Applied Intelligence 21, 289–299, 2004

Rough Self Organizing Map

SANKAR K. PAL, BISWARUP DASGUPTA AND PABITRA MITRA
Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India
sankar@isical.ac.in
biswarupdg@yahoo.com
pabitra_r@isical.ac.in

Abstract. A rough self-organizing map (RSOM) with fuzzy discretization of feature space is described here. Discernibility reducts obtained using rough set theory are used to extract domain knowledge in an unsupervised framework. Reducts are then used to determine the initial weights of the network, which are further refined using competitive learning. Superiority of this network in terms of quality of clusters, learning time and representation of data is demonstrated quantitatively through experiments over the conventional SOM.

Keywords: soft computing, pattern recognition, self organization, rough sets, fuzzy sets, data mining, case generation

1. Introduction

Rough set theory [1] provides an effective means for classificatory analysis of data tables. The main goal of rough set theoretic analysis is to synthesize or construct approximations (upper and lower) of concepts from the acquired data. The key concepts here are those of "information granule" and "reducts". Information granule formalizes the concept of finite precision representation of objects in real life situations, and the reducts represent the *core* of an information system (both in terms of objects and features) in a granular universe. An important use of rough set theory has been in generating logical rules for classification and association [2]. These logical rules correspond to different important granulated regions of the feature space, which represent data clusters.

Recently rough sets have been integrated with neural networks [3]. In the framework of rough-neuro integration research has been done in the use of rough sets for encoding weights of knowledge-based networks. However, mainly layered networks in supervised learning framework have been considered so far [4]. This article is an attempt to incorporate rough set methodology in the framework of unsupervised networks.

Self-organizing map (SOM) [5] is an unsupervised network which has been recently popular for unsupervised mining of large data sets. The process of self-organization generates a network whose weights represent prototypes of the input data. These prototypes may be considered as cases representing the entire data set. Unlike the ones produced by existing case generation methodologies, they are not just subset of the original data but evolved in the self organizing process. Neuro-fuzzy systems have also been used for generation of cases [6, 7]. This includes mainly the use of layered network in supervised framework. In the present investigation we consider unsupervised framework using a SOM. Since SOM suffers from the problem of slow convergence and local minima, we integrate rough set theory with SOM synergistically to offer a fast and robust solution to the initialization and local minima problem; thereby designing Rough-SOM (RSOM). Here rough set theoretic knowledge is used to encode the weights as well as to determine the network size. Fuzzy set theory is used for discretization of feature space. Performance of the network is measured in terms of learning time, representation error, cluster quality and network compactness. All these characteristics have been demonstrated with three

data sets and compared with that of the conventional SOM.

2. Rough Sets

2.1. Definitions

Here, we present some preliminaries of rough set theory, which are relevant to this article.

An information system is a pair $S = \langle U,\ A\rangle$, where U is a non-empty finite set called the universe and A is a non-empty finite set of attributes. An attribute a can be regarded as a function from the domain U to some value set V_a.

An information system may be represented as an attribute-value table, in which rows are labeled by objects of the universe and columns by the attributes.

With every subset of attributes $B \subseteq A$, one can easily associate an equivalence relation I_B on U:

$$I_B = \{(x, y) \in U\text{: for every } a \in B, a(x) = a(y)\}.$$

Then

$$I_B = \bigcap_{a \in B} I_a.$$

If $X \subseteq U$, the sets $\{x \in U : [x]_B \subseteq X\}$ and $[x \in U : [x]_B \cap X \neq \Phi]$, 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 S and denoted by $\underline{B}X$, $\bar{B}X$ respectively.

$X(\subseteq U)$ is B-exact or B-definable in S if $\underline{B}X = \bar{B}X$. It may be observed that $\underline{B}X$ is the greatest B-definable set contained in X, and $\bar{B}X$ is the smallest B-definable set containing X.

We now define the notions relevant to knowledge reduction. The aim is to obtain irreducible but essential parts of the knowledge encoded by the given information system; these would constitute reducts of the system. So one is, in effect, looking for the maximal sets of attributes taken from the initial set (A, say), which induce the same partition on the domain as A. In other words, the essence of the information remains intact, and superfluous attributes are removed. Reducts have been nicely characterized in [2] by discernibility matrices and discernibility functions. Consider $U = \{x_1, \ldots, x_n\}$ and $A = \{a_1, \ldots, a_m\}$ in the information system $S = \langle U, A\rangle$. By the discernibility matrix $M(S)$ of S is meant an $n \times n$ matrix such that

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

A discernibility function f_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:

$$f_s(\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_m) = \wedge\{\vee c_{ij} : 1 \leq i, j \leq n, j < i, c_{ij} \neq \phi\} \tag{2}$$

where $\vee(c_{ij})$ is the disjunction of all variables $\bar{a}$ with $a \in c_{ij}$. It is seen in [2] that $\{a_{i1}, \ldots, a_{ip}\}$ is a reduct of S if and only if $a_{i1} \wedge \cdots \wedge a_{ip}$ is a prime implicant (constituent of the disjunctive normal form) of f_s.

2.2. Indiscernility of Patterns and Fuzzy Discretization of the Feature Space

A primary notion of rough set is of indescernibility relation. For continuous valued attributes the feature space needs to be discretized for defining indiscernibility relations and equivalence classes. Discretization is a widely studied problem in rough set theory and in this article we use fuzzy set theory for effective discretization. Use of fuzzy sets has several advantages over 'hard' discretization, like modelling of overlapped clusters, linguistic representation of data. We discretize each feature into three levels low, medium and high; finer discretizations may lead to better accuracy at the cost of higher computational load.

Each feature of a pattern is described in terms of their fuzzy membership values in the linguistic property sets *low* (L), *medium* (M) and *high* (H). Let these be represented by L_j, M_j and H_j respectively. The features for the ith pattern $\mathbf{F}_i$ are mapped to the corresponding three-dimensional feature space of $\mu_{\text{low}(F_{ij})}(\mathbf{F}_i)$, $\mu_{\text{medium}(F_{ij})}(\mathbf{F}_i)$ and $\mu_{\text{high}(F_{ij})}(\mathbf{F}_i)$ by Eq. (3). An n-dimensional pattern $\mathbf{F}_i = [F_{i1}, F_{i2}, \ldots, F_{\text{in}}]$ is represented as an $3n$-dimensional vector [8]

$$\mathbf{F}_i = \left[\mu_{\text{low}(F_{\text{i1}})}(\mathbf{F}_i), \ldots\ldots, \mu_{\text{high}(F_{\text{in}})}(\mathbf{F}_i)\right], \tag{3}$$

where the μ values indicate the membership functions of the corresponding linguistic Π-sets *low*, *medium* and *high* along each feature axis. This effectively discretizes each feature into three levels.

Then consider only those attributes which have a numerical value greater than some threshold TH (= 0.5,

say). This implies clamping only those features demonstrating high membership values with unity, while the others are fixed at zero. An attribute-value table is constructed comprising the above binary valued $3n$-dimensional feature vectors.

We use the Π-fuzzy sets (in the one dimensional form), with range [0,1], represented as

$$\Pi(F_j; c, \lambda) = \begin{cases} 2(1 - \|F_j - c\|/\lambda)^2 & \text{for } \lambda/2 \le \|F_j - c\| \le \lambda, \\ 1 - 2(\|F_j - c\|/\lambda)^2 & \text{for } 0 \le \|F_j - c\| \le \lambda/2, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where λ (>0) is the radius of theΠ-function with c as the central point. The details of the above method may be found in [8].

Let us now explain the procedure for selecting centers (c) and radii (λ) of the overlapping Π-sets. Let m_j be the mean of the pattern points along jth axis. Then m_{jl} and m_{jh} are defined as the mean (along 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 (for the training set) considering numerical values only. For the three linguistic property sets along the jth axis, the centers and the corresponding radii of the corresponding Π-functions are defined as

$$\begin{aligned} c_{\text{low}(Fj)} &= m_{jl} \\ c_{\text{medium}(Fj)} &= m_j \\ c_{\text{high}(Fj)} &= m_{jh} \\ \lambda_{\text{low}(Fj)} &= c_{\text{medium}(Fj)} - c_{\text{low}(Fj)} \\ \lambda_{\text{high}(Fj)} &= c_{\text{high}(Fj)} - c_{\text{medium}(Fj)} \\ \lambda_{\text{medium}(Fj)} &= c_{\text{high}(Fj)} - c_{\text{low}(Fj)} \end{aligned} \tag{5}$$

respectively. 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 properties. The nature of membership functions is illustrated in Fig. 1.

2.3. *Methodology for Generation of Reducts and Dependency Rules*

Let there be m sets $O_1, \ldots, O_m$ of objects in the attribute-value table (obtained by the procedure described in the previous section) having identical attribute values, and $card\,(O_i) = n_{ki}, i = 1, \ldots, m$ such that $n_{k_1} > n_{k_2} > \cdots > n_{km}$ and

$$\sum_{i=1}^{m} n_{ki} = n_k.$$

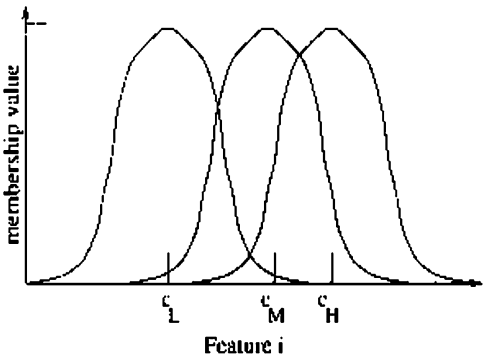

Figure 1. Π-Membership functions for linguistic property sets *low*(L), *medium*(M) and *high*(H) for each feature axis.

The attribute-value table can now be represented as an $m \times 3n$ array. Let $n'_{k_1}, n'_{k_2}, \ldots, n'_{\text{km}}$ denote the distinct elements among $n_{k1}, n_{k2} \cdots n_{km}$ such that $n'_{k_1} > n'_{k_2} > \cdots > n'_{k_m}$.

Let a heuristic threshold be defined as [4]

$$\text{Tr} = \left\lceil \frac{\sum_{i=1}^{m} \frac{1}{n_{k'_i} - n_{k'_{i+1}}}}{TH} \right\rceil \tag{6}$$

so that all entries having frequency less than Tr are eliminated from the table, resulting in the reduced attribute-value table S. Note that the main motive of introducing this threshold function lies in reducing the size of the model. One attempts to eliminate noisy pattern representatives (having lower values of n_{ki}) from the reduced attribute-value table. From the reduced attribute-value table obtained, reducts are determined using the methodology described below.

Let $\{x_{i1}, \ldots, x_{ip}\}$ be the set of those objects of U that occur in S. Now a discernibility matrix (denoted $M(B)$) is defined as follows:

$$c_{ij} = \{a \in B : a(xi) \neq a(x_j)\} \quad \text{for } i, j = 1, \ldots, n. \tag{7}$$

For each object $x_j \in x_{i1}, \ldots, x_{ip}$, the discernibility function $f_{d_i}^{x_j}$ is defined as

$$f_{x_j} = \wedge\{\vee(c_{ij}) : 1 \le i, j \le n, j < i, c_{ij} \neq \phi\}, \tag{8}$$

where $\vee(c_{ij})$ is the disjunction of all members of c_{ij}. One thus obtains a dependency rule r_i, viz. $P_i \rightarrow cluster_i$, where P_i is the disjunctive normal form (d.n.f.) of f_{x_j}, $j \in i_1, \ldots, i_p$.

3. Rough-SOM

3.1. Self-Organizing Maps

The Kohonen feature map is a two-layered network. The first layer of the network is the input layer. The second layer, called the competitive layer, is usually organized as a two-dimensional grid. All interconnections go from the first layer to the second (Fig. 2).

All the nodes in the competitive layer compare the inputs with their weights and compete with each other to become the winning unit having the lowest difference. The basic idea underlying what is called competitive learning is roughly as follows: Assume a sequence of input vectors $\{x = x(t) \in R^n$, where t is the time coordinate $\}$ and a set of variable reference vectors $\{m_i(t)$: $m_i \in R^n$, $i = 1, 2, \ldots, k$ where k is the number of units in the competitive layer $\}$. Initially the values of the reference vectors (also called weight vectors) are set randomly. At each successive instant of time t, an input pattern $x(t)$ is presented to the network. The input pattern $x(t)$ is then compared with each $m_i(t)$ and the best matching $m_i(t)$ is updated to match even more closely the current $x(t)$.

If the comparison is based on some distance measure $d(x, m_i)$, altering m_i must be such that, if $i = c$ the index of the best-matching reference vector, then $d(x, m_c)$ is reduced, and all the other reference vectors m_i, with $i \neq c$, are left intact. In this way the different reference vectors tend to become specifically "tuned" to different domains of the input variable x.

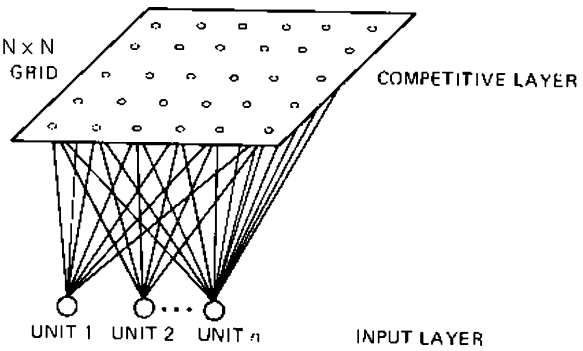

Figure 2. The basic network structure for the Kohonen feature map.

3.1.1. Learning. The first step in the operation of a Kohonen network is to compute a matching value for each unit in the competitive layer. This value measures the extent to which the weights or reference vectors of each unit match the corresponding values of the input pattern. The matching value for each unit i is $\|x - m_i\|$ which is the distance between vectors x and m_i and is computed by

$$\sqrt{\sum_j (x_j - m_{ij})^2} \quad \text{for } j = 1, 2, \ldots, n. \tag{9}$$

The unit with the lowest matching value (the best match) wins the competition. In other words, the unit c is said to be the best matched unit if

$$\|x - m_c\| = \min_i \{\|x - m_i\|\}, \tag{10}$$

where the minimum is taken over all units i in the competitive layer. If two units have the same matching value, then by convention, the unit with the lower index value i is chosen.

The next step is to self-organize a two-dimensional map that reflects the distribution of input patterns. In biophysically inspired neural network models, correlated learning by spatially neighboring cells can be implemented using various kinds of lateral feedback connections and other lateral interactions. Here the lateral interaction is enforced directly in a general form, for arbitrary underlying network structures, by defining a neighborhood set N_c around the winning cell. At each learning step, all the cells within N_c are updated, whereas cells outside N_c are left intact. The update equation is:

$$\Delta m_{ij} = \begin{cases} \alpha(x_j - m_{ij}) & \text{if unit } i \text{ is in the neighborhood } N_c, \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

and

$$m_{ij}^{\text{new}} = m_{ij}^{\text{old}} + \Delta m_{ij}. \tag{12}$$

Here α is the learning parameter. This adjustment results in both the winning unit and its neighbors, having their weights modified, becoming more like the input pattern. The winner then becomes more likely to win

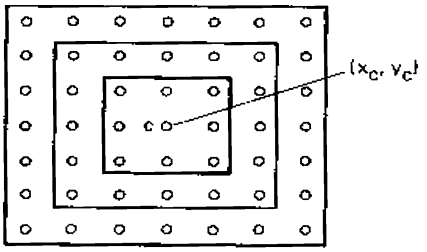

Figure 3. Neighborhood N_c, centered on unit $c(x_c, y_c)$. Three different neighborhoods are shown at distance $d = 1, 2$ and 3.

the competition should the same or a similar input pattern be presented subsequently.

3.1.2. Effect of Neighborhood. The width or radius of N_c can be time-variable; in fact, for good global ordering, it has experimentally turned out to be advantageous to let N_c be very wide in the beginning and shrink monotonically with time (Fig. 3). This is because a wide initial N_c, corresponding to a coarse spatial resolution in the learning process, first induces a rough global order in the m_i values, after which narrowing of N_c improves the spatial resolution of the map; the acquired global order, however, is not destroyed later on. This allows the topological order of the map to be formed.

3.2. Incorporation of Rough Sets in SOM

As described in Section 2.3, the dependency rules generated using rough set theory from an information system are used to discern objects with respect to their attributes. However the dependency rules generated by rough set are coarse and are therefore needed to be fine-tuned. Here we have used the rough set dependency rules to get a crude knowledge of the cluster boundaries of the input patterns to be fed to a self-organizing map. These crude knowledge is used to encode the initial weights of the nodes of the map, which is then trained using the usual learning process (Section 3.1.1). Since an initial knowledge about the cluster boundaries is encoded into the network, the learning time reduces greatly with improved performance.

The steps involved in the process are summarized below:

1. From the initial data set, use fuzzy discretization process to create the information system.
2. For each object in the information table, generate the discernibility function

$$f_A(\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_{3n}) = \wedge\{\vee c_{ij} \| 1 \le j \le i \le n, c_{ij} \ne \phi\} \quad (13)$$

   where $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_{3n}$ are the $3n$ Boolean variables corresponding to the attributes $a_1, a_2, \ldots, a_{3n}$ of each object in the information system. The expression f_A is reduced to the set of all prime implicants of f_A that determines the set of all reducts of A.
3. The self-organizing map is created with $3n$ inputs (Section 2.2), which correspond to the attributes of the information table, and a competitive layer of $N \times N$ grid of units where N is the total number of implicants present in discernibility functions of all the objects of the information table.
4. Each implicant of the function f_A is mapped to a unit in the competitive layer of the network and high weights are given to those links that come from the attributes, which occur in the implicant expression. The idea behind this is that when an input pattern belonging to an object, say O_i, is applied to the inputs of the network, one of the implicants of the discernibility function of O_i will be satisfied and the corresponding unit in the competitive layer will fire and emerge as the winning unit. All the implicants of an object O_i are placed in the same layer while the implicants of different objects are placed in different layers separated by the maximum neighborhood distance. In this way the initial knowledge obtained with rough set methodology is used to train the SOM. This is explained with the following example.

   Let the reduct of an object O_i be

$$O_i : (F_{1\text{low}} \wedge F_{2\text{medium}}) \vee (F_{1\text{high}} \wedge F_{2\text{high}})$$

   where $F_{(\cdot)\text{low}}$, $F_{(\cdot)\text{medium}}$ and $F_{(\cdot)\text{high}}$ represent the low, medium and high values of the corresponding features.

   Then the implicants are mapped to the nodes of the layer in the following manner. Here high weights (H) are given only to those links which come from the features present in the implicant expression. Other links are given low weights.

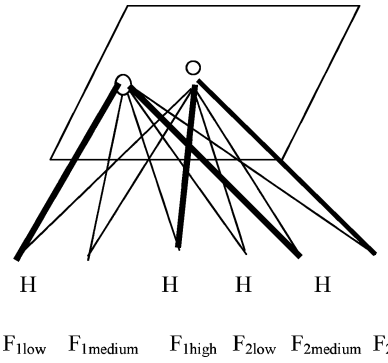

Figure 4. Mapping of reducts in the competitive layer of RSOM.

4. Experimental Results

4.1. Data Sets Used

We have considered three data sets for our experiment. The first data set (Fig. 5) consists of 2 features containing 417 points from 2 horse-shoe shaped clusters. The second data set is the speech data "Vowel" that deals with 871 Indian Telegu vowel sounds [9]. These were uttered in a consonant-vowel-consonant context by three male speakers in the age group of 30 to 35 years. The data set has three features: F_1, F_2 and F_3 corresponding to the first, second and third vowel format frequencies obtained through spectrum analysis of the speech data. Figure 6 depicts the projection in the F_1–F_2 plane, of the six vowel classes δ, a, i, u, e, o. These overlapping classes are denoted by $c_1, c_2, \ldots, c_6$. The third data set is the medical data consisting of nine input features and four pattern classes, and it deals with various *Hepatobiliary disorders* of 536 patient cases [10]. The input features are the results of different biochemical tests, viz., Glutamic Oxalacetic Transaminate (GOT, Karmen unit), Glutamic Pyruvic Transaminase (GPT, Karmen unit), Lactate Dehydrase (LDH, iu/l), Gamma Glutamyl Transpeptidase (GGT, mu/ml), Blood Urea Nitrogen (BUN, mg/dl), Mean Corpuscular Haemoglobin (MCH, pg), Total Bilirubin (Tbil, mg/dl) and Creatinine (CRTNN, mg/dl). The hepatobiliary disorders Alcoholic Liver Damage (ALD), Primary Hepatoma (PH), Liver Cirrhosis (LC) and Cholelithiasis (C), constitute the four classes. These are referred to as c_1, c_2, c_3, c_4.

As an illustration of the parameters of the fuzzy membership functions and the rough set reducts, we mention them below only for the horse-shoe data.

$$c_{\text{low}(F1)} = 0.223095$$
$$c_{\text{medium}(F1)} = 0.499258$$
$$c_{\text{high}(F1)} = 0.753786$$
$$\lambda_{\text{low}(F1)} = 0.276163$$

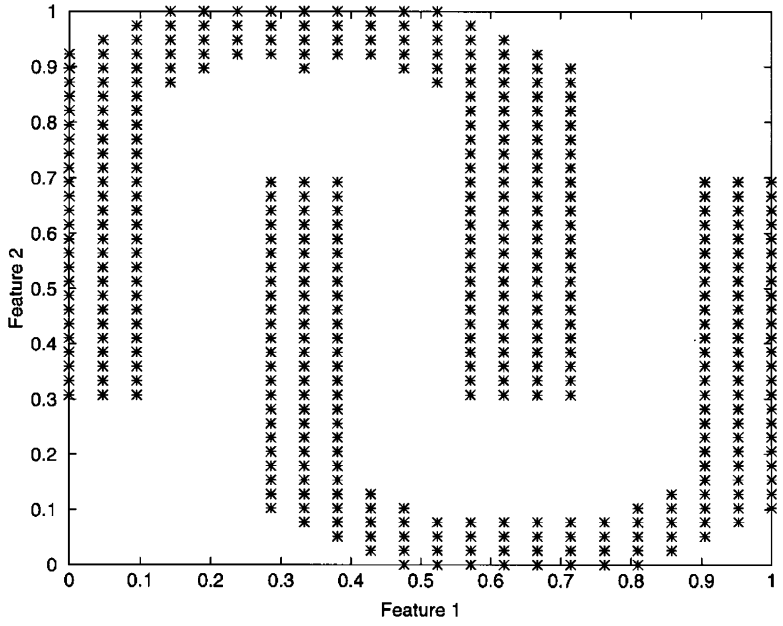

Figure 5. Horse-shoe data.

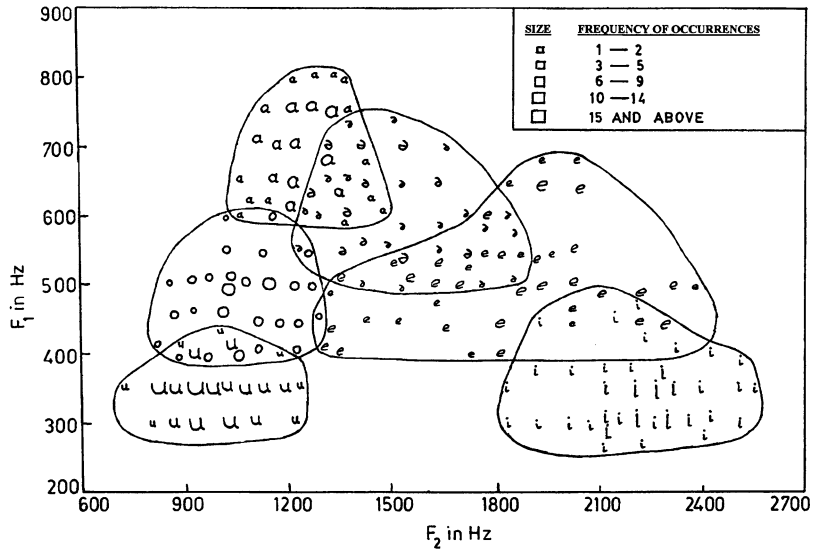

Figure 6. Vowel data.

$$\lambda_{\text{medium}(F1)} = 0.254528$$
$$\lambda_{\text{high}(F1)} = 0.265345$$
$$c_{\text{low}(F2)} = 0.263265$$
$$c_{\text{medium}(F2)} = 0.511283$$
$$c_{\text{high}(F2)} = 0.744306$$
$$\lambda_{\text{low}(F2)} = 0.248019$$
$$\lambda_{\text{medium}(F2)} = 0.233022$$
$$\lambda_{\text{high}(F2)} = 0.240521$$
$$O_1 : (F_{1\text{low}} \wedge F_{2\text{medium}}) \vee (F_{1\text{high}} \wedge F_{2\text{medium}})$$
$$O_2 : (F_{1\text{low}} \wedge F_{2\text{high}})$$
$$O_3 : (F_{1\text{high}} \wedge F_{2\text{low}})$$
$$O_4 : (F_{1\text{medium}} \wedge F_{2\text{high}}) \vee (F_{1\text{medium}} \wedge F_{2\text{low}}).$$

4.2. *Results*

To demonstrate the effectiveness of the proposed knowledge-encoding scheme (RSOM), its performance is compared with that of the randomly initialized self-organized map. The following quantities are considered for comparison.

4.2.1. Quantization Error. The quantization error (q_E) measures how fast the weight vectors of the winning nodes in the competitive layer are aligning themselves with the input vectors presented during training. It is calculated by the following equation:

$$q_E = \frac{\sum_{p=1}^{n} \left(\sum_{\text{all winning nodes}} \sqrt{\left(\sum_{j} (x_{pj} - m_j)^2 \right)} \right)}{\text{number of patterns}}, \tag{14}$$

Here $j = 1, \ldots, m$, m being the number of input features to the net, x_{pj} is the jth component of pth pattern and n is the total number of patterns. Hence, higher the quantization error (q_E), more is the difference between the reference vectors and the input vectors of the nodes in the competitive layer.

4.2.2. Entropy and β-Index. For measuring the quality of cluster structure we have used two indices, namely, an Entropy measure [11] and β-index [12]. These are defined below.

Entropy:

Let the distance between two weight vectors p, q be

$$D_{pq} = \left[\sum_{j} \left(\frac{x_{pj} - x_{qj}}{\max_j - \min_j} \right)^2 \right]^{\frac{1}{2}}, \tag{15}$$

where x_{pj} and x_{qj} denote the weight values for p and q respectively along the jth direction, and $j =$

$1, \ldots, m$, m being the number of features input to the net. $\max_j, \min_j$ are respectively the maximum and minimum values computed over all the samples along jth axis.

Let the similarity between p, q be defined as

$$\text{sim}(p, q) = e^{-\beta D_{pq}}, \tag{16}$$

where $\beta = \frac{-\ln 0.5}{\bar{D}}$, a positive constant such that

$$\text{sim}(p, q) = \begin{cases} 1 & \text{if } D_{pq} = 0, \\ 0 & \text{if } D_{pq} = \infty, \\ 0.5 & \text{if } D_{pq} = \bar{D}. \end{cases}$$

$\bar{D}$ is the average distance between points computed over the entire dataset. Entropy is defined as

$$E = -\sum_{p=1}^{l}\sum_{q=1}^{l} (\text{sim}(p, q) \times \log \text{sim}(p, q) + (1 - \text{sim}(p, q)) \times \log(1 - \text{sim}(p, q))). \tag{17}$$

If the data is uniformly distributed in the feature space entropy is maximum. When the data has well-formed clusters uncertainty is low and so is entropy.

β-index:

β-index [12] is defined as:

$$\beta = \frac{\sum_{i=1}^{k}\sum_{p=1}^{n_i}\left(X_p^i - \bar{X}\right)^T\left(X_p^i - \bar{X}\right)}{\sum_{i=1}^{k}\sum_{p=1}^{n_i}\left(X_p^i - \bar{X}^i\right)^T\left(X_p^i - \bar{X}^i\right)}, \tag{18}$$

where n_i is the number of points in the ith ($i = 1, \ldots, k$) cluster, X_p^i is the pth pattern ($p = 1, \ldots, n_i$) in cluster i, $\bar{X}^i$ the mean of n_i patterns of the ith cluster, $\sum_i n_i = n$, where n is the total number of patterns, and $\bar{X}$ is the mean value of the entire set of patterns. Note that β is nothing but the ratio of the total variation and within-cluster variation. This type of measure is widely used for feature selection and cluster analysis. For a given data and k (number of clusters) value, the higher the homogeneity within the clustered regions, higher would be the β value.

4.2.3. Frequency of Winning Nodes (f_k). Here we have used the number of winning of top k nodes (f_k) in the competitive layer, where k is the number of rules (characterizing the clusters) obtained using rough sets. Here $k = 4$ for horse-shoe data, $k = 14$ for vowel data and $k = 7$ for medical data. f_k reflects the error if all but k nodes would have been pruned. In other words, it measures the number of sample points correctly represented by these nodes.

4.2.4. Number of Iterations. We compute the number of iterations at which the error does not change much.

The comparative results for the three data sets are presented in Table 1.

The following conclusions can be made from the obtained results:

1. *Better cluster quality:* As seen from Table 1 RSOM has lower value of entropy; thus implying lower intracluster distance and higher intercluster distance in the clustered space compared to the conventional SOM. RSOM also has higher value of β-index, indicating more homogeneity within its clustered regions. The quantization error of RSOM is also far less than that of SOM.
2. *Less learning time:* The number of iterations required to achieve the error level is far less in RSOM compared to SOM. The convergence curves of the quantization errors are presented in Figs. 7–9 for the data sets used. It is seen that RSOM starts from a very low value of quantization error compared to SOM.

Table 1. Comparison of RSOM with SOM.

Data	Initialization	Quantization error	Iteration at which error converged	Entropy	f_K	β-index
Horse-shoe	Random	0.038	5000	0.7557	83	0.99
	Rough	0.022	50	0.6255	112	0.99
Vowel	Random	32.588	8830	0.6717	245	0.06
	Rough	0.081	95	0.6141	316	0.96
Medical	Random	28.855	8666	0.6744	110	0.61
	Rough	0.246	102	0.6121	125	0.71

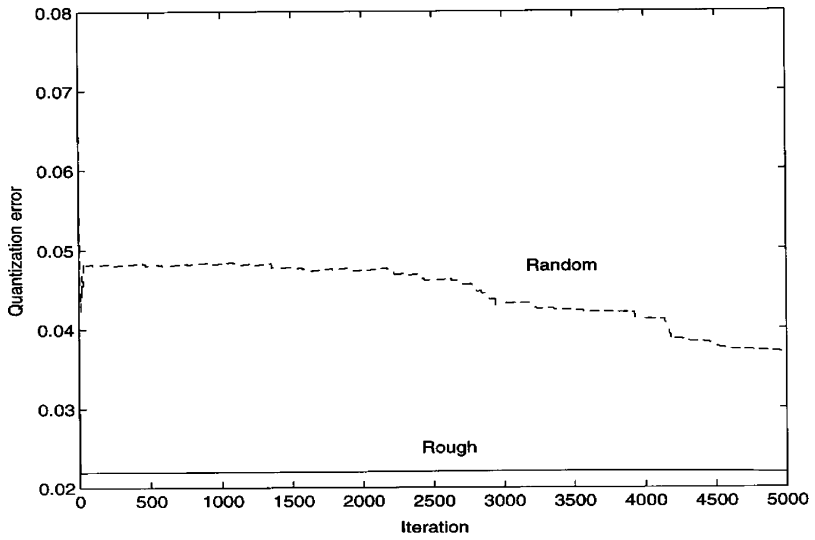

Figure 7. Variation of quantization error with iteration for horse-shoe data.

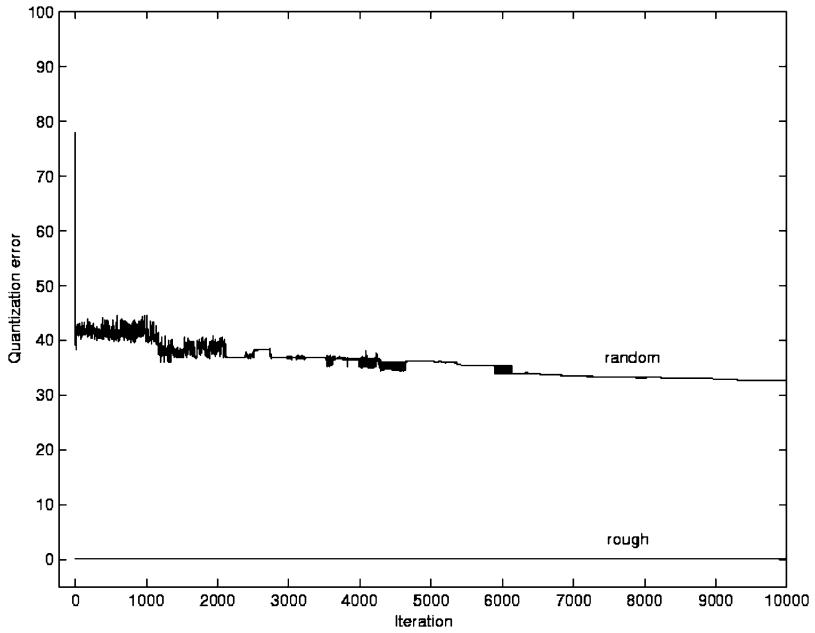

Figure 8. Variation of quantization error with iteration for vowel data.

3. *Compact representation of data:* It is seen that in the case of RSOM fewer nodes in the competitive layer dominate i.e., they win for most of the samples in the training set. On the other hand, in conventional SOM this number is higher. This is quantified by the frequency of winning of the top k nodes. It is observed that this value is much higher for RSOM; thus signifying less error if all but k nodes would have been pruned. In other words, RSOM achieves a more compact representation of the data.

As a demonstration of the nature of distribution of the frequency of winning nodes, we have shown the results for only the horse-shoe data as in Figs. 10 and 11. Separation between the clusters is seen to be more prominent in Fig. 11. These

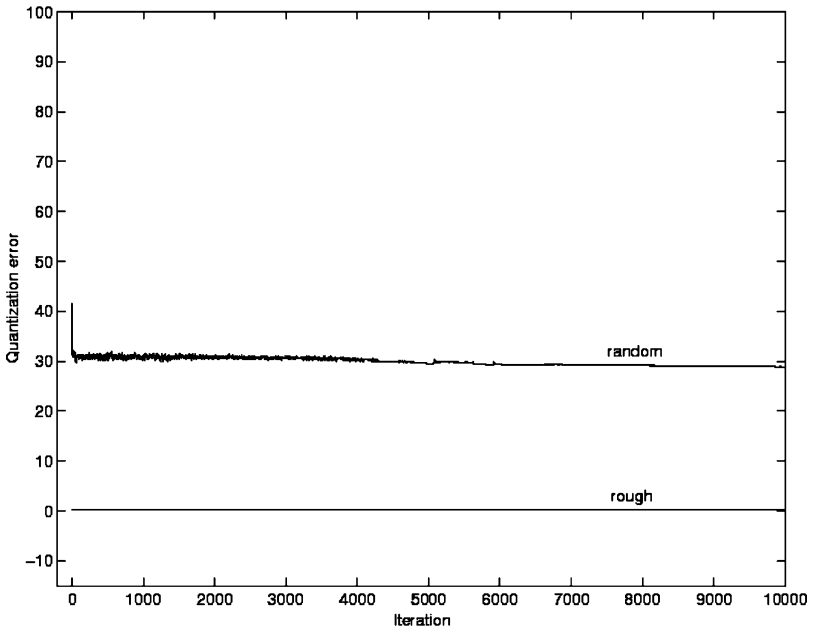

Figure 9. Variation of quantization error with iteration for medical data.

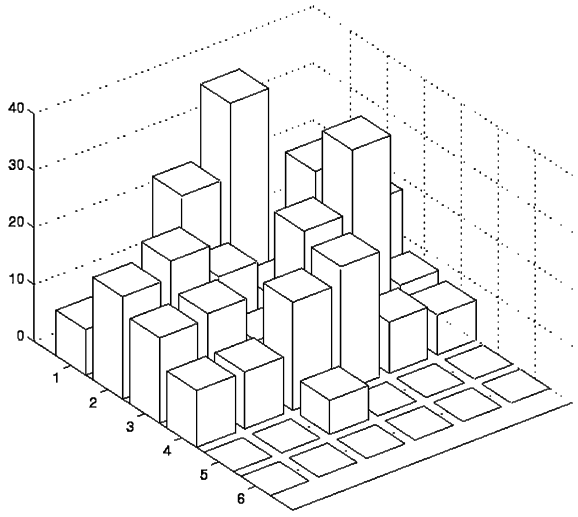

Figure 10. Plot showing the frequency of winning nodes using random weights for the horse-shoe data.

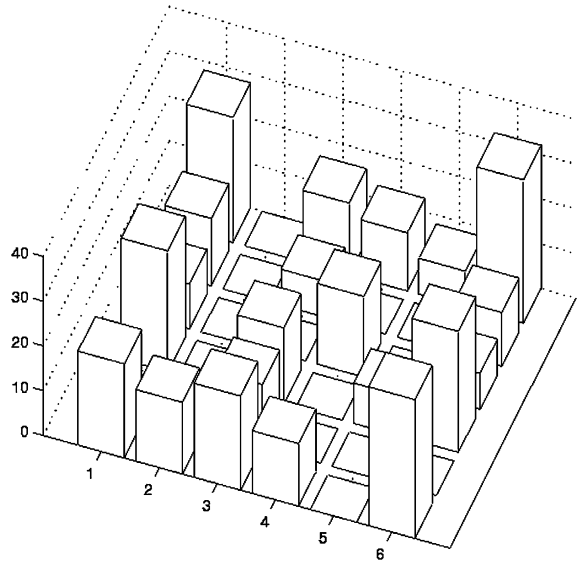

Figure 11. Plot showing the frequency of winning nodes using rough set knowledge for the horse-shoe data.

winning nodes may be viewed as the prototype points (cases) representing the two classes. Unlike the conventional methods, here the cases/ prototypes selected are not just a subset of the original data points, rather they represent some collective information generated by the network after learning the entire data set.

5. Conclusions

A self-organizing map incorporating the theory of rough sets with fuzzy discretization is designed. Rough set theory is used to encode domain knowledge in the

form of crude rules, which are mapped for initialization of the weights of SOM. Superiority of the model (RSOM), compared to random initialization of weights of SOM, is demonstrated for different data sets in terms of learning time, quality of clusters and quantization error. Here the clusters obtained by RSOM are found to be more compact with prominent boundaries i.e. the resulting SOM is sparse with fewer separated winning nodes. Therefore the cases, as represented by the weight vectors of the winning nodes, constitute a compact case base.

Since RSOM achieves compact clusters, this will enable one to extract non-ambiguous rules. Its significance in mining large data sets is evident.

References

1. Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning About Data*, Kluwer Academic: Dordrecht, 1991.
2. A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems," in *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory*, edited by R. Slowinsky, Kluwer Academic: Dordrecht, 1992, pp. 331–363.
3. "Special Issue on Rough-neuro Computing," *Neurocomputing*, edited by S.K. Pal, W. Pedrycz, A. Skowron, and R. Swiniarski, 2001, vol. 36.
4. M. Banerjee, S. Mitra, and S.K. Pal, "Rough fuzzy MLP: Knowledge encoding and classification," *IEEE Trans. Neural Networks*, vol. 9, pp. 1203–1216, 1998.
5. T. Kohonen, *Self-Organizing Maps, Springer Series in Information Sciences*, Springer: Heidelberg, 2001.
6. S.K. Pal, D.S. Yeung, and T.S. Dillon (Eds.), *Soft Computing in Case Based Reasoning*, Springer: London, 2001.
7. R.K. De and S.K. Pal, "A connectionist model for selection of cases," *Information Sciences*, vol. 132, pp. 179–194, 2001.
8. S.K. Pal and S. Mitra, "Multi-layer perceptron, fuzzy sets and classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 683–697, 1992.
9. S.K. Pal and D. Dutta Majumdar, "Fuzzy sets and decision making approaches in vowel and speaker recognition," *IEEE Trans. System, Man and Cybernatics*, vol. SMC-7, no. 8, pp. 625–629, 1977.
10. S. Mitra, "Fuzzy MLP based expert system for medical diagnosis," *Fuzzy Sets and Systems*, vol. 65, pp. 285–296, 1994.
11. S.K. Pal and S. Mitra, *Neuro-fuzzy Pattern Recognition: Methods in Soft Computing*, John Wiley: New York, 1999.
12. S.K.Pal, A. Ghosh, and B. Uma Shankar, "Segmentation of remotely sensed images with fuzzy thresholding, and quantitative evaluation," *International Journal of Remote Sensing*, vol. 21, no. 11, pp. 2269–2300, 2000.

Sankar K. Pal is a Professor and Distinguished Scientist at the Indian Statistical Institute, Calcutta. He is also the Founding Head of Machine Intelligence Unit. He received the M. Tech. and Ph.D. degrees in Radio physics and Electronics in 1974 and 1979 respectively, from the University of Calcutta. In 1982 he received another Ph.D. in Electrical Engineering along with DIC from Imperial College, University of London. During 1986–87, he worked at the University of California, Berkeley and the University of Maryland, College Park, as a Fulbright Post-doctoral Visiting Fellow; and during 1990–92 & in 1994 at the NASA Johnson Space Center, Houston, Texas as an NRC Guest Investigator. He is a Distinguished Visitor of IEEE Computer Society (USA) for the Asia-Pacific Region since 1997, and held several visiting positions in Hong Kong and Australian universities during 1999–2003.

Prof. Pal is a Fellow of the IEEE, USA, Third World Academy of Sciences, Italy, International Association for Pattern Recognition, USA, and all the 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, Web Intelligence and Bioinformatics. He is a co-author of ten books and about three hundred research publications.

He has 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 (USA), 1994 IEEE Trans. Neural Networks Outstanding Paper Award (USA), 1995 NASA Patent Application Award (USA), 1997 IETE—Ram Lal Wadhwa Gold Medal, 1998 Om Bhasin Foundation Award, 1999 G. D. Birla Award for Scientific Research, 2000 Khwarizmi International Award (1st winner) from the Islamic Republic of Iran, the 2001 INSA-Syed Husain Zaheer Medal, and the FICCI Award 2000–2001 in Engineering and Technology.

Prof. Pal has been serving (has served) as an Editor, Associate Editor and a Guest Editor of IEEE Trans. Pattern Analysis and Machine Intelligence, IEEE Trans. Neural Networks, IEEE Computer, Pattern Recognition Letters, Neurocomputing, Applied Intelligence, Information Sciences, Fuzzy Sets and Systems, Fundamenta Informaticae and Int. J. Computational Intelligence and Applications; and a Member, Executive Advisory Editorial Board, IEEE Trans. Fuzzy Systems, Int. Journal on Image and Graphics, and Int. Journal of Approximate Reasoning.