Case Generation Using Rough Sets with Fuzzy Representation

Sankar K. Pal, *Fellow*, *IEEE*, and Pabitra Mitra, *Student Member*, *IEEE*

Abstract—In this article, we propose a rough-fuzzy hybridization scheme for case generation. Fuzzy set theory is used for linguistic representation of patterns, thereby producing a fuzzy granulation of the feature space. Rough set theory is used to obtain dependency rules which model informative regions in the granulated feature space. The fuzzy membership functions corresponding to the informative regions are stored as cases along with the strength values. Case retrieval is made using a similarity measure based on these membership functions. Unlike the existing case selection methods, the cases here are cluster granules and not sample points. Also, each case involves a reduced number of relevant features. These makes the algorithm suitable for mining data sets, large both in dimension and size, due to its low-time requirement in case generation as well as retrieval. Superiority of the algorithm in terms of classification accuracy and case generation and retrieval times is demonstrated on some real-life data sets.

Index Terms—Case-based reasoning, linguistic representation, rough dependency rules, granular computing, rough-fuzzy hybridization, soft computing, pattern recognition, data mining.

1 INTRODUCTION

A case-based reasoning (CBR) system adapts old solutions to meet new demands, explains and critiques new situations using old instances (called cases), and performs reasoning from precedents to interpret new problems [1]. A case may be defined as a contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving goals of the system. Selection and generation of cases are two important components of a CBR system. While case selection deals with selecting informative prototypes from the data, case generation concerns itself with the construction of "cases" that need not necessarily include any of the given data points. The cases in the latter one may be constituted by some description of a collection of points, represented by information granules. Since computation needs to be performed only on information granules, not on the individual points, retrieval time will be reduced. The present article concerns with the problem of case generation and its related merits in rough-fuzzy granular framework.

Early CBR systems mainly used case selection mechanisms based on the nearest-neighbor principle. These algorithms involve case pruning/growing methodologies, as exemplified by the popular IB3 algorithm [2]. A summary of the above approaches may be found in [3]. Recently, fuzzy logic and other soft computing tools have been integrated with CBR for developing efficient methodologies and algorithms [4]. For case selection and retrieval, the role of fuzzy logic has been mainly in providing similarity measures [5] and modeling ambiguous situations [6], [7]. A neuro-fuzzy method for selecting cases has been proposed in [8], where a fuzzy case similarity measure is used, with repeated growing and pruning of cases, until the case base becomes stable. All the operations are performed using a connectionist model with adaptive link structure. One may note that the literature on case generation is relatively scanty in both classical and soft computing framework.

Rough set theory was developed by Pawlak [9], [10] for classificatory analysis of data tables. The main goal of rough set theoretic analysis is to synthesise approximation (upper and lower) of concepts from the aquired data. While fuzzy set theory assigns to each object a grade of belongingness to represent an imprecise set, the focus of rough set theory is on the ambiguity caused by limited discernibility of objects in the domain of discourse. The key concepts here are those of "information granule" and "reducts." An information granule is a clump of objects (points) in the universe of discourse drawn together by indistinguishability, similarity, proximity, or functionality. Information granules reduce the data by identifying equivalence classes, i.e., objects that are indiscernible, using the available attributes. Only one of the elements of the equivalence class is needed to represent the entire class. Reduction can also be done by keeping only those attributes that preserve the indiscernibility relation. So, one is, in effect, looking for minimal subsets of attributes that induce the same partition on the domain as done by the original set. In other words, the essence of information remains intact and superfluous attributes are removed. The above sets of attributes are called reducts. An information granule formalizes the concept of finite precision representation of objects in real-life situations, and reducts represent the *core* of an information system (both in terms of objects and features) in a granular universe. It may be noted that cases also represent the informative and irreducible part of a problem. Hence, rough set theory is a natural choice for case selection in domains which are data rich, contain uncertainties, and allow tolerance for imprecision. Additionaly, rough sets have the capability of handling complex objects (e.g., proofs, hierarchies, frames, rule bases); thereby strengthening further the necessity of rough-CBR systems. Some of the attempts being made in this regard are available in [11], [12]. Recently, rough sets and fuzzy sets have been integrated in soft computing framework, the aim being to develop a model of uncertainty stronger than either

• *The authors are with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India. E-mail: {sankar, pabitra_r}@isical.ac.in.*

Manuscript received 30 May 2001; revised 25 Feb. 2002; accepted 7 Jan. 2003. For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 114239.

TABLE 1
Hiring: An Example of a Decision Table

	Diploma (*i*)	*Experinece* (*e*)	*French* (*f*)	*Reference* (*r*)	*Decision*
x_1	MBA	Medium	Yes	Excellent	Accept
x_2	MBA	Low	Yes	Neutral	Reject
x_3	MCE	Low	Yes	Good	Reject
x_4	MSc	High	Yes	Neutral	Accept
x_5	MSc	Medium	Yes	Neutral	Reject
x_6	MSc	High	Yes	Excellent	Reject
x_7	MBA	High	No	Good	Accept
x_8	MCE	Low	No	Excellent	Reject

[13]. Therefore, rough-fuzzy CBR system has significant potential.

In this article, we use rough-fuzzy hybridization for designing a methodology for case generation of a CBR system. Each pattern (object) is represented by its fuzzy membership values with respect to three overlapping linguistic property sets "low," "medium," and "high," thereby generating a fuzzy granulation of the feature space which contains granules with ill-defined boundaries. Discernibility of the granulated objects in terms of attributes is then computed in the form of a discernibility matrix. Using rough set theory, a number of decision rules are generated from the discernibility matrix. The rules represent *rough clusters* of points in the original feature space. The fuzzy membership functions corresponding to the region, modeled by a rule, are then stored as a case. A strength factor, representing the a priori probability (size) of the cluster, is associated with each case. In other words, each case has three components, namely, the membership functions of the fuzzy sets appearing in the reducts, the class labels, and the strength factor. In the *retrieval* phase, these fuzzy membership functions are utilized to compute the similarity of the stored cases with an unknown pattern.

It may be noted that, unlike most case selection schemes, the cases generated by our algorithm need not be any of the objects (patterns) encountered, rather they represent regions having dimensions equal to or less than that of the input feature space. That is, all the input features (attributes) may not be required to represent a case. This type of variable and reduced length representation of cases results in the decrease in retrieval time. Furthermore, the proposed algorithm deals only with the information granules, not the actual data points. Because of these characterstics, its significance to data mining applications is evident.

The effectiveness of the methodology is demonstrated on some real-life data sets, large both in dimension and size. Cases are evaluated in terms of the classification accuracy obtained using 1-NN rule. A comparison is made with the conventional IB3 and IB4 algorithms [2], and random case selection method. The proposed methodology is found to perform better in terms of 1-NN accuracy, case generation time, and average case retrieval time.

2 Rough Sets

Let us present here some preliminaries of rough set theory which are relevant to this article. For details, one may refer to [10] and [14].

2.1 Information Systems

An *information system* can be viewed as a pair $\mathcal{S} = < U, A >$, or a function $f : U \times A \rightarrow V$, where U is a nonempty finite set of *objects* called the *universe*, A a nonempty finite set of *attributes*, and V a value set such that $a : U \rightarrow V_a$ for every $a \in A$. The set V_a is called the *value set* of a.

In many situations, there is an outcome of classification that is known. This a posteriori knowledge is expressed by one distinguished attribute called decision attribute. Information systems of this kind are called decision systems. A *decision system* is any information system of the form $\mathcal{A} = (U, A \cup \{d\})$, where $d \notin A$ is the *decision attribute*. The elements of A are called *conditional attributes*. An information (decision) system may be represented as an *attribute-value (decision) table*, in which rows are labeled by objects of the universe and columns by the attributes. Table 1 is an example of representing a decision system

$$\mathcal{A}' = (U, \{Diploma,\ Experience,\ French,\ Reference\} \cup \{Decision\})$$

for hiring personnels.

2.2 Indiscernibility and Set Approximation

A decision system (i.e., a decision table) expresses all the knowledge available about a system. This table may be unnecessarily large because it could be redundant at least in two ways. The same or indiscernible objects may be represented several times, or some attributes may be superfluous. The notion of equivalence relation is used to tackle this problem.

With every subset of attributes $B \subseteq A$, one can easily associate an *equivalence relation* I_B on U: $I_B = \{(x, y) \in U : \text{for every } a \in B,\ a(x) = a(y)\}$. I_B is called *B-indiscernibility relation*. If $(x, y) \in I_B$, then objects x and y are indiscernible from each other by attributes B. The equivalence classes of the partition induced by the B-indiscernibility relation are denoted by $[x]_B$. These are also known as *granules*. For example, in the case of the decision system represented by Table 1, if we consider the attribute set $B = \{Diploma, Experience\}$, the relation I_B defines the following partition of the universe

$$I_B = I_{\{Diploma,\ Experience\}} = \{\{x_3, x_8\}, \{x_4, x_6\}, \{x_5\}, \{x_1\}, \{x_2\}, \{x_7\}\}.$$

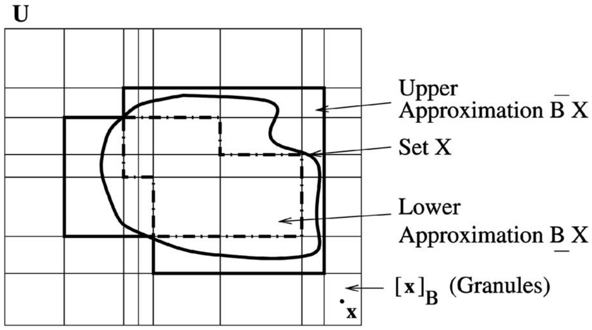


Fig. 1. Rough representation of a set with upper and lower approximations.

Here, $\{x_3, x_8\}, \{x_4, x_6\}, \{x_5\}, \{x_1\}, \{x_2\}, \{x_7\}$ are the granules obtained by the relation I_B.

The partition induced by the equivalence relation I_B can be used to build new subsets of the universe. Subsets that are most often of interest have the same value of the outcome attribute, i.e., belong to the same class. It may happen, however, that a concept (e.g., "*Reject*" in Table 1) cannot be defined crisply using the attributes available. It is here that the notion of rough set emerges. Although we cannot delineate the concept crisply, it is possible to delineate the objects which definitely "belong" to the concept and those which definitely "do not belong" to the concept. These notions are formally expressed as follows.

Let $\mathcal{A} = (U, A)$ be an information system and let $B \subseteq A$ and $X \subseteq U$. We can approximate X using only the information contained in B by constructing the lower and upper approximations of X. If $X \subseteq U$, the sets $\{x \in U : [x]_B \subseteq X\}$ and $\{x \in U : [x]_B \cap X \neq \emptyset\}$, where $[x]_B$ denotes the equivalence class of the object $x \in U$ relative to I_B, are called the *B-lower* and *B-upper approximation* of X in $\mathcal{S}$ and denoted by $\underline{B}X, \overline{B}X$, respectively. The objects in $\underline{B}X$ can be certainly classified as members of X on the basis of knowledge in B, while objects in $\overline{B}X$ can only be classified as possible members of X on the basis of B. This is illustrated in Fig. 1. Considering the decision system *Hiring* (Table 1), if $B = \{Diploma, Experience\}$ and X is the concept *Reject*, then: $\underline{B}X = \{x_2, \{x_3, x_8\}, x_5\}$ and $\overline{B}X = \{x_2, \{x_3, x_8\}, \{x_4, x_6\}, x_5\}$.

2.3 Reducts

Indiscernibility relation reduces the data by identifying equivalence classes, i.e., objects that are indiscernible, using the available attributes. Only one element of the equivalence class is needed to represent the entire class. Reduction can also be done by keeping only those attributes that preserve the indiscernibility relation and, consequently, set approximation. So, one is, in effect, looking for *minimal* sets of attributes taken from the initial set A, so that the minimal sets induce the *same* partition on the domain as done by A. In other words, the essence of the information remains intact and superfluous attributes are removed. The above sets of attributes are called *reducts*.

Reducts have been nicely characterized in [14] by *discernibility matrices* and *discernibility functions*. Consider $U = \{x_1, \ldots, x_n\}$ and $A = \{a_1, \ldots, a_m\}$ in the information system $\mathcal{S} =< U, A >$. By the discernibility matrix $\mathrm{M}(\mathcal{S})$ of $\mathcal{S}$ is meant an $n \times n$-matrix (symmetrical with empty diagonal) with entries c_{ij}s as follows:

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j)\}. \tag{1}$$

A discernibility function $f_{\mathcal{S}}$ is a function of m Boolean variables $\bar{a}_1, \ldots, \bar{a}_m$ corresponding to the attributes $a_1, \ldots, a_m$, respectively, and defined as follows:

$$f_{\mathcal{S}}(\bar{a}_1, \ldots, \bar{a}_m) = \bigwedge\Big\{\bigvee(c_{ij}) : 1 \leq i, j \leq n,\ j < i,\ c_{ij} \neq \emptyset\Big\}, \tag{2}$$

where $\bigvee(c_{ij})$ is the disjunction of all variables $\bar{a}$ with $a \in c_{ij}$. It is seen in [14] that $\{a_{i_1}, \ldots, a_{i_p}\}$ is a reduct in $\mathcal{S}$ if and only if $a_{i_1} \wedge \ldots \wedge a_{i_p}$ is a prime implicant (constituent of the disjunctive normal form) of $f_{\mathcal{S}}$.

2.4 Dependency Rule Generation

A principal task in the method of rule generation is to compute reducts relative to a particular kind of information system, the decision system. Relativized versions of discernibility matrices and functions shall be the basic tools used in the computation. d-reducts and d-discernibility matrices are used for this purpose [14]. The methodology is described below.

Let $\mathcal{S} =< U, A >$ be a decision table, with $A = C \cup d$, and d and C its sets of decision and condition attributes respectively. Let the value set of d be of cardinality l, i.e., $V_d = \{d_1, d_2, \ldots, d_l\}$, representing l classes. Divide the decision table $\mathcal{S} = < U, A >$ into l tables $\mathcal{S}_i = < U_i, A_i >$, $i = 1, \ldots, l$, corresponding to the l decision attributes $d_1, \ldots, d_l$, where $U = U_1 \cup \ldots \cup U_l$ and $A_i = C \cup \{d_i\}$.

Let $\{x_{i1}, \ldots, x_{ip}\}$ be the set of those objects of U_i that occur in $\mathcal{S}_i, i = 1, \ldots, l$. Now, for each d_i-reduct $B = \{b_1, \ldots, b_k\}$ (say), a discernibility matrix (denoted by $\mathrm{M}_{d_i}(B)$) can be derived from the d_i-discernibility matrix as follows:

$$c_{ij} = \{a \in B : a(x_i) \neq a(x_j)\}, \tag{3}$$

for $i, j = 1, \ldots, n$.

For each object $x_j \in x_{i_1}, \ldots, x_{i_p}$, the discernibility function $f_{d_i}^{x_j}$ is defined as

$$f_{d_i}^{x_j} = \bigwedge\Big\{\bigvee(c_{ij}) : 1 \leq i, j \leq n,\ j < i,\ c_{ij} \neq \emptyset\Big\}, \tag{4}$$

where $\bigvee(c_{ij})$ is the disjunction of all members of c_{ij}. Then, $f_{d_i}^{x_j}$ is brought to its disjunctive normal form (d.n.f). One thus obtains a dependency rule r_i, *viz.* $d_i \leftarrow P_i$, where P_i is the disjunctive normal form (d.n.f) of $f_{d_i}^{x_j}, j \in i_1, \ldots, i_p$.

The dependency factor df_i for r_i is given by

$$df_i = \frac{card(POS_{B_i}(d_i))}{card(U_i)}, \tag{5}$$

where $POS_{B_i}(d_i) = \bigcup_{X \in I_{d_i}} \underline{B}_i(X)$, and $\underline{B}_i(X)$ is the lower approximation of X with respect to B_i. B_i is the set of condition attributes occuring in the rule $r_i : d_i \leftarrow P_i$. $POS_{B_i}(d_i)$ is the positive region of class d_i with respect to attributes B_i, denoting the region of class d_i that can be surely described by attributes B_i. Thus, df_i measures the information about decision attributes d_i derivable from the condition attributes of a rule B_i. df_i has values in the interval $[0, 1]$, with the maximum and minimum values corresponding to complete dependence and independence of d_i on B_i, respectively.

TABLE 2
Two Decision Tables Obtained by Splitting the *Hiring* Table $\mathcal{S}$ (Table 1)

	i	e	f	r	*Decision*
x_1	MBA	Medium	Yes	Excellent	Accept
x_4	MSc	High	Yes	Neutral	Accept
x_7	MBA	High	No	Good	Accept

(a)

	i	e	f	r	*Decision*
x_2	MBA	Low	Yes	Neutral	Reject
x_3	MCE	Low	Yes	Good	Reject
x_5	MSc	Medium	Yes	Neutral	Reject
x_6	MSc	High	Yes	Excellent	Reject
x_8	MCE	Low	No	Excellent	Reject

(b)

(a) $\mathcal{S}_{Accept}$ and (b) $\mathcal{S}_{Reject}$.

Example 1. The methodology for rough set rule generation is illustrated here. Let us consider the *Hiring* decision system $\mathcal{A}' = (U, \{Diploma(i), Experience(e), French(f), Reference(r)\} \cup \{Decision\})$ of Table 1. $V_{Decision} = \{Accept, Reject\}$ is the value set of the atrribute $Decision$; $V_{Decision}$ is of cardinality two. The original decision table (Table 1) is thus split into two decision tables $\mathcal{S}_{Accept}$ (Table 2a), and $\mathcal{S}_{Reject}$ (Table 2b). Since all the objects in each table are distinct, they could not be reduced further. Next, for each decision table, the discernibility matrices $\mathbf{M}_{Accept}(C)$ and $\mathbf{M}_{Reject}(C)$ are obtained using (3). Among them, only the matrix $\mathbf{M}_{Accept}(C)$ is shown in Table 3, as an illustration. The discernibility function obtained from $\mathbf{M}_{Accept}(C)$ is

$$\begin{aligned} f_{Accept} &= (i \vee e \vee r) \wedge (e \vee f \vee r) \wedge (i \vee f \vee r) \\ &= (e \wedge i) \vee (e \wedge f) \vee (i \wedge f) \vee r \\ &\quad \text{(disjunctive normal form).} \end{aligned}$$

The following dependency rules are obtained from f_{Accept}

$$\begin{aligned} &Accept \leftarrow e \wedge i \\ &Accept \leftarrow e \wedge f \\ &Accept \leftarrow i \wedge f \\ &Accept \leftarrow r. \end{aligned}$$

3 Linguistic Representation of Patterns and Fuzzy Granulation

As is evident from the previous section, the rough set theory deals with a set of objects in a granular universe. In the present section, we describe a way of obtaining the granular feature space using fuzzy linguistic representation of patterns. Only the case of numeric features is mentioned here. (Features in descriptive and set forms can also be handled in this framework.) The details of the methodologies involved may be found in [15], [16].

TABLE 3
Discernibility Matrix $\mathbf{M}_{Accept}$ for the Split *Hiring* Decision Table $\mathcal{S}_{Accept}$ (Table 2a)

Objects	x_1	x_4	x_7
x_1		i, e, r	e, f, r
x_4			i, f, r
x_7			

Let a pattern (object) $\mathbf{F}$ be represented by n numeric features (attributes), i.e., $\mathbf{F} = [F_1, F_2, \ldots, F_n]$. In other words, $\mathbf{F}$ is a point in an n-dimensional vector space. Each feature is then described in terms of its fuzzy membership values corresponding to three linguistic fuzzy sets, namely, *low* (L), *medium* (M), and *high* (H). Thus, an n-dimensional pattern vector is represented as a $3n$-dimensional vector [15], [16]

$$\begin{aligned} \mathbf{F} =& [\mu^1_{low}(F_1), \mu^1_{medium}(F_1), \mu^1_{high}(F_1); \mu^2_{low}(F_2), \mu^2_{medium}(F_2), \\ & \mu^2_{high}(F_2); \ldots ; \mu^n_{low}(F_n), \mu^n_{medium}(F_n), \mu^n_{high}(F_n)], \end{aligned} \tag{6}$$

where $\mu^j_{low}(F_j)$, $\mu^j_{medium}(F_j)$, and $\mu^j_{high}(F_j)$ indicate the membership values of F_j to the fuzzy sets *low*, *medium*, and *high* along feature axis j. $\mu(F_j) \in [0, 1]$. It means each feature F is represented by three $[0, 1]$-valued membership functions representing three fuzzy sets or characterizing three fuzzy granules along each axis; thereby constituting 3^n fuzzy granules in an n-dimensional feature space. These functions introduce an expert's bias, to an extent, in the representation of the original n-dimensional points.

For each input feature F_j, the fuzzy sets *low*, *medium*, and *high* are characterized individually by a π-membership function whose form is [17], [18]

$$\mu(F_j) = \pi(F_j; c, \lambda) = \begin{cases} 2\left(1 - \frac{|F_j - c|}{\lambda}\right)^2, & \text{for } \frac{\lambda}{2} \le |F_j - c| \le \lambda \\ 1 - 2\left(\frac{|F_j - c|}{\lambda}\right)^2, & \text{for } 0 \le |F_j - c| \le \frac{\lambda}{2} \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where $\lambda (> 0)$ is the radius of the π-function with c as the central point. For each of the fuzzy sets *low*, *medium*, and *high*, λ and c take different values. These values are chosen so that the membership functions for these three fuzzy sets have overlapping nature (intersecting at membership value 0.5), as shown in Fig. 2.

Let us now explain the procedure for selecting the centers (c) and radii (λ) of the overlapping π-functions. Let m_j be the mean of the pattern points along jth axis. Then, m_{j_l} and m_{j_h} are defined as the mean (along the jth axis) of the pattern points having coordinate values in the range $[F_{j_{min}}, m_j)$ and $(m_j, F_{j_{max}}]$, respectively, where $F_{j_{max}}$ and $F_{j_{min}}$ denote the upper and lower bounds of the dynamic range of feature F_j. The centers and the radii of the three π-functions are defined as

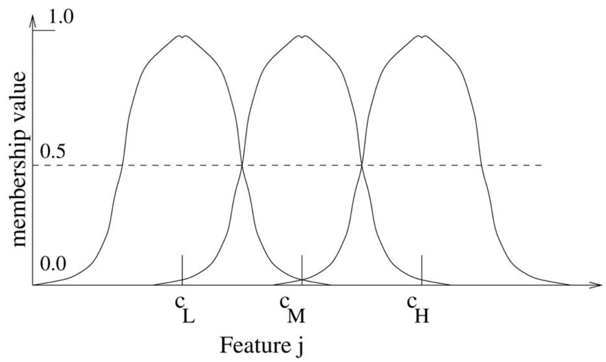


Fig. 2. π-Membership functions for linguistic fuzzy sets *low* (L), *medium* (M), and *high* (H) for each feature axis.

$$\begin{aligned} & low(F_j) && = m_{j_l} \\ & c_{medium(F_j)} && = m_j \\ & c_{high(F_j)} && = m_{j_h} \\ & \lambda_{low(F_j)} && = c_{medium(F_j)} - c_{low(F_j)} \\ & \lambda_{high(F_j)} && = c_{high(F_j)} - c_{medium(F_j)} \\ & \lambda_{medium(F_j)} && = 0.5\,(c_{high(F_j)} - c_{low(F_j)}). \end{aligned} \tag{8}$$

Here, we take into account the distribution of the pattern points along each feature axis while choosing the corresponding centers and radii of the linguistic fuzzy sets.

The aforesaid three overlapping functions along each axis generate the fuzzy granulated feature space in n-dimension. The granulated space contains 3^n granules with fuzzy boundaries among them. Here, the granules (clumps of similar objects or patterns) are attributed by the three fuzzy linguistic values "low," "medium," and "high." The degree of belongingness of a pattern to a granule (or the degree of possessing a property low, medium, or high by a pattern) is determined by the corresponding membership function.

Furthermore, if one wishes to obtain crisp granules (or crisp subsets), α-cut, $0 < \alpha < 1$, [18] of these fuzzy sets may be used. (α-cut of a fuzzy set is a crisp set of points for which membership value is greater than or equal to α.)

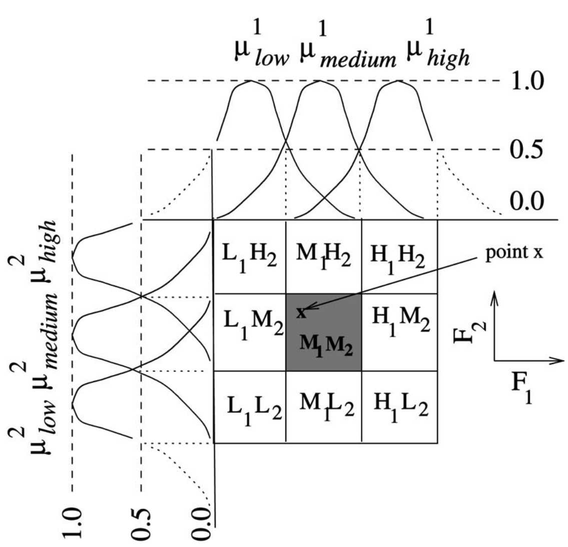


Fig. 3. Generation of crisp granules from linguistic (fuzzy) representation of the features F_1 and F_2. The dark region (M_1, M_2) indicates a crisp granule obtained by 0.5-cuts on the μ^1_{medium} and μ^2_{medium} functions.

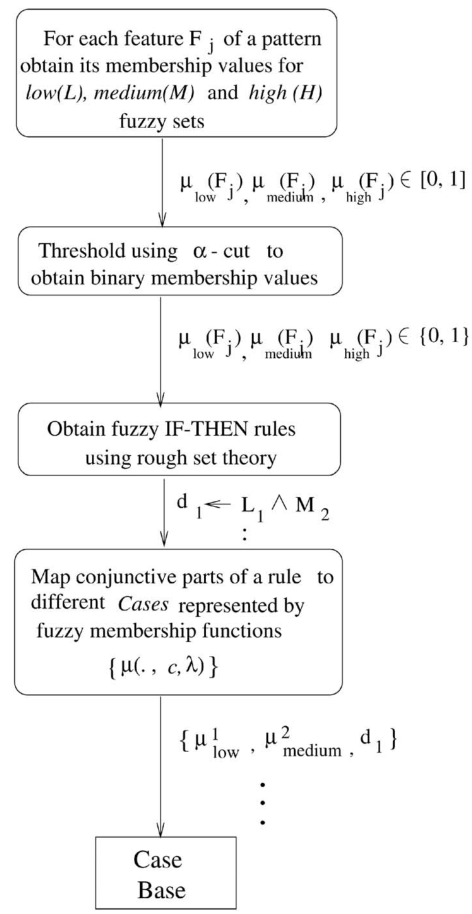


Fig. 4. Schematic diagram of rough-fuzzy case generation.

Note that the concept of fuzzy granulation has been explained earlier in different ways and effectively used in rough-fuzzy framework [19], [20].

4 Case Generation Methodology

Here, we describe a methodology for case generation on the fuzzy granulated space as obtained in the previous section. This involves two tasks, namely, 1) generation of fuzzy rules using rough set theory and 2) mapping the rules to cases. For obtaining linguistic rules, we have converted fuzzy membership values of the patterns to binary ones, i.e., fuzzy membership functions to binary functions using an α-cut. This is illustrated in Fig. 3, where 0.5-cut is used to obtain $3^2 = 9$ crisp granules (subsets) of the two-dimensional feature space from the linguistic representation of the input features. Although we have adopted this procedure, one can generate rules directly from fuzzy granules [21], [22].

The schematic diagram for the generation of case is shown in Fig. 4. One may note that, the inputs to the case generation process are fuzzy membership functions, the output "cases" are also fuzzy membership functions, but the intermediate rough set theoretic processing is performed on binary functions representing crisp sets (granules). For example, the inputs to Block 2 are fuzzy membership functions. Its outputs are binary membership functions which are used for rough processing in Block 3 and Block 4. Finally, the outputs of Block 4, representing cases, are again fuzzy membership functions. Each task is discussed below.

4.1 Thresholding and Rule Generation

Consider the $3n$ fuzzy membership values of a n-dimensional pattern $\mathbf{F}_i$. Then, select only those attributes having values greater than or equal to Th (= 0.5, say). In other words, we obtain a 0.5-cut of all the fuzzy sets to obtain binary membership values corresponding to the sets *low*, *medium*, and *high*.

For example, consider the point x in Fig. 3. Its $3n$-dimensionall fuzzy represention is $F = [0.4, 0.7, 0.1, 0.2, 0.8, 0.4]$. After binarization it becomes $F_b = [0, 1, 0, 0, 1, 0]$, which denotes the crisp granule (or subset) at the center of the 3×3 granulated space.

After the binary membership values are obtained for all the patterns, we constitute the decision table for rough set rule generation. As the method considers multiple objects in a class, a separate $n_k \times 3n$-dimensional attribute-value decision table is generated for each class d_k (where n_k indicates the number of objects in d_k). Let there be m sets $O_1, \ldots, O_m$ of objects in the table having identical attribute values, and $card(O_i) = n_{k_i}, i = 1, \ldots, m$, such that $n_{k_1} \geq \ldots \geq n_{k_m}$ and $\sum_{i=1}^{m} n_{k_i} = n_k$. The attribute-value table can now be represented as an $m \times 3n$ array. Let $n_{k'_1}, n_{k'_2}, \ldots, n_{k'_m}$ denote the distinct elements among $n_{k_1}, \ldots, n_{k_m}$ such that $n_{k'_1} > n_{k'_2} > \ldots > n_{k'_m}$. Let a heuristic threshold function be defined as [23]

$$Tr = \left\lceil \frac{\sum_{i=1}^{m} \frac{1}{n_{k'_i} - n_{k'_{i+1}}}}{Th} \right\rceil, \quad (9)$$

so that all entries having frequency less than Tr are eliminated from the table, resulting in the reduced attribute-value table. The main motive of introducing this threshold function lies in reducing the size of the case base and in eliminating the noisy patterns. From the reduced attribute-value table, thus obtained, rough dependency rules are generated using the methodology described in Section 2.4. More details on the methodologies for rule generation from high frequency reducts in large databases are available in [24]. Indiscernibility matrices of large size may also be handled using the "value reduct" approach [10].

4.2 Mapping Dependency Rules to Cases

We now describe the technique for mapping rough dependency rules to cases. The algorithm is based on the observation that each dependency rule (having frequency above some threshold) represent a cluster in the feature space. It may be noted that only a subset of features appears in each of the rules, this indicates the fact that the entire feature set is not always necessary to characterize a cluster. A *case* is contructed out of a *dependency rule* in the following manner:

1. Consider the antecedent part of a rule; split it into atomic formulas containing only conjunction of literals.
2. For each atomic formulae, generate a case—containing the centers and radii of the fuzzy linguistic variables ("low," "medium," and "high") which are present in the formula. (Thus, multiple cases may be generated from a rule.)
3. Associate with each such case generated, the precedent part of the rule and the case strength equal to the dependency factor of the rule (5). The strength factor reflect the size of the corresponding cluster and the significance of the case.

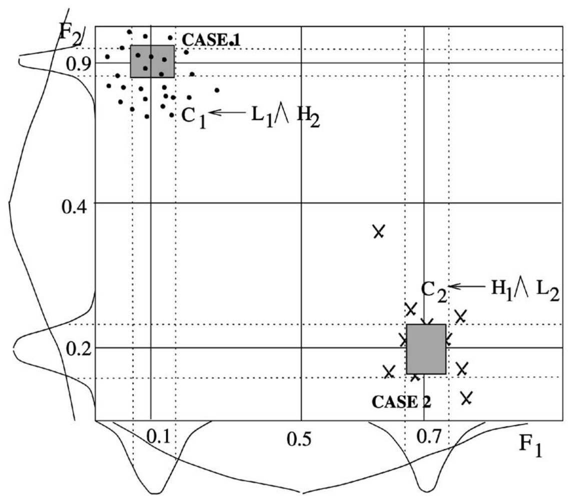


Fig. 5. Rough-Fuzzy case generation for a two-dimensional data.

Thus, a case has the following structure:

case{
Feature i: $fuzzset_i$: center, radius;
......
$Class\ k$
$Strength$
}

where $fuzzset$ denote the fuzzy sets "low," "medium," and "high." Note that the use of these fuzzy sets for representing the cases introduce "expert's bias" in the generated "case knowledge." The method is explained below with the help of an example.

One may note that while 0.5-cut is used to convert the $3n$ fuzzy membership functions of a pattern to binary ones for rough set rule generation (Section 4.1), the original fuzzy functions are retained in order to use them in for representing the generated cases (Section 4.2). These are also illustrated in Fig. 4, where the outputs μ^1_{low}, μ^2_{medium} are fuzzy sets (membership functions).

Example 2. Consider a data having two features F_1, F_2 and two classes as shown in Fig. 5. The granulated feature space has $3^2 = 9$ granules. These granules are characterized by three membership functions along each axis, and have ill-defined boundaries. Let the following two dependency rules be obtained from the reduced attribute table:

$$class_1 \leftarrow L_1 \wedge H_2,\ df = 0.5$$
$$class_2 \leftarrow H_1 \wedge L_2,\ df = 0.4.$$

Let the parameters of the fuzzy sets "low," "medium," and "high" be as follows:
Feature 1: $c_L = 0.1, \lambda_L = 0.5, c_M = 0.5, \lambda_M = 0.7, c_H = 0.7, \lambda_H = 0.4$.
Feature 2: $c_L = 0.2, \lambda_L = 0.5, c_M = 0.4, \lambda_M = 0.7, c_H = 0.9, \lambda_H = 0.5$.

Therefore, we have the following two cases:

$case$ 1{
Feature No: 1, $fuzzset$ (L): center=0.1, radius=0.5
Feature No: 2, $fuzzset$ (H): center=0.9, radius=0.5}

$Class = 1$
$Strength = 0.5$
}
$case\ 2${
Feature No: 1, $fuzzset$ (H): center=0.7, radius=0.4
Feature No: 2, $fuzzset$ (L): center=0.2, radius=0.5
$Class = 2$
$Strength = 0.4$
}

5 Case Retrieval

Each case thus obtained in the previous section is a collection of fuzzy sets $\{fuzzsets\}$ described by a set of one-dimensional π-membership functions with different c and λ values. To compute the similarity of an unknown pattern $\mathbf{F}$ (of dimension n) to a case $\mathbf{p}$ (of variable dimension n_p, $n_p \leq n$), we use

$$sim(\mathbf{F}, \mathbf{p}) = \sqrt{\frac{1}{n_p} \sum_{j=1}^{n_p} \left(\mu^{j}_{fuzzset}(F_j) \right)^2}, \tag{10}$$

where $\mu^j_{fuzzset}(F_j)$ is the degree of belongingness of the jth component of $\mathbf{F}$ to $fuzzset$ representing the case $\mathbf{p}$. When $\mu^j = 1$ for all j, $sim(\mathbf{F}, \mathbf{p}) = 1$ (maximum) and when $\mu^j = 0$ for all j, $sim(\mathbf{F}, \mathbf{p}) = 0$ (minimum). Therefore, (10) provides a collective measure computed over the degree of similarity of each component of the unknown pattern with the corresponding one of a stored case. Higher the value of the similarity, the closer the pattern $\mathbf{F}$ is to case $\mathbf{p}$. Note that fuzzy membership functions in (10) take care of the distribution of points within a granule; thereby providing a better similarity measure between $\mathbf{F}$ and $\mathbf{p}$ than the conventional Euclidean distance between two points.

For classifying (or to provide a label to) an unknown pattern, the case closest to the pattern, in terms of $sim(\mathbf{F}, \mathbf{p})$ measure, is retrieved and its class label is assigned to that pattern. Ties are resolved using the parameter $Case\ Strength$.

6 Results and Comparison

Experiments were performed on three real-life data sets, including the ones with a large number of samples and dimension. All the data sets are available in the UCI Machine Learning Archive [25]. The characterstics of the data sets are summarized below:

1. *Forest Covertype*: Contains 10 dimensions, seven classes, and 586,012 samples. It is a Geographical Information System data representing forest cover type (pine/fir, etc.) of the US. The variables are cartographic and remote sensing measurements. All the variables are numeric.
2. *Multiple features*: This data set consists of features of handwritten numerals ("0"-"9") extracted from a collection of Dutch utility maps. There are 2,000 patterns, 649 features (all numeric), and 10 classes total.
3. *Iris*: The data set contains 150 instances, four features, and three classes of Iris flowers. The features are numeric.

TABLE 4
Rough Dependence for the **Iris** Data

C_1	$\leftarrow$	$L_1 \wedge H_2 \wedge L_3$	$df = 0.81$
C_2	$\leftarrow$	$M_1 \wedge L_2 \wedge M_4$	$df = 0.81$
C_3	$\leftarrow$	$H_1 \wedge H_4$	$df = 0.77$

The cases generated using the rough-fuzzy methodology are compared with those obtained using the following three case selection methodologies:

1. Instance-based learning algorithm, IB3 [2].
2. Instance-based learning algorithm with reduced number of features, IB4 [26]. The feature weighting is learned by random hill climbing in IB4. A specified number of features having high weights is selected.
3. Random case selection.

A comparison is made on the basis of the following:

1. 1-NN classification accuracy using the generated/selected cases. For all the data, 10 percent of the samples are used as training set for case generation and 90 percent of the samples are used as a test set.
2. Number of cases stored in the case base.
3. Total CPU time required for case generation.
4. Average CPU time required to retrieve a case for the patterns in test set.

For the purpose of illustration, we present the rough dependency rules and the corresponding generated cases in Tables 4 and 5, respectively, for the Iris data, as an example. Comparitive results of the rough-fuzzy case generation methodology with other case selection algorithms are presented in Tables 6, 7, and 8 for the Iris, Forest Covertype, and Multiple features data, respectively, in terms of number of cases, 1-NN classification accuracy, average number of features per case (n_{avg}), and case generation (t_{gen}) and retrieval (t_{ret}) times. It can be seen from the tables that the cases obtained using the proposed rough-fuzzy methodology are much superior to random selection method and IB4, and close to IB3 in terms of classification accuracy. The method requires significantly less time compared to IB3 and IB4 for case generation. As is seen from the tables, the average number of features stored per case (n_{avg}) by the rough-fuzzy technique is much less than the original data dimension (n). As a consequence, the average retrieval time required is very low. IB4 also stores cases with a reduced number of features and has a low retrieval time, but its accuracy is much less compared to the proposed method. Moreover, all the cases involve equal number of features, unlike ours.

7 Conclusions and Discussion

We have presented a case generation methodology based on rough-fuzzy hybridization. Fuzzy set theory is used to represent a pattern in terms of its membership to linguistic variables. This gives rise to efficient fuzzy granulation of the

TABLE 5
Cases Generated for the **Iris** Data

case 1{
Feature No: 1, *fuzzset*(L): center=5.19, radius=0.65
Feature No: 2, *fuzzset* (H): center=3.43, radius=0.37
Feature No: 3, *fuzzset* (L): center=0.37, radius=0.82
Class=1
Strength=0.81
}
case 2{
Feature No: 1, *fuzzset*(M): center=3.05, radius=0.34
Feature No: 2, *fuzzset* (L): center=1.70, radius=2.05
Feature No: 4, *fuzzset* (M): center=1.20, radius=0.68
Class=2
Strength=0.81
}
case 3{
Feature No: 1, *fuzzset*(H): center=6.58, radius=0.74
Feature No: 4, *fuzzset* (H): center=1.74, radius=0.54
Class=3
Strength=0.77
}

feature sapce. On the granular universe thus obtained, rough sets are used to form reducts which contain *informative* and *irreducible* information both in terms of features and patterns. The fuzzy linguistic rules obtained from the reducts represent different clusters in the granular feature space. Granular clusters (regions), modeled by the rules, are mapped to different cases, represented by fuzzy membership functions. Note that the above representation introduces expert's bias in the case knowledge, i.e., in the cases ultimately generated.

Since the rough set theory is used to obtain cases through crude rules (i.e., it deals with information granules, and not the original data), case generation time is reduced. Also, since each case involves a reduced number of relevant features, only the informative regions and the relevant subset of features are stored. As a result, case retrieval time decreases significantly. The algorithm is suitable for mining data sets, large both in dimension and size, where the requirement is to achieve an approximate but effective solution fast.

Note that we have used three fuzzy property sets "low," "medium," and "high." One may consider hedges like "very," "more or less" to generate more granules, i.e., finer granulated space. However, this will enhance the computational requirement for both case generation and retrieval.

TABLE 6
Comparison of Case Selection Algorithms for **Iris** Data

Algorithm	No. of Cases	n_{avg}	Classification accuracy	t_{gen} (sec)	t_{ret} (sec)
Rough-fuzzy	3	2.67	98.17	0.2	0.005
IB3	3	4	98.00	2.50	0.01
IB4	3	4	90.01	4.01	0.01
Random	3	4	87.19	0.01	0.01

TABLE 7
Comparison of Case Selection Algorithms for **Forest** Data

Algorithm	No. of Cases	n_{avg}	Classification accuracy	t_{gen} (sec)	t_{ret} (sec)
Rough-fuzzy	542	4.10	67.01	244	4.4
IB3	545	10	66.88	4055	52.0
IB4	545	4	50.05	7021	4.5
Random	545	10	41.02	17	52.0

TABLE 8
Comparison of Case Selection Algorithms for **Multiple Features** Data

Algorithm	No. of Cases	n_{avg}	Classification accuracy	t_{gen} (sec)	t_{ret} (sec)
Rough-fuzzy	50	20.87	77.01	1096	10.05
IB3	52	649	78.99	4112	507
IB4	52	21	41.00	8009	20.02
Random	50	649	50.02	8.01	507

ACKNOWLEDGMENTS

Financial assistance from the Council of Scientific and Idustrial Research, New Delhi, India, through grant no. 22/346/02-EMR II is gratefully acknowledged.

REFERENCES

[1] J.L. Kolodner, *Case-Based Reasoning.* San Mateo, Calif.: Morgan Kaufmann, 1993.
[2] D.W. Aha, D. Kibler, and M.K. Albert, "Instance-Based Learning Algorithms," *Machine Learning,* vol. 6, pp. 37-66, 1991.
[3] D.R. Wilson and T.R. Martinez, "Reduction Techniques for Instance-Based Learning Algorithms," *Machine Learning,* vol. 38, no. 3, pp. 257-286, 2000.
[4] S.K. Pal, T.S. Dillon, and D.S. Yeung, *Soft Computing in Case-Based Reasoning.* London: Springer Verlag, 2000.
[5] H.-D. Burkhard and M.M. Richter, "On the Notion of Similarity in Case Based Reasoning and Fuzzy Theory," *Soft Computing in Case-Based Reasoning,* S.K. Pal, T.S. Dillon, and D.S. Yeung, eds., pp. 29-45, London: Springer Verlag, 2000.
[6] D. Dubois and H. Prade, "Fuzzy Set Modeling in Case Based Reasoning," *Int'l J. Intelligent Systems,* vol. 13, pp. 345-373, 1998.
[7] W. Dubitzky, A. Schuster, J. Hughes, and D. Bell, "An Advanced Case Knowledge Architecture Based on Fuzzy Objects," *Applied Intelligence,* vol. 7, pp. 187-204, 1997.

[8] R.K. De and S.K. Pal, "A Connectionist Model for Selection of Cases," *Information Sciences,* vol. 132, pp. 179-194, 2001.

[9] Z. Pawlak, "Rough Sets," *Int'l J. Computer and Information Sciences,* vol. 11, pp. 341-356, 1982.

[10] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data.* Dordrecht: Kluwer Academic, 1991.

[11] L. Polkowski, A. Skowron, and J. Komorowski, "Approximate Case-Based Reasoning: A Rough Mereological Approach," *Proc. Fourth German Workshop Case-Based Reasoning, System Deveolpment, and Evaluation,* pp. 144-151, 1996.

[12] *Proc. Int'l Workshop Rough Set Theory and Granular Computing.* May 2001.

[13] *Rough Fuzzy Hybridization: New Trends in Decision Making,* S.K. Pal and A. Skowron, eds. Singapore: Springer Verlag, 1999.

[14] A. Skowron and C. Rauszer, "The Discernibility Matrices and Functions in Information Systems," *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory,* R. Slowiński, ed. pp. 331-362, Dordrecht: Kluwer Academic, 1992.

[15] S.K. Pal and S. Mitra, "Multi-Layer Perceptron, Fuzzy Sets and Classification," *IEEE Trans. Neural Networks,* vol. 3, pp. 683-697, 1992.

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

[17] L.A. Zadeh, "Calculus of Fuzzy Restrictions," *Fuzzy Sets and Their Applications to Cognitive and Decision Processes,* L.A. Zadeh, K.S. Fu, K. Tanaka, and M. Shimura, eds. London: Academic Press, 1975.

[18] L.A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 3, pp. 28-44, 1973.

[19] W. Pedrycz and Z.A. Sosnowski, "Designing Decision Trees with the Use of Fuzzy Granulation," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 30, no. 2. pp. 151-159, 2000.

[20] L.A. Zadeh, "Toward a Theory of Fuzzy Information Granulation and Its Centrality in Human Reasoning and Fuzzy Logic," *Fuzzy Sets and Systems,* vol. 90, pp. 111-127, 1997.

[21] T.Y. Lin, "Granular Computing: Fuzzy Logic and Rough Sets," *Computing with Words in Information/Intelligent Systems,* L.A. Zadeh and J. Kacprzyk, eds. pp. 183-200, Physica-Verlag, Heidelberg 1999.

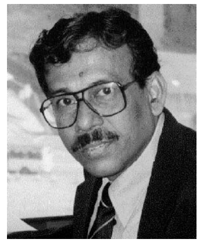
[22] L.A. Zadeh, "Fuzzy-Algorithmic Approach to the Definition of Complex or Imprecise Concepts," *J. Man-Machine Studies,* vol. 8, pp. 249-291, 1976.

[23] M. Banerjee, S. Mitra, and S.K. Pal, "Rough Fuzzy MLP: Knowledge Encoding and Classification," *IEEE Trans. Neural Networks,* vol. 9, no. 6, pp. 1203-1216, 1998.

[24] T.Y. Lin and R. Chen, "Finding Reducts in Very Large Databases," *Proc. Joint Conf. Information Science Research,* pp. 350-352, 1997.

[25] C.L. Blake and C.J. Merz UCI Repository of Machine Learning Databases, Univ. of California, Irvine, Dept. of Information and Computer Sciences, http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[26] D.W. Aha, "Tolerating Noisy, Irrelevant, and Novel Attributes in Instance-Based Learning Algorithms," *Int'l J. Man-Machine Studies,* vol. 36, pp. 267-287, 1992.



Sankar K. Pal (M'81-SM'84-F'93) (04230363) received the MTech and PhD degrees in radio physics and electronics in 1974 and 1979, respectively, from the University of Calcutta. In 1982, he received another PhD degree in electrical engineering along with the DIC degree from Imperial College, University of London. He is a professor and distinguished scientist at the Indian Statistical Institute, Calcutta. He is also the founding head of the Machine Intelligence Unit. From 1986-1987, he worked at the University of California, Berkeley, and the University of Maryland, College Park, as a Fulbright postdoctoral visiting fellow, and from 1990-1992 and in 1994, at the NASA Johnson Space Center, Houston, Texas as an NRC guest investigator. He served as a distinguished visitor of IEEE Computer Society (US) for the Asia-Pacific Region from 1997-1999, and he held several visiting positions in Hong Kong and Australian universities from 1999-2002. Professor Pal is a fellow of the IEEE, the Third World Academy of Sciences, Italy, the International Association for Pattern Recognition, and all four National Academies for Science/Engineering in India. His research interests include pattern recognition and machine learning, image processing, data mining, soft computing, neural nets, genetic algorithms, fuzzy sets, and rough sets. He is a coauthor of 10 books and about 300 research publications. He received the 1990 S.S. Bhatnagar Prize (which is the most coveted award for a scientist in India) and many prestigious awards in India and abroad including the 1993 Jawaharlal Nehru Fellowship, 1993 Vikram Sarabhai Research Award, 1993 NASA Tech Brief Award (US), 1994 IEEE Transactions on Neural Networks Outstanding Paper Award (US), 1995 NASA Patent Application Award (US), 1997 IETE—Ram Lal Wadhwa Gold Medal, 1998 Om Bhasin Foundation Award, 1999 G.D. Birla Award for Scientific Research, 2000 Khwarizmi International Award (first winner) from the Islamic Republic of Iran, the 2001 INSA-Syed Husain Zaheer Medal, and the FICCI Award 2000-2001 in Engineering and Technology. Professor Pal served/serving as an associate editor and a guest editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Neural Networks*, *IEEE Computer*, *Pattern Recognition Letters*, *Neurocomputing*, *Applied Intelligence*, *Information Sciences*, *Fuzzy Sets and Systems*, *Fundamenta Informaticae*, and the *International Journal of Computational Intelligence and Applications*; and as a member, executive advisory editorial board, *IEEE Transactions on Fuzzy Systems*, *International Journal on Image and Graphics*, and the *International Journal of Approximate Reasoning*.

Pabitra Mitra received the BTech degree in electrical engineering from the Indian Insitute of Technology, Kharagpur, in 1996. He worked as a scientist with the Institute for Robotics and Intelligent Systems, India. Currently, he is a senior research fellow of the Indian Statistical Institute, Calcutta. His research interests are in the areas of data mining and knowledge discovery, pattern recognition, learning theory, and soft computing. He is a student member of the IEEE and the IEEE Computer Society.