external world. The type of description and interaction dictates the level of granularity: The most suitable

level is selected. The level of granularity also influences the computational efficiency.

3) The development of the encoding and decoding mechanisms. Granular worlds rarely exist and operate independently without interaction with the environment. Hence communication mechanisms are needed. There are two categories of the communications tasks. One involves two granular worlds built on the some formalism. The other has no constraint on the formalism of the granular information. Communication mechanisms for granular worlds at different level of granularity are referred to as encoding and decoding. For example, A/D and D/A conversions are encoding and decoding mechanisms for communications between analog world and digital world.

4) Interoperability between different formal platforms of information granules. This issue is crucial to the design of hybrid models operating within the realm of various formalisms of information granularity.

Pedrycz *et al.* [2] identified 5 types of data: intervals, fuzzy sets, rough sets, shadowed sets, and probabilistic sets. These types of data are viewed as different granularities of information.

2.4.2 Intervals

Interval arithmetic was introduced by Warmus [12, 13]. Intervals are connected subset in the real set R, so the interval analysis can be considered as a special case of set analysis. An interval I can be represented by an ordered real pair $[a, b]$, where a is its lower bound and b is its upper bound. Note that the interval $[a, b]$ may be closed or not, i.e., the lower and upper bounds may or may not belong to the interval. When $a = b$, the interval I is converted to a real.

Using the lower and upper bounds, the width and center of the interval can be defined as:

Definition 2.15

$$\text{width}([a, b]) = b - a \tag{2.26}$$

$$\text{center}([a, b]) = (a + b) / 2 = a + \text{width}([a, b]) / 2 \tag{2.27}$$

Dual operations {+, -, *, /} for two intervals $[a, b]$ and $[c, d]$ result in another interval, whose value depends on the value and sign of a, b, c, and d. For the 6 possible combinations of a, b, c, and d, where $a < b$ and $c < d$, Table 2.1 gives their set operations AND and OR.

Table 2.1 SET OPERATIONS FOR TWO INTERVALS

Situation	AND ∩	OR ∪
$a > d$	φ	$[c, d] \cup [a, b]$
$c > b$	φ	$[a, b] \cup [c, d]$
$a > c, b < d$	$[a, b]$	$[c, d]$
$c > a, d < b$	$[c, d]$	$[a, b]$
$a < c < b < d$	$[c, b]$	$[a, d]$
$c < a < d < b$	$[a, d]$	$[c, b]$

Based on Table 2.1, interval arithmetic operations are as follows:

$$[a,b]+[c,d]=[a+c,b+d] \quad (2.28)$$

$$[a,b]-[c,d]=[a-d,b-c] \quad (2.29)$$

$$[a,b]*[c,d]=[\min(ac,ad,bc,bd),\max(ac,ad,bc,bd)] \quad (2.30)$$

$$[a,b]/[c,d]=[a,b]*[1/d,1/c], 0 \notin [c,d] \quad (2.31)$$

$$\alpha[a,b]=\begin{cases}[\alpha a,\alpha b], \alpha>0 \\ [\alpha b,\alpha a], \alpha<0\end{cases} \quad (2.32)$$

Note that the distributivity doesn't hold for + and *, because two intervals may have the same lower and upper bounds [11]. For three intervals *I*, *J* and *K*,

$$I*[J+K] \subset I*J+I*K \quad (2.33)$$

2.4.3 Rough sets

In the past decade, the field of rough sets has grown rapidly. Rough set data analysis (RSDA), developed by Z. Pawlak and his co–workers in the early 1980s [92], has become a recognized and widely researched method. RSDA is generally regarded as part of the "Soft Computing" paradigm [93]. However, while other soft methods require additional model assumptions such as the representability of the collected sample, prior probabilities, fuzzy functions, or degrees of belief, RSDA is unique in the sense that it is "non– invasive" i.e. that it uses only the information given by the operationalised data, and does not rely on other model assumptions. In other words, instead of using external numbers or other additional parameters, rough set analysis utilizes solely the structure of the given data.

An information system S is a formal structure viewed as a four-tuple of the form $S = \langle X, Q, V, f \rangle$, whose components assume the following roles. X is a finite universe including all elements we are interested in some problem description, $X = \{x_1, x_2, \ldots, x_n\}$, Q is a finite set of attributes used in the description of elements of X; $Q = \{q_1, q_2, \ldots, q_n\}$. V describes values of all attributes, that is $V = \bigcup_{i=1}^{n} V_i$ with Vi forming a set of values of the i-th attribute. "f" is called a decision function and reads as follows:

$$f : X \times Q \rightarrow V \quad (2.34)$$

We have $f(x, q_i) \in V_i$ where q_i denotes an i-th attribute in Q and x is confined to X. Any pair (q, v), is called a description of the information. An equivalence relation (indiscernibility) in X brings together all objects that are not distinguishable in the selected subset of the attributes A,

$$IND(A) = \{(x, y) \in X \times X \mid f(x, a) = f(y, a), for \forall a \in A\} \quad (2.35)$$

As the objects which satisfy the above relation cannot be distinguished one from another, we can form a notion of an equivalence class $[x]_A$ induced with respect to A

$$[x]_A = \{y \in X \mid (x, y) \in IND(A)\} \quad (2.36)$$

An ordered pair AS = (X, $IND(A)$) is called an approximation space.

Lower approximation of X in AS is defined as:

$$X_- = \{x \in X \mid [x]_A \subseteq X\} \quad (2.37)$$

Upper approximation of X in AS is defined as

$$X_+ = \{x \in X \mid [x]_A \cap X \neq \phi\} \quad (2.38)$$

A rough set is a construct represented by means of two approximations, $\langle X_-, X_+ \rangle$. We would like to stress that rough sets are defined vis-à-vis a certain approximation space.

The selection of the approximation space implies the definition of the rough set. In other words, the same concept X may lead to different descriptions depending upon the assumed approximation space AS.

2.4.4 Shadowed sets and probabilistic sets

The underlying motivation behind shadowed sets [138] is the one about the localization of uncertainty of the member ship grades and its "centralized distribution across a fuzzy set. Formally, a shadowed set A defined in X is a granular construct realizing the mapping

$$A: X \rightarrow \{0, [0, 1], 1\} \quad (2.39)$$

$A(x) = 0$ states that x is excluded from the concept A, $A(x) = 1$ expresses that x fully belongs to A. $A(x) = [0,1]$ quantifies a situation when nothing is known about the membership of the element (x) to A. The set of arguments of X where this property holds is call a shadow of A.

In 1981, Hirota [94] proposed probabilistic sets in which for a given element of the universe of discourse X the grades of membership there are governed by some probability function, denoted as $p_x(u)$, $u \in [0, 1]$. This model captures the non-uniqueness of the membership grades and attaches to each of them some useful probabilistic characteristics.

2.4.5 Recent developments in granular computing

A recent development is that Ronald R. Yager [95] introduces a new aggregation technique based on the ordered weighted averaging (OWA) operators. An OWA operator of dimension n is a mapping $F:\mathbf{R}^n \rightarrow \mathbf{R}$, that has an associated n vector $w = (w_1, w_2, \ldots, w_n)^T$ such as $w_i \in [0,1]$, $1 \leq i \leq n$, and

$$w_1 + \cdots + w_n = 1. \tag{2.40}$$

Furthermore

$$F(a_1, \ldots, a_n) = w_1 b_1 + \cdots + w_n b_n \tag{2.41}$$

where bj is the *j-th* largest element of the bag $< a_1, \ldots, a_n >$.

The OWA operator orders the set A and applies a weighted average to the ordering of the elements, as opposed to the elements themselves. Obviously, this is a nonlinear function. In addition, the maximum, minimum, mean and median functions can all be implemented by an appropriate selection of weights. An important refinement of this idea is the Induced Ordered Weighted Averaging (IOWA) operator [96]. The IOWA operator aggregates ordered pairs (a, u), where a is a numeric value to be averaged, and $u \in U$ is an ordering value. The pairs are ordered by the value of u, and then the ordered weighted average of the values a is taken. The importance of this refinement is the U need only be an ordinal universe of discourse. Thus, values of a linguistic variable defined over U are perfectly valid candidates for IOWA ordering values.

Another research direction in granular computing is the type-2 fuzzy sets. The knowledge used to construct a fuzzy logic system is often uncertain. The uncertainties may arise form the following sources: 1) the words used in the antecedents and the consequents of the rules can mean different things to different people, 2) consequents obtained by polling a group of experts may differ, 3) the training data are noisy, and 4) the measurements that activate the FLS are noisy [97-99]. Zadeh introduced type-2 fuzzy sets [19] to deal with these uncertainties. Formally, a type-2 fuzzy set A is an ordered pair

$(x, \mu(x))$, in which x is a real number, and $\mu(x)$ is a fuzzy set representing the membership of x in A. [100] elaborated the use of type-2 fuzzy sets. In a recent work [101], Wu *et al.* designed a new method, based on the bound sets, which could relieve the computation burden of an interval type-2 fuzzy logic system during its operation. Furthermore, a type-3 fuzzy set would be a fuzzy set whose membership grads are type-2 fuzzy sets, and so on. Type-n fuzzy sets are a way of capturing types of uncertainty that, while falling under the general category of fuzziness, are not represented well by type-1 fuzzy sets.

Many approaches have been developed to convert numeric data into information granules. In [39], Pedrycz elaborates granulation with fuzzy sets and fuzzy clustering. In a recent paper [42], Bargiela *et al.* propose recursive information granulation. Bortolan *et al.* propose fuzzy descriptive models, an interactive framework of information granulation [41]. [40] introduces a hybrid two-phase approach that starts from a rough specification of the support of the fuzzy sets that is followed by detailed computation involving a specific type of membership function and an estimation of its parameters. In 1993, Sugeno and Yasukawa (SY) propose qualitative fuzzy modeling [43], which creates a linguistically interpretable fuzzy rule based model from input-output sample data. Tikk *et al.* [44] improve the SY modeling by proposing algorithms for trapezoid approximation, for the determination of the number of rules. In a recent work, Pedrycz [102, 2] proposes a new scheme of building information granules in the collaborative fashion to capture both the relational and directional aspects of information granulation. Auephanwiriyakul *et al.* [103] develope a linguistic fuzzy c-means that works with vectors of fuzzy numbers. It employs distance calculations, a fuzzy membership update equation and fuzzy center vector update equations that are extensions of Euclidean distance and the standard FCM membership and center update equations via the extension principle. As a generalization of classical automata, Ying [104] introduces a new kind of fuzzy automata whose inputs are instead strings of fuzzy subsets of the input alphabet. Finding the reduct is the core theme in rough set theory, Yin [134] proposes a granular data model to compute a reduct; each granule is represented by a bit string. Ling *et al.* [135] extend the quotient space model to the fuzzy granular world, the result provide a powerful mathematical model and tool for granule computing. In [136], Q. Liu constructs a granular deductive reasoning system, in which an object is presented by an ordered pair: an assertion and a semantic set corresponding to the assertion. Temporal and spatio-temporal aggregations are costly operations for maintaining time-evolving data [141], Zhang *et al.* propose that aggregates can be maintained using multiple levels of temporal granularities: older data using coarser granularities while more recent data using finer.

In [139], Zadeh gives a summary for using Granular Computing as a basis for a computational theory of perception.

2.5 Fuzzy mathematics

2.5.1 Basic concepts

The Mathematics based on fuzzy set theory is a very broad topic, subsuming arithmetic operations on fuzzy numbers, possibility theory, fuzzy relations, fuzzy logic, and many others. One classic unifying work on this topic is a textbook by Kandel [75]. Our discussion in this section will be limited to arithmetic operations on fuzzy numbers and linguistic terms. A fuzzy number is a fuzzy set that represents a number that is not precisely known. Given this basic definition, there is a considerable amount of work that has been done in defining arithmetic operations whose operands are fuzzy numbers. Also, the traditional numeric operators are useless in this context. A statement such as "hot + cold = warm" makes perfect linguistic sense, but if the "+" operator is the numeric sum, then the statement becomes an absurdity. Hence, we require new definitions for operators on fuzzy numbers.

There are two methods in general use for defining arithmetic operators on fuzzy numbers. The first and best-known is the extension principle, which "fuzzifies" an operation, shifting its domain and range from the set of real numbers to the set of fuzzy numbers. The second method of defining fuzzy arithmetic operators is restricted to a class of fuzzy numbers called LR fuzzy numbers.

Definition 2.16

An LR fuzzy number is a fuzzy number whose membership function is of the form

$$A(x) = \begin{cases} L[(a-x)/\alpha] & (a-\alpha) \le x \le a \\ R[(x-a)/\beta] & a \le x \le (a+\beta) \\ 0 & \textit{otherwise} \end{cases} \qquad (2.42)$$

where a is the unique value for which $A(a)=1$, $\alpha,\beta > 0$ are respectively the left and right spread of the fuzzy number, and L, R are shape functions for which $L(0) = R(0) = 1$ and $L(1) = R(1) = 0$.

Fuzzy arithmetic operators on these kinds of fuzzy numbers can be defined using the α- cut of $A(x)$. Since α- cuts are necessarily closed intervals on the real line, the definitions of classic interval analysis can be used directly. Indeed, this method is far more computationally efficient than using the extension principle, which generally reduces to solving a nonlinear programming problem [76, 77]. Dubois and Prade have produced a large body of work on LR fuzzy numbers [78-82].

One important property of arithmetic operations is called closure. We say that the set of real numbers is closed under an operation $f(x)$ ($x = (x_1, x_2, \ldots, x_n)$) if, for any $x \in \mathbf{R}^n$, $f(x) \in \mathbf{R}$. This simply means that a function of n real values will evaluate to a real number. In the context of fuzzy numbers, we wish to know that given an operator $\circ$, if $A \circ B$ is closed on the set of fuzzy numbers. In [24], Zimmermann answers this question in the affirmative; any increasing or decreasing function of two fuzzy numbers will take its values exclusively from the set of fuzzy numbers. This means that the operations defined by either the extension principle or the α-cut method will be closed on the set of fuzzy numbers [77].

A related question is: given a fuzzy mathematical operator, is a finite set of fuzzy numbers closed under this operator? This question arises from the nature of linguistic variables: the term set T of the linguistic variable is finite, and thus the semantic rule M associates the elements of T with a finite set of fuzzy numbers. In general, fuzzy mathematical operators defined by the methods discussed are not closed for a finite set of fuzzy numbers [83]. This relates to the problem of description pointed out by Zadeh in [60]. Description is essentially the inverse of the semantic rule of a LV. Given a fuzzy number F in a universe of discourse U and a LV defined on U, what term $t \in T$ of the LV best represents F? The difficulty with this problem is that F can be any convex, normal fuzzy set on U. The mapping M^{-1}: $\mathcal{T} \rightarrow T$ where $\mathcal{T}$ is the set of fuzzy numbers on U, does not normally exist. When using fuzzy mathematical operators, M^{-1} must normally be computed as an approximation. Further criticisms of the standard fuzzy mathematical operators may be found in [77] and [84].

A paper by Delgado *et al.* [85] resolves this difficulty. In this paper, the authors introduce the idea of changing the semantic rule of a linguistic variable following a mathematical operation on that LV. In other words, while the term set of a LV does not change under an n arithmetic operation, the fuzzy sets associated with each term do change. Using this idea, Delgado *et al.* [85] provided definitions for addition, subtraction and scalar multiplication, all of which are closed on a term set. This method is also call label arithmetic.

2.5.2 Fuzzy set distance measures

The linguistic gradient operator relies heavily on the idea of a distance metric.

Definition 2.17

A distance metric is a function δ on a universe of discourse U, which must satisfy 3 properties [83]:

1) $\delta : U^2 \rightarrow R^+$ (2.43)

2) $\delta(x_1, x_1) = 0$ (2.44)

3) $\delta(x_1, x_2) < \delta(x_3, x_4)$, if $x_3 < x_1 < x_2 < x_4$ (2.45)

where x_i ($i = 1, 2, 3, 4$) is a fuzzy set.

The first property requires the distance has to be a non-negative real number; the second property means that the distance from a fuzzy number to itself is 0; the third requires the distance of two fuzzy numbers is greater than the distance of any other two numbers between them.

[83] pointed to the Minkowski family of distance metrics (includes the Euclidean distance as a special case) as an excellent group of candidates for δ. Numerous other definitions for fuzzy distance metrics have also been proposed. One such definition is based on the Hausdorf metric [86]. This definition is based on interval arithmetic, and measures the sum of the distances between the corresponding α-cuts of the two fuzzy sets:

$$\delta(u,v) = \int_0^1 \max(|u_l(\alpha) - v_l(\alpha)|, |u_u(\alpha) - v_u(\alpha)|)d\alpha \tag{2.46}$$

where u and v are fuzzy sets, $u(\alpha)$ is the closed interval formed by an α-cuts of u, and $u_l(\alpha)$ and $u_u(\alpha)$ are the lower and upper bounds of the interval $u(\alpha)$, respectively. A more general form of the distance between two intervals is used in definition 2 of [87]. The distance formula itself does not change, but the underlying measure of the distance between two closed intervals includes the Hausdorf metric as a special case. Two other distance measures are also proposed in this paper. One uses the difference between the expected values of two fuzzy numbers. The other is specific to trapezoidal fuzzy numbers, and uses geometrical principles to generate a distance metric [87].

2.5.3 Linguistic space

In modern control theory, a system is represented as a set of first-order differential equations on the state variables of the system. This state space approach is often less mathematically complex, and at the same time enables the designer to analyze key system characteristics such as stability, controllability and observability [88]. A state space is a space defined by the cross product of the state variables of some system [19]. In crisp control systems, all n state variables take real or integer values, so the space is normally $\mathbf{R}^n$ or $\mathbf{Z}^n$. However, in fuzzy systems, linguistic variables are used. So the state space of a fuzzy system is the cross product of the linguistic variables involved in the system. In [89], a discussion of the 2-dimensional case is presented. This book summarizes the idea of a linguistic trajectory, which is an analogue of the system trajectory in crisp systems. The book also introduces the idea of a partition of the underlying universe of discourse induced by a linguistic variable. However, the n-dimensional case is not covered, as the graphical techniques used do not extend well to higher-dimensional spaces. C. Liu *et al.* [90] defines a linguistic state space, and formalizes the ordering of linguistic terms on each axis of the space by using a standard vector. This standard vector contains all atomic terms of a linguistic variable, which are arranged in a "reasonable" order. For an n-dimensional state space, this means that there will be n standard vectors, each associated with one dimension of the space. The standard vector provides an ordering of the linguistic terms to define other operations on the linguistic terms.

2.6 Linguistic arithmetic and linguistic gradient operator

2.6.1 Linguistic arithmetic

Dick proposed a set of linguistic arithmetic operators [35] and linguistic gradient operator [1]. These papers introduce a distance metric and a difference vector on a term set, a generalized linguistic variable, linguistic arithmetic operators, linguistic gradient operator, and linguistic neural network. The linguistic gradient operator is the foundation of our research. Therefore, I would like to discuss them in details here.

The following assumptions were set forth in [1]. These assumptions are also held for the remaining chapters, except otherwise specified:

1. All rulebases of the fuzzy systems in question are complete and consistent
2. The term sets of all n linguistic variables contain only atomic terms.
3. The semantic rule of each linguistic variable associates each term of the linguistic variable with a LR fuzzy number.
4. When ordered in a reasonable manner, the term set is symmetric about some "zero" term.
5. The semantic rule of each linguistic variable induces a uniform partition on the respective universe of discourse.
6. The rules of the fuzzy system are to be of the form below:

   If x_1 is A_1 and x_2 is A_2 and … and x_n is A_n Then y_1 is B_1 and y_2 is B_2 and … and y_m is B_m.

   where x_i is the i-th input variable defined on a universe of discourse U_i, A_i is a fuzzy set on U_i, y_j is the j-th input variable defined on a universe of discourse V_j, B_j is a fuzzy set on V_j.

As an analogue of the traditional state-space representations for crisp system, a linguistic state space is a mathematical representation of an n-input, single-output fuzzy system. The linguistic state space is defined by the input variables of the system, and the system output is a function in this space. It is a discrete, finite space, the coordinates of which are linguistic values.

Definition 2.18

The linguistic state space of a fuzzy system is a space $U_i \times \ldots \times U_i \times \ldots \times U_n$, in which the i-th dimension of the space takes its possible values from the term set of the ith linguistic variable of the fuzzy system, and U_i is the universe of discourse for the i-th LV of the fuzzy system in a closed interval form $[-a, a]$.

Obviously, the term sets of the linguistic variables must be ordered to make the coordinates have meaning. The "reasonable" ordering of terms is defined to be the same as the ordering of the fuzzy numbers associated with each term by the semantic rule M. Fuzzy sets are ordered by their families of α-cuts as follows:

Definition 2.19

For two fuzzy sets A and B generated by the semantic rule M of a linguistic variable L, $A < B$ iff $\forall \alpha \in (0,1], \inf({}^{\alpha}A) < \inf({}^{\alpha}B)$ and $\sup({}^{\alpha}A) < \sup({}^{\alpha}B)$, where ${}^{\alpha}A$ is the α-cut of fuzzy set A, inf denotes the infimum, and sup denotes the supremum.

Since we assume that every fuzzy set in the system is an LR fuzzy number, every α-cut necessarily forms a closed interval on the universe of discourse. Furthermore, this ordering of the fuzzy sets meets the criteria of "reasonableness" set forth in [90]. Because the rulebase is complete and consistent, it is a right-unique mapping from the rule antecedents to the rule consequents, i.e., it is a function in the linguistic state space.

As classic gradient, the linguistic gradient evaluates the change rate of a function at a point in linguistic state space. Hence, before defined the linguistic gradient operator, a definition of difference vector for the term set of a linguistic variable (an axis of the linguistic space), based on a distance metric [91] for that term set, is required. The difference between two terms is defined by counting transitions:

Definition 2.20

The distance λ between two terms in a standard vector W equals the number of terms appearing between the source and goal terms plus 1. λ is 0 if the source and goal are the same term.

Based on the distance metric, the difference D between two terms of a term set is defined as a one-dimensional vector:

Definition 2.21

The difference D between two terms, t_1 and t_2, in a standard vector W is a one-dimension vector,

$$D(t_1,t_2)=\begin{cases}\lambda(t_1,t_2) & \text{if } t_1 < t_2\\ 0 & \text{if } t_1 = t_2\\ -\lambda(t_1,t_2) & \text{if } t_1 > t_2\end{cases} \qquad (2.47)$$

To express the difference in linguistic form, the linguistic difference is defined as

Definition 2.22

The linguistic difference δ between two terms in a standard vector W is

$$\delta = \begin{cases} NL & \text{if } D <= -3 \\ NM & \text{if } D = -2 \\ NS & \text{if } D = -1 \\ AZ & \text{if } D = 0 \\ PS & \text{if } D = 1 \\ PM & \text{if } D = 2 \\ PZ & \text{if } D >= 3 \end{cases} \tag{2.48}$$

where W_δ = (NL, NM, NS, AZ, PS, PM, PL) is the standard vector of δ, and *D* is the numeric difference. Although these atomic terms can be named arbitrarily, as widely used in literature, they are named NL, NM, NS, AZ, PS, PM, and PL, which stand for negative large, negative middle, negative small, zero, positive small, positive middle, and positive large, respectively. The semantic rule of δ is given in Fig. 2.3

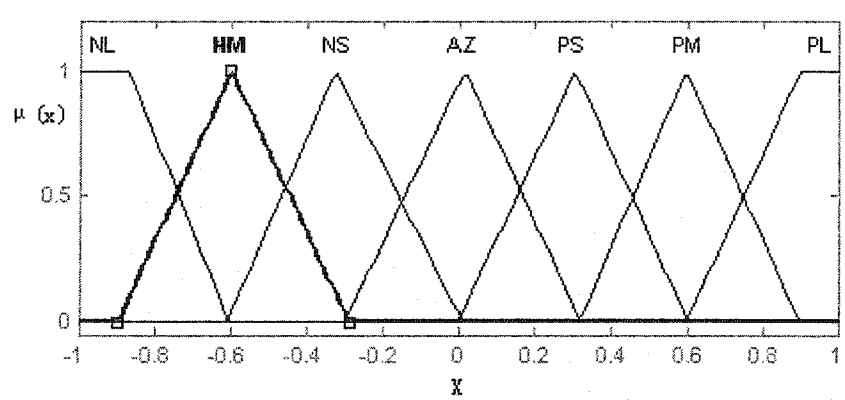


Figure 2.3 Semantic rule of δ

This definition only works for a standard vector with exactly 7 entries. Later, it will be extended to include term sets of arbitrary size so long as those term sets obey the assumptions.

In section 2.2.3, the syntactic rule of a linguistic variable is a context free grammar, associating a fuzzy set with an atomic term. This means that any adaptation must be undertaken by altering the semantic rules of each linguistic variable via numeric procedure.

The definition of generalized linguistic variable replaces the context free grammar with a phrase-structure grammar to generate new terms. New terms are generated by adding hedges and applying crossover operator.

2.6.2 Linguistic gradient operator

The linguistic gradient operator acts in the linguistic rule space formed by a fuzzy rulebase. As the gradient operator in calculus, the linguistic gradient operator determines the direction of the greatest local increase in the consequent function C, so that it makes possible the extraction of additional information about a rule base, which may help determine various characteristics of the rulebase. This additional information is obtained without performing any numerical computation, thus follows the "computing with words" paradigm.

The linguistic gradient was inspired by gradient in digital image processing, thus their computations are also very similar. The gradient operator in digital image processing is based on some difference operators. Computing the gradients for an image is convolution over the image.

Analogue to defining image gradient operator, some sort of difference operator on term set are require, thus the difference ratio and the term set difference vector are defined.

Definition 2.23

Given a linguistic variable L with term set T, and two terms t_1, $t_2 \in T$, the difference ratio D_r is defined as

$$D_r(t_1, t_2) = \frac{D(t_1, t_2)}{|T|} \qquad (2.49)$$

where D is the numeric difference and $|T|$ is the cardinality of T.

Definition 2.24

Given an ordered term set T, two terms of interest – source and goal – and the relation "the difference between source and goal", the term set difference vector Δ is defined as:

$$\Delta(source, goal) = \begin{cases} NL, D_r \leq -5/14 \\ NM, -5/14 < D_r \leq -3/14 \\ NS, -3/14 < D_r < 0 \\ AZ, D_r = 0 \\ PS, 0 < D_r < 3/14 \\ PM, 3/14 \leq D_r < 5/14 \\ PL, D_r \geq 5/14 \end{cases} \qquad (2.50)$$

The standard vector Δ is W_Δ= (NL, NM, NS, AZ, PS, PM, PL), with AZ the zero term. Definition 2.24 maps the numeric difference into a standard linguistic variable, thereby insuring that differences for different term sets are comparable quantities, and returning linguistic quantities. The ranges used in equ. 2.50 are based on the similarity of partition intervals in different term sets that meet the assumptions set forth in section 2.6.1. Justification can be found in [1]. Semantic rule of Δ is shown in Fig. 2.4.

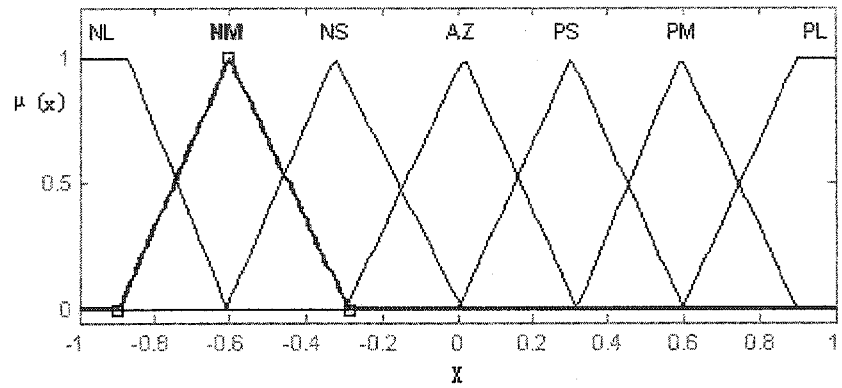


Figure 2.4 Semantic rule for Δ

The linguistic gradient is a mapping from the universe of discourse of consequence to the n-dimensional rule space. The result of the gradient operator is an n-vector of linguistic difference values.

Definition 2.25

The linguistic gradient is a mapping

$$V \Rightarrow \Delta_1 \times \Delta_2 \times ... \times \Delta_n \tag{2.51}$$

where V is the universe of discourse of $C(x)$, and Δ_i is the term set difference vector defined for the i-th dimension of the n-dimensional rule space. The mapping is defined as

$$\Delta C(\vec{x}) = (\frac{\partial y}{\partial x_1},...,\frac{\partial y}{\partial x_n}) \tag{2.52}$$

where

$$\begin{aligned}\frac{\partial y}{\partial x_i} &= \Delta\left[C(\vec{x}_l), C(x_{l1}, x_{l2},..., x_{l(i-1)}, x_{(l-1)i}, x_{l(i+1)},..., x_{\ln})\right] \\ &+ \Delta\left[C(x_{l1}, x_{l2},..., x_{l(i-1)}, x_{(l+1)i}, x_{l(i+1)},..., x_{\ln}), C(\vec{x}_l)\right]\end{aligned} \tag{2.53}$$

where “+” is defined in Table 2.2, l denotes an index on the standard vector for each dimension, $i = (1, 2, \ldots, n)$. The subscript l is used to identify the neighbours of the point x in the i-th dimension of the rule space.

Table 2.2 LABEL ADDITION

	NL	NM	NS	AZ	PS	PM	PL
NL	NL	NL	NM	NM	NS	NS	AZ
NM	NL	NM	NM	NS	NS	AZ	PS
NS	NM	NM	NS	NS	AZ	PS	PS
AZ	NM	NS	NS	AZ	PS	PS	PM
PS	NS	NS	AZ	PS	PS	PM	PM
PM	NS	AZ	PS	PS	PM	PM	PL
PL	AZ	PS	PS	PM	PM	PL	PL

These difference values do not generally belong to the term sets of the linguistic variable underlying each dimension of the rule space. They belong to a single term set, and are therefore directly comparable. A definition of the direction of the linguistic gradient is: the gradient vector is the sum of n vector V_i, where V_i lies on the i-th axis of the rule space, and the magnitude of V_i is the i-component of the linguistic gradient vector. The linguistic gradient will point in the direction of the maximum rate of change in $C(x)$. This yields new information concerning the structure of a fuzzy rulebase.

Because the linguistic gradient operator acts at a linguistic level, we are able to avoid using numerical technique.

Example 1 in Chapter III section 3.2 demonstrates how to compute the linguistic gradients.

2.7 Introduction to Artificial Neural Network and Genetic Algorithm

ANN and GA are two important paradigms in SC. We introduce basic concepts of ANN and GA in this section.

2.7.1 Neural Network

The field of Neural Networks has a history over 60 years. McClulloch and Pitts in 1943 [105] developed a basic threshold neuron. Since then, many researchers have published a large number of works on neural networks. Hebb developed a learning algorithm for a neural network in 1949, and Rosenblatt designed the perceptron in 1958 [106, 107]. In 1986, Rumelhart *et al.* proposed the backpropagation algorithm [108], which allows multilayer neural networks to learn from experience. Multilayer perceptron has become the most popular neural network architecture in use today [109].

Inspired by biological nervous systems, many researchers have been exploring artificial neural networks, which model the brain as a continuous-time nonlinear dynamic system in connectionist architectures that are expected to mimic brain mechanisms to simulate intelligent behavior.

A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity storing experiential knowledge and making it available for use. Thereby, it endows machines with some of the cognitive abilities that biological organisms possess.

ANNs store information among the synaptic connections. Here a neuron is an elementary processor that performs primitive types of operations, like summing the weighted inputs coming to it and then amplifying or thresholding the sum. These networks can be trained by examples and sometimes generalize well for unknown test cases. Performance is improved over time by iteratively updating the weights in the network. Neural networks are naturally parallel computing devices. Various models are designated by the network topology, node characteristics, and the status updating rules.

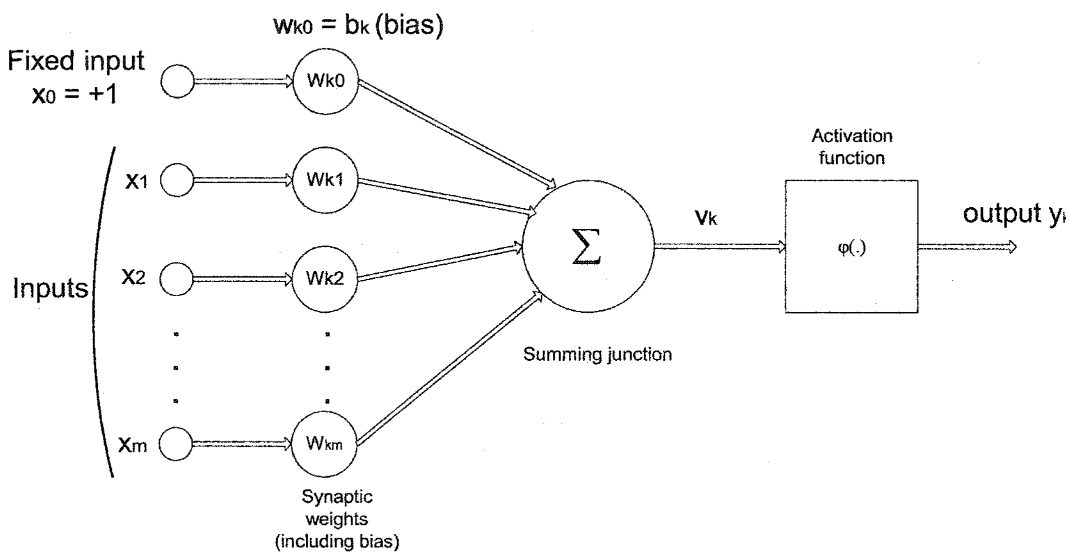


Figure 2.5 Nonlinear model of a neuron

A neuron is an information-processing unit that is fundamental to the operation of a neural network. Fig. 2.5 shows the model of a neuron, which consists of three basic elements [109]:

- A set of synapses, each of which is characterized by a weight of its own. A signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight wkj.
- An adder for summing the products of the input signals and the respective synapse weights.
- An activation function for limiting the amplitude of the output of a neuron. Typically, the normalized amplitude range of the output of a neuron is [0, 1], or alternatively [-1, 1].

Such a neuron is referred as the McCulloch-Pitts model. In this model, the output of a neuron takes on the value of 1 if the induced local field of that neuron is nonnegative and 0 otherwise. This is called the all-or-none property of the McCulloch-Pitts model. A simple generalization of McCulloch-Pitts neuron, by replacing the threshold function with a more general nonlinear function, enhances the power of the networks built from such neurons. A synchronous assembly of such neurons is capable of universal computation for suitably chosen weights. Such an assembly can perform any computation that an ordinary digital computer can.

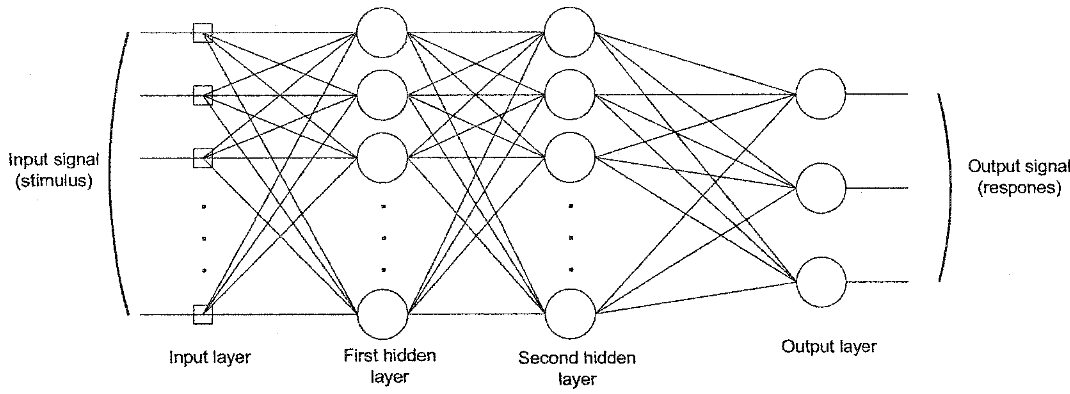


Figure 2.6 Architecture graph of a multilayer perceptron with two hidden layers

Typical multilayer perceptron (MLP) consists of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. Fig. 2.6 depicts the architecture of a MLP.

Each hidden or output neuron of a multilayer perceptron is designed to perform two computations: 1. The computation of the function signal appearing at the output of a neuron, which is expressed as a continuous nonlinear function of the input signal and synaptic weights associated with that neuron; 2. The computation of an estimate of the gradient vector, which is needed for the backward pass through the network.

The Backpropagation Algorithm consists of a forward pass, which maps input to output, and a backward pass, which updates the weights of each node according to the difference between actual output and expected output.

The MLP has been proven to be universal approximator [109]. It can approximate any nonlinear mapping after training. Tasks that MLP can perform include pattern classification, clustering or categorization, function approximation, prediction or forecasting, optimization, retrieval by content, and control.

The back-propagation algorithm for the design of a MLP may be viewed as the application of a recursive technique known in statistics as stochastic approximation [109]. There are various ways for the design of a neural network. Radial basis function network designed as a curve-fitting problem in a high-dimensional space [110]. Hopfield network [111] has a deterministic behavior. It is described by a set of nonlinear differential equations that define the exact evolution of the model as a function of time.

2.7.2 Genetic Algorithm

Most symbolic AI systems are very static. Most of them can usually only solve one given specific problem, since their architecture was designed for whatever that specific problem was in the first place. Thus, if the given problem were somehow to be changed, these systems could have a hard time adapting to them, since the algorithm that would originally arrive to the solution may be either incorrect or less efficient. In 1975, GA was created by J. Holland [34] to combat these problems. GA is based on natural biological evolution. The architecture of systems that implement GA is more able to adapt to a wide range of problems. A GA functions by generating a large set of possible solutions to a given problem. It then evaluates each of those solutions, and decides on a "fitness level" for each solution set. These solutions then breed new solutions. The parent solutions that were more "fit" are more likely to reproduce, while those that were less "fit" are less likely to do so. In essence, solutions are evolved over time. This way we evolve our search space scope to a point where you can find the solution. GA can be incredibly efficient if programmed correctly.

GAs are convenient to program or understand, since they are biological based. Here is the general algorithm for a GA:

1. Create a Random Initial State. An initial population is created from a random selection of solutions, which are analogous to chromosomes. This is unlike the situation for Symbolic AI systems, where the initial state in a problem is already given instead.
2. Evaluate Fitness. A value for fitness is assigned to each solution (chromosome) depending on how close it actually is to solving the problem. These “solutions” are not “answers” to the problem. They are possible characteristics that the system would employ in order to reach the answer.
3. Reproduce and Children Mutate. Those chromosomes with a higher fitness value are more likely to reproduce offspring, which can mutate after reproduction. The offspring is a product of the father and mother, whose composition consists of a combination of genes from them. This process is known as "crossing over".
4. Next Generation. If the new generation contains a solution that produces an output that is close enough or equal to the desired answer then the problem has been solved. If this is not the case, then the new generation will go through the same process as their parents did. This will continue until a solution is reached.

It's important to understand that the functioning of such an algorithm does not guarantee success. We are in a stochastic system and a genetic pool may be too far from the solution. For example, a too fast convergence may halt the process of evolution.

These algorithms are nevertheless extremely effective, and are used in fields as diverse as stock exchange, production scheduling or programming of assembly robots in the automotive industry.

2.7.3 Hybrid Systems

An active trend is various forms of hybrid system of fuzzy logic, neural networks and genetic algorithms. Neuro-fuzzy computing [113, 114, 116-119], which is a judicious integration of the merits of neural and fuzzy approaches, enables one to build more intelligent decision-making systems. This incorporates the generic advantages of artificial neural networks like massive parallelism, robustness, and learning in data-rich environments into the system. The modeling of imprecise and qualitative knowledge as well as the transmission of uncertainty is made possible through the use of fuzzy logic.

ANFIS by Jang [115] implements a Sugeno-like fuzzy system [52] in a five-layer network structure. Backpropagation is used to learn the antecedent membership functions, while least mean squares algorithm determines the coefficients of the linear combinations in the consequent of the rule. The linguistic neural network (LNN) [35] replaces the numeric connection weights of a standard neural network with linguistic values, thus promotes an understanding of the network at the level of individual nodes. The induced local field, activation function, weights updating algorithm is also similar to counterpart in MLP, only replace the numeric variables with linguistic variables, and use the linguistic mathematics operators instead of numeric operators [1]. Hayashi et al. [120] fuzzified the delta rule for multilayer perceptron (MLP) using fuzzy numbers at the input, output, and weight levels. But there were problems with the stopping rule. Mitra and Pal have used the fuzzy LP for inferencing and rule generation [121].

Applications of genetic algorithms combined with fuzzy control are being investigated not only at the academic level but also at the commercial level [21]. Genetic algorithms are particularly suited for tuning the membership functions in terms of placing them in the universe of discourse. Properly configured GA/Fuzzy architectures search the complete universe of discourse and find adequate solutions according to the fitness function [112]

Ishibuchi et al. [122] select a small number of significant fuzzy IF–THEN rules to construct a compact and efficient fuzzy classification system. GAs are used to solve this combinatorial optimization problem, with an objective function for simultaneously maximizing the number of correctly classified patterns and minimizing the number of fuzzy rules. Wang and Yen [123] have designed a hybrid algorithm that uses GAs for extracting important fuzzy rules from a given rulebase to construct a parsimonious fuzzy model with high generalization ability. The parameters of the model are estimated using the Kalman filter.

Chapter 3 FUZZY RULEBASE SIMILARITY CONFIDENCE LEVEL

This chapter presents the major contribution of this thesis, the Similarity Confidence Level (SCL) for fuzzy rulebases. The SCL is a novel similarity measure for comparing two fuzzy rulebases based on linguistic gradients [1], and is inspired by a content-based image retrieval algorithm [36].

Manipulating fuzzy rulebases is similar to digital image processing in many ways. First, both rulebases and images are presented in discrete finite spaces, which have well defined borders. Second, their outputs (for rulebases, the output is the consequent; for grey image, the output is the grey value) are also discrete interval scales. There are also some differences between them. One is that, as our assumptions in Chapter II held, the consequent variable may take negative values, but the grey value of an image can only take non-negative values. Furthermore, a fuzzy rulebase may have an arbitrary number of inputs, while images have limited dimensions. Overall, image processing algorithms can provide inspiration for defining our fuzzy rulebase similarity operator.

Our objective is to measure the structural similarity between fuzzy rulebases within the computing with words (CW) paradigm, i.e. our algorithm will only compare linguistic outputs, avoiding comparison over the defuzzified reasoning surface. The question is how we can obtain structural information from rulebases and how to compare this structural information. To solve this problem, we direct our attention to a sub-area of image processing called content-based image retrieval (CBIR). One major task of CBIR is to recognize contents in multimedia data which relates to recognition of semantics of the contents of media data [124]. For this purpose, CBIR has found mechanisms to extract some sort of structure invariants, and developed many similarity measures based on these invariants. Our fuzzy rulebases similarity operator is inspired by a CBIR algorithm proposed by J. Rose, *et al.* [36], in which the similarity of two image is computed in 5 steps: First, the gradients are computed at each point of the two images to be compared; Second, threshold these gradients, images are converted into binary map; Third, project binary numbers to horizontal, vertical, and diagonal axes; Fourth, compute the Euclidean distance between projection vectors; Finally, normalize the result. Our algorithm is very similar to it, however, we made some changes according to the characteristics of fuzzy rulebases.

3.1 Definition of SCL

The SCL for two fuzzy rulebases is a mapping from the universes of discourse of the two fuzzy rulebases' consequents to the interval [0, 1]:

$$SCL : V_1 \times V_2 \Rightarrow [0,1] \tag{3.1}$$

where V_1 and V_2 are the universe of discourse of consequent LVs. The calculation of SCL consists of 6 major phases: computing the gradient maps of the two rulebases, thresholding the gradient maps, projecting the thresholded vectors to each axis, regranulating the projection vectors, calculating the Euclidean distance, and normalizing the result.

Suppose that we have two n-dimensional fuzzy rulebases, F_1 and F_2, of which corresponding universes of discourse are identical. The SCL(F_1, F_2) is calculated as follows:

1) Compute the gradient vector maps of F_1 and F_2

We denote the cardinality of the *i*-th dimension in rulespace as c_i. Plainly, the number of points in rulespace (equivalently, the number of rules in the rulebase) is given by $c_1 \cdot c_2 \cdot \ldots \cdot c_n$, where c_i is the cardinality of the *i*-th dimension. For each point in rulespace, the gradient vector $\nabla C(\bar{x})$ is calculated using the algorithm described in Chapter II Section 2.5.2. Note that $\nabla C(\bar{x})$ is the vector sum of n vectors $v_i(\bar{x})$, where $v_i(\bar{x})$ lies on the *i*-th axis of the rule space.

2) Threshold the gradient vectors

The thresholded gradient fields will be coarser granulations of the gradient vector field; each component may take the values –1, 0, or 1. This simplification of $\nabla C(\bar{x})$ allows us to focus on the most important differences between two rulebases, while suppressing minor ones. This also retains the basic directional information contained in each $v_i(\bar{x})$; experimentation indicates that discarding the direction of the gradient vector at this point is undesirable.

A fuzzy inference system may have different linguistic variables for different universes of discourse. For instance, one LV might use 6 membership functions, while another might use 3 MFs. We have dealt with these differing granulations by restricting the linguistic gradient to a single, common term set. However, the magnitude of the linguistic gradient will still tend to be greater for coarser granulations, all things being equal. Hence, we adopt a thresholding rule for linguistic gradients that still takes different values of c_i into account. We set the linguistic threshold for each dimension in a rulebase according to c_i.

For all i and $\vec{x}$, threshold $v_i(\vec{x})$,

$$v_i(\vec{x}) = \begin{cases} 1, if \ |v_i(\vec{x})| \geq threshold_i, v_i(\vec{x}) > AZ \\ 0, if \ |v_i(\vec{x})| < threshold_i \\ -1, if \ |v_i(\vec{x})| \geq threshold_i, v_i(\vec{x}) < AZ \end{cases} \quad (3.2)$$

where $threshold_i$ is the threshold for the i-th dimension given by

$$threshold_i = \begin{cases} PS, & \text{if } c_i > 4 \\ PM, & \text{if } c_i = 3, 4 \\ PL, & \text{if } c_i = 2 \end{cases} \quad (3.3)$$

where c_i is the cardinality of the output linguistic variable. The conditions in equ.3.3 are determined by experiment. Because the possible values of c_i are 2-9, we use all possible combinations of these values for the conditions to run the validating experiment. The combination used in equ.3.3 gives the best result.

3) Compute the projection to every axis

The i-th dimension projection $P_i = (p_{i1}, p_{i2}, \ldots, p_{ici})$ is a $2c_i$-dimension vector, where

$$p_{ij} = \left(\frac{c_i \sum_{allx_i = j, ST v_i(\vec{x}) = -1} v_i(\vec{x})}{\prod_{k=1}^{n} c_k}, \frac{c_i \sum_{allx_i = j, ST v_i(\vec{x}) = 1} v_i(\vec{x})}{\prod_{k=1}^{n} c_k} \right), j = 1, \ldots c_i \quad (3.4)$$

where n is the number of inputs, c_i is the cardinality of the i-th dimension, $v_i(\vec{x})$ is the linguistic gradient lies in i-th axis for partition $\vec{x}$. $v_i(\vec{x})$ is used in Equ. 3.4 instead of $\nabla C(\vec{x})$, because the projection from $\nabla C(\vec{x})$ to the i-th axis depends on $v_i(\vec{x})$ solely, projections of other vectors $v_j(\vec{x}), j \neq i$ to this axis equals 0. p_{ij} is a 2-d vector, one dimension of which represents the number of -1s, the other represents the number of 1s. This vector is calculated by counting the number of -1s and 1s separately, and normalized by divided the number of points involved, which is equal the product of all input cardinalities other than the i-th.

4) Regranulate the projected vectors

Two different fuzzy systems may use different LVs for the same universe of discourse. One rulebase might granulate a particular universe of discourse into 3 membership functions, while the other might use 4. However, in order to compare the projections for these 2 rulebases, we need to have a common granularity for corresponding universes of discourse. Thus, in order to generalize our algorithm beyond requiring common granulations in the original rulebases, we will *regranulate* one of the projections to match the other. During this process, we must retain as much information as possible.

Regranulation is a term defined by Dick in [142] to describe the process of changing the granulation of some universe(s) of discourse. An example of regranulation is image resizing, in which the number of points representing the image changed, but important information is retained so that we can still recognize the image.

Hereafter, the first digit of a subscript denotes a rulebase, the second digit denotes a dimension in that rule space and the third digit denotes an index on that dimension.

Suppose that P_{1i} and P_{2i}, representing the i-th axis projection vectors of F_1 and F_2 are of cardinality c_{1i} and c_{2i}, respectively. We want to regranulate P_{2i} to $P'_{2i} = (p'_{2i1}, p'_{2i2}, \ldots, p'_{2i(c1i)})$.

Our regranulation is very similar to the first-order image resizing algorithm, in which the value in a new point is the weighted average of its nearest neighbours [128]. The intersections of membership functions of the i-th input of F_1 divide the i-th universe of discourse into c_{1i} intervals, $[a_{1i1}, b_{1i1}]$, $[a_{1i2}, b_{1i2}]$, …, and $[a_{1i(c1i)}, b_{1i(c1i)}]$, corresponding to $p_{1i1}, p_{1i2}, \ldots p_{1i(c1i)}$, respectively. Similarly, the i-th universe of discourse of F_2 is divided into c_{2i} intervals, $[a_{2i1}, b_{2i1}]$, $[a_{2i2}, b_{2i2}]$, …, and $[a_{2i(c2i)}, b_{2i(c2i)}]$, where a_{1i1} and a_{2i1} are the lower bound of the universe of discourse, $b_{1i(c1i)}$ and $b_{2i(c2i)}$ are the upper bound of the universe of discourse. We calculate the regranulated projection vectors as a weighted sum of the original projection values, based on the overlap of the numeric partitions underlying the rulespace.

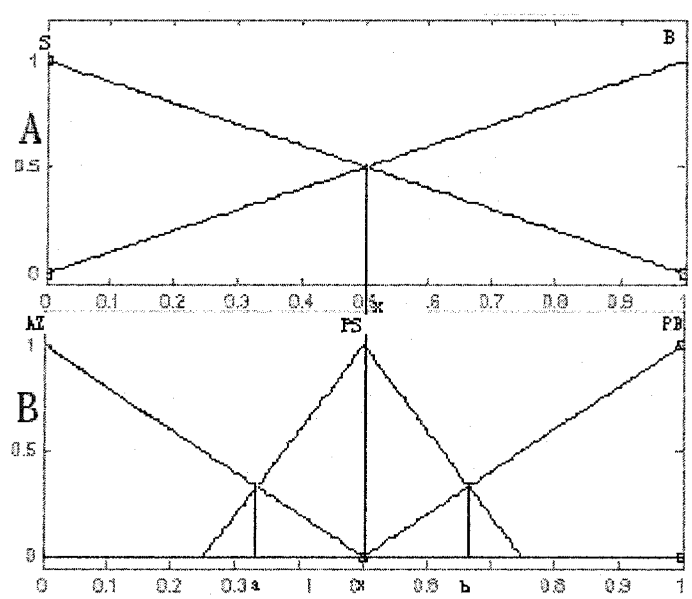


Figure 3.1 Example of regranulation

By using interval algebra, for the i-th dimension, the $P'_{2i} = (p'_{2i1}, p'_{2i2}, \ldots, p'_{2ic1i})$ is computed as follows

$$p'_{2ij} = \min\left(1, \sum_{k=1}^{c_{2i}}\left(p_{2ik} \times \frac{width([a_{2ik}, b_{2ik}] \cap [a_{1ij}, b_{1ij}])}{width([a_{2ik}, b_{2ik}])}\right)\right), j = 1,2,\ldots,c_{1i} \quad (3.5)$$

where c_{ni} is the cardinality of i-th dimension for F_n, a_{nij} and b_{nij} are the lower bound and upper bound of the j-th interval on i-th dimension for F_n, respectively.

Fig 3.1 gives an example of regranulation. Suppose that two LVs A and B are corresponding inputs of two fuzzy rulebases. A has two atomic terms Small and Big, the intersection of whose MFs divides the universe of discourse into two intervals [0, x] and [x, 1]. Similarly, the intersections of MFs of B divide the universe of disourse of B into 3 intervals [0, a], [a, b], and [b, 1]. If these intervals are drawn on the same real segment, [0, x] will contain [0, a] and overlap with [a, b], [x, 1] will contain [b, 1] and overlap with [a, b]. The length of projection vector of A is 4 (2 times its cardinality) and that of B is 6. Then we can use following equations to regranulate the projection vector of B:

$$p'_{Small} = \min\left(1, (p_{AZ} + \frac{x-a}{b-a} p_{PS})\right) \quad (3.6)$$

$$p'_{Big} = \min\left(1, (\frac{b-x}{b-a} p_{PS} + p_{PB})\right) \quad (3.7)$$

Since the original projection vectors are normalized, the regranulated value is saturated when it reaches 1. In general, regranulating F_1 or F_2 will result in different SCL values, but the difference is very small. Experimental result on this issue will be presented in Chapter IV. After the projection vectors being regranulated, the remaining steps are straightforward.

5) Calculate the Euclidean distance

Calculate the Euclidean distance using

$$dist = \sqrt{\sum_{i=1}^{n}(\sum_{j=1}^{c_i}(p_{1ij} - p'_{2ij})^2)} \tag{3.8}$$

where n is the number of inputs, c_i is the cardinality of the i-th dimension, p is the projection vector, p' is the regranulated projection vector.

6) Normalize the result

The Euclidean distance depends on the sum of the cardinalities of the input LVs for each of the rulebases in question; hence, it has to be normalized so that the result is a meaningful representation of similarity. The Similarity Confidence Level between two fuzzy rulebases is defined as:

$$SCL = \frac{Max - dist}{Max} \tag{3.9}$$

where *Max* is the maximum possible value of *dist*, given by

$$Max = \sqrt{2 \times \sum_{i=1}^{n}(c_{1i})} \tag{3.10}$$

The SCL ranges from 0 to 1. 1 means that the two rulebases being compared are structurally identical, while 0 means their structures are completely different.

3.2 Examples

Example 1

This example demonstrates the computation of linguistic gradients. F_1 is a Mamdani fuzzy system taken from [125]. Semantic rules of the linguistic variables are given in Fig.

3.2, and the rulebase is presented in Table. 3.2. The linguistic gradient is computed for each point in F_I as follows:

For the point (dist, angle) = (ZE, ZE) in F_I,

$$\begin{aligned}&\nabla C(ZE,ZE)\\&=(NULL+\Delta(PM,PS),NULL+\Delta(PS,PS))\\&=(PM,AZ)\end{aligned}$$

As with digital images, the rulespace has a defined border, and the points in rulespace along this border must receive special treatment. We elect to insert a NULL value into Equ. 2.53 at every point where an out-of-bounds value would be called for, and extend the label addition defined in Table 2.2 to include this situation. Label additions involve NULL are defined in Table 3.1

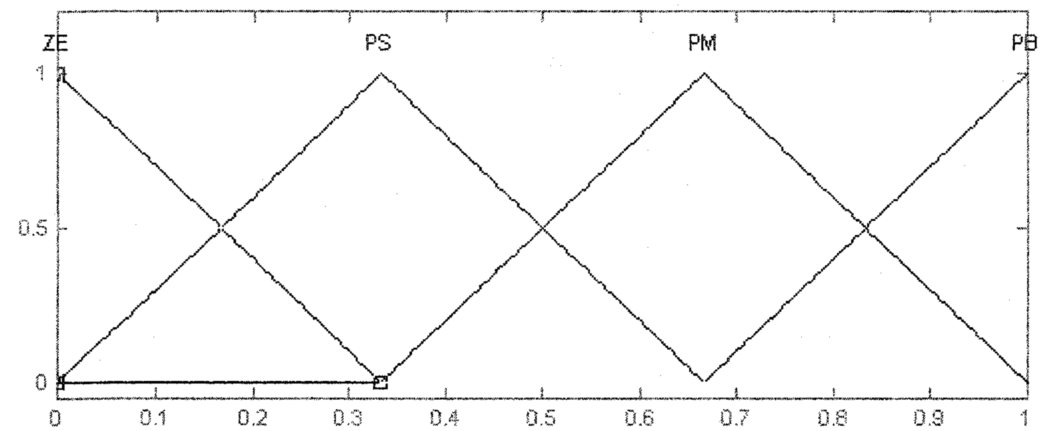
Table 3.1 LABEL ADDITIONS INVOLVING NULL

	NL	**NM**	**NS**	**AZ**	**PS**	**PM**	**PL**
NULL	NL	NM	NS	AZ	PS	PM	PL

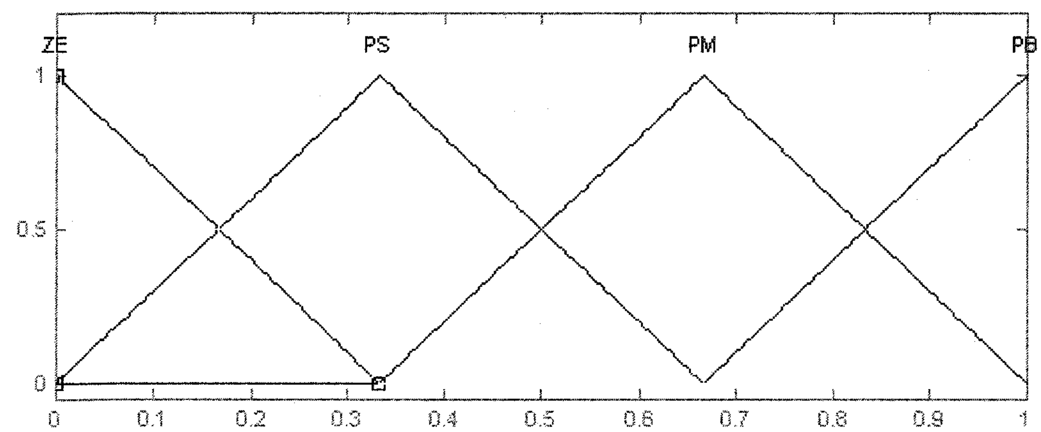
The point (dist, angle) = (*PS, PS*) is a normal case. The linguistic gradient at this point is given by

$$\begin{aligned}&\nabla C(PS,PS)\\&=(\Delta(PM,PS)+\Delta(PB,PM),\Delta(PM,PM)+\Delta(PS,PM))\\&=(PM,NS)\end{aligned}$$

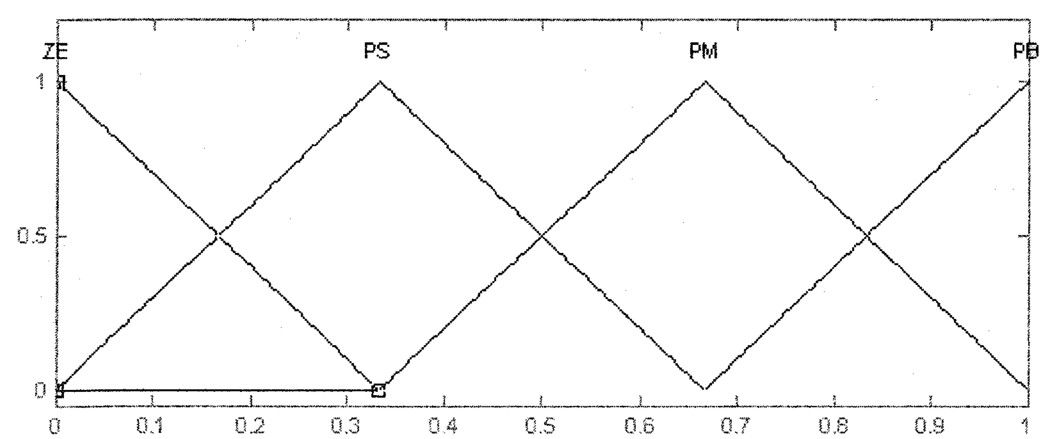
Gradient vectors for other points can be calculated in the same way. The gradient map for F_I is shown in Table 3.3.



(a) Input variable "dist"



(b) Input variable "angle"



(c) Output variable "adj-var"

Figure 3.2 Membership funcions of F_1

Table 3.2 RULEBASE OF F_1

Adj-var		Dist			
		ZE	PS	PM	PB
Angle	PB	ZE	PS	PS	ZE
	PM	ZE	PS	PS	PS
	PS	PS	PM	PB	PB
	ZE	PS	PM	PB	PB

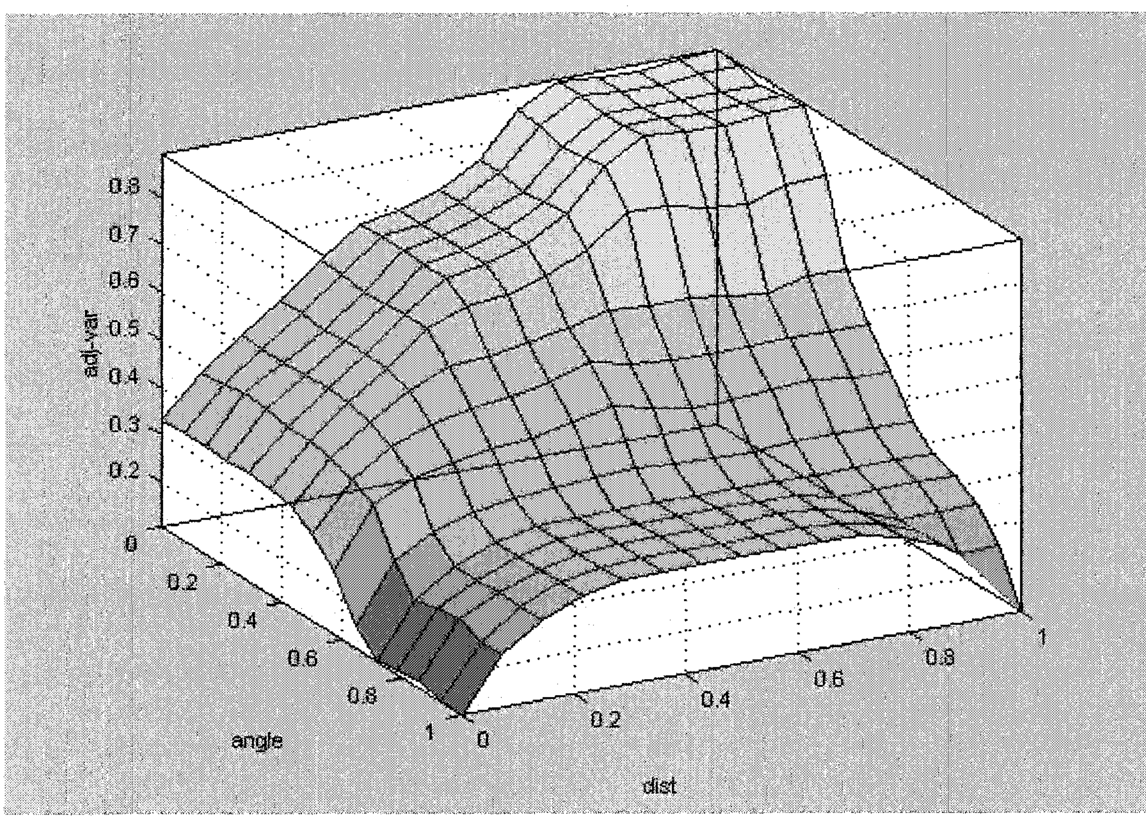


Figure 3.3 Reasoning Surface of F_1

Table 3.3 GRADIENT MAP OF F_1

∇C(adj-var)		Dist			
		ZE	PS	PM	PB
Angle	PB	PM,AZ	PS,AZ	NS,AZ	NM,NM
	PM	PM,NS	PS,NS	AZ,NM	AZ,NL
	PS	PM,NS	PM,NS	PS,NM	AZ,NM
	ZE	PM,AZ	PM,AZ	PS,AZ	AZ,AZ

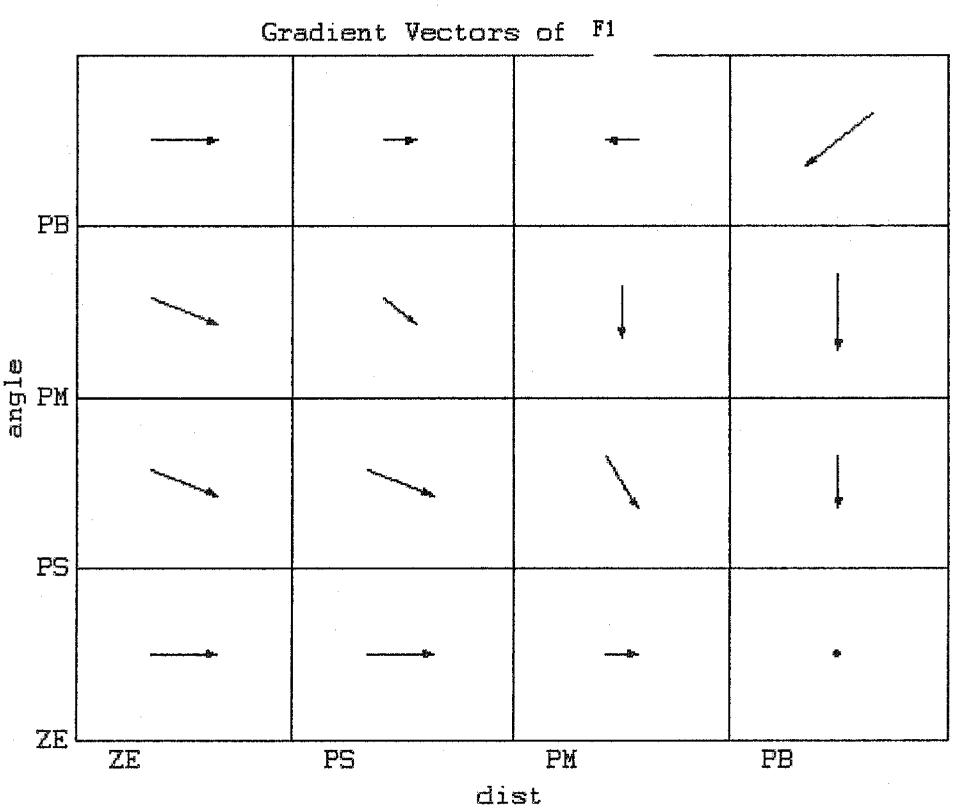


Figure 3.4 Linguistic Gradient Vector Field of F_1

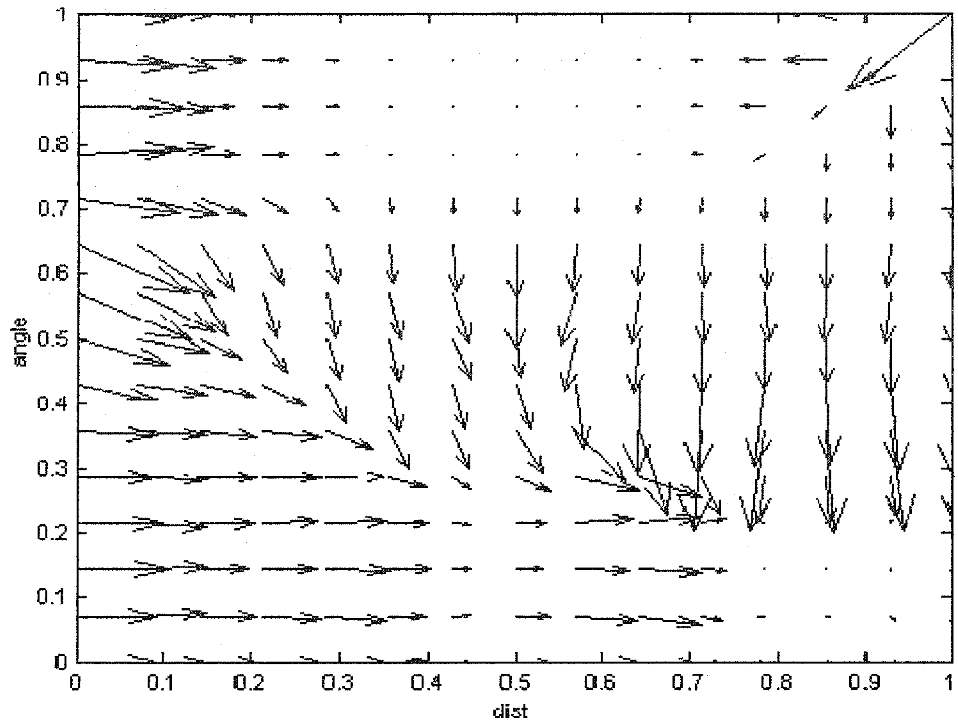


Figure 3.5 Quiver plot of the control surface of F_1

The gradient vector field of F_1 is shown in Figure 3.4, while the numerical gradient vector field for the control surface of F_1 as computed by the Matlab ® Fuzzy Logic Toolbox is shown in Figure 3.5. The relationship between them is obvious. The gradient vector field map is a coarse approximation of the numerical gradient vector field.

Example 2

In this example, we calculate the SCL for the two Mamdani fuzzy inference systems F_1 [125] and F_2 [126], for which membership functions are shown in Fig. 3.2 and Fig. 3.6, and rulebases are shown in Table 3.2 and Table 3.5, respectively. The computation of the gradient map of F_1 is shown in *Example 1.* The respective gradient maps for F_1 and F_2 are shown in Table 3.3 and Table 3.6.

After the gradient maps are created, they are thresholded as follows:

For F_1,

- $c_{11} = 4$, $\therefore$ $threshold_{11}$ = PM;
- $c_{12} = 4$, $\therefore$ $threshold_{12}$ = PM;

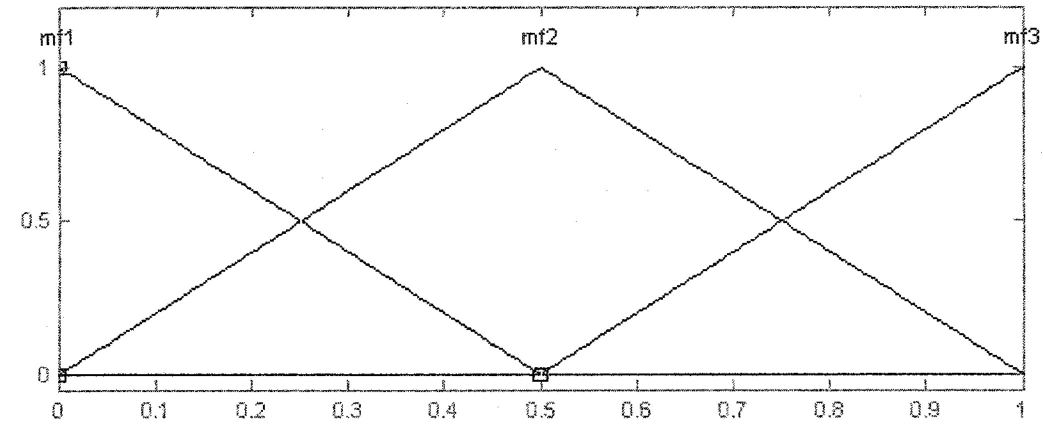
For F_2,

- $c_{21} = 3$, $\therefore$ $threshold_{21}$ = PM;
- $c_{22} = 4$, $\therefore$ $threshold_{22}$ = PM;

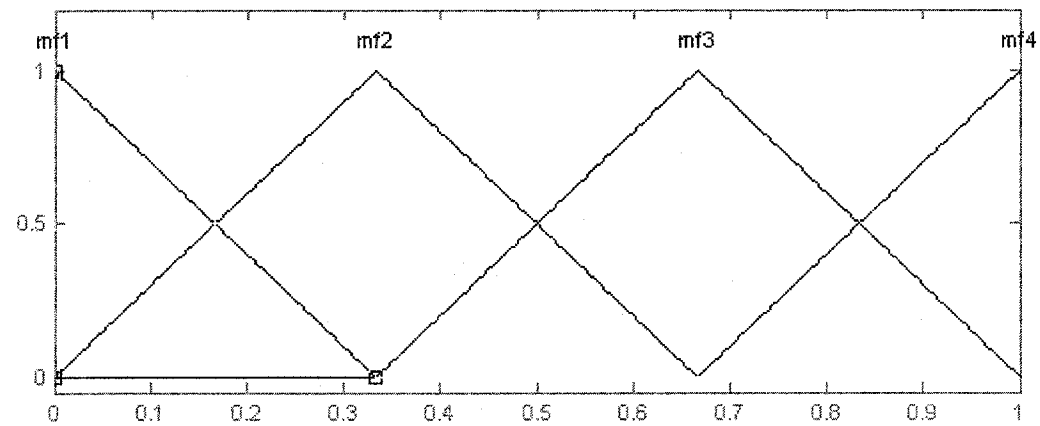
where $threshold_{11}$ and $threshold_{12}$ are thresholds for the dimensions dist and angle of F_1, $threshold_{21}$ and $threshold_{22}$ are thresholds for the dimensions x_1 and x_2 of F_2 respectively. The thresholded gradient maps for F_1 and F_2 are given in Table 3.4 and Table 3.7, respectively.

Table 3.4 THRESHOLDED GRADIENT MAP OF F_1

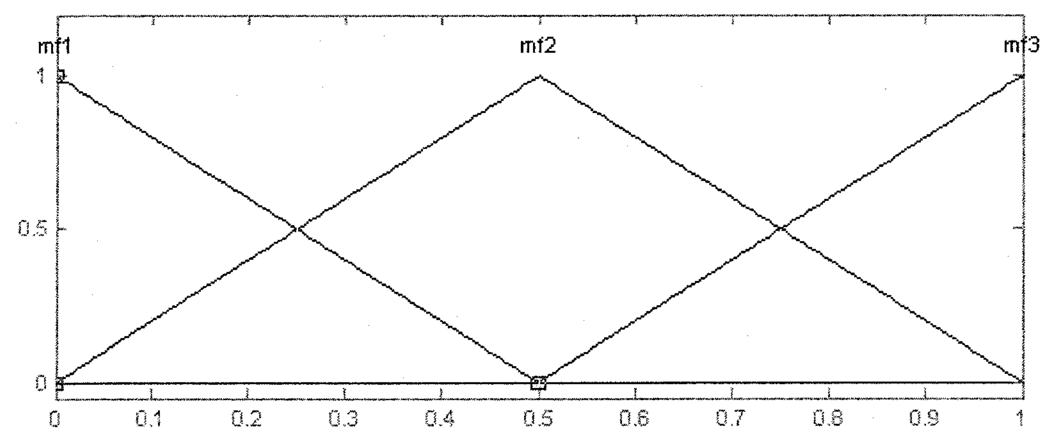
thresholded ∇C(adj-ar)		dist				
		ZE	PS	PM	PB	
Angle	PB	1,0	0,0	0,0	-1,-1	P_{12}={(0,0) (2,0) (2,0) (1,0)}/4
	PM	1,0	0,0	0,-1	0,-1	
	PS	1,0	1,0	0,-1	0,-1	
	ZE	1,0	1,0	0,0	0,0	
P_{11} =		{(0, 4)	(0, 2)	(0, 0)	(1, 0)}/4	



(a) Input variable “x1”



(b) Input variable “x2”



(c) Output variable “output”

Figure 3.6 Membership functions for F_2

Table 3.5 RULEBASE OF F_2

output		x1		
		mf1	mf2	mf3
x2	mf4	mf2	mf2	mf3
	mf3	mf3	mf3	mf3
	mf2	mf1	mf2	mf2
	mf1	mf1	mf1	mf2

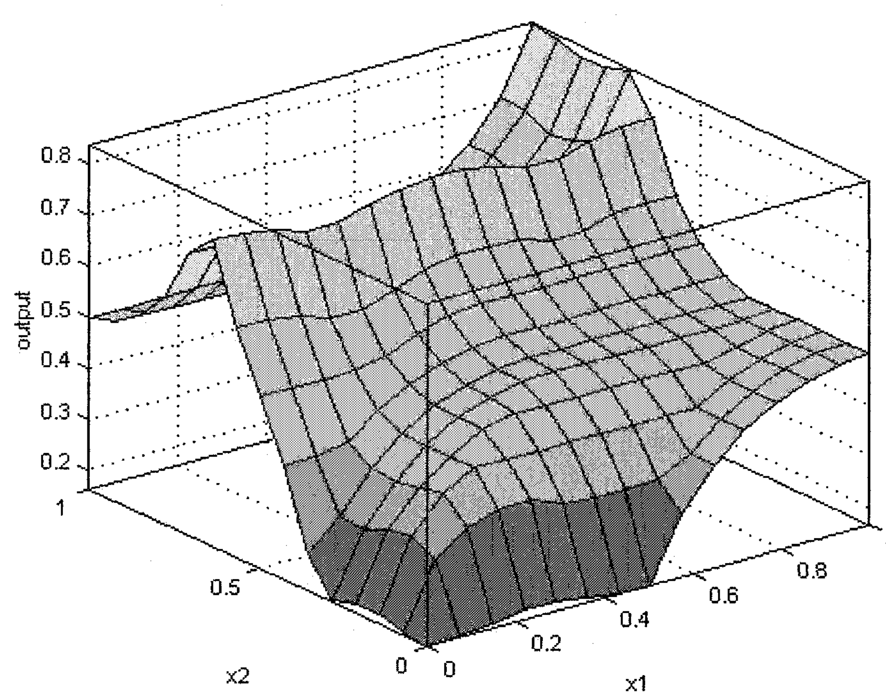


Figure 3.7 Reasoning Surface of F_2

Table 3.6 GRADIENT MAP OF F_2

∇C(output)		x1		
		mf1	mf2	Mf3
x2	mf4	AZ,NM	PS,NM	PM,AZ
	mf3	AZ,PS	AZ,AZ	AZ,PS
	mf2	PM,PM	PS,PM	AZ,PS
	mf1	AZ,AZ	PS,PM	PM,AZ

Table 3.7 THRESHOLDED GRADIENT MAP OF F_2

Thresholded ∇C(output)		X1		
		Mf1	Mf2	Mf3
x2	Mf4	0,-1	0,-1	1,0
	Mf3	0,0	0,0	0,0
	Mf2	1,1	0,1	0,0
	Mf1	0,0	0,1	1,0

P_{21} = {(0, 1) (0, 0) (0, 2)}/3

P_{22}={(0,1) (0,2) (0,0) (2,0)}/4

The projection vectors are calculated by counting the number of 1s and -1s separately along each axis in the thresholded gradient map and then normalizing as in Equ. 3.4. We obtain 2 projection vectors for each rulebase, P_{11} and P_{12} for F_1, P_{21} and P_{22} for F_2,

P_{11} = [(0, 1) (0, 0.5) (0, 0) (0.25, 0)]

P_{12} = [(0, 0) (0.5, 0) (0.5, 0) (0.25, 0)]

P_{21} = [(0, 0.25) (0, 0) (0, 0.5)]

P_{22} = [(0, 0.3333) (0, 0.6667) (0, 0) (0.6667, 0)]

P_{11} and P_{21} are not of the same length so one of them needs to be regranulated. Here we regranulate P_{11} to the same granularity as P_{21}. The intersections of membership functions of dist in F_1 are 0.16665, 0.5 and 0.83335. They divide the universe of discourse of dist into 4 intervals: [0, 0.16665], [0.16665, 0.5], [0.5, 0.83335] and [0.83335, 1]. Similarly, the universe of discourse of x1 in F_2 is divided into 3 intervals, [0, 0.25], [0.25, 0.75] and [0.75, 1]. P_{11} is regranulated using Equ. 3.5 as follows:

$$p'_{111} = p_{111} + p_{112} \times (0.25 - 0.16665) / (0.5 - 0.16665)$$

$$= (0, 1) + (0, 0.5) \times 0.08335 / 0.33335$$

$$= (0, 1)$$

$$p'_{112} = p_{112} \times (0.5 - 0.25) / (0.5 - 0.16665) + p_{113} \times (0.75 - 0.5) / (0.83335 - 0.5)$$

$$= (0,0.5) \times 0.25 / 0.33335 + (0,0) \times 0.25 / 0.33335$$

$$= (0, 0.3750) + (0, 0)$$

$$= (0, 0.3750)$$

$$p'_{113} = p_{113} \times (0.83335 - 0.75) / (0.83335 - 0.5) + p_{114}$$

$= (0, 0) \times 0.08335 / 0.33335 + (0.25, 0)$

$= (0.25, 0)$

Combining these 3 pairs together, we obtain

$P'_{11} = \{p'_{111}, p'_{112}, p'_{113}\}$

$= \{(0, 1)\ (0, 0.3750)\ (0.25, 0)\}$

The Euclidean distance between (P'_{11}, P_{12}) and (P_{21}, P_{22}) is

$dist = \text{sqrt}((0-0)^2 + (1-0.25)^2 + (0-0)^2 + (0.3750-0)^2 + (0.25-0)^2 + (0-0.5)^2 + (0-0)^2 + (0-0.3333)^2 + (0.5-0)^2 + (0-0.6667)^2 + (0.5-0)^2 + (0-0)^2 + (0.25-0.6667)^2 + (0-0)^2) = 1.4983$

The maximum possible value for a 3×4 rulebase is

$$Max = \sqrt{2 \times (3+4)} = 3.7417$$

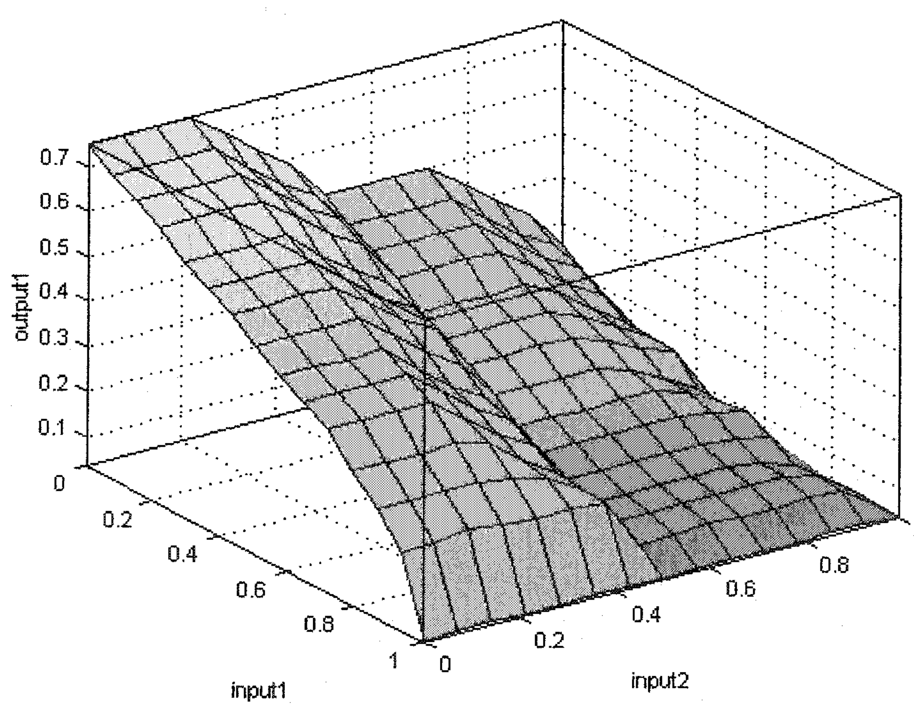
Finally, we obtain the SCL

$$SCL(F_1, F_2) = \frac{3.7417 - 1.4983}{3.7417} = 0.5996$$

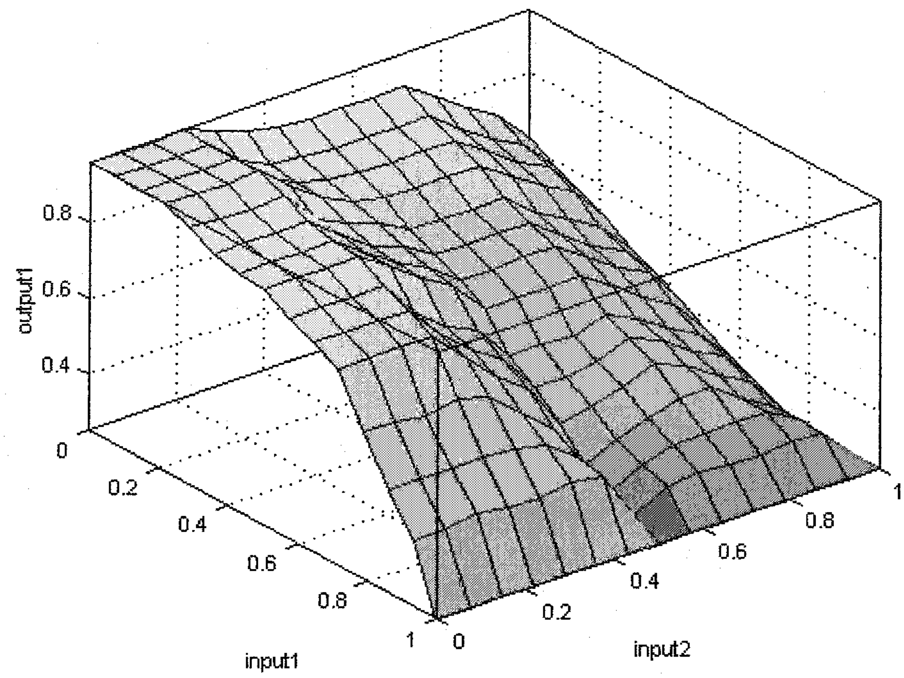
If we regranulate P_{21} instead of P_{11}, the SCL is 0.6210, which is close to the above result.

Example 3

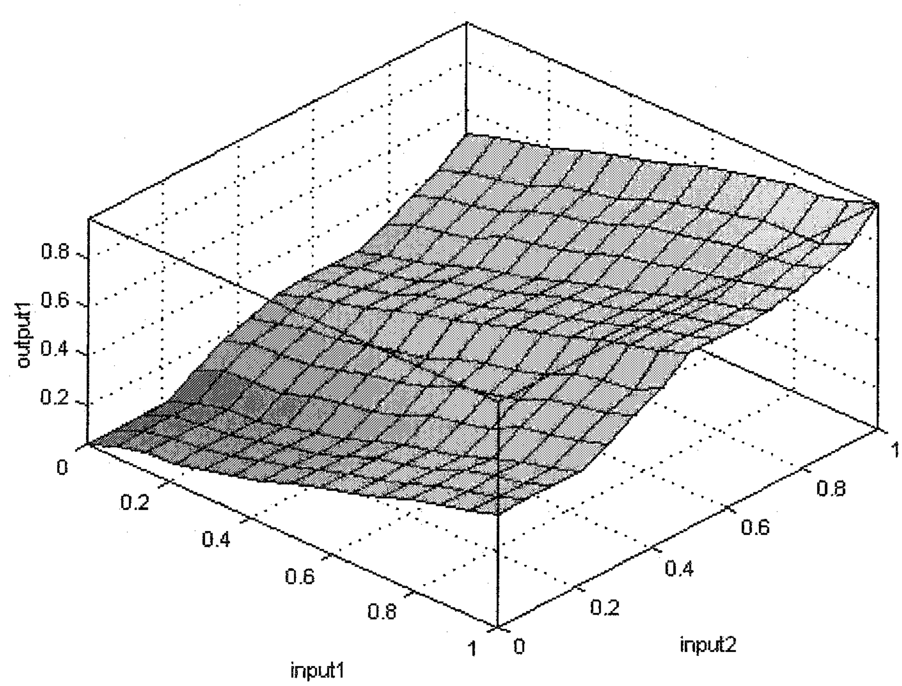
Fig.3.3 shows the reasoning surfaces of 3 Mamdani systems F_3, F_4 and F_5. The reasoning surface of F_3 looks very close to that of F_4 – the most important difference between them is their output range. F_3's output range is [0, 0.8]; F_4's output range is [0,1]. The reasoning surface of F_5 is very different from that of F_3 and F_4. Table 3.6 gives their pair wise SCL values and RMS differences. SCL (F_3, F_4) = 0.9000 means that F_3 and F_4 are structurally close, while SCL (F_3, F_5) = 0.1101 and SCL (F_4, F_5) = 0.1180 means both F_3 and F_4 are very different from F_5 in structure. Although the pair wise RMS differences are fairly close, the SCL has the ability to reveal crucial structural differences. That is because the RMS comes from both the dynamic behaviours (shapes of the reasoning surfaces) and the difference of output range, while SCL depends only on the dynamic behaviours.



(a) Reasoning Surface of F_3



(b) Reasoning Surface of F_4



(c) Reasoning Surface of F_5

Figure 3.8 Reasoning Surface of F_3, F_4 and F_5

Table 3.8 PAIR WISE RMS AND SCL OF F_3, F_4 AND F_5

Rulebases Pair	RMS	SCL
F_3, F_4	0.3173	0.9000
F_4, F_5	0.4392	0.1180
F_3, F_5	0.3794	0.1101

Chapter 4 EXPERIMENTAL RESULT

This chapter describes our validation experiment for the SCL. We had expected a negative correlation between SCL and RMS, so we designed a test based on a set of 2-dimensional fuzzy rulebases collected from the literature. After normalizing and applying some restrictions, we totally got 603 pairs of 2-d rulebases. This test took pair-wise RMS and SCL of these rulebases as two variables, and computed the Spearman rank correlation coefficient between them. The result showed that the SCL significantly negatively correlate to RMS.

Another important aspect of an algorithm is computational performance. Although the computational complexities of SCL and RMS are of similar order, SCL is computed at a higher level of granularity, so we were expecting a higher performance for SCL. In our experiment, the SCL demonstrated a much faster running time than RMS.

In what follows, we first describe the validation strategy. We then present our experimental design and method, followed by the results. Finally, we give the analysis and experimental result for computational performance.

4.1 Validation Strategy

Validating a new measure needs a ground truth. Widely used ground truths are human expert ranking or other measures. Unfortunately, we couldn't find a qualified expert to help us in our research, so we have to use another measure to validate SCL. As mentioned in Chapter 1, there is no such measure for fuzzy rulebases in literature, thus, we choose to compare SCL with a more traditional difference measure for two systems, the RMS difference, over a fuzzy rulebases library, in which all rulebases' universes of discourse are normalized. Example 3 shows that RMS difference tends to be larger for two fuzzy rulebases that have different universes of discourse, even though they have similar reasoning surface. In our experiment, we normalized all rulebases, made their corresponding universes of discourse identical, such that their pair wise RMS consisted of only dynamic part. Since SCL compares the dynamic behaviours of two fuzzy rulebases, over a normalized fuzzy rulebases library, RMS can be a suitable ground truth.

The root-mean-square (RMS) of a variate X, sometimes called the quadratic mean, is the square root of the mean squared value of x: [6]

$$R(x) = \begin{cases} \sqrt{\dfrac{\sum_{i=1}^{n} x_i^2}{n}} & \text{for a discrete distribution} \\ \sqrt{\dfrac{\int P(x)x^2 dx}{\int P(x)dx}} & \text{for a continuous distribution} \end{cases} \tag{4.1}$$

The RMS is often used to quantify the deviation of a signal from a given baseline or fit. In the case of crisp inputs and outputs, a fuzzy inference system implements a nonlinear mapping from its input space to output space [3, 4]. Therefore when the number of inputs and respective universe of discourse of two fuzzy systems are identical, we can use the RMS difference as an index to measure the similarity of these two systems, the smaller the RMS difference, the more similar the two systems are.

The Spearman rank correlation coefficient is a nonparametric (distribution-free) rank statistic proposed by Spearman in 1904 as a measure of the strength of the associations between two variables [5]. It can be used to discover the strength of a link between two sets of data. The Spearman rank correlation coefficient is defined by

$$r' = 1 - 6\sum \frac{d^2}{N(N^2-1)} \tag{4.2}$$

where d is the difference in statistical rank of corresponding variables, N is the number of samples. The Spearman rank correlation coefficient is an approximation to the exact correlation coefficient

$$r = \frac{\sum xy}{\sqrt{\sum x^2 \sum y^2}} \tag{4.3}$$

computed from the original data.

There are some advantages to a rank-based test. Because it uses ranks, the Spearman coefficient is much easier to compute than the exact correlation coefficient. We do not need to assume any parametric family of distributions for the data. The Spearman coefficient is also robust: not sensitive to occasional errors in the data or to outliers.

We use the Spearman coefficient because, while both RMS and SCL map to numeric values, the underlying quantities they represent have fundamentally different granulations (i.e. they compare numeric reasoning surfaces and linguistic rulebases). The rank-based Spearman statistic is therefore more appropriate than Pearson's coefficient. Spearman correlations with an absolute value of roughly 0.3 or greater (out of a possible range of [–1, 1]) are generally taken to denote significant relationships. Moreover, in our experiment, all universes of discourse of the rulebases have been, such that the RMS only contains of dynamic part. Since SCL and RMS reveal contradistinctive properties of a pair of rulebases, they should negatively correlate.

A further technique is required to test the significance of the relationship. The significance gives the probability of the relationship we have found being a chance event.

The significance value is determined by the Spearman rank correlation coefficient and the degree of freedom (population of data set minus 2). This value must be looked up on the Spearman Rank significance table. Usually, a significance level of lower than 5% or 1% indicates that the correlation is not a result of chance [5].

For summary, the criterion for validating SCL is that in the experiment, we can obtain a Spearman rank correlation coefficient between SCL and RMS less than -0.3 with significance less that 0.01, over a normalized fuzzy rulebases library.

4.2 Constraints

We collected all 89 fuzzy rulebases that appeared in the proceedings of the FUZZ-IEEE '98 for validation purpose. All universe of discourse of these rulebases were normalized. For symmetric dimensions, the universe of discourse was normalized to [-1, 1]; for asymmetric dimensions, the universe of discourse was normalized to [0, 1]. Not all possible pairs of rulebases were comparable. Some restrictions applied:

1. As the assumptions indicated in chapter II, this research is limited to MISO Mamdani systems only;
2. In order to calculate the RMS difference and the SCL, 2 rulebases must have these properties in common: number of dimensions, universe of discourse of corresponding dimension;
3. Output ranges of the two rulebases should be identical. Although we can calculate the RMS difference and SCL of a pair of rulebases with different output ranges, the result is meaningless for our objective;

4. Rulebases of more than 2 dimensions are seldom reported in the fuzzy systems literature, and so our collection contains very few of them. Because a data set with too few samples is statistically meaningless, the experiment on higher dimensional fuzzy systems has to be left for future research.

Another constraint concerns incomplete rulebases. When computing RMS, if a particular input fired no rule, we computed the system output in the same way as that in Matlab®, in which the system output is set as the mean value of the output range. When computing linguistic gradient, if one rule was missing, it was set according to following rules:

1). If the output LV was symmetrical, the missed rule was assigned the zero term.

2) If the output LV was asymmetrical, the missed rule was assigned a pseudo term. This pseudo term was a singleton located at the middle of the output universe of discourse and the term set distances between this pseudo-term and its neighbour terms are set to 0.5.

Given these constraints, we were able to compare 603 pairs of 2-d rulebases.

4.3 Experimental Result

Programs implementing the SCL and validation were developed in C++ and Matlab®. A C++ fuzzy library reads the Matlab fis files, does the Mamdani inferencing, computes the RMS difference, and generates the gradient vector map. A group of Matlab scripts implement the rest of the algorithm described in Chapter III, and compute the Spearman rank correlation coefficient.

First, we calculated the RMS differences and SCL of the 603 pairs of rulebases. Then, we calculated the Spearman Rank Correlation Coefficient between the RMS differences and SCL.

Regranulating rulebase of higher granularity and regranulating that of less granularity resulted in slightly different SCL values. In this experiment, when regranulating the rulebase which has a greater sum of all input cardinalities, we got a Spearman Rank correlation Coefficient of –0.5639, which gave a significant level much less than 0.01. In fact, the significance factor is of the order e–11. When regranulating the rulebase which has a less sum of input cardinalities, we got a Spearman Rank Correlation Coefficient of –0.5651 with a significant level of the order e–11. These results fulfill our validation criteria, and so we are prepared to assert that the SCL is a valid measure of the difference between two fuzzy rulebases.

4.4 Performance Analysis

Performance is an important aspect of algorithm analysis. One of the advantages of CW scheme is that it provides high computational efficiency [38]. Efficiency means using fewer resources including CPU cycles (time) and computer memory (space). Time and space are both important, but we are usually more interested in time efficiency rather than space efficiency these days - relative memory prices are quite low compared to the past.

In the situation of continuous mapping, for the convenience of computation, we have to discretize the input variables. One disadvantage of this method is that it causes transformation distortion, because during the discretization process, part of the input space is ignored. We were facing this problem and paying tremendous time cost to minimize the error when computing RMS difference between 2 fuzzy systems.

Computing RMS includes 4 steps. First, discretize the universe of discourse of every input; second, for all points in the discretized input space, compute the outputs; then, compute the differences of corresponding points between two fuzzy systems; finally, compute the RMS using Equ 4.1. The pseudo code for computing RMS is listed as follows:

Algorithm 1

```
function RMS(fis1, fis2)
cursor = inf    //inf: an array storing the infimums of the universe of discourse
num, denom = 0
while (cursor[n] < sup[n])  //sup: supremum array; n: number of inputs
        num = num + (fireFis(fis1, cursor) - fireFis(fis2, cursor))^2
        denom = denom + 1
        cursor[1] = cursor[1] + sr  //sr: step size
        for i = 1 to n
                if cursor[i] > sup[i] then
                        cursor[i] = inf[i]
                        cursor[i+1] = cursor[i+1] + sr
                end if
        end for
end while
```

```
return sqrt(num / denom)
end function


function fireFis(fis, x)
    num = 0, denom = 0
    fireAllRules(fis, x)
    for cursor = consequent.inf to consequent.sup step sd  //sd is defuzzification
accuracy
        y = consequent.eval(cursor)
        num = num + y * cursor
        denom = denom + y
    end for
    return num / denom
end function
```

From above code, we can observe that RMS is computed over a high-dimensional grid of points. The computation of RMS involves great amount of numeric operations. The complexity for computing RMS is $O(\frac{width(V_o)}{s_d}\prod_{i=1}^{n}\frac{width(V_i)}{s_r})$, where s_d is the accuracy for defuzzification, s_r is the step size for computing RMS difference, n is the dimension of the input space, V_i and V_o are the universes of discourse of the input and output, respectively. Simulating an actual continuous system in a discrete space induces distortion. Obviously, the smaller we set the step size, the more precise the simulation is. In [127], Ross gives a fuzzy inferencing example, in which using 63 sampling points for every fuzzy MF gives a relatively good precision, while lower sampling rates result in big distortion. In our experiment, where each LV contains at least 2 atomic terms, it is reasonable to set width(V)/s greater than 100, even though it will significantly affect the time performance. Large step sizes improve the efficiency, but without structure knowledge, may miss important local behavior.

SCL is computed at every point in the linguistic input space. The complexity for computing SCL is $O(\prod_{i=1}^{n} c_i)$. Generally, n ranges from 1-7 and the maximum value of c_i is 7-9. Complexities of both RMS and SCL increase exponentially with respect to the number of dimensions. However, in all cases, c_i is greatly less than width(V_i)/s_r, which reflects the fact that RMS and SCL are computed at different level of granularity. Consequently, we could expect that the SCL is much more time efficient that RMS.

Our experiment testified our expectation. The experiment was run on a computer with AMD2200+ CPU, 512M RAM, 30G HDD, and windows XP. Parameters were set as s_d = 0.005, s_r=0.01. In our test data set, n was 2, the maximum c_i was 9. Note that, as described in Section 1.3, RMS was computed using compiled C++ code, while the SCL was a mixture of C++ and Matlab® scripts, which are slower. For computing RMS of the 603 pairs of FISes, it took over 10 hours for the computer to run the program. Contrastively, it only took about 65 seconds for computing linguistic gradients and SCL of the same data set. The experiment has shown that SCL is much more efficient than RMS. This computational efficiency is extremely important when querying high-volume rulebase library.

Chapter 5 CONCLUSION

Soft computing (SC) is a set of computing paradigms including Neural Network (NN), Fuzzy set theory, approximate reasoning, derivative-free optimization methods such as genetic algorithms (GA) and simulated annealing (SA). SC is a complement to conventional AI approaches. SC has become a part of “modern AI”. One of the most successful applications of SC methods is the application of fuzzy systems in fields where intuition and judgment play critical roles. So far, the most promising and valuable application of fuzzy systems is control systems, such as temperature controller, traffic controller and process controller. This success lies on two characteristics of fuzzy systems: first, being universal approximators, they are able to model and control highly non-linear plants. Second, a qualitative description of the process to be controlled is sufficient for fuzzy systems.

The idea of computing with words was introduced by Zadeh [38] to emulate human thinking. Computing with words employs words in computing and reasoning, arriving at conclusions expressed as words from premises expressed in a natural language. The importance of computing with words derives from two facts. Firstly, there are many problems in which the available information is not precise enough. Secondly, there are many situations in which there is a tolerance for imprecision which can be exploited through the use of CW to achieve tractability, robustness, low solution cost and better rapport with reality [129].

Fuzzy rulebase is the key component of a fuzzy system. Therefore, analyzing the fuzzy rulebase rather than the numeric system I/O makes possible the comprehension of the nature of the system. Since fuzzy rulebases are expressed in linguistic if-then rules, it is more appropriate to analyze them at the linguistic level of granularity, following the CW paradigm.

In this thesis, we introduce the SCL as a novel similarity measure for fuzzy rulebases. Analogous to a CBIR algorithm, SCL uses linguistic gradient operator to extract structural information from rulebases, then compares rulesbases based on this information. The proposed algorithm is able to reveal the structure similarity between 2 fuzzy rulebases rather than comparing the defuzzified system output, and thus follows the CW framework. By avoiding time-consuming numeric operations over high-dimensional input space, The SCL has a much higher computational efficiency than RMS difference. Hence, it is more suitable for time-critical applications such as querying libraries of rulebases. Thus, it provides the user a better heuristic alternative to the traditional method.

A validation experiment has been performed, and demonstrated a strong negative correlation between SCL and RMS difference. Our objective is achieved. However, this experiment was based on 2-dimensional fuzzy rulebases. Further validation experiments need to be done on higher-dimensional rulebases. In addition to further database testing, comparison of SCL results with human expert rankings would be a useful next step.

The inputs of a rulebase are independent, so their order is unimportant in practice. However, when comparing two rulebases, if the order of inputs of one of them is changed, both the RMS difference and SCL will change. This is similar to the rotation of image around the diagonal axis. Many CBIR algorithms addressing this issue have been proposed [130, 131]. In the future, we intend to develop a new algorithm to extract an input-order invariant and use it to design a new similarity operator.

One possible application for such a measure is to use this measure as an index in libraries of reusable software components. In this approach, the description of a component in the library could be a fuzzy rulebase, and a query to the library could take the form of another fuzzy rulebase representing the desired functionality. This application will be a future topic of research.

REFERENCE

[1] Dick, S., Rodriguez, and W., Kandel, A., "A granular counterpart to the gradient operator," *Soft Computing*, vol. 6 no. 2, pp. 124-140, Apr 2002.

[2] Bargiela, A., and Pedrycz, W., *Granular Computing An Introduction*, Kluwer Academic Publishers, Norwell, MA, USA, 2003.

[3] Wang, L.X., *A Course in Fuzzy Systems and Control*, Upper Saddle River, NJ: Prentice Hall PTR, 1997.

[4] Klir, G. J., Yuan, B., *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Upper Saddle River, NJ: Prentice Hall PTR, 1995.

[5] Lehmann, E.L. and D'Abrera, H.J.M. *Nonparametrics: Statistical Methods Based on Ranks*, revised. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[6] Eric W. Weisstein. "Root-Mean-Square", From *MathWorld* - A Wolfram Web Resource. http://mathworld.wolfram.com/Root-Mean-Square.html

[7] Zadeh, L. A. "Fuzzy Sets," *Information and Control,* vol. 8, pp.338-353, 1965.

[8] Sugeno, M.. "Fuzzy measures and fuzzy integrals: a survey," in Gupta ,M. M., Saridis, G. N., and Gaines, B. R., editors, *Fuzzy automata and decision processes*, pages 89-102, North-Holland, New York, 1997.

[9] Yager, R.R. "On the measure of fuzziness and negation, Part I: membership in the unit interval," *International Journal of Man-Machine Studies*, vol. 5, pp. 221-229, 1979.

[10] Buckley, J. J., Eslami, E. *An Introduction to Fuzzy Logic and Fuzzy Sets*, Physica-Verlag, New York, 2002.

[11] Dong, W., Shah, H. "Vertex method for computing functions of fuzzy variables," *Fuzzy Sets Systems*, vol.24, pp.65-87, 1987.

[12] Warmus, M., "Calculus of Approximations," *Bull. Acad. Polon. Sci., Cl. III*, vol. IV, No. 5, pp. 253-259. 1956.

[13] Warmus, M., "Approximations and Inequalities in the Calculus of Approximations. Classification of Approximate Numbers," *Bull. Acad. Polon. Sci., Ser. math., astr. et phys.*, vol. IX, No. 4, pp. 241-245, 1961.

[14] Bellman, R., and Giertz, M. "On the analytic formalism of the theory of fuzzy sets," *Information Sciences*, vol. 5, pp. 149-156, 1973.

[15] Voxman, W., and Goetschel, R. "A note on the characterization of the max and min operators," *Information Sciences,* vol. 30, pp. 5-10, 1983.

[16] Schweizer, B., and Sklar, A. "Statistical metric spaces," *Pacific J. of Mathematics,* vol., 10, pp. 313-334, 1960.

[17] Schweizer, B., and Sklar, A.. "Associate functions and statistical triangle inequalities," *Publicationes Mathematicae Debrecen,* vol. 8, pp. 169-186, 1961.

[18] Schweizer, B., and Sklar, A. "Associate functions and abstract semi-groups," *Publicationes Mathematicae Debrecen,* vol. 10, pp. 69-81, 1963.

[19] Zadeh, L. A. "The concept of a linguistic variable and its application to approximate reasoning," *Information Sciences,* vol. 8, pp. 199-249, 301-357; vol. 9, pp. 43-80, 1975.

[20] Yager, R. R. "A characterization of the extension principle," *Fuzzy Sets and Systems,* vol. 18, pp. 205-217, 1986.

[21] Munakata, T., and Jani, Y., "Fuzzy Systems: an Overview," *Fuzzy systems*, vol.37, no.3, pp68-76, March, 1994.

[22] Zadeh, L., *Preface, in (Marks-II R.J.) Fuzzy logic technology and applications.* IEEE Technical Activities Board, 1994.

[23] Novak, V., *Fuzzy sets and their applications.* Adam Hilger, Bristol, 1989.

[24] Zimmermann, H.-J., *Fuzzy set theory and its applications*. Second ed. Boston, MA: Kluwer Academic Publishers, 1996.

[25] Klir, G.J., Yuan, B., eds. *Fuzzy sets, fuzzy logic and fuzzy system: Selected papers by Lotfi A. Zadeh*. World Scientific Singapore, 1996.

[26] Nguyen, H.T., Walker, A. *First course in fuzzy logic.* CRC Press, 1999.

[27] Hajek, P. *Metamathematics of fuzzy logic*. Kluwer 1998.

[28] Turunen, E., *Mathematics behind fuzzy logic.* Physica Verlag 1999.

[29] Novak, V., Perfilieva, I., Mockor, J., *Mathematical principles of fuzzy logic.* Kluwer 2000.

[30] Hadipriono, F. and Sun, K. "Angular fuzzy set models for linguistic values," *Civ. Eng. Syst*. Vol.7. no.3, pp.148-156, 1990.

[31] Takagi, H. and Hayashi, I., "NN-driven fuzzy reasoning," *Int. J. Approximate Reasoning*, vo.5, pp.191-212, 1991.

[32] Karr, C. L. and Gentry, E. J., "Fuzzy control of pH using genetic algorithms," *IEEE trans. Fuzzy Syst*, vol.1, no.1, pp.46-53, 1993.

[33] De Luca, A. and Termini, S. "A definition of a non-probalilistic entropy in the setting of fuzzy sets theory," *Inf. Control*, vol.20, pp.301-312, 1972.

[34] Holland J.H., *Adaptation in natural and artificial system,* Ann Arbor, The University of Michigan Press, 1975.

[35] Dick, S., and Kandel, A., "Granular computing in neural networks," in Pedrycz, W., editor, *Granular computing: an emerging paradigm*, pp: 275 – 305, Physica-Verlag, 2001.

[36] Rose, J., Shah, M., "Content-Based Image Retrieval Using Gradient Projections," in *Proc. IEEE Southeastcon '98,* pp. 118-121, Orlando, FL, USA, April 24-26, 1998.

[37] Jang, J.-S.R., Sun, C.-T., and Mizutani, E., *Neuro-Fuzzy and Soft Computing*, Upper Saddle River, NJ: Prentice Hall, 1997.

[38] Zadeh, L.A., "Fuzzy Logic = Computing with Words," *IEEE Transactions on Fuzzy Systems*, vol. 4 no.2, pp.103-111, May 1996.

[39] Pedrycz, W., "Neural Networks in the Framework of Granular Computing," *Int. J. Appl. Math. Comput. Sci.*, vol. 10, no. 4, pp 723-745, 2000.

[40] Pedrycz, W., Vukovich, G., "On Elicitation of Membership Functions," *IEEE Trans. On Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 32, no.6, pp 761-767, Nov 2002.

[41] Botolan, G., Pedrycz, W., "Fuzzy Descriptive Models: An Interactive Framework of Information Granulation," *IEEE Trans. On Fuzzy Systems*, vol. 10, no. 6, pp. 743-755, Dec 2002.

[42] Bargiela, A., Pedrycz, W., "Recursive Information Granulation: Aggregation and Interpretation Issues," *IEEE Trans. On Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 33, no. 1, pp 96-112, Feb 2003.

[43] Sugeno, M., Yasukawa, T., "A Fuzzy Logic Based Approach to Qualitative Modeling," *IEEE Trans. Fuzzy Syst.*, vol.1, pp 7-21, Jan 1993.

[44] Tikk, D., Biro, G., Gedeon, T.D., Koczy, L.T., Yang, J.D., "Improvements and Critique on Sugeno's and Yashukawa's Qualitative Modeling," *IEEE Trans. On Fuzzy Systems*, vol. 10, no. 5, pp 596-606, Oct 2002.

[45] Höppner, F., Klawonn, F., and Eklund, P., "Learning Indistinguishability from Data," *Soft Computing*, vol. 6 no.1, pp 6-13, 2002.

[46] Pedrycz, W., "Conditional Fuzzy c-Means," *Pattern Recognition Letters*, vol.17 no. 6, pp 625-632, 1996.

[47] Setnes, M., Babuska, R., Kaymak, U., and Lemke, R. "Similarity Measures in Fuzzy Rule Base Simplification," *IEEE Trans. SMC Part B*, vol. 28, no.3, pp 376-386, 1998.

[48] Joentgen, A., Mikenina, L., Weber, R., and Zimmermann, H.J., "Dynamic fuzzy data analysis based on similarity between functions," *Fuzzy Sets and Systems*, vol.105, pp 81-90, 1999.

[49] Black, M. "Vagueness: an exercise in logical analysis," *Philosophy of Science*, vol. 4, pp. 427-455, 1937.

[50] Zadeh, L.A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. On Systems, Man and Cybernetics*, vol.3 no.1, pp.28-44, Jan. 1973.

[51] Mamdani, E.H. "Application of Fuzzy Algorithm for Control of Simple Dynamic Plant," Proceedings of the Institute of Electrical Engineers, vol.121, pp.1585-1588, 1974.

[52] Takagi, T., and Sugeno, M. "Fuzzy Identification of systems and its Appliction to Modeling and Control," IEEE Trans. On systems, Man and Cybernetics, vol.15, pp.116-132, 1985.

[53] Sugeno, M. and Kand, G.T., "Fuzzy Modelling and control of Multilayer Incinerator," *Fuzzy Sets and Systems*, vol.18, pp.329-346, 1986.

[54] Tanaka, K. and Sugeno, M., "Stability Analysis and Design of Fuzzy Control Systems," *Fuzzy Sets and Systems*, vol.45, pp.135-156, 1992.

[55] Tanaka, K., Ikeda, T., and Wang, H.O., "Quadratic Stability and Stabilization of Fuzzy Control Systems," in *Proceedings of the 1996 Biennial Conference of the NAFIPS*. 1996, pp.245-249.

[56] Thathachar, M.A.L., and Viswanath, P., "On the Stability of Fuzzy Systems," *IEEE Trans. On Fuzzy Systems*, vol.5, no.1, pp.145-151, Feb. 1997.

[57] Wang, L.X., and Mendel, J.M., "Generating fuzzy rules by learning from examples," *In Proc. 1991 IEEE International Symposium on Intelligent Control*, Arlington, VA, USA, pp. 263–271, 1991.

[58] Wang, L.X., "Stable Adaptive Fuzzy Controllers with Application to Inverted Pendulum Tracking," *IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics*, vol. 26, no. 5, pp.677-691, Oct. 1996.

[59] Kim, W.C., Ahn, S.C., and Kwon, W.H., "Stability Analysis and Stabilization of Fuzzy State Space Models," Fuzzy Sets and Systems, vol.71, pp.131-142, 1995.

[60] Zadeh, L.A., "Quantitative Fuzzy Semantics," Information Sciences, vol. 3, pp159-176, 1971.

[61] Fukami, S., Mizumoto, M., and Tanaka, K., "Some Considertations on Fuzzy conditional inference," Fuzzy Sets and Systems, vol.4, pp.243-273, 1980.

[62] Batur, C., and Kasparian, V., "Self-organizing model based expert controller," in *Proc. IEEE International Conference on Systems Engineering*, Fairborn, OH, USA, pages 411–425, 1989.

[63] Langari, G., and Tomizuka, M., "Self organizing fuzzy linguistic control with application to arc welding," In Proc. IROS '90: IEEE International Workshop on Intelligent Robots and Systems '90, Towards a New Frontier of Applications, Ibaraki, Japan, vol.2, pages 1007–1021, 1990.

[64] Linkens, D.A., and Abbod, M.F., "Self-organizing fuzzy logic control for real-time processes," In *Proc. International Conference on Control '91*, Edinburgh, UK, (IEE), vol.2, pp. 971–977, 1991.

[65] Linkens, D.A., and Abbod, M.F., "On-line self-organizing fuzzy logic controller with patient simulators," In *Proc. IEEE Colloquium on Knowledge- Based Control*: Principles and Applications (Digest No.091), London, UK, pp. 1–5, 1991.

[66] Linkens, D.A., and Abbod, M.F., "Fast, self-organizing control for industrial processes," *In Algorithms and Architectures for Real-Time Control: Proc. IFAC Workshop*, Bangor, UK, pp. 153–160, 1991.

[67] Mamdani, E.H., and Assilian, S., "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," *Int. J. Man-Mach. Stud.*, vol. 7, pp. 1-13, 1974.

[68] Zhang, B.S., and Edmunds, J.M., "Self-organising fuzzy logic controller," *IEEE Proc. D (Control Theory and Applications)*. vol.139, no.5, pages 460–464, 1992.

[69] Wang, L.X., and Mendel, J.M., "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man and Cybernetics*. vol.22, no.6, pages 1414–1441, 1992.

[70] Yamaguchi T., Takagi, T., and Mita, T., "Self-organizing control using fuzzy neural networks," *International Journal of Control*. vol.56, no.2, pages 415–454, 1992.

[71] Arabshahi, P., Robert J. Marks II, and Reed, R., "Adaptation of Fuzzy Inferencing: A Survey," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*. vol.E75-A, no.12, pp.1826–1834, 1992.

[72] Wang, L.X., "Stable Adaptive Fuzzy Control of Nonlinear Systems," *IEEE Trans ON Fuzzy Systems*, vol. 1, no.1, pp. 146-155 May 1993.

[73] Turksen, I.B., and Zhong, Z., "An approximate analogical reasoning approach based on similarity measures," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, no. 6, pp.1049-1056, 1988.

[74] Raha, S., Pal, N. R., Ray, K. S., "Similarity-Based Approximate Reasoning: Methodology and Application," *IEEE Trans. On Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol.32, no.4, pp. 541-547, July 2002.

[75] Kandel, A., *Fuzzy Mathematical Techniques with Applications*, Reading, MA: Addison-Wesley Pub. Co., 1986.

[76] Hong, D.H., and Hwang, C., "A T-Sum Bound of LR Fuzzy Numbers," *Fuzzy Sets and Systems*, vol.91, pp.239-252, 1997.

[77] Giachetti, R.E., and Young, R.E., "A Parameteric Representation of Fuzzy Numbers and their Arithmetic Operators," *Fuzzy Sets and Systems*, vol.91, pp.185-202, 1997.

[78] Dubois, D. and Prade, H.M., "Systems of Linear Fuzzy Constraints," *Fuzzy Sets and Systems*, vol.3, pp.37-48, 1980.

[79] Dubois, D. and Prade, H.M., "On Several Definitions of the differential of a Fuzzy Mapping," *Fuzzy Sets and Systems*, vol.24, pp.117-120, 1987.

[80] Dubois, D. and Prade, H.M., "Additions of Interactive Fuzzy Numbers," *IEEE Trans. Automatic Control*, vol.26 no.4, pp.926-936, Aug 1981.

[81] Dubois, D. and Prade, H.M., "Varous Kinds of Interactive Addition of Fuzzy Numbers. Application to Decision Analysis in Presence of Linguistic Probabilisties," in *Proceedings of the 1979 IEEE Conference on Decision and control*, pp.783-787, 1979.

[82] Dubois, D. and Prade, H.M., "On Distances between Fuzzy Points and Their Use for Plausible Reasoning," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp.300-303, 1979.

[83] Koczy, L.T., Hirota, K., "Ordering, Distance and Closeness of Fuzzy Sets," *Fuzzy Sets and Systems*, vol.59, pp.281-293, 1993.

[84] Klir, G.J., "Fuzzy Arithmetic with Requisite Constraints," *Fuzzy Sets and Systems*, vol.91, pp.165-175, 1997.

[85] Delgado, M., Verdegay, J.L., Vila, M.A., "On Aggregation Operations of Linguistic Labels," *International Journal of Intelligent Systems*, vol.8, pp.351-370, 1993.

[86] Ma, M., Turksen, I.B., Kandel, A., "Fuzzy Partition and Fuzzy Rule Base," *Inf. Sci.* vol. 108, no. 1, pp. 109-122, 1998.

[87] Heilpern, S., "Representation and Application of Fuzzy Numbers," *Fuzzy Sets and Systems*, vol.91, pp.259-268, 1997.

[88] Ogata, K., *Modern Control Engineering*, 3rd Ed. Upper Saddle River, NJ: Prentice-Hall, 1997.

[89] Driankov, D., Hellendoorn, H., Reinfrank, M., *An Introduction to Fuzzy Control*, New York: Springer-Verlag, 1993.

[90] Liu, C., Shindhelm, A., Li, D., Jin, K., "A Numerical Approach to Linguistic Variables and Linguistic Space," in *Proceeding of the 1996 IEEE International Conference on fuzzy Systems*, pp.954-959, 1996.

[91] Stilman, B., "Linguistic Geometry: Methodology and Techniques," *Cybernetics and Systems*, vol. 26, pp. 535-597, 1995.

[92] Pawlak, Z., "Rough Sets," *Internat. J. Comput. Inform. Sci.*, vol. 11, pp. 341–356. 1982.

[93] Munakata, T., *Fundamentals of the new artificial intelligence: Beyond traditional paradigms*. Springer–Verlag., 1998.

[94] Hirota, K., "Concepts of probabilistic sets," In: *Fuzzy Sets and Systems* Vol. 5, pp. 31-46. 1981.

[95] Yager, R.R., "On ordered weighted averaging aggregation operators in multi-criteria decision making," *IEEE Transactions on Systems, Man and Cybernetics* vol.18 pp. 183-190. 1988.

[96] Yager, R.R., Filev, D.P., "Induced Ordered Weighted Averaging Operators," *IEEE Trans. Syst., Man, Cybern. Part B-Cybernetics*, vol. 229 no. 2, pp. 141-150, 1999.

[97] Liang, Q., and Mendel, J.M., "Interval Type-2 Fuzzy Logic Systems: Theory and Design," *IEEE trans. on Fuzzy Syst.*, vol. 8, no.5, pp. 551-563, Oct. 2000.

[98] Mendel, J.M., *Computing with words when words can mean different things to different people*, Presented at the Int. ICSC Congr. Computation Intelligence: Methods Application, 3rd Annual Symp. Fuzzy Logic Applicaton, Rochester, NY, June 1999.

[99] Mendel, J.M., *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[100] Karnik, N. N., Mendel, J. M., *An introduction to Type-2 Fuzzy Logic Systems, Signal & Image Processing Institute*, Los Angeles, CA: Department of Electrical Engineering & Systems, University of California, Technical Report, 1998.

[101] Wu, H., Mendel, J. M., "Uncertainty Bounds and Their Use in the Design of Interval Type-2 Fuzzy Logic Systems," *IEEE Trans. On Fuzzy Systems*, vol. 10, no. 5, pp. 622-639, Oct 2002.

[102] Pedrycz, W., "Relational and Directional Aspects in the Construction of Information Granules," *IEEE Trans. On Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol.32, no.5, pp. 605-614, Sep 2002.

[103] Auephanwiriyakul, S., Keller, J. M., "Analysis and Efficient Implementation of a Linguistic Fuzzy C-Means," *IEEE Trans. On Fuzzy Systems*, vol. 10, no. 5, Oct 2002.

[104] Ying, M., "A Formal Model of Computing With Words," *IEEE Trans. On Fuzzy Systems*, vol. 10, no. 5, pp.640-652, Oct 2002.

[105] McCulloch, W.S., Pitts, W., "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, vol.5, pp. 115-133, 1943.

[106] Hebb, D.O., *The Organization of Behavior: A Neuropsychological Theory*, New York: John Wiley & Sons, 1949.

[107] Rosenblatt, F., "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review*, vol.65 no.6, pp386-408, 1958.

[108] Rumelhart, D.E., Hinton, G.E., Williams, R.J., "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Rumelhart, D.E., McClelland, J.L., Eds. Cambridge, MA: MIT Press, pp.318-362, 1986.

[109] Haykin, S., *Neural Networks: A Comprehensive foundation*, 2nd Ed. Upper Saddle River, NJ: Prentice Hall, 1999.

[110] Powell, M.J.D., "Radial basis functions for multivariable interpolation: A review," in *Proc. Of IMA Conference on Algorithms for the Approximation of Functions and Data*, pp.143-167, RMCS, Shrivenham, England, 1985.

[111] Hopfield, J.J., "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences, USA*, vol.81, pp.3088-3092, 1984.

[112] Goldberg, D.E., "Genetic and evolutionary algorithm come of age," *Commun. ACM,* vol. 37, no. 2, pp. 113-119, Mar, 1994.

[113] Pal, S.K., and Mitra, S., *Neuro-fuzzy Pattern Recognition: Methods in Soft Computing*. New York: Wiley, 1999.

[114] Buckley, J.J., and Feuring, T., *Fuzzy and Neural: Interactions and Applications, Studies in Fuzziness and Soft Computing*. Heidelberg, Germany: Physica-Verlag, 1999.

[115] Jang, J.R., "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, 1993.

[116] Lin, C.T., and Lee, C.S., *Neural Fuzzy Systems—A Neuro–Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[117] Kasabov, N., *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. Cambridge, MA: MIT Press, 1996.

[118] Pedrycz, W., *Computational Intelligence: An Introduction*. Boca Raton, FL: CRC, 1998.

[119] Wang, L.X., *Adaptive Fuzzy Systems and Control.* Englewood Cliffs, NJ: Prentice-Hall, 1994.

[120] Hayashi, Y., Buckley, J. J., and Czogala, E., "Fuzzy neural network with fuzzy signals and weights," *Int. J. Intell. Syst.*, vol. 8, no. 4, pp. 527–537, 1993.

[121] Mitra, S. and Pal, S.K., "Logical operation based fuzzy MLP for classification and rule generation," Neural Networks, vol. 7, pp. 353–373, 1994.

[122] Ishibuchi, H., Nozaki, K., Yamamoto, N. and Tanaka, H., "Selecting fuzzy If–Then rules for classification problems using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 260–270, 1995.

[123] Wang, L. and Yen, J., "Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter," *Fuzzy Sets Syst.*, vol. 101, pp. 353–362, 1999.

[124] Yoshitaka, A., Ichikawa, T., "A Survey on Content-Based Retrieval for Multimedia Databases," *IEEE Trans on Knowledge and Data Eng.*, vol.11 no.1, pp.81-93, Jan 1999.

[125] Choi, J.W., "Navigation Strategy of an Intelligent Mobile Robot Using Fuzzy

Logic," in *Proceedings of the 1998 IEEE International Conference on Fuzzy Systems*, May 4-9, 1998, Anchorage, AK, USA, pp. 602-605.

[126] Holve, R., "Investigation of Automatic Rule Generation for Hierarchical Fuzzy Systems," in *Proceedings of the 1998 IEEE International Conference on Fuzzy Systems*, May 4-9, 1998, Anchorage, AK, USA, pp. 973-978.

[127] Ross, T.J., *Fuzzy Logic with Engineering Applications*, New York: The McGraw-Hill companies, Inc, 1995.

[128] Gonzalez, R.C., Woods, R.E., *Digital Image Processing*, Second Edition, NJ: Prentice Hall, 2001.

[129] Zadeh, L.A., "Outline of a Computational theory of Perceptions Based on Computing with Words," *Soft Computing & Intelligent Systems: theory & applications*, Sinha, N.K., Gupta, M.M. ed, FL: Academic Press, 2000.

[130] Schmid, C., and Mohr, R., "Local Grayvalue Invariants for Image Retrieval," *IEEE Trans. PAMI*, vol.19, no.5, pp.530-535, May 1997.

[131] Swain, M.J., and Ballard, B.H., "Color Indexing," *Int'l J. Computer Vision*, vol.7, no.1, pp.11-32, 1991.

[132] Zadeh, L.A., "Fuzzy logic, Neural Networks, and Soft Computing," *Commu. of the ACM*, vol.37, no.3, pp. 73-84, Mar 1994.

[133] Zhang, Y.Q., Kandel A., "Fuzzy moves using compensatory granular reasoning," *Int'l J of Computational Intelligence and Applications*, vol.3, no.3, pp.249-312, Sep 2003.

[134] Yin, P., "Finding the reduct by granular computing using bit maps – decision rules mining," *in Proceeding of the SPIE – Int. Soc. Opt. Eng.*(USA), vol.5098, pp.82-93, 2003.

[135] Ling, Z., Bo, Z., "Theory of fuzzy quotient space (methods of fuzzy granular computing)," *Journal of Software*, vol.14, no.4, pp.770-776, Apr 2003.

[136] Liu, Q., "Granular language and its reasoning," *in Proc. SPIE – Int. Soc. Opt. Eng.* (USA), vol.5098, pp. 279-366, 2003.

[137] Pedrycz, W., "Computational intelligence and visual computing: an emerging technology for software engineering," *Soft Computing*, vol.7 no.1, pp.33-44, 2002.

[138] Pedrycz, W., Vukovich, G., "Granular computing with shadowed sets," *Int. J. Intell. Syst.*, vol.17, no.2, pp. 173-270, Feb 2002.

[139] Zadeh, L.A., "Granular computing as a basis for a computational theory of perceptions," *in 2002 IEEE World Congress on Computational Intelligence. FUZZ-IEEE'02 Proceedings*, vol.1, pp. 564-569, 2002.

[140] Chiang, Y.T., Chiang, Y.C., Hsu, T.S., Liau, C.J., Wang, D.W., "How much privacy? A system to safe guard personal privacy while releasing databases," *RSCTC 2002 Proceedings*, pp.226-259. 2002.

[141] Zhang, D., Gunopulos, D., Tsotras, V.J., Seeger, B., "Temporal and spactio-temporal aggregations over data streams using multiple time granularities," *Information Systems*, vol.28, no. 1-2, pp. 61-84. March-Apr 2003.

[142] Dick, S., Schencker, A., Last, M., Bunke, H., Kandel, A., "Re-Granulating a Fuzzy Rulebase," in *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, Melbourne, Austraila, Dec. 2-5, 2001, pp. 372-375