An Algorithmic Approach for Fuzzy Inference

C. J. Kim, *Member, IEEE*

***Abstract*— To apply fuzzy logic, two major tasks need to be performed: the derivation of production rules and the determination of membership functions. These tasks are often difficult and time consuming. This paper presents an algorithmic method for generating membership functions and fuzzy production rules; the method includes an entropy minimization for screening analog values. Membership functions are derived by partitioning the variables into the desired number of fuzzy terms and production rules are obtained from minimum entropy clustering decisions. In the rule derivation process, rule weights are also calculated. This algorithmic approach alleviates many problems in the application of fuzzy logic to binary classification.**

***Index Terms*—Algorithms, clustering, entropy, fuzzy logic, inference.**

I. Introduction

FUZZY logic has been applied with reasonable success to many control problems for which only conventional control methods had previously been utilized. In such control problems, the value of fuzzy logic is that vague meanings and relationships, expressed in ordinary language, can be effectively formulated. The fuzzy inference procedure includes the translation of an analog value into membership grades, which are defined by the membership function of fuzzy terms.

Although fuzzy logic theory was introduced in the 1960's, its application to industrial control emerged in the early 1970's with a procedure for the control of a steam engine [1]; since then, fuzzy logic has been applied in other control areas. Currently, fuzzy logic is involved in many industrial and commercial applications, even in home appliances. To apply fuzzy logic, we must define fuzzy production rules, fuzzy terms, and membership functions. It is often difficult and time-consuming to derive these rules and membership functions. By devising an automatic procedure for deriving membership functions and production rules, therefore, we can make fuzzy logic applications much easier to produce. Advanced applications (such as learning fuzzy control) need an adaptive method of representing fuzzy knowledge, so an attempt to automate fuzzy logic applications is a timely response to an important subject.

An estimation model for fuzzy membership functions was introduced using fuzzy ensemble membership apportionment learning estimators [2]. However, this model estimates only membership functions and does not produce fuzzy rules. Recently, a "table-lookup" scheme for fuzzy rule generation for numerical input–output pairs was suggested [3]. This scheme, which aims to extract a rule for each input–output pair, however, determines the partitions of the domain interval and membership functions in an *ad hoc* manner. Artificial intelligence (AI) and neural network techniques have also been applied to extract fuzzy rules from numerical data, however, they require that the number of divisions in the input variable be defined in advance [4].

Clearly, an automatic process which can generate both membership functions and production rules directly from experienced sample data would be of considerably more value. The primary objective of this paper is to develop an algorithm which is capable of automating fuzzy logic applications in binary classification and decision-making problems. Using an algorithmic approach that utilizes the concept of entropy minimization, membership functions are generated and, based upon them, fuzzy production rules and rule weights can be determined. The rule weight devised in this paper, unlike the "relative weight" used for medicine and biology [5], is assigned by an algorithmic process.

II. Algorithmic Application of Fuzzy Logic

In fuzzy logic applications, membership functions, usually of triangular or trapezoidal shape, have typically been determined by human experts. Accordingly, experience and common sense are the two leading guidelines for determining the membership functions. Similarly, fuzzy production rules have been devised from expert opinion. The fundamental problem with this approach is that the production rules derived by the expert using experience and common sense are not always the most suitable ones for an automatic controller. Furthermore, there is no way to assess whether or not a rule correctly and optimally represents most of the experienced sample data. We propose to develop an algorithmic approach which, without human intervention, can be utilized universally for fuzzy logic applications in classification problem. Guided by a theorem of maximum information extraction, this algorithmic approach generates membership functions and fuzzy production rules from experienced sample data.

The automation of membership function derivation can be considered as an attempt to draw a structured *linguistic variable* in which the fuzzy terms and their meanings can be characterized by an algorithm. One of the basic tools for fuzzy logic is the linguistic variable, i.e., a variable whose values are not numbers, but words in a natural or artificial language [6]. A linguistic variable is characterized by a quintuple $(r, T(r), U, G, M)$ where

Manuscript received March 8, 1994; revised March 26, 1997.

The author is with the Department of Electrical Engineering, University of Suwon, Suwon, Kyung-Gi, 445–743 Korea.

Publisher Item Identifier S 1063-6706(97)07506-1.

r name of the variable, its "label," or sometimes its value R;
$T(r)$ term set of r; that is, the set of names in r;
U range of $T(r)$;
G a syntactic rule for generating R, the values of r;
M a semantic rule for associating each R with its meaning.

A particular R (that is, a name generated by G) is called a *term*. For example, if a linguistic variable r is defined with the label "age" in $U = [0, 100]$, then the terms of this linguistic variable, generated by the rule $G(r)$, could be called "old," "middle," "young," and so on. Therefore, $T(r)$ defines the term set of the variable r with $T(\text{age}) = \{\text{old, middle, young}\}$. $M(r)$ is the rule that assigns meanings to these terms. A linguistic variable r is called *structured* if the term set $T(r)$ and the meaning $M(r)$ can be characterized algorithmically. For a structured variable, $M(r)$ and $T(r)$ can be regarded as algorithms, which generates the terms and the meanings associated with them.

The above description of a linguistic variable can be rephrased as follows: $G(r)$ determines the fuzzy terms from a variable and $M(r)$ determines the membership functions of the fuzzy terms. Once the number of fuzzy terms is decided, the only unknown item in the linguistic variable is the rule $M(r)$. The algorithmic approach in this paper will decide the rule for membership function formation; in a theoretical sense, therefore, this paper attempts to draw a structured linguistic variable.

In industrial control application of fuzzy logic, a set of terms drawn from linguistic variables have been used to describe the states of the process. In particular, the error value and the change in error value are quantized into a number of points covering the range in U and the values are then assigned as grades of membership in seven subsets [7], [8]. The following seven-term set seems to be an industry standard for fuzzy logic applications: positive big (PB); positive medium (PM); positive small (PS); zero (ZE); negative small (NS); negative medium (NM); and negative big (NB). Therefore, if we devise an algorithm for assigning membership grades to the standard terms, we can apply our scheme to any application area of fuzzy logic. In addition, this approach provides an automatic mechanism for generating fuzzy production rules from the term set $T(r)$ and the meaning $M(r)$.

III. Entropy Principle in Classification

The main idea behind the automatic generation of membership functions and production rules is the concept of analog value screening. Using the entropy principle, the analog values of a parameter in the sample data can be clustered. Optimal division of the analog space will yield fuzzy terms for each parameter; the partition point (the entropy minimum point) will decide the range of the membership functions. Using the same screening method, but with binary parameter values, fuzzy production rules can be drawn. Because the rule extraction process is performed over each individual fuzzy term, the final production rule will consist of the integration of independent rules.

We first discuss the entropy concept relative to the classification of two-class ("true" and "false") samples. When we look at the samples in the "true" class, for example, we try to discover what it is that makes them "true." In other words, we try to find similarities among the parameters for "true" cases, which distinguish them from samples which are "false." This means that we try to find attributes or groups of attributes possessed by "true" samples and not by "false" samples. These attributes or groups of attributes then become part of the boundary separating the "true" samples from the "false" samples. To optimally separate "true" and "false" samples, we usually use a measure of information. The quantity of information gain or loss is a basic element for entropy calculation for analog screening.

The main purpose of entropy minimization analysis in information theory is to determine the gain or loss of information in a given data set. This information quantity compares the contents of available data to some prior state of expectation. The higher one's prior estimate of the probability for an outcome, the lower the information gained by observing its occurrence. In general, on the basis of what we already know, the more probable the event is, the lesser the information content is if and when the event occurs. In other words, when information gain is minimized, we reach (at an optimal point) for predicting the occurrence. A quantity of information is defined as proportional to the negative of the logarithm of probability [9].

If we assume that the probability that the ith sample x_i is true is $P(x_i)$ and if we actually observe the sample x_i in the future and discover that it is true, then we gain the following information:

$$I(x_i) = -k \ln P(x_i).$$

If we discover that it is false, on the other hand, we still gain the following information:

$$I(\sim x_i) = -k \ln[1 - P(x_i)].$$

Entropy is defined as the expected value of information. The entropy of a set of possible outcomes of a trial in which one and only one outcome is true is expressed as the summation of the products of all probabilities and their logarithms. Thus, the expected value of the information to be gained by observing x_i can be expressed as follows (with $P_i = P(x_i)$):

$$S(x_i, \sim x_i) = -k[P_i \ln P_i + (1 - P_i) \ln(1 - P_i)].$$

The entropy of all the samples (N) is expressed by

$$S = -k \sum_{i=1}^{N} [P_i \ln P_i + (1 - P_i) \ln(1 - P_i)]. \tag{1}$$

This entropy is smallest when the amount of information that we can expect to gain from further observation is least. Therefore, given all available information, it is possible to cluster using the minimum entropy principle. In entropy minimum state, all of the information has been extracted from the available sample data. This observation is very important to the algorithmic approach: when samples are the only source of information, maximum extraction of information is

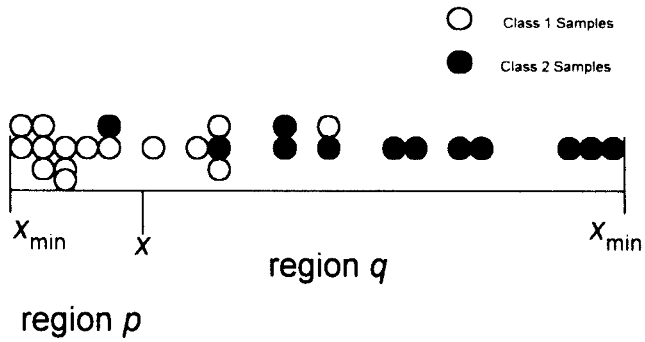

Fig. 1. Illustration of threshold value calculation.

essential for an automated process. Therefore, in classification problems, the entropy principle is a useful tool for optimal clustering. In the next section, we will discuss the basis for this algorithmic approach, viz., analog screening.

IV. Analog Screening with Entropy Principle

The clustering point in samples is called a threshold value between classes. If we divide once-clustered samples, again using the same entropy principle, we can subcluster the samples. The thresholds optimally divide the sample space; the divided regions will yield the fuzzy terms. Membership functions are shaped from the thresholds. To draw fuzzy production rules, minimum entropy clustering with changes in the entropy equation can be applied. To begin, we consider the entropy equation for sample clustering.

Assume that we are seeking a threshold value for samples in the range $x_{\min}$ to $x_{\max}$ for a two-class problem (see Fig. 1). By moving an imaginary threshold value x between $x_{\min}$ and $x_{\max}$ we can calculate the entropy for each value of x for region $p[x_{\min}, x]$ and region $q[x, x_{\max}]$, which is [10]

$$S(x) = p(x)S_p(x) + q(x)S_q(x) \tag{2}$$

where

$p(x)$ fraction of all samples in the p region;
$q(x)$ fraction of all samples in the q region;
$p(x) + q(x) = 1$.

Entropies of the p and q regions $S_p(x)$ and $S_q(x)$ can be expressed by [cf. (1)]

$$S_p(x) = -(p_1(x)\ln p_1(x) + p_2(x)\ln p_2(x)) \tag{3}$$
$$S_q(x) = -(q_1(x)\ln q_1(x) + q_2(x)\ln q_2(x)) \tag{4}$$

where

$p_k(x)$ probability that the Class k sample is in the region p;
$q_k(x)$ corresponding conditional probability for the region q.

We calculate the entropies of (3) and (4) using relatively unbiased estimates of $p_k(x)$, $q_k(x)$, $p(x)$, and $q(x)$. The relatively unbiased estimates for $p_k(x)$ and $p(x)$ are

$$P_k(x) = \frac{n_k(x) + w(x)}{n(x) + w(x)} \tag{5}$$
$$P(x) = \frac{n(x) + v(x)}{n + v(x)} \tag{6}$$

where

$n_k(x)$ number of Class k samples located in the region p;
$n(x)$ total number of samples located in the region p;
n total number of samples in the p and q regions.

The variables w and v are *a priori* weights; both are set to one, which permits the simplification

$$P_k(x) = \frac{n_k(x) + 1}{n(x) + 1} \tag{7}$$
$$P(x) = \frac{n(x) + 1}{n + 1}. \tag{8}$$

Equations for $q_k(x)$ and $q(x)$ can be derived similarly. Using the estimates and the entropy equation, we calculate the entropy for each value of x. A value of x whose entropy is the minimum, $X = S_{\min}(x_{\min}, x_{\max})$ is the optimal threshold value in the range of $[x_{\min}, x_{\max}]$.

Along with the entropy calculation, there is the problem of assigning a probability in cases where only one digit (or variable) has been observed "true" on z of n occasions. What makes it difficult to assign a probability is the feeling that what is observed is more likely than what is not and that what is observed more often is more likely than what is observed less often. This probability can be expressed as

$$P = \lim_{n\to\infty} \frac{z}{n}.$$

As n becomes larger and larger, z/n comes closer and closer to P. But it is not clear in what sense z/n is approaching a limit, which we presume to exist and call P. In such cases, it is possible to use the mean probability $\bar{P}$ to represent P. Mean probability in the class separation is defined by [10]

$$\bar{P} = \frac{z + t}{n + t + f} \tag{9}$$

where

t number of distinguishable "true" states;
f number of distinguishable "false" states.

This mean probability when there are only two classes $(t = 1$ and $f = 1)$ becomes

$$\bar{P} = \frac{z + 1}{n + 2}. \tag{10}$$

The mean probability is used in the entropy equation for production rule derivation and in rule weight calculation. The process for the analog screening of threshold values for membership functions and the production rule derivation will be explored in detail in the next section.

V. Membership Function and Production Rule Generation

A. Membership Function Generation

Using the entropy equations (3) and (4) with the estimates given by (7) and (8), we calculate the entropy for all the x's. The value of x, which yields the minimum entropy, is taken to be the threshold value (X) of the two partitions. We call this the first threshold and indicate by X_{11}. This threshold value is calculated in the range of $x_{\min}$ and $x_{\max}$. If we replace the variables $x_{\min}$ and $x_{\max}$ by X_{01} and X_{02}, respectively, then we can indicate X_{11} by $X_{11} = S_{\min}(X_{01}, X_{02})$. With only one threshold value, there can be two nonoverlapping fuzzy terms with rectangular-shaped membership functions [see Fig. 2(a)]

$[X_{01}, X_{11}]$: NG (negative)
$[X_{11}, X_{02}]$: PO (positive).

We can draw another threshold line to subdivide each side more precisely. Using the same procedure for entropy calculation, we can compute secondary threshold values from the positive and negative sides (as shown below)

$$X_{21} = S_{\min}(X_{01}, X_{11})$$
$$X_{22} = S_{\min}(X_{11}, X_{02}).$$

Assuming trapezoid shapes at the both ends with threshold values calculated above, we now have three terms: PO, ZE (zero), and NG [see Fig. 2(b)]

$[X_{01}, X_{11}]$: NG
$[X_{21}, X_{22}]$: ZE
$[X_{11}, X_{02}]$: PO.

To generate the seven fuzzy terms (the final partition), we need one more level of thresholds. We can calculate four tertiary threshold values; each of them separates the three terms more precisely. The third level threshold values are

$$X_{31} = S_{\min}(X_{01}, X_{21})$$
$$X_{32} = S_{\min}(X_{21}, X_{11})$$
$$X_{33} = S_{\min}(X_{11}, X_{22})$$
$$X_{34} = S_{\min}(X_{22}, X_{02}).$$

Again, assuming trapezoidal shapes at both ends, we have a total of seven membership functions arrived at by mechanically connecting the threshold positions, as shown in Fig. 2(c). We label them (from the left) NB, NM, NS, ZE, PS, PM, and PB. These relationships, up to the fourth level, are illustrated in Fig. 3.

Therefore, up to the fourth level we can draw the following general threshold formula for the kth threshold in the nth level X_{nk}. For $k = 1, \cdots, 2^{(n-2)}$:

- if $k = 1, X_{nk} = S_{\min}(X_{01}, X_{(n-1)k})$;
- if $k = 2^{(n-2)}, X_{nk} = S_{\min}(X_{(n-1)(k/2)}, X_{11})$;
- otherwise, $X_{nk} = S_{\min}(X_{(n-1)l}, X_{(n-1)m})$, where $k = l + m$.

For $k = 2^{(n-2)} + 1, \cdots, 2^{(n-1)}$, with $n > 1$:

- if $k = 2^{(n-2)} + 1, X_{nk} = S_{\min}(X_{11}, X_{(n-1)(k-1)})$;
- if $k = 2^{(n-1)}, X_{nk} = S_{\min}(X_{(n-1)(k/2)}, X_{02})$;
- otherwise, $X_{nk} = S_{\min}(X_{(n-1)l}, X_{(n-2)m})$, where $k = l + m + 1$.

Also, we may draw the following relationship between the threshold level and the number of fuzzy terms (and membership functions):

$$L = 2^n - 1 \quad (n > 1) \tag{11}$$

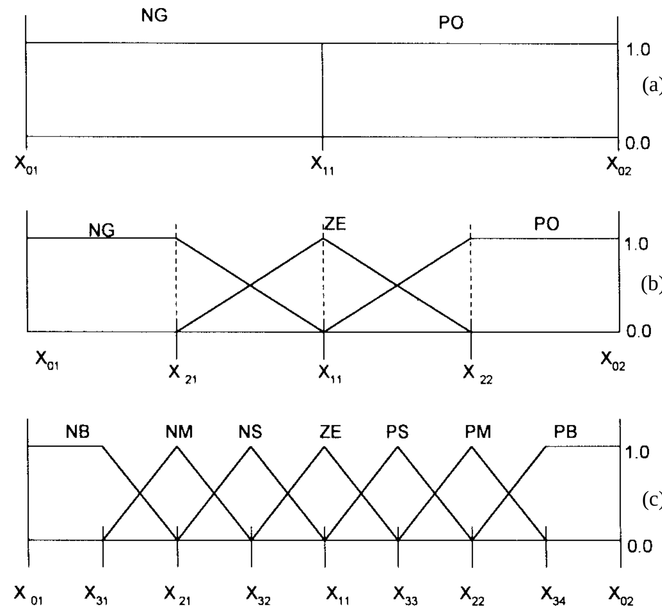

Fig. 2. Repeated partitions and corresponding fuzzy terms. (a) The first partition. (b) The second partition. (c) The third partition.

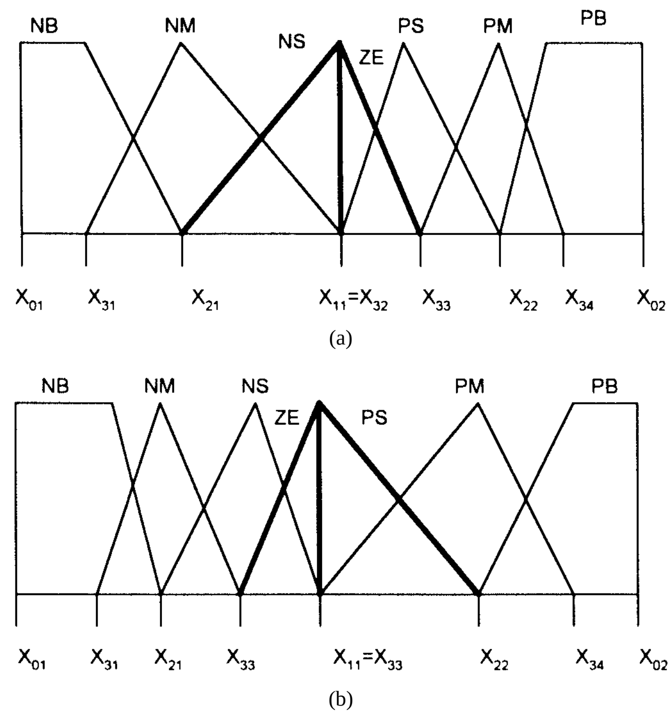

Fig. 3. Illustration of membership functions when two threshold values are identical. (a) $X_{11} = X_{32}$ case. (b) $X_{11} = X_{33}$ case.

where

L number of fuzzy terms;
n threshold level (primary, secondary, tertiary, etc.).

To generate the "universal" seven fuzzy terms, we always go down to the third threshold level. However, for well-separated samples, three levels of threshold calculation may cause an over partition of the sample space. With our algorithmic generation of membership functions, it is possible for two thresholds to share the same value. In this case, samples are

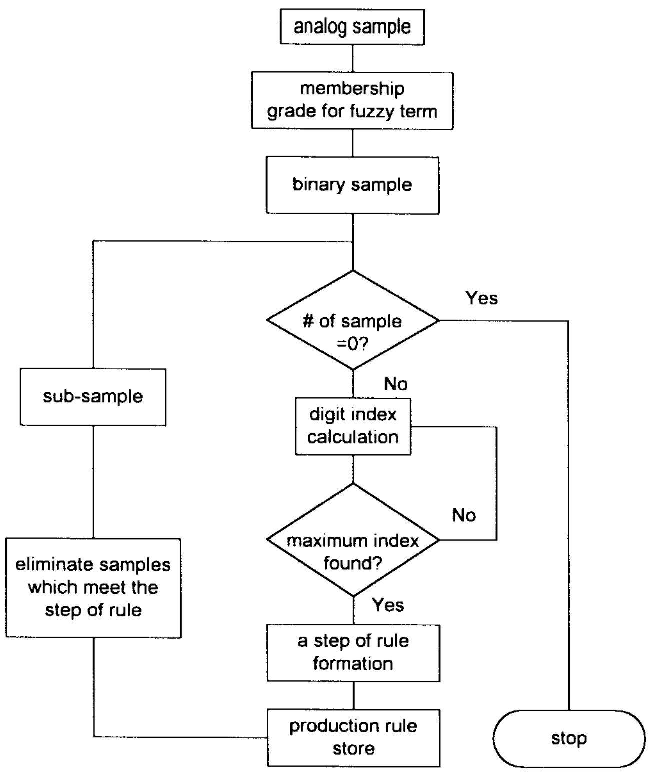

Fig. 4. Flow chart for production rule derivation.

well classified with the primary and secondary thresholds alone and further clustering is redundant for separation purposes. However, this situation does not cause a problem, as illustrated in Fig. 4. The fuzzy membership functions are drawn as before, but with a slight change in their shape. The thick lines in Fig. 4(a) cover NS and ZE in the case of $X_{11} = X_{32}$ and, in Fig. 4(b), ZE and PS for $X_{11} = X_{33}$.

These membership functions are not realistic and must be interpreted with care. In other words, the membership functions do not give a true picture of the real situation. ZE, for instance, does not correspond to the true zero value of the input; reality and expert opinion are totally ignored for the determination of the membership functions. This artificiality, however, does not lead to any problem for rule generation or inference. In reality, membership functions are meaningful only when they accurately represent the sample data from which the production rules are derived.

B. Fuzzy Production Rule Generation

Fuzzy production rules relate input and output variables. The production rules for two class problems will be generated from the acquired seven fuzzy terms using the entropy principle. Since the entropy minimization principle has been proven effective for decision rule derivation with binary values [9], we will transform the analog values of the samples into binary values. This binary transformation will be performed separately for each fuzzy term from NB to PB. If the analog value has the highest membership grade in a term, it will be assigned value one; if not, it will be assigned value zero. This procedure produces, therefore, seven sets of binary data. Rule derivation is performed for each term and, for each term, each step of the rule is derived with an association rule weight, which gives the reliability of the rule.

The rules for each fuzzy term resemble a decision tree in which the branch points indicate the divided search route. The rule for each fuzzy term has the form "If $\cdots$, else if $\cdots$, else if $\cdots$, end if." Therefore, a general production rule from sample data will appear as the one shown in Appendix F. There is a rule for each fuzzy term and each rule of the "if $\cdots$, else if $\cdots$, $\cdots$, end" form allows only one fuzzy term. This rule formation is somewhat different from the conventional one, which may include as many fuzzy terms as chosen (seven in our case) in a single rule.

Membership function generation is a result of a repeated calculation of thresholds for optimal separation and production rule generation will also choose an optimal rule from numerous candidates. The entropy of a production rule for a fuzzy term, using the mean probability of (9) and (10), is

$$S = -k \sum_{i=1}^{m} y_i \bar{p}_i \ln \bar{p}_i \tag{12}$$

where

m total number of steps, i.e., the total number of separating points in the decision rule;

y_i number of samples covered by step i of all possible samples;

k a constant.

Theoretically, therefore, we check all possible rules and calculate their entropy to select a rule whose entropy is smallest. As can be seen from (12), however, there are simply too many combinations of variables or rules to check. For an n digit binary number, there are 2^n combinations; if we have, for example, 15 digit binary numbers, there are $2^{15} = 32\,768$ different ways. Even allowing that we usually do not have that many samples, we still have too many combinations to investigate. If we have 68 samples of 15 digit binary numbers (the same number of samples used in the example in Section VII), for instance, the actual number of ways is $(68)^2 = 4624$. A practical way to apply the entropy principle for production rule derivation is obviously needed.

To simplify the evaluation production rules, we tried to find some easily derivable relations among the entropy equations. From (12), we can see that the closer $\bar{p}$ is to one or zero, the smaller the entropy S is. Also, from (9) and (10), it is apparent that the bigger z is, the bigger $\bar{p}$ is. Therefore, if we can find the biggest z, we can find the rule with minimum entropy; to do this, we will use the concept of *digit index*. The digit index is defined as the ratio of correct separation of two classes using only a single digit (or feature). In other words, the index is a measure which locates the digit that assures the minimum number of wrong classifications.

For digit index determination, we first calculate a quantity called the *digit count*. We count the number of one's in the Class 1 samples and the number of zero's in the Class 2 samples, or vice versa. Then we divide each number by the total number of samples in each class. The result is the so-called digit count d. If we have n digits (or features), then,

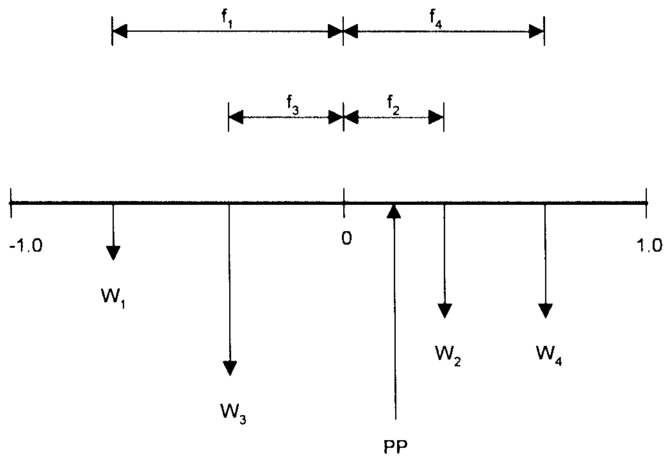

Fig. 5. Illustration of the pivot and balance defuzzification.

TABLE I
PRODUCTION RULE SAMPLE DATA

sample #	variable 1	variable 2	variable 3	class 1
1	0.210	1.477	2.420	1
2	0.180	1.435	5.012	1
3	0.203	1.184	5.245	1
4	0.106	1.154	6.012	1
5	0.202	1.057	7.034	1
6	0.185	-0.673	4.992	1
7	-0.170	4.628	3.420	2
8	0.724	1.114	5.940	2
9	0.035	3.944	5.120	2
10	0.167	4.262	3.420	2
11	0.169	4.000	6.011	2
12	0.045	1.251	5.093	2
13	0.017	3.904	9.024	2
14	-0.001	4.703	4.062	2
15	-0.118	4.640	5.872	2

by this calculation, we can have n digit counts; d_n is the digit count for the digit n. Next, we add all the digit counts of each class. If this value is close to 1.0, that digit (or feature) is not important for separation: the one's and zero's have the same weight in both classes. If the value is not close to 1.0, there are less one's or zero's in a class. We formalize this idea by defining *digit* (or *feature*) *index* as follows:

$$I_n = \left|\sum d_n - 1.0\right|. \tag{13}$$

To generate production rules, we use the digit (or feature) whose index is the maximum. Then, we apply the rule to the samples and eliminate those samples which satisfy the rule. For example, if we chose the rule as "$1\times\times$ for Class 1" ($\times$ indicates "don't care"), then we delete all the samples (of both classes) whose value for the first digit is one. We repeat the following sequence until all the samples are accounted for: digit index calculation; formation of the rule; elimination of the samples which satisfy the rule. Fig. 5 shows a flow chart of this simplified generation procedure.

C. Production Rule and Rule Weight

Procedures which include a mix of fuzzy logic and neural networks have been developed to provide adaptability to fuzzy logic applications. A crucial aspect of adaptive fuzzy logic is centered on the change of the shape of membership functions or the determination of rule weights [11]. The determination of rule weights will be discussed below in conjunction with a discussion of production rule derivation.

The production rule generated by the simplified procedure from sample data is optimal but not perfect; the rule inevitably yields incorrect separation at each step. At each step, therefore, we calculate the reliability or weight using the mean probability of the two-case problem given by (10). As an illustration, we discuss an example adopted from [9] and changed to simulate an imaginary fault identification situation. We have fifteen three-variable samples to be classified into two classes. We will derive a production rule for fuzzy term PS and the region of the terms PS is assumed to be $[0.165, 0.212]$ for the first variable, $[3.877, 4.774]$ for the second, and $[4.890, 6.036]$ for the third. The sample data are shown in Table I.

For binary conversion, the samples in each variable are translated into binary values (D's) depending whether they are the members of the term PS. If the sample values are within the range of the PS, they are translated to 1, otherwise, 0. Thus, the following binary table for the fuzzy term PS will be resulted as indicated in Table II.

TABLE II
BINARY TABLE FOR THE FUZZY TERM PS

sample #	D1	D2	D3	class
1	1	0	0	1
2	1	0	1	1
3	1	0	1	1
4	0	0	1	1
5	1	0	0	1
6	1	0	1	1
7	0	1	0	2
8	0	0	1	2
9	0	1	1	2
10	1	1	0	2
11	1	1	1	2
12	0	0	1	2
13	0	1	0	2
14	0	1	0	2
15	0	1	1	2

TABLE III
PROCESS OF DRAWING THE DIGIT INDEX

	D1		D2		D3	
	Class1	Class2	Class1	Class2	Class1	Class2
# of 1/0	5	7	0	2	4	4
digit count	5/6 + 7/9		0/6 + 2/9		4/6 + 4/9	
digit index	0.61		0.78		0.11	

TABLE IV

sample #	D1	D2	D3	Class
1	1	0	0	1
2	1	0	1	1
3	1	0	1	1
4	0	0	1	1
5	1	0	0	1
6	1	0	1	1
8	0	0	1	2
12	0	0	1	2

Now, we follow the steps to produce rules for the fuzzy term PS.

Step 1: First, we find the digit count and the digit index. For the samples of each digit (D1, D2, and D3), we count the number of 1's in the Class 1 samples and the number of 0's in the Class 2 samples. Then we divide the number of 1's by the number of samples in Class 1, and the number of 0's by the number of samples in Class 2. The process of drawing the digit index is shown in Table III.

From the above process, the second digit has the largest digit index, so we start the separation process with the second digit. Therefore, we have "×1× for Class 2" as the first step of the rule. Then $z_1 = 7$ (from Class 2) and $n_1 = 7$ (from both classes) and the mean probability, using (10), is $\bar{P}_i = (z_1 + 1)/(n_1 + 2) = 8/9 = 0.89$. This value is taken to be the weight of the first rule step. Eliminating the samples having 1 for their second variable, we have Table IV.

Step 2: The process of drawing the digit index is shown in Table V.

We see that the digit index is biggest for the first variable so "1×× for Class 1" becomes the second step of the rule. Then $z_2 = 5$ and $n_2 = 5$ and, therefore, the weight of the step 2 of the rule is $\bar{P}_2 = 6/7 = 0.86$. The remaining samples are shown in Table VI.

Step 3: The process of drawing the digit index is shown in Table VII.

As we see that all the digit indexes are same, we can choose "××1 for Class 2." Then, $z_3 = 2$ and $n_3 = 3$ and, therefore, the rule weight for step 3 of the rule is $\bar{P}_3 = 3/5 = 0.6$.

After the third step, rule derivation stops. The production rule for fuzzy term PS consists of three steps. This fuzzy production rule can be written as shown at the bottom of the page.

The above procedure must be performed for each of the other six fuzzy terms; the final production rule will be a set of seven independent production rules.

TABLE V

	D1		D2		D3	
	Class1	Class2	Class1	Class2	Class1	Class2
# of 1/0	5	2	0	2	4	0
digit count	5/6 + 2/2		0/6 + 2/2		4/6 + 0/2	
digit index	0.83		0.00		0.33	

TABLE VI
REMAINING SAMPLES

sample #	D1	D2	D3	Class
4	0	0	1	1
8	0	0	1	2
12	0	0	1	2

TABLE VII

	D1		D2		D3	
	Class1	Class2	Class1	Class2	Class1	Class2
# of 1/0	0	2	0	2	1	0
digit count	0/1 + 2/2		0/1 + 2/2		1/1 + 0/2	
digit index	0.00		0.00		0.00	

VI. INFERENCE AND DEFUZZIFICATION FOR ALGORITHMIC APPROACH

So far, we have discussed the algorithmic procedure for membership function and production rule generation. However, this algorithmic approach is not complete unless suitable inference and defuzzification methods can also be provided.

A. Inference

Inference is a mechanism by means of which a conclusion is drawn from sample data and production rules. It is designed to evaluate the rules whose conditional parts are satisfied. A popular inference method is "max" in which the final membership grade for an output is the union of the fuzzy membership grades which are the outputs of the individual production rules. The values of the membership grades are determined by the degrees of membership in the conditional part of the rules [6]. If OR is used to form the conditional part, the grade value is determined by the maximum of the membership grades; if AND is used, it is determined by the minimum of the grades. However, this inference method does not take into consideration cases where the rules are assigned weights; to accommodate production rules with rule weights, therefore, a new inference method is required.

Two methods of inference will be described. Either method is applied to each of the seven sets of production rules—one for each fuzzy term. The following explanation applies to all the fuzzy terms and corresponding production rules. The first method is to check for a matched (nonzero) premise from a first fuzzy term, for example, PB. If a step (each step has the fuzzy term PB in the conditional part) of the rule is matched, then the firing strength, the corresponding weight, and the class identification for the fuzzy term PB $(f_{\mathrm{PB}}, W_{\mathrm{PB}}, C_{\mathrm{PB}})$ are recorded. Then, we move to the next fuzzy term, PM, for example. The process is applied to all the fuzzy terms. Therefore, this method yields seven set of firing strength, corresponding weights, and class identification represented by $\{(f_j, W_j, C_j),\ j = \mathrm{PB}, \mathrm{PM}, \mathrm{PS}, \mathrm{ZE}, \mathrm{NS}, \mathrm{NM}, \mathrm{NB}\}$. This method is called the "overall match" method and is illustrated in Fig. 6(a).

The other inference method is called the "round match" method because it is based on a "round" which checks each step of each fuzzy term rule. Unlike the "overall match" method, this method does not check all the steps of a fuzzy term rule. Instead, this method checks the first step of the first term rule, and then the first step of the second fuzzy term

```
IF variable 2 is PS, THEN Class 2                  (weight = 0.89)   (step 1)
   ELSE IF variable 1 is PS, THEN Class 1          (weight = 0.86)   (step 2)
      ELSE IF variable 3 is PS, THEN Class 2       (weight = 0.60)   (step 3)
ENDIF
```

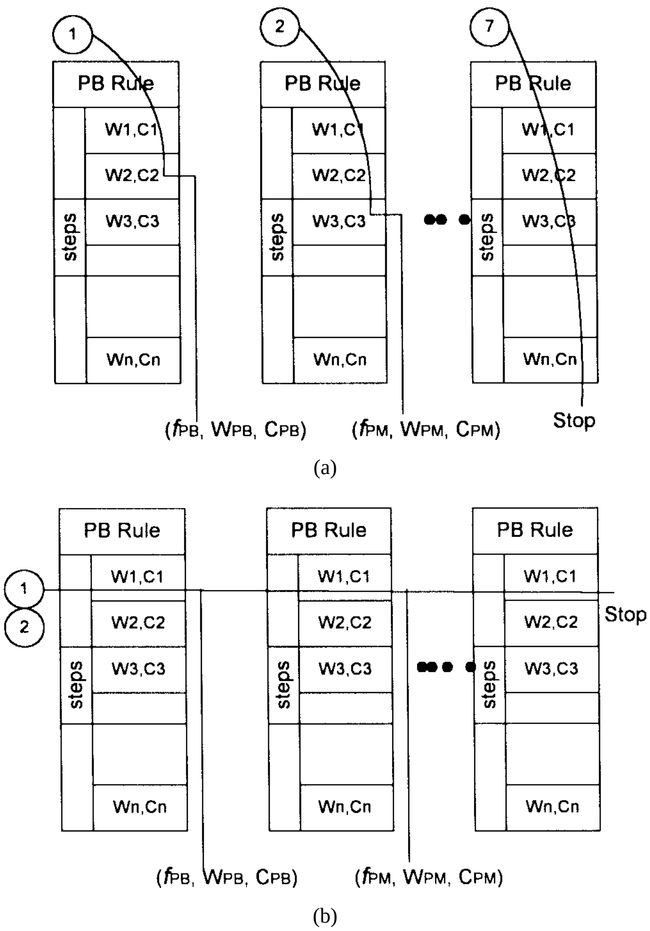

Fig. 6. Match and fire process for (a) overall match and (b) round match method.

rule, and then the first step of the third fuzzy term rule, and so on. Depending upon the result of the match at each step, the round will either continue or stop. This means that if, for example, the first steps of any one or more fuzzy rules (for example PB rule and PM rule) are matched, the process stops. Again, the firing strength, the corresponding weight, the class identification $(f_{\rm PB}, W_{\rm PB}, C_{\rm PB})$, and $(f_{\rm PM}, W_{\rm PM}, C_{\rm PM})$ for term PB and PM, respectively, will result. If we do not have any matched set in the first round of the fuzzy term rules, we move to the second round. This process will go on until there is a matched set or all rounds are finished. The process is illustrated in Fig. 6(b).

B. Defuzzification

Usually, more than one fuzzy rule may be matched and fired at one time, so there should be a conflict resolution measure. This output decoding method is called defuzzification. Defuzzification is the process of converting the result of the inference into a nonfuzzy value which best represents the membership function of an inferred fuzzy classification actor. One of the most famous method of defuzzification is center of area method which can be represented by

$$x_d = \frac{\sum_{i=1}^{m} x_i f_k(x_i)}{\sum_{i=1}^{m} f_k(x_i)} \tag{14}$$

TABLE VIII
TEST RESULTS (pp WITH TWO METHODS)

no.	class	by overall-match	by round-match	rounds
2	fault	-0.179	1.000	1
3	fault	0.401	0.687	1
5	fault	0.699	0.642	1
8	fault	-0.196	0.373	1
9	fault	0.460	0.670	1
10	fault	0.229	0.691	1
11	fault	0.605	0.681	1
12	fault	0.299	0.478	1
14	fault	-0.316	-0.092	1
15	fault	0.361	0.281	1
16	fault	0.187	0.148	1
20	fault	0.473	0.319	1
21	fault	0.465	0.282	1
28	not-fault	-0.621	-0.390	1
31	not-fault	-0.613	-0.350	1
35	not-fault	-0.676	-0.467	1
38	not-fault	-0.810	-0.482	1
44	not-fault	-0.608	-0.354	1
45	not-fault	-0.674	-0.451	1
46	not-fault	-0.810	-0.482	1
47	not-fault	-0.327	-0.169	1
54	not-fault	-0.600	-0.237	1
56	not-fault	-0.728	-0.487	1
58	not-fault	-0.533	-0.298	1
59	not-fault	-0.701	-0.487	1
60	not-fault	-0.761	-0.485	1
61	not-fault	-0.707	-0.482	1
62	not-fault	-0.701	-0.293	1
63	not-fault	-0.361	-0.169	1
64	not-fault	-0.417	-0.169	1
65	not-fault	-0.373	-0.169	1
66	not-fault	-0.446	-0.183	1
67	not-fault	-0.347	-0.286	1
68	not-fault	-0.714	-0.275	1

where

m	number of quantized levels of variable;
x_i	value of a variable at the quantized level i;
$f_k(x_i)$	membership degree of fuzzy term k at the value x_i;
x_d	fuzzified value.

The other popular method is the mean of the maximum method which can be represented by

$$x_d = \sum_{i=1}^{l} \frac{x_i}{l} \tag{15}$$

where l is the number of quantized x values which reach their maximum membership degrees.

For classification problems and for systems with rule weights, however, the conventional defuzzification method is not appropriate. Due to the nature of the problem, the output of the classification process should not be an analog value but a binary value, i.e., the output is not a quantity but a discrete status [13], [14]. This unique characteristic of classification problem and the introduction of rule weights,

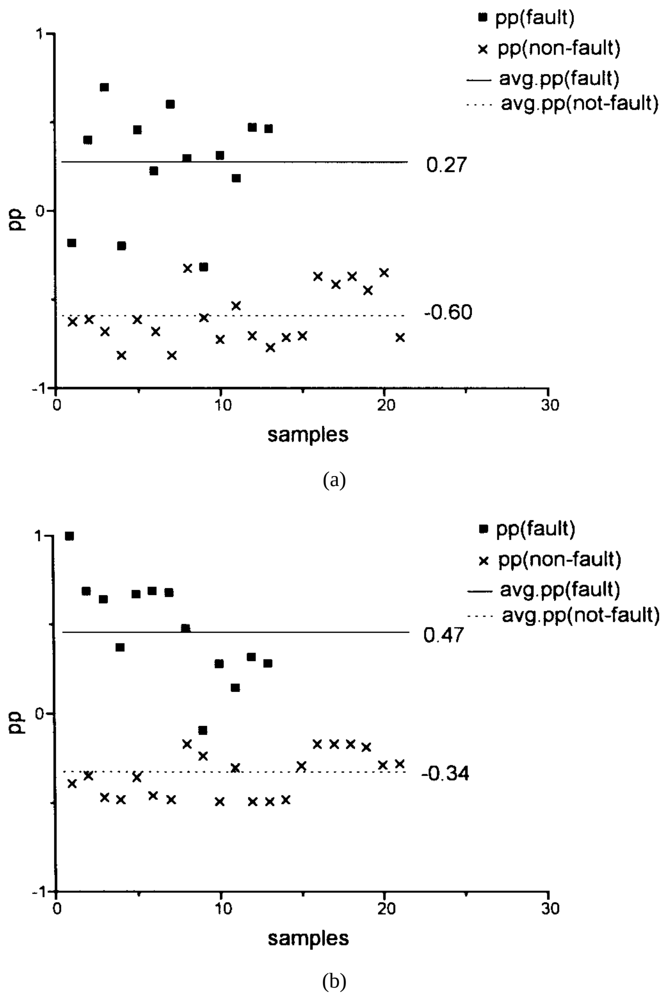

Fig. 7. Comparison of two match methods.

therefore, require the development of a new defuzzification method, which we shall call the "lever and pivot" method.

The lever and pivot concept finds solutions in the unique environment of the classification problem: binary output and multiple sets of firing strength, weight, and class identification. The basic idea of this method is to place the "weights" in the location designated by the firing "length" and then to move the pivot to the position which balances the lever (see Fig. 7). A firing strength determines the distance from the center of the lever and the weight of the step of the fired rule acts as a measuring weight. We place the weights on the left side of the lever if the class identification $C_k = 2$ and on the right side if $C_k = 1$. If we scale the lever so that it is centered on zero, Class 1 includes all points on the right side of the pivot point *pp*, and Class 2 includes all points on the negative side. Therefore, the sign of the final defuzzified output, the pivot point *pp*, decides the class of the sample: Class 1 if *pp* is positive and Class 2 if negative.

The pivot point pp for defuzzification, therefore, can be expressed by the following:

$$pp = \frac{\sum_{C_1} f_n W_n - \sum_{C_2} f_m W_m}{\sum_{C_1 \,\&\, C_2} W_l} \qquad (16)$$

where

f_k firing strength of the matched rule of the term k $k = \mathrm{PB}, \mathrm{PM}, \cdots, \mathrm{NB}$;

W_k weight of the matched step of the fuzzy rule of the term k $k = \mathrm{PB}, \mathrm{PM}, \cdots, \mathrm{NB}$;

C_k class identification of the fuzzy rule of the term k 1 for Class 1 and 2 for Class 2.

Appendix A
The 15 Variables

no	variable	Remarks
1	m(odd)	mean of odd harmonic current
2	m(odd)	mean of absolute odd harmonic current
3	s(odd)	standard deviation of odd harmonic current
4	m(eve)	mean of even harmonic current
5	m(eve)	mean of absolute even harmonic current
6	s(eve)	standard deviation of even harmonic current
7	m(sub)	mean of sub-harmonic current
8	m(sub)	mean of absolute sub-harmonic current
9	s(sub)	standard deviation of sub-harmonic current
10	m(3rd)	mean of third harmonic current
11	m(3rd)	mean of absolute third harmonic current
12	s(3rd)	standard deviation of third harmonic current
13	m(2nd)	mean of second harmonic current
14	m(2nd)	mean of absolute second harmonic current
15	s(2nd)	standard deviation of second harmonic current

VII. Sample Execution of the Algorithmic Approach

This example, which uses actual sample data, serves two purposes: 1) to test the algorithm's overall functionality relative to membership function and rule generation and 2) to test if the derived production rules are appropriate for the actual problem. The sample data consists of 27 samples of fault and 41 samples of normal event, which appear faulty; the data is taken from electric power distribution networks [15]. These data were collected for discrimination studies of low current and high impedance faults from normal events such as switching, big motor-load connection, capacitor bank switching, and so on [16]. As the time-domain amplitude change of the fault current are low, feature parameters are chosen from frequency-domain variables. Harmonic parameters are selected from 0 to 640 Hz range excluding the driving frequency of 60 Hz. To find statistical measures (or feature) of the parameters, mean, standard deviation, and mean of the absolute value are calculated. The length of data for measuring these statistical measures is limited to 30 cycles of 60-Hz waveform. Fifteen variables of the statistical measures of the harmonic contents are shown in Appendix A.

We arranged fault data and normal data to form a training set and a testing set; this was accomplished using diagonal numbers from a table of random units [17]. If we meet an even number in the table, starting from the first sample, we put the sample into the training set; otherwise, we put it in the testing set. We stop the process once we have half of the fault samples in the training set. A similar procedure is performed for normal event samples. Appendix B shows the overall sample data with 15 variables; the truing set and the testing set are shown in Appendix C and D, respectively.

APPENDIX B
SAMPLE DATA (A)

#	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	class
1	-0.170	9.304	19.114	0.026	2.008	4.393	-0.140	3.271	6.660	-0.074	4.181	8.537	-0.049	0.980	1.975	F
2	0.215	29.866	34.525	-0.197	7.288	9.267	0.262	10.630	13.892	0.071	13.657	15.691	0.095	3.652	4.642	F
3	0.106	24.667	30.498	-0.089	5.720	7.411	0.167	10.649	14.381	0.060	11.200	13.758	0.026	2.831	3.543	F
4	0.037	12.703	18.098	0.060	3.500	4.957	0.127	6.849	12.536	0.028	5.729	8.195	-0.034	1.829	2.469	F
5	0.024	27.285	32.204	-0.008	6.469	7.965	0.223	10.861	13.954	-0.005	12.481	14.683	0.025	3.063	3.786	F
6	0.024	21.807	27.810	-0.084	4.505	5.835	0.201	7.169	9.275	-0.014	9.920	12.580	0.050	2.020	2.530	F
7	0.233	14.640	20.383	0.166	3.395	4.609	0.152	6.164	9.033	-0.096	6.629	9.153	-0.074	1.579	2.095	F
8	0.180	6.101	7.874	-0.162	3.121	4.106	0.118	3.998	5.753	0.080	2.737	3.530	0.048	1.711	2.169	F
9	0.045	6.457	8.075	-0.022	2.843	3.764	0.004	5.419	8.268	0.017	2.916	3.647	0.018	1.462	1.903	F
10	0.125	5.771	9.247	-0.130	2.248	3.261	0.170	3.297	5.121	0.051	2.611	4.170	0.054	1.285	1.839	F
11	0.038	2.506	2.942	-0.037	1.362	1.801	0.024	1.914	3.078	0.020	1.128	1.321	0.014	0.678	0.883	F
12	0.078	2.303	2.746	-0.066	1.292	1.784	0.104	1.937	3.248	0.034	1.028	1.225	0.035	0.700	0.938	F
13	0.035	3.589	4.960	-0.043	1.985	2.853	0.033	2.747	4.541	0.016	1.596	2.215	0.012	1.120	1.562	F
14	0.004	1.251	1.477	-0.001	0.306	0.377	0.005	0.365	0.449	0.003	0.532	0.597	-0.004	0.144	0.194	F
15	-0.001	2.816	3.946	0.004	1.291	2.020	0.008	2.245	4.384	0.002	1.251	1.729	0.004	0.699	1.092	F
16	0.017	2.323	3.694	-0.021	0.698	1.242	0.028	1.465	3.331	0.007	1.026	1.615	0.010	0.378	0.698	F
17	0.045	1.382	1.744	-0.046	0.624	0.941	0.050	1.020	1.878	0.020	0.606	0.760	0.017	0.335	0.510	F
19	0.128	4.623	7.126	-0.128	1.796	2.700	0.142	2.582	4.245	0.053	2.071	3.184	0.045	0.958	1.441	F
20	0.047	5.774	8.134	-0.023	2.157	3.138	0.032	3.540	5.688	0.023	2.605	3.647	0.010	1.217	1.722	F
21	0.004	14.352	17.298	0.017	4.264	5.565	-0.064	7.425	11.357	0.007	6.507	7.785	-0.016	2.204	2.792	F
22	0.072	21.133	25.265	-0.039	4.904	6.279	-0.008	7.887	10.004	0.027	9.590	11.340	0.022	2.542	3.157	F
23	-0.105	12.539	16.741	0.036	3.767	5.016	-0.103	6.390	9.620	-0.049	5.703	7.574	0.007	1.998	2.523	F
24	0.133	4.242	7.379	0.134	2.189	2.877	0.302	3.030	5.536	0.057	1.834	3.305	0.041	1.230	1.624	F
25	0.031	2.725	3.394	-0.034	1.699	2.211	0.078	2.325	3.622	0.015	1.207	1.504	0.024	0.944	1.237	F
26	0.112	7.644	10.628	-0.131	2.676	3.647	0.075	5.051	7.978	0.046	3.455	4.771	0.047	1.435	1.931	F
27	0.000	7.907	11.275	-0.006	2.339	3.232	0.005	3.747	5.833	0.001	3.542	4.962	-0.009	1.305	1.758	F
28	0.703	1.231	1.435	-0.719	0.782	0.505	0.822	0.994	0.931	0.303	0.505	0.539	0.282	0.327	0.272	NF
29	0.672	3.365	4.011	-0.682	0.743	0.571	0.803	0.985	0.975	0.292	0.968	1.066	0.274	0.325	0.311	NF
30	0.645	3.308	3.937	-0.667	0.669	0.394	0.774	0.845	0.644	0.282	0.944	1.038	0.264	0.286	0.222	NF
31	0.637	3.230	3.829	-0.653	0.655	0.368	0.759	0.817	0.603	0.276	0.871	0.955	0.259	0.281	0.217	NF
32	0.639	3.203	3.759	-0.651	0.652	0.346	0.755	0.802	0.564	0.274	0.802	0.877	0.259	0.284	0.219	NF
33	0.640	2.171	2.776	-0.653	0.699	0.441	0.760	0.990	0.980	0.279	0.626	0.700	0.259	0.297	0.233	NF
34	0.649	1.114	1.184	-0.662	0.662	0.222	0.766	0.766	0.349	0.280	0.493	0.510	0.262	0.262	0.098	NF
35	0.647	1.089	1.154	-0.660	0.660	0.218	0.767	0.767	0.336	0.280	0.475	0.489	0.263	0.263	0.099	NF
36	0.653	1.021	1.057	-0.671	0.671	0.201	0.780	0.780	0.319	0.283	0.437	0.441	0.266	0.266	0.097	NF
37	0.657	0.878	0.845	-0.673	0.673	0.180	0.781	0.781	0.268	0.284	0.357	0.335	0.268	0.268	0.093	NF
38	0.667	0.886	0.853	-0.679	0.679	0.172	0.788	0.788	0.270	0.288	0.358	0.334	0.269	0.269	0.091	NF
39	0.654	2.464	3.118	-0.674	0.718	0.528	0.783	0.989	1.031	0.285	0.694	0.782	0.267	0.312	0.285	NF
40	0.665	3.258	3.816	-0.672	0.673	0.346	0.780	0.824	0.576	0.285	0.817	0.892	0.267	0.285	0.221	NF
41	0.659	3.275	3.868	-0.673	0.676	0.367	0.782	0.832	0.600	0.287	0.871	0.956	0.267	0.289	0.227	NF
42	0.655	3.379	4.019	-0.670	0.676	0.394	0.779	0.851	0.650	0.285	0.953	1.047	0.266	0.291	0.232	NF
43	0.653	3.346	3.992	-0.668	0.672	0.401	0.777	0.850	0.660	0.284	0.977	1.074	0.265	0.292	0.230	NF
44	0.658	3.310	3.940	-0.670	0.672	0.397	0.776	0.847	0.650	0.282	0.954	1.049	0.266	0.290	0.227	NF
45	0.646	2.492	3.172	-0.669	0.688	0.461	0.779	0.983	0.998	0.281	0.714	0.820	0.266	0.296	0.247	NF

SAMPLE DATA (B)

46	0.655	1.007	1.050	-0.667	0.667	0.193	0.774	0.774	0.326	0.283	0.435	0.438	0.265	0.265	0.092	NF
47	1.931	2.003	1.375	-1.977	1.977	0.336	2.297	2.297	0.444	0.835	0.835	0.499	0.785	0.785	0.147	NF
48	1.095	1.343	1.196	-1.118	1.118	0.298	1.297	1.297	0.389	0.473	0.495	0.390	0.444	0.444	0.133	NF
49	0.415	0.943	1.070	-0.423	0.426	0.241	0.491	0.502	0.313	0.179	0.293	0.308	0.168	0.177	0.121	NF
50	0.007	0.866	1.056	-0.006	0.152	0.189	0.010	0.230	0.284	0.003	0.267	0.306	0.004	0.081	0.101	NF
51	0.118	0.713	0.871	0.122	0.183	0.182	0.143	0.285	0.326	0.051	0.285	0.318	0.049	0.087	0.097	NF
5	0.031	0.822	0.974	0.033	0.170	0.198	0.038	0.254	0.306	0.014	0.357	0.400	-0.012	0.073	0.092	NF
53	0.182	0.924	1.063	0.188	0.230	0.214	0.218	0.321	0.331	0.079	0.409	0.454	0.075	0.101	0.098	NF
54	0.424	1.047	1.183	0.436	0.436	0.221	0.506	0.518	0.344	0.184	0.454	0.490	0.172	0.176	0.098	NF
55	0.635	1.022	1.077	0.650	0.650	0.203	0.756	0.757	0.323	0.275	0.433	0.439	0.258	0.258	0.096	NF
56	0.774	1.029	0.985	0.788	0.788	0.196	0.914	0.914	0.305	0.334	0.422	0.397	0.313	0.313	0.097	NF
57	0.835	1.003	0.877	0.851	0.851	0.172	0.987	0.987	0.271	0.361	0.394	0.331	0.337	0.337	0.091	NF
58	0.833	0.936	0.739	0.850	0.850	0.159	0.987	0.987	0.237	0.361	0.370	0.269	0.337	0.337	0.090	NF
59	0.787	1.055	1.019	0.805	0.805	0.236	0.934	0.954	0.444	0.340	0.371	0.297	0.319	0.319	0.114	NF
60	0.728	1.121	1.160	0.745	0.745	0.181	0.867	0.867	0.273	0.315	0.380	0.349	0.296	0.296	0.095	NF
61	0.677	1.145	1.231	0.695	0.695	0.191	0.806	0.806	0.289	0.294	0.405	0.396	0.275	0.275	0.100	NF
62	0.608	0.882	0.891	0.621	0.626	0.333	0.718	0.733	0.431	0.262	0.383	0.386	0.246	0.250	0.139	NF
63	1.351	1.399	0.962	1.380	1.380	0.249	1.602	1.602	0.354	0.585	0.595	0.415	0.548	0.548	0.108	NF
64	1.653	1.655	0.893	1.688	1.688	0.185	1.959	1.959	0.288	0.715	0.715	0.386	0.670	0.670	0.089	NF
65	1.575	1.575	0.769	1.608	1.608	0.175	1.867	1.867	0.272	0.682	0.682	0.318	0.639	0.639	0.093	NF
66	1.259	1.263	0.650	1.287	1.287	0.187	1.495	1.495	0.263	0.544	0.544	0.257	0.511	0.511	0.096	NF
67	0.875	0.935	0.680	0.896	0.896	0.190	1.042	1.042	0.273	0.379	0.387	0.273	0.356	0.356	0.095	NF
68	0.554	0.817	0.834	0.568	0.568	0.189	0.659	0.661	0.298	0.240	0.354	0.353	0.225	0.226	0.092	NF

Once training and testing samples are arranged, we derive membership functions and production rules from the training samples. The testing samples are used to check the performance of the production rules. For each of the 15 variables, three levels of thresholds are calculated for eight fuzzy terms using the entropy minimization process; the seven threshold values of each variable are shown in Appendix E. We then proceed to locate left edge points, center points, and right edge points of the membership functions. For convenience, all the membership functions are assigned triangular shape except for the sets at the end points, which are assigned trapezoidal shape. For the triangular sets, the degree of membership of the element at the center point is 1.0 and the degrees of membership at the right and left edge points are 0.0.

For production rule generation, training samples are converted into binary values for each of the fuzzy terms. Each sample value is assigned the value "1" for the fuzzy set in which it has highest membership degree. Digit index calculation finds which variable is most important and how many steps should be covered to construct fuzzy term rules. All the mean probabilities (rule weights) are also calculated. Appendix F shows the production rules of the training sample data. Each fuzzy term has a rule (fuzzy term rule) with a with a few or more steps (or branches) of the decision process.

The final step of the example is to test the production rule (which contains seven fuzzy term rules) with the testing sample data. The testing sample data are first fuzzified using the fuzzy membership functions of the seven terms. Then, using the derived rules, two inference methods are applied simultaneously to compare results. We apply pivot balance defuzzification to the output fuzzy sets. Table VIII shows the results of the test. Using the "round match" method, the first round was matched and fired for every sample and, thus, the match stopped after the first round. A minus sign after a result indicates that the sample was assigned to the normal (nonfault) event class by the method.

For fault class samples, we have three incorrect classifications with the "overall match" method and only one incorrect classification with the "step match" method. Samples in the "overall match" method differ more between classed with respect to *pp* values; this may have implications for the "security" of the classification. This is illustrated in Fig. 7. The fault samples and normal samples in the "overall match" are placed farther away from the reference than those in the "round match" method: the distance between the *pp*'s of fault and normal samples is 0.87 in the "overall match" and 0.81 in the "round match" method. The "overall match" method may be used, therefore, for more sensitive fault identification, while the "round match" method may be better for greater security against false identification.

The significance of this example is not that this algorithmic approach works well in classification problem, but that the algorithm appropriately generates membership functions and production rules. Our example is sufficient to show that that

APPENDIX C
TRAINING DATA

#	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	class
1	-0.170	9.304	19.114	0.026	2.008	4.393	-0.140	3.271	6.660	-0.074	4.181	8.537	-0.049	0.980	1.975	F
4	0.037	12.703	18.098	0.060	3.500	4.957	0.127	6.849	12.536	0.028	5.729	8.195	-0.034	1.829	2.469	F
6	-0.024	21.807	27.810	-0.084	4.505	5.835	0.201	7.169	9.275	-0.014	9.920	12.580	0.050	2.020	2.530	F
7	-0.233	14.640	20.383	0.166	3.395	4.609	0.152	6.164	9.033	-0.096	6.629	9.153	-0.074	1.579	2.095	F
13	0.035	3.589	4.960	-0.043	1.985	2.853	0.033	2.747	4.541	0.016	1.596	2.215	0.012	1.120	1.562	F
17	0.045	1.382	1.744	0.046	0.624	0.941	0.050	1.020	1.878	0.020	0.606	0.760	0.017	0.335	0.510	F
19	0.128	4.623	7.126	0.128	1.796	2.700	0.142	2.582	4.245	0.053	2.071	3.184	0.045	0.958	1.441	F
22	0.072	21.133	25.265	0.039	4.904	6.279	0.008	7.887	10.004	0.027	9.590	11.340	0.022	2.542	3.157	F
23	0.105	12.539	16.741	0.036	3.767	5.016	0.103	6.390	9.620	-0.049	5.703	7.574	0.007	1.998	2.523	F
24	0.133	4.242	7.379	0.134	2.189	2.877	0.302	3.030	5.536	0.057	1.834	3.305	0.041	1.230	1.624	F
25	0.031	2.725	3.394	0.034	1.699	2.211	0.078	2.325	3.622	0.015	1.207	1.504	0.024	0.944	1.237	F
26	0.112	7.644	10.628	0.131	2.676	3.647	0.075	5.051	7.978	0.046	3.455	4.771	0.047	1.435	1.931	F
27	0.000	7.907	11.275	-0.006	2.339	3.232	0.005	3.747	5.833	0.001	3.542	4.962	-0.009	1.305	1.758	F
29	0.672	3.365	4.011	-0.682	0.743	0.571	0.803	0.985	0.975	0.292	0.968	1.066	0.274	0.325	0.311	NF
30	0.645	3.308	3.937	0.667	0.669	0.394	0.774	0.845	0.644	0.282	0.944	1.038	0.264	0.286	0.222	NF
32	0.639	3.203	3.759	-0.651	0.652	0.346	0.755	0.802	0.564	0.274	0.802	0.877	0.259	0.284	0.219	NF
33	0.640	2.171	2.776	-0.653	0.699	0.441	0.760	0.990	0.980	0.279	0.626	0.700	0.259	0.297	0.233	NF
34	0.649	1.114	1.184	0.662	0.662	0.222	0.766	0.766	0.349	0.280	0.493	0.510	0.262	0.262	0.098	NF
36	0.653	1.021	1.057	-0.671	0.671	0.201	0.780	0.780	0.319	0.283	0.437	0.441	0.266	0.266	0.097	NF
37	0.657	0.878	0.845	-0.673	0.673	0.180	0.781	0.781	0.268	0.284	0.357	0.335	0.268	0.268	0.093	NF
40	0.665	3.258	3.816	-0.672	0.673	0.346	0.780	0.824	0.576	0.285	0.817	0.892	0.267	0.285	0.221	NF
41	0.659	3.275	3.868	-0.673	0.676	0.367	0.782	0.832	0.600	0.287	0.871	0.956	0.267	0.289	0.227	NF
42	0.655	3.379	4.019	-0.670	0.676	0.394	0.779	0.851	0.650	0.285	0.953	1.047	0.266	0.291	0.232	NF
43	0.653	3.346	3.992	-0.668	0.672	0.401	0.777	0.850	0.660	0.284	0.977	1.074	0.265	0.292	0.230	NF
48	1.095	1.343	1.196	-1.118	1.118	0.298	1.297	1.297	0.389	0.473	0.495	0.390	0.444	0.444	0.133	NF
49	0.415	0.943	1.070	-0.423	0.426	0.241	0.491	0.502	0.313	0.179	0.293	0.308	0.168	0.177	0.121	NF
50	0.007	0.866	1.056	-0.006	0.152	0.189	0.010	0.230	0.284	0.003	0.267	0.306	0.004	0.081	0.101	NF
51	-0.118	0.713	0.871	0.122	0.183	0.182	-0.143	0.285	0.326	-0.051	0.285	0.318	-0.049	0.087	0.097	NF
52	-0.031	0.822	0.974	0.033	0.170	0.198	0.038	0.254	0.306	-0.014	0.357	0.400	-0.012	0.073	0.092	NF
53	0.182	0.924	1.063	0.188	0.230	0.214	0.218	0.321	0.331	0.079	0.409	0.454	0.075	0.101	0.098	NF
55	0.635	1.022	1.077	0.650	0.650	0.203	0.756	0.757	0.323	0.275	0.433	0.439	0.258	0.258	0.096	NF
57	0.835	1.003	0.877	0.851	0.851	0.172	0.987	0.987	0.271	0.361	0.394	0.331	0.337	0.337	0.091	NF
59	0.787	1.055	1.019	0.805	0.805	0.236	0.934	0.954	0.444	0.340	0.371	0.297	0.319	0.319	0.114	NF

the generated production rules with rule weights based on the fuzzy membership functions are relevant to real world situations.

VIII. CONCLUSION

An algorithmic method to automate the procedure for fuzzy logic application to binary classification is presented. This approach is based on an entropy minimization principle to optimally generate, using only sample data, membership functions, and fuzzy production rules. Membership function generation using the clustering principle is discussed and the production rule derivation, along with the rule weight determination, is illustrated. The relevance and appropriateness of the membership functions and production rules was evident from the detailed example. In the future, an advanced algorithmic method may be evolved for generic fuzzy logic applications.

ACKNOWLEDGMENT

The author would like to thank the referees of this paper, whose comments and suggestions helped improve the quality of the presentation.

REFERENCES

[1] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proc. Inst. Elect. Eng.*, vol. 121, no. 12, pp. 1585–1588, Dec. 1974.

[2] B. V. Dasarathy, "Practical models for fuzzy membership estimation," in *Proc. 2nd Int. Workshop Indust. Fuzzy Contr. Intell. Syst.*, College Station, TX, Dec. 1992, pp. 187–193.

APPENDIX D
TEST DATA

#	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	class
2	0.215	29.866	34.525	-0.197	7.288	9.267	0.262	10.630	13.892	0.071	13.657	15.691	0.095	3.652	4.642	F
3	0.106	24.667	30.498	0.089	5.720	7.411	0.167	10.649	14.381	0.060	11.200	13.758	0.026	2.831	3.543	F
5	0.024	27.285	32.204	-0.008	6.469	7.965	0.223	10.861	13.954	0.005	12.481	14.683	0.025	3.063	3.786	F
8	0.180	6.101	7.874	-0.162	3.121	4.106	0.118	3.998	5.753	0.080	2.737	3.530	0.048	1.711	2.169	F
9	0.045	6.457	8.075	-0.022	2.843	3.764	0.004	5.419	8.268	0.017	2.916	3.647	0.018	1.462	1.903	F
10	0.125	5.771	9.247	-0.130	2.248	3.261	0.170	3.297	5.121	0.051	2.611	4.170	0.054	1.285	1.839	F
11	0.038	2.506	2.942	-0.037	1.362	1.801	0.024	1.914	3.078	0.020	1.128	1.321	0.014	0.678	0.883	F
12	0.078	2.303	2.746	-0.066	1.292	1.784	0.104	1.937	3.248	0.034	1.028	1.225	0.035	0.700	0.938	F
14	0.004	1.251	1.477	-0.001	0.306	0.377	-0.005	0.365	0.449	0.003	0.532	0.597	-0.004	0.144	0.194	F
15	-0.001	2.816	3.946	0.004	1.291	2.020	0.008	2.245	4.384	-0.002	1.251	1.729	0.004	0.699	1.092	F
16	0.017	2.323	3.694	-0.021	0.698	1.242	0.028	1.465	3.331	0.007	1.026	1.615	0.010	0.378	0.698	F
20	0.047	5.774	8.134	-0.023	2.157	3.138	0.032	3.540	5.688	0.023	2.605	3.647	0.010	1.217	1.722	F
21	0.004	14.352	17.298	0.017	4.264	5.565	0.064	7.425	11.357	0.007	6.507	7.785	-0.016	2.204	2.792	F
28	0.703	1.231	1.435	-0.719	0.782	0.505	0.822	0.994	0.931	0.303	0.505	0.539	0.282	0.327	0.272	NF
31	0.637	3.230	3.829	0.653	0.655	0.368	0.759	0.817	0.603	0.276	0.871	0.955	0.259	0.281	0.217	NF
35	0.647	1.089	1.154	0.660	0.660	0.218	0.767	0.767	0.336	0.280	0.475	0.489	0.263	0.263	0.099	NF
38	0.667	0.886	0.853	0.679	0.679	0.172	0.788	0.788	0.270	0.288	0.358	0.334	0.269	0.269	0.091	NF
44	0.658	3.310	3.940	-0.670	0.672	0.397	0.776	0.847	0.650	0.282	0.954	1.049	0.266	0.290	0.227	NF
45	0.646	2.492	3.172	-0.669	0.688	0.461	0.779	0.983	0.998	0.281	0.714	0.820	0.266	0.296	0.247	NF
46	0.655	1.007	1.050	-0.667	0.667	0.193	0.774	0.774	0.326	0.283	0.435	0.438	0.265	0.265	0.092	NF
47	1.931	2.003	1.375	1.977	1.977	0.336	2.297	2.297	0.444	0.835	0.835	0.499	0.785	0.785	0.147	NF
54	0.424	1.047	1.183	-0.436	0.436	0.221	0.506	0.518	0.344	0.184	0.454	0.490	0.172	0.176	0.098	NF
56	0.774	1.029	0.985	-0.788	0.788	0.196	0.914	0.914	0.305	0.334	0.422	0.397	0.313	0.313	0.097	NF
59	0.787	1.055	1.019	-0.805	0.805	0.236	0.934	0.954	0.444	0.340	0.371	0.297	0.319	0.319	0.114	NF
60	0.728	1.121	1.160	-0.745	0.745	0.181	0.867	0.867	0.273	0.315	0.380	0.349	0.296	0.296	0.095	NF
61	0.677	1.145	1.231	-0.695	0.695	0.191	0.806	0.806	0.289	0.294	0.405	0.396	0.275	0.275	0.100	NF
62	0.608	0.882	0.891	-0.621	0.626	0.333	0.718	0.733	0.431	0.262	0.383	0.386	0.246	0.250	0.139	NF
63	1.351	1.399	0.962	-1.380	1.380	0.249	1.602	1.602	0.354	0.585	0.595	0.415	0.548	0.548	0.108	NF
64	1.653	1.655	0.893	-1.688	1.688	0.185	1.959	1.959	0.288	0.715	0.715	0.386	0.670	0.670	0.089	NF
65	1.575	1.575	0.769	-1.608	1.608	0.175	1.867	1.867	0.272	0.682	0.682	0.318	0.639	0.639	0.093	NF
66	1.259	1.263	0.650	-1.287	1.287	0.187	1.495	1.495	0.263	0.544	0.544	0.257	0.511	0.511	0.096	NF
67	0.875	0.935	0.680	-0.896	0.896	0.190	1.042	1.042	0.273	0.379	0.387	0.273	0.356	0.356	0.095	NF
68	0.554	0.817	0.834	-0.568	0.568	0.189	0.659	0.661	0.298	0.240	0.354	0.353	0.225	0.226	0.092	NF

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

[4] C. T. Lin and C. S. Lee, "Neural network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, no. 2, pp. 1320–1336, Feb. 1991.

[5] E. Sanchez, "Fuzzy logic knowledge systems and artificial neural networks in medicine and biology," in *An Introduction to Fuzzy Logic Application in Intelligent Systems*, R. R. Yager and L. A. Zadeh, Eds. Boston, MA: Kluwer, 1992, pp. 235–251.

[6] H.-J. Zimmermann, *Fuzzy Set Theory and Its Applications*, 2nd ed. Boston, MA: Kluwer, 1991.

[7] P. J. King and E. H. Mamdani, "The application of fuzzy control systems to industrial processes," in *Proc. Workshop Discrete Syst. Fuzzy Reasoning*, Queen Mary College, Univ. London, U.K., Jan. 1976, pp. 235–243.

[8] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man Mach. Studies*, vol. 7, pp. 1–13, 1975.

[9] R. Christensen, *Fundamentals of Inductive Reasoning*. Lincoln, MA: Entropy Ltd., 1980.

[10] ——, *Entropy Minimax Sourcebook*. Lincoln, MA: Entropy Ltd., 1980, vol. I–IV.

APPENDIX E
THRESHOLD VALUES

Variable No.	X_{31}	X_{21}	X_{32}	X_{11}	X_{33}	X_{22}	X_{34}
1	0.000	0.026	0.033	0.165	0.168	0.212	0.256
2	1.116	3.402	3.426	3.877	3.922	4.774	5.625
3	0.865	1.249	3.434	4.890	4.947	6.036	7.124
4	-1.116	-1.070	-1.024	-0.155	-0.149	-0.027	0.127
5	0.437	0.627	0.663	1.340	1.340	1.518	1.687
6	0.174	0.203	0.232	0.783	0.796	1.058	1.319
7	-0.001	0.026	0.207	0.355	0.355	0.402	0.447
8	0.232	0.268	0.305	0.996	1.030	1.340	1.668
9	0.271	0.329	0.388	1.495	1.522	2.047	2.571
10	0.001	0.006	0.010	0.075	0.076	0.095	0.114
11	0.520	0.991	1.003	1.232	1.254	1.667	2.079
12	0.705	1.104	1.125	1.533	1.533	2.086	2.610
13	-0.070	0.010	0.012	0.056	0.056	0.075	0.093
14	0.085	0.320	0.332	0.567	0.567	0.666	0.759
15	0.092	0.106	0.121	0.398	0.404	0.536	0.667

APPENDIX F
GENERATED RULES

Term	Step#	digit	binary	class	$\overline{P}$	Steps of the Production Rule
PB	1	15	1	1	0.933	IF s(2nd) is PB, THEN fault.
	2	13	1	2	0.944	ELSE IF m(2nd) is PB, THEN not-fault.
	3	4	1	2	0.667	ELSE IF m(eve) is PB, THEN not-fault.
	4	15	0	2	0.667	ELSE IF s(2nd) is not PB, THEN not-fault.
						ENDIF
PM	1	4	1	1	0.769	IF m(eve) is PM, THEN fault.
	2	11	1	1	0.667	ELSE IF m(3rd) is PM, THEN fault.
	3	2	1	1	0.667	ELSE IF m(odd) is PM, THEN fault.
	4	13	1	2	0.667	ELSE IF m(2nd) is PM, THEN non-fault.
	5	8	1	2	0.667	ELSE IF m(sub) is PM, THEN non-fault.
	6	15	0	2	0.810	ELSE IF s(2nd) is not PM, THEN not-fault.
						ENDIF
PS	1	4	1	1	0.800	IF m(eve) is PS, THEN fault.
	2	8	1	1	0.667	ELSE IF m(sub) is PS, THEN fault.
	3	3	1	1	0.667	ELSE IF s(odd) is PS, THEN fault.
	4	10	1	2	0.667	ELSE IF m(3rd) is PS, THEN not-fault.
	5	15	0	2	0.667	ELSE IF s(2nd) is not PS, THEN not-fault.
						ENDIF
ZE	1	8	1	2	0.938	IF m(sub) is ZE, THEN not-fault.
	2	4	1	2	0.750	ELSE IF m(eve) is ZE, THEN not-fault.
	3	10	1	1	0.800	ELSE IF m(3rd) is ZE, THEN fault.
	4	11	1	1	0.667	ELSE IF m(3rd) is ZE, THEN fault.
	5	15	0	1	0.688	ELSE IF s(2nd) is not ZE, THEN fault.
						ENDIF
NS	1	5	1	1	0.941	IF m(eve) is NS, THEN fault.
	2	13	1	1	0.900	ELSE IF m(2nd) is NS, THEN fault.
	3	8	1	2	0.750	ELSE IF m(sub) is NS, THEN not-fault.
	4	10	1	1	0.667	ELSE IF m(3rd) is NS, THEN fault.
	5	15	0	1	0.600	ELSE IF s(2nd) is not NS, THEN fault.
						ENDIF
NM	1	14	1	2	0.933	IF m(2nd) is NM, THEN not-fault.
	2	9	1	2	0.833	ELSE IF s(sub) is NM, THEN not-fault
	3	7	1	1	0.857	ELSE IF m(sun) is NM, THEN fault.
	4	15	1	2	0.667	ELSE IF s(2nd) is NM, THEN not-fault.
	5	3	1	2	0.667	ELSE IF s(odd) is NM, THEN not-fault.
	6	13	1	1	0.750	ELSE IF m(2nd) is NM, THEN fault.
	7	15	0	1	0.800	ELSE IF s(2nd) is not NM, THEN fault.
						ENDIF
NB	1	12	1	2	0.889	IF s(3rd) is NB, THEN not-fault.
	2	10	1	1	0.875	ELSE IF m(3rd) is NB, THEN fault.
	3	13	1	1	0.667	ELSE IF m(2nd) is NB, THEN fault.
	4	7	1	1	0.667	ELSE IF m(sub) is NB, THEN fault.
	5	15	0	2	0.500	ELSE IF s(2nd) is not NB, THEN not-fault.
						ENDIF

[11] E. Cox, "Adaptive fuzzy systems," *IEEE Spectrum*, vol. 30, pp. 27–31, Feb. 1993.

[12] M. H. Smith, "Evaluation of performance and robustness of a parallel dynamic switching fuzzy systems," in *Proc. 2nd Int. Workshop Indust. Fuzzy Contr. Intell. Syst.*, College Station, TX, Dec. 1992, pp. 163–172.

[13] C. J. Kim and B. D. Russell, "A structure for fuzzy decision-making system for power system protection," in *Proc. IEEE 2nd Int. Conf. Fuzzy Syst.*, San Francisco, CA, Mar. 28–Apr. 1, 1993, pp. 998–1003.

[14] ——, "Application of fuzzy expert model for power systems protection," in *Proc. 5th Int. Fuzzy Syst. Assoc. World Congress*, Seoul, Korea, July 1993, pp. 1074–1077.

[15] C. J. Kim, "An intelligent decision-making system for detecting high impedance faults," Ph.D. dissertation, Texas A&M Univ., College Station, TX, Dec. 1989.

[16] C. J. Kim and B. D. Russell, "Classification of faults and switching events by inductive reasoning and expert system methodology," *IEEE Trans. Power Delivery*, vol. 4, pp. 1631–1637, July 1989.

[17] *CRC Standard Mathematical Tables*, 27th ed. Boca Raton, FL: CRC, 1984.

C. J. Kim (M'90) received the B.S. and M.S. degrees in electrical engineering at Seoul National University, Korea, in 1980 and 1982, respectively, and the Ph.D. degree from Texas A&M University, College Station, TX, in 1989.

From 1990 to 1991, he worked as a Post-Doctoral Research Associate for Power System Automation Laboratory (PSAL), Texas A&M University, College Station. From 1992 to 1994, he worked as a Research Faculty Member for PSAL. Since 1994 he has been with the Department of Electrical Engineering, University of Suwon, Kyung-Gi, Korea, where he holds an Assistant Professor position. His research interests include information technology and intelligent systems and their application to power system control, monitoring, and protection.