CHAPTER 6 Decision Inference with Fuzzy Logic

6.1 Introduction

In this chapter, the origin of fuzzy logic is first introduced. Then, the standard procedure of table-lookup scheme is illustrated in details. Lastly, the application of the table-lookup scheme in solving ECG classification problem is discussed.

Two main theoretical developments following the publication of Dr. Zadeh's first seminal paper on fuzzy sets are the Mamdani and Sugeno fuzzy models [79-81]. Zadeh's main contention is that, although probability theory is appropriate for measuring randomness of information, it is inappropriate for measuring the meaning of information. Indeed, much of the confusion surrounding the use of English words and phrases is related to lack of clarity (vagueness) rather than randomness. This is a crucial point for analyzing language structures and can also be important in creating a measure of confidence in production rules. Zadeh proposes possibility theory as a measure of vagueness, just as probability theory measures randomness [79].

Zadeh's theory expresses lack of precision in a quantitative fashion by introducing a set membership function that can take on real values between 0 and 1 [80]. This notion of a fuzzy set can be described as follows: let S be a set and s a member of that set. A fuzzy subset F of S is defined by a membership function mF(s) that measures the "degree" to which is belongs to F. A standard example of a fuzzy set, as presented in Figure 6.1, is for S to be the set of positive integers and F to be the fuzzy subset of S called small integers. Now various integer values can have a "possibility" distribution defining their "fuzzy membership" in the set of small integers: mF(1)=1.0, mF(2)=1.0, mF(3)=0.9, F(4)=0.8, ..., mF(50) = 0.001, and etc. With the statement of positive integer $s \in S$ is a small integer, the membership function, mF(s) creates a possibility distribution across all the positive integers, S.

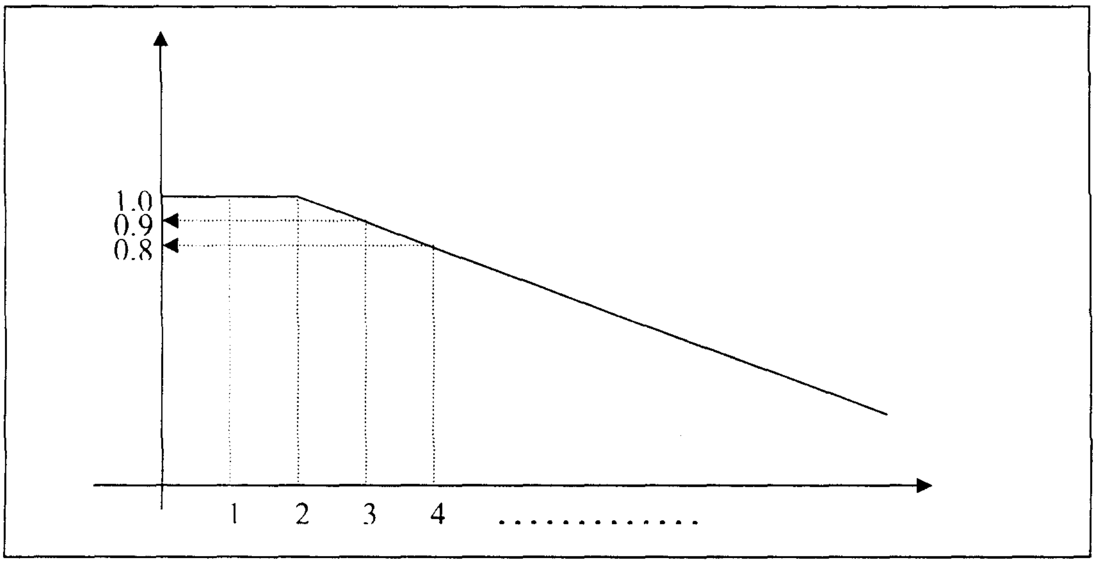

Figure 6.1: The fuzzy set representation for "small integers".

Fuzzy set theory is not concerned with how these possibility distributions are created, but rather with the rules for computing the combined possibilities over expressions that contain fuzzy variables. Thus, it includes rules for combining possibility measures for expressions containing fuzzy variables. When we design information processing systems such as controllers and classifiers, two kinds of information are usually available. One is numerical data and the other is linguistic knowledge from domain experts. Various pattern classification methods have been proposed for designing classification systems from numerical data [82-85]. Those methods usually cannot utilize linguistic knowledge for designing classification systems. On the other hand, fuzzy rule-based systems [86, 87] are traditionally designed from linguistic knowledge of human experts. Recently various methods have been proposed for automatically designing fuzzy rule-based systems from numerical data without human experts [88-95] i.e. the table-lookup scheme. We have adopted a training algorithm using a table-lookup scheme as described in the Section 6.2.

6.2 Table-Lookup Scheme Training Procedure

In classical set theory, a subset A of a set X can be defined by its characteristic function χ_A as a mapping from the elements of X to the elements of the set $\{0,1\}$, χ_A: X $\rightarrow$ $\{0,1\}$. This mapping may be represented as a set of ordered pairs, with exactly one ordered pair present for each element of X. The first element of the ordered pair is an element of the set X, and the second element is an element of the set $\{0,1\}$. The value zero is used to represent non-membership, and the value one is used to represent membership. The truth or falsity of the statement "x is in A" is determined by the ordered pair (x, χ_A(x)). The statement is true if the second element of the ordered pair is 1, and the statement is false if it is 0. Similarly, a fuzzy subset A of a set X can be defined as a set of ordered pairs, each with the first element from X, and the second element from the interval [0,1], with exactly one ordered pair present for each element of X. This defines a mapping, μ_A, between elements of the set X and values in the interval [0,1]. The value zero is used to represent complete non-membership, the value one is used to represent complete membership, and values in between are used to represent intermediate degrees of membership. The set X is referred to as the universe of discourse for the fuzzy subset A. Frequently, the mapping μ_A is described as a function, the membership function of A. The degree to which the statement "x is in A" is true is determined by finding the ordered pair (x, μ_A(x)). The degree of truth of the statement is the second element of the ordered pair. It should be noted that the terms membership function and fuzzy subset get used interchangeably.

In this section, we develop a very simple method for adaptive fuzzy system that performs a one-pass operation on the numerical input-output pairs and linguistic fuzzy IF-THEN rules. The key idea of this method is to generate fuzzy rules from input-output pairs, collect the generated rules and linguistic rules into common fuzzy rule base, and construct a final fuzzy logic system based on the combined fuzzy rule base.

Generation of the fuzzy rules from numerical data is illustrated in the following. Suppose we are given a set of desired input-output data pairs:

$(x_1^{(1)}, x_2^{(1)}; y^{(1)}), (x_1^{(2)}, x_2^{(2)}; y^{(2)}), \ldots$

where x_1 and x_2 are inputs, and y is the output. The task here is to generate a set of fuzzy IF-THEN rules from the desired input-output pairs and use these fuzzy IF-THEN rules to determine a fuzzy logic system $f:(x_1, x_2) \rightarrow y$.

We illustrate this approach in the following three steps:

Step 1 We divide the input and output spaces into fuzzy regions. Every node i in this layer is a fixed node with a node function

$$L_{1,i} = \mu_{Ai}(x_1) \qquad \text{if } i = 1,2,3 \qquad \text{or} \tag{6.1}$$

$$L_{1,i} = \mu_{B(i-3)}(x_2) \qquad \text{if } i = 4,5,6 \qquad \text{or} \tag{6.2}$$

$$L_{1,i} = \mu_{C(i-6)}(y) \qquad \text{if } i = 7,8,9 \tag{6.3}$$

where $L_{1,i}$ is the membership grade of a fuzzy set A (= A1, A2, A3, B1, B2, B3, C1, C2, C3) ,it specifies the degree to which the input-output pairs $(x_1, x_2; y)$ satisfies the quantifier A.

Here the membership function (MF) for A is the generalized bell function:

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - a_A}{c_A} \right|^{2b_A}} \tag{6.4}$$

where $\{a, b, c\}$ is the parameter set. As the values of these parameters change, the bell-shaped function varies accordingly, thus exhibiting various forms of membership functions (MFs) for fuzzy set A. Parameters in this layer are referred to as premise parameters.

Step 2 We first assign the input-output pairs $(x_1, x_2; y)$ to the regions with maximum degrees, thus obtain one rule from each pair of desired input-output data. We assign a degree to each rule j generated from data pairs as follows:

$$L_{2,j} = \max\{\mu_{Ai}(x_1)\}.\max\{\mu_{Bi}(x_2)\}.\max\{\mu_{Ci}(y)\}, \text{ for } i = 1,2,3 \ldots \quad (6.5)$$

We accept only the rule j from a conflict group that has maximum degree.

Table 6.1-6.3 illustrates the table-lookup representations of a fuzzy rule base. We fill the boxes of the base with fuzzy rules according to the following strategy: a combined fuzzy rule base is assigned rules from either those generated from numerical data or linguistic rules (we assume that a linguistic rule also has a degree which is assigned by the human expert and reflects the expert's belief of the importance of the rule); if there is more than one rule in one box of the fuzzy rule base, use the rule that has maximum degree. In this way, both numerical and linguistic information are codified into a common framework- the combined fuzzy rule base. If a linguistic rule is an "and" rule, it fills only one box of the fuzzy rule base, but if a linguistic rule is an "or" rule (that is, a rule for which the THEN part follows if any condition of the IF part is satisfied), it fills all the boxes in the rows or columns corresponding to the regions of the IF part.

Step 3 We use the following defuzzification strategy to determine the output control y for given inputs (x_1, x_2). First, for given inputs (x_1, x_2), we combine the antecedents of the j th fuzzy rule using product operations to determine the degree,

$$w^j{}_{O^j} = \mu_{I_1^j}(x_1)\mu_{I_2^j}(x_2) \tag{6.6}$$

where O^j denotes the output region of Rule j, and $I_k{}^j$ denotes the input region of Rule j for the k th component. Then we use the centre average defuzzification formula to determine the output

$$L_3 = y = \frac{\sum_{j=1}^{M} w^j{}_{O^j}\,\overline{y}^j}{\sum_{j=1}^{M} w^j{}_{O^j}} \tag{6.7}$$

where $\overline{y}^j$ denotes the centre value of region O^j, and M is the number of fuzzy rules in the combined fuzzy rule base.

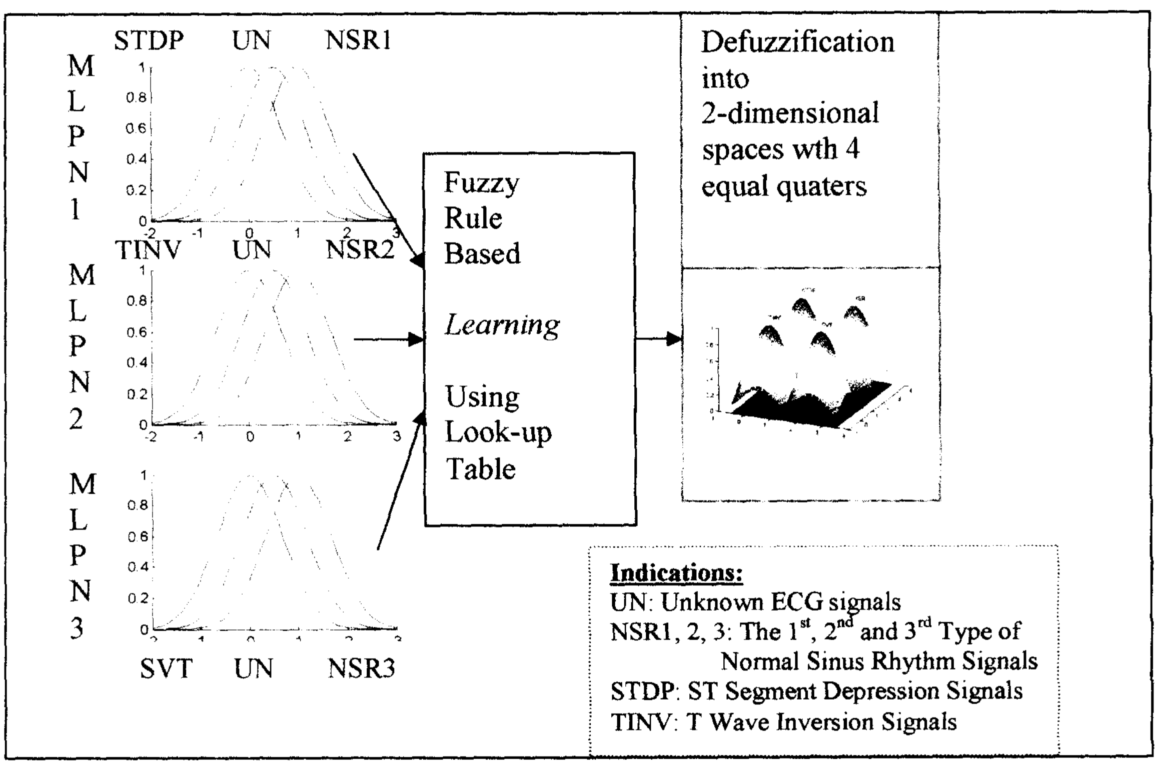

Figure 6.2: Partition of the input and output spaces of the fuzzy inference system into fuzzy regions and the corresponding membership functions.

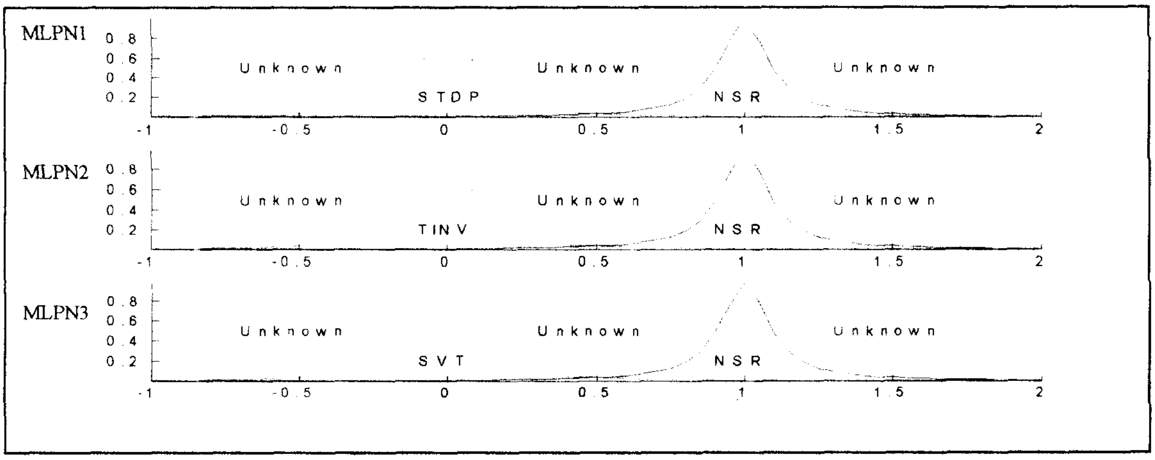

Figure 6.3: Modified partition of the input spaces of the fuzzy inference system into fuzzy regions and the corresponding membership functions.

Fuzzy partition of the input and output spaces into fuzzy regions and the corresponding membership functions are as shown in Figure 6.2 and 6.3. In this section, we describe the table look-up scheme for generating fuzzy if-then rules directly from the numerical output values of the three trained neural networks. Advantages of our method over other rule extraction methods [96-104] are as follows: (1) extracted fuzzy if-then rules are always linguistically interpretable, (2) a certainty grade is assigned to each rule and (3) our method is applicable to arbitrary trained neural networks. That is, our method is a general algorithm to extract fuzzy if-then rules by handling the mapping of the numerical output values from the trained neural networks as a black box model.

Note that the activation function for the neuron is unipolar sigmoidal, i.e. has range of [0,1], at the hidden layer only, but the output neuron has linear activation function, i.e. has range of $[-\infty, \infty]$. Therefore, any ECG signal that failed to be recognized by the neural networks will fall into the ranges of output value greater than one, approximately 0.5 or below zeros. Hence, we modified the fuzzy membership functions for the input spaces of the fuzzy inference system as shown in Figure 6.3.

x_2 \ x_1	STDP	Unknown	NSR
TINV	SVT	SVT	SVT
Unknown	SVT	SVT	SVT
NSR	SVT	SVT	Unknown

Table 6.1: Table-lookup illustration of the fuzzy rule base for x_3 is SVT.

x_2 \ x_1	STDP	Unknown	NSR
TINV	STDP	TINV	Unknown
Unknown	STDP	Unknown	NSR
NSR	Unknown	SVT	NSR

Table 6.2: Table-lookup illustration of the fuzzy rule base for x_3 is Unknown.

x_2 \ x_1	STDP	Unknown	NSR
TINV	STDP	TINV	Unknown
Unknown	STDP	NSR	NSR
NSR	Unknown	Unknown	NSR

Table 6.3: Table-lookup illustration of the fuzzy rule base for x_3 is NSR.

6.3 Conclusions

In this study, there is no proper way of determining the shape of the fuzzy regions. We heuristically tune the fuzzy regions until satisfactory results are obtained. We first applied a 10-step time-average filter to the output of each MLPN, then we tuned the range of the fuzzy regions such that most of these output falls within desirable fuzzy regions. Note that we use the time-average filter only for tuning the fuzzy regions. Once the fuzzy regions are fixed, the filter will be removed.

We feel that it is improper to divide the output spaces of the hybrid neural fuzzy system into fuzzy regions that are arranged in a hierarchical fashion (such as "slow, medium, fast" or "cold, warm, hot" etc.). This is because any particular ECG signal may have some feature that is common to all the other ECG signals. Each fuzzy region should overlap with all the other fuzzy regions as shown by the modified membership functions in Figure 6.4.

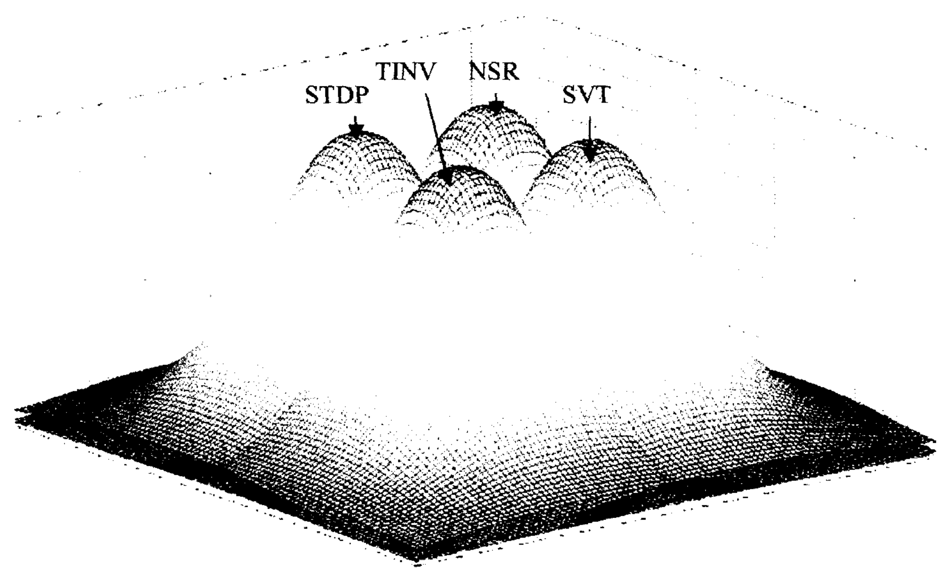

Figure 6.4: Divisions of output spaces into modified fuzzy regions and the corresponding membership functions.