Theoretical Computer Science 466 (2012) 37–63

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Fuzzy rough granular self-organizing map and fuzzy rough entropy

Avatharam Ganivada [a], Shubhra Sankar Ray [a,b,*], Sankar K. Pal [a]

[a] *Center for Soft Computing Research, India*

[b] *Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India*

ARTICLE INFO

Article history:
Received 22 February 2012
Received in revised form 7 July 2012
Accepted 28 August 2012
Communicated by A. Skowron

Keywords:
Fuzzy rough sets
Rule based layered network
Fuzzy reflexive relation
Unsupervised learning
Cluster validity index
Natural computing
Microarray
Gene-function

ABSTRACT

A fuzzy rough granular self-organizing map (FRGSOM) involving a 3-dimensional linguistic vector and connection weights, defined in an unsupervised manner, is proposed for clustering patterns having overlapping regions. Each feature of a pattern is transformed into a 3-dimensional granular space using a π-membership function with centers and scaling factors corresponding to the linguistic terms *low*, *medium* or *high*. The three-dimensional linguistic vectors are then used to develop granulation structures, based on a user defined α-value. The granulation structures are labeled with integer values representing the crisp decision classes. These structures are presented in a decision table, which is used to determine the dependency factors of the conditional attributes using the concept of fuzzy rough sets. The dependency factors are used as initial connection weights of the proposed FRGSOM. The FRGSOM is then trained through a competitive learning of the self-organizing map. We also propose a new "fuzzy rough entropy measure", based on the resulting clusters and using the concept of fuzzy rough sets. The effectiveness of the FRGSOM and the utility of "fuzzy rough entropy" in evaluating cluster quality are demonstrated on different real life datasets, including microarrays, with varying dimensions.

1. Introduction

Granulation is an important step in the human cognition process. This is also a computational paradigm, among others such as self-organization, self-replication, evolution, perception, functioning of brain, group behavior, cell membranes, and morphogenesis, which are abstracted from natural phenomena. A good survey on natural computing explaining its different facets is provided in [1]. Granulation of an object (i.e., a data set, a decision table containing the data set, an image, etc.) involves the collection of granules from the object, where each granule represents a group of data points that have very similar characteristics. These characteristics may refer to similarity, equality and proximity between the data points. The concept of information granulation is described in [2], where perceptions are considered to have fuzzy boundaries and the values of attributes they can take are granulated (a clump of indistinguishable data points/patterns).

Granular Computing (GrC) is an approach, where computation and operations are performed on information granules. The information granules can be developed by fuzzy sets, rough sets, neural networks, etc. For example, the fuzzy set theory is used to describe features in terms of information granules *low*, *medium* or *high*. The development of information granules, using fuzzy set theory and rough set theory, can be found in [2,6]. In neural networks, the Kohonen self-organizing map (SOM) [13] is a pattern discovery technique that organizing data in clusters/groups, reflecting the underlying structures, through the competitive learning. Each such group can be represented as an information granule. The granules can be

* Corresponding author at: Center for Soft Computing Research, India. Tel.: +91 9874287750; fax: +91 33 2578 8699.
E-mail address: shubhra@isical.ac.in (S.S. Ray).

0304-3975/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.08.021

further incorporated into different types of neural network techniques like multi-layer perceptron, self-organizing map, support vector machines, etc., to develop efficient granular neural networks for discovering the patterns in the object.

Different types of granular neural network architectures are described in [3]. Herbert and Yao [12] proposed a granular computing framework for hierarchical self-organizing maps, where training is performed using a bidirectional update propagation method. Mitra and Pal [5] developed a self organizing neural network architecture as a fuzzy classifier. A knowledge based unsupervised network, called Rough SOM, is developed in [4] for discovering underlying clusters in a given data. Here, rough sets are used to encode the weights as well as to determine the size of the network, and fuzzy sets are used for discretization of feature space. Lingras et al. [14] proposed an adaption of self-organizing maps, based on the properties of rough sets, to find interval set representation of clusters of web users on three educational web sites. Investigations have also been carried out in integrating fuzzy sets and rough sets to handle uncertainty and provide a high degree of flexibility [11,20,21,9,10].

In this article, we propose a network by integrating fuzzy sets, fuzzy rough sets and the Kohonen self-organizing map (SOM) [13], where two facets of natural computation, viz, granulation and self organization are integrated. A preliminary version of this investigation is reported in [18]. In this investigation, fuzzy sets are used to develop linguistic input vectors or information granules, fuzzy rough sets are used to extract the crude domain knowledge about data in the form of rules, using a fuzzy reflexive relation, and SOM is used for clustering the data by integrating fuzzy sets and fuzzy rough sets. Here, the significance of the fuzzy reflexive relation is that it measures the similarity between any two patterns in a universe. There is also a possibility of outliers to be present in the data due to noise. This is effectively reduced by using the concept of fuzzy rough sets based on the fuzzy reflexive relation. The concept of an information granule is incorporated, in our methodology, in two different ways: (i) each feature value of data is defined as an information granule in terms of *low*, *medium* or *high* using fuzzy sets and (ii) by using an user defined α-value (α-cut), between 0 and 1, to generate the granulation structures (information granules). These granulation structures, when presented to a decision table, help in extracting domain knowledge about the data in the form of rules. These rules are encoded into SOM and they are used as network parameters (initial connection weights). The resultant network is called a fuzzy rough granular self-organizing map (FRGSOM) and it consists of two layers: input layer and SOM's output layer. The arrangement of the nodes in the input layer and SOM's output layer of the network, incorporation of rules into the network, and training of the network are as follows:

The number of nodes in the input layer is determined by a $3n$-dimensional linguistic vector (*low*, *medium* and *high*), describing a n-dimensional pattern, using fuzzy sets. The number of nodes in the SOM's output layer is considered to be the same as the expected number of clusters. The nodes are then arranged in a two-dimensional grid. The connection weights between nodes in the input layer and nodes in the SOM's output layer are initialized by dependency factors. The initial knowledge based network (FRGSOM) is trained through the competitive learning process of the conventional SOM. After completion of the training process, the network determines the underlying granulation structures/clusters of the data. These are formed at the nodes of the SOM's output layer in a topological order.

We also propose a new entropy measure, called fuzzy rough entropy, based on the lower and upper approximations of a set, and provide some of its properties. In general, in real life data sets, pattern classes have overlapping boundaries, resulting in uncertainty with respect to class belongingness of patterns. The uncertainty can be handled by defining degrees of membership of the patterns, belonging to lower and upper approximations of a set, corresponding to each cluster. Several researchers have defined entropy measures, based on fuzzy set theory and rough set theory, in the past few years. Sen and Pal [22] proposed different classes of entropy measures using fuzzy rough sets. Entropy measure is also determined in [23], using fuzzy rough sets with T-norms, to quantify the uncertainty in T-generalized fuzzy rough sets. In this investigation, the lower and upper approximations are defined by the concept of fuzzy rough sets. We use the proposed entropy measure to quantify the uncertainty in each cluster. The performance of FRGSOM and some related clustering methods like rough self-organizing map (Rough SOM), fuzzy self-organizing map (FSOM) and self-organizing map (SOM) is then evaluated, based on the proposed fuzzy rough entropy, β-index [31] and Davies–Bouldin index (DB-index) [30] for different real life data sets, viz., Telugu vowel data, medical data and microarray data.

The paper is organized as follows: we describe the methodology of FRGSOM in Section 2. A new fuzzy rough entropy measure is proposed, based on the concept of fuzzy rough sets, and its properties are discussed in Section 3. In Section 4, a brief description of the various real life data sets, that are used in our experiments, is given. The FRGSOM is compared with Rough SOM, FSOM and SOM on these data sets and using the proposed fuzzy rough entropy measure, β-index [31] and Davies–Bouldin index (DB-index) [30]. Finally, Section 5 concludes this investigation.

2. Proposed fuzzy rough granular self-organizing map

Here, we describe the process of integrating fuzzy sets and fuzzy rough sets with SOM [13] in order to develop the proposed FRGSOM. The main steps of our methodology are:

1. *Represent input vector of SOM in terms of fuzzy granules:*
   The input vector of SOM is described in terms of fuzzy granules *low*, *medium* and *high*, using the concept of fuzzy sets. The procedure of defining the fuzzy granules using fuzzy sets is explained in Section 2.1.
2. *Granulate the linguistic input data based on α-cut:* The linguistic input data is granulated in two phases. While the first phase computes the pairwise similarity matrix among the patterns using t-norm and implication operator, the second

phase generates the granulation structures. The value of α, used to generate granulation structures, is chosen between 0 to 1 in the second phase. The complete procedure of defining granular structures is explained in Section 2.2.

3. *Introduce the concept of fuzzy rough sets to extract domain knowledge of data:* The granulation structures are first labeled according to decision class information and then presented to a decision table. Based on the decision table, a fuzzy reflexive relation is defined to measure a feature-wise similarity between two patterns in the universe, thereby approximating each such structure using lower and upper approximations. The lower approximation is used to define a positive degree of a pattern in a universe and a dependency degree of a conditional attribute. The lower approximations, positive degrees, and dependency degrees are then used to extract the domain knowledge about the data. These are discussed in detail in Section 2.3.
4. *Incorporate the domain knowledge in SOM:*
   The domain knowledge is encoded in the form of connection weights between the nodes of the input layer and the nodes of the output layer of self-organizing map (SOM). The knowledge encoding procedure and the network configuration of the FRGSOM are explained with an example in Section 2.4.
5. *Train the FRGSOM and cluster the data:*
   The FRGSOM is trained with the competitive learning of SOM and the data is clustered. These are explained in Section 2.5.

2.1. Input vector representation of SOM in terms of fuzzy granules

In general, human minds can perform a wide variety of physical and mental tasks without any measurement or computation. Familiar examples of such tasks include parking a car, driving in heavy traffic, and understanding speech. For performing such tasks, one needs perceptions of size, distance, weight, speed, time, direction, smell, color, shape, force, etc. But a fundamental difference between such measurements on the one hand and perception on the other hand is that the measurements are crisp numbers, whereas perceptions are fuzzy numbers or, more generally, fuzzy granules [15].

A fuzzy granule is a group of patterns defined by the generalized constraint form "X isr R", where, 'R' is a constrained relation, 'r' is a random set constraint and a combination of probabilistic and possibilistic constraints, and 'X' is a fuzzy set valued random variable which takes the values *low*, *medium* and *high*. Using fuzzy-set theoretical techniques, a pattern point x, belonging to the universe U, may be assigned a grade of membership with a membership function, $\mu_A(x)$, to a fuzzy set A. This is defined as

$$A = \{(\mu_A(x), x)\}, \quad x \in U, \mu_A(x) \in [0, 1]. \tag{1}$$

In Eq. (1), the membership values are defined by the π-membership function, with range [0, 1] and $x \in \mathbf{R}^n$, and is defined as [16]

$$\pi(x, C, \lambda) = \begin{cases} 2\left(1 - \frac{\|x-C\|_2}{\lambda}\right)^2, & \text{for } \frac{\lambda}{2} \le \|x - C\|_2 \le \lambda, \\ 1 - 2\left(\frac{\|x-C\|_2}{\lambda}\right)^2, & \text{for } 0 \le \|x - C\|_2 \le \frac{\lambda}{2}, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $\lambda > 0$ is a scaling factor (radius) of the π function with C as a central point, and $\|\cdot\|_2$ denotes the Euclidean norm.

2.1.1. Choice of parameters of π functions for numerical features

Let $\{F_{ij}\}$, for $i = 1, 2, \ldots, s$; $j = 1, 2, \ldots, n$; represent a set of s patterns with n features for a given data set, and $F_{j_{min_m}}$ and $F_{j_{max_m}}$ denote the minimum and maximum values along the jth feature, considering all the s patterns. In general, real-life data contains outliers which can affect the parameters, center and scaling factor of π-membership function. The effect of outliers can be reduced by taking an average of feature values of all the patterns (s patterns) along the jth feature, F_j. It is considered as the center of the linguistic term *medium* and denoted by r_{m_j}. Then, the average values (along the jth feature F_j) of the patterns, having the label values in the ranges $[F_{j_{min_m}}, r_{m_j})$ and $(r_{m_j}, F_{j_{max_m}}]$, are defined as the means of the linguistic terms *low* and *high*, and denoted by r_{l_j} and r_{h_j}, respectively. Similarly, considering the patterns having label values in the ranges $[F_{j_{min_m}}, r_{m_j})$ and $(r_{m_j}, F_{j_{max_m}}]$, along the jth axis, we define $F_{j_{min_l}} = F_{j_{min_m}}$, $F_{j_{max_l}} = r_{m_j}$, $F_{j_{min_h}} = r_{m_j}$, and $F_{j_{max_h}} = F_{j_{max_m}}$. The center C and the corresponding scaling factor λ for linguistic terms *low*, *medium* and *high*, along the jth feature, F_j, are as defined in [17].

2.1.2. Incorporation of granular concept

An n-dimensional pattern can be represented as a $3n$-dimensional linguistic vector [5]. If $F_{i1}, F_{i2}, \ldots, F_{in}$ represent the n features of the ith pattern F_i, then the fuzzy granule of the features is defined as

$$\overrightarrow{F}_i = [\mu_{low(F_{i1})}(\overrightarrow{F}_i), \mu_{medium(F_{i1})}(\overrightarrow{F}_i), \mu_{high(F_{i1})}(\overrightarrow{F}_i), \ldots, \mu_{high(F_{in})}(\overrightarrow{F}_i)], \tag{3}$$

where μ indicates the value of the π-membership function along each feature axis, and corresponds to a fuzzy granule *low*, *medium* or *high*.

2.2. Granulation of linguistic input data based on an α-cut

Here, we recall some preliminaries on granulations based on rough sets, fuzzy sets and fuzzy rough sets. Then, the proposed granulation of linguistic data, based on an α-cut, is explained. In rough sets, the granulation structure is typically a partition of the universe. For preliminaries on granulations, based on rough sets, one may refer to [6–8].

In fuzzy sets, patterns belong to a set, and a couple of patterns belong to a relation with a given degree. A fuzzy relation R in U is a mapping $U \times U \rightarrow [0, 1]$, where a mapping is expressed by a membership function $R(x, y)$ of a relation R, i.e., $R = \{((x, y), R(x, y)) \mid (R(x, y)) \in [0, 1], x \in U, y \in U\}$. For each $y \in U$, the R-foreset of y is a fuzzy set R_y, defined by $R_y(x) = R(x, y)$, for all $x \in U$.

In fuzzy rough set theory, a similarity between any two patterns in U is modeled by a fuzzy relation R, which is defined as

$$\begin{aligned} R(x, x) &= 1 \quad \text{(reflexive)}, \\ R(x, y) &= R(y, x) \quad \text{(symmetry), and} \\ T(R(x, y)R(y, z)) &\leq R(x, z) \quad (T\text{-transitivity}), \end{aligned}$$

for all x, y, z in U. Given a t-norm (or a T-norm), if R does not satisfy symmetry and T-transitivity properties then R is called a fuzzy reflexive relation (fuzzy T-equivalence relation). In general, for the fuzzy T-equivalence relation, we call R_y a fuzzy T-equivalence class (fuzzy equivalence granule) of y. The following fuzzy logical counterparts of connectives [19] are used in generalization of lower and upper approximations in fuzzy rough set theory. For all x and $y \in [0, 1]$, an operator T, mapping from $[0, 1]^2$ to $[0, 1]$, satisfies $T(1, x) = x$. We use T_M and T_L to represent t-norms, and these are defined as

$$T_M(x, y) = min(x, y), \tag{4}$$

$$T_L(x, y) = max(0, x + y - 1) \quad \text{(Lukasiewicz } t\text{-norm)}. \tag{5}$$

On the other hand, a mapping $I : [0, 1] \times [0, 1] \rightarrow [0, 1]$ satisfies $I(0, 0) = 1, I(1, x) = x$ for all $x \in [0, 1]$, where I is an implicator. For all $x, y \in [0, 1]$, the implicators I_M and I_L are defined as

$$I_M(x, y) = max(1 - x, y), \tag{6}$$

$$I_L(x, y) = min(1, 1 - x + y) \quad \text{(Lukasiewicz implicator)}. \tag{7}$$

Algorithm 1. *Similarity Matrix*

Input: $x(i, k), i = 1, 2, \ldots, s$;
$k = 1, 2, \ldots, 3n$.
/*a set of $3n$-dimensional granular data*/
Output: $m(i, j), i, j = 1, 2, \ldots, s$.
/*a similarity matrix*/
Method:

```
for i ← 1 to s do
  for j ← 1 to s do
    for k ← 1 to 3n do
      X ← x(i, k);
      Y ← x(j, k);
      /* Use Eq. 5 */
      l1 ← 1 − X + Y;
      l2 ← 1 − Y + X;
      I1 ← (l1 < 1)?l1 : 1;
      I2 ← (l2 < 1)?l2 : 1;
      /* Use Eq. 7 */
      M(k) ← ((I1 + I2 − 1) > 0)?(I1 + I2 − 1) : 0;
    end fork
    for k ← 1 to 3n do
      m(i, j) ← min[M(k)];
    end for k
  end for j
end for i
```

Algorithm 2. *Granulation Structures*

Input: $x(i, k), i = 1, 2, \ldots, s$;
$k = 1, 2, \ldots, 3n$. /*linguistic input data*/
$m(i, j), i, j = 1, 2, \ldots, s$.
/*a similarity matrix*/
α /*a real value between 0 to 1*/

Output: p /* number of groups*/
$array(p)$ /*number of patterns in each group*/
$U(i_1, j_1, k), i_1 = 1, 2, \ldots, p;$
$j_1 = 1, 2, \ldots, p(i_1),$ /*granulation structures*/
Method:

```
p ← 0;
for i ← 1 to s do
   /*use < continue > statement*/
   array(p) ← 0;
   p ← p + 1;
   for j ← 1 to s do
      if m(i, j) > α then
         /*use < continue > statement*/
         flag(j) ← 1;
         g(p, array(p)) ← j;
      end if
   end for j
end for i
for i1 ← 1 to p do
   for j1 ← 1 to p(i1) do
      val ← g(i1, array(j1));
      for k ← 1 to 3n do
         U(i1, j1, k) ← x(val, k);
      end for k
   end for j1
end for i1
```

2.2.1. Determination of granulation structures

As mentioned earlier, the linguistic input data is granulated to find the granulation structures of the data in two phases, using two different algorithms. While the first phase computes a pairwise similarity matrix of the size $s \times s$, where s is the total number of patterns (see Algorithm 1), the second phase generates the granulation structures (see Algorithm 2). Algorithm 1 is used to define a pairwise similarity matrix among the linguistic input data, using fuzzy logic connectives (see Eqs. (5) and (7)). The similarity matrix is then used to develop the granulation structures based on an α-value, where the value of α is chosen between 0 and 1. The method of determining granulation structures (i.e., p groups in Algorithm 2) is shown in Algorithm 2. The resultant structures (p groups) can be viewed as partitions or clusters. These partitions are arranged in decreasing order according to the size of the group. Here, the size is defined by the number of points in a group. It may be noted that, for different α-values, between 0 to 1, the number of granulation structures will be different. We performed experiments with different α-values and for every α-value, we select the top c groups, based on their size, in all p groups where c represents the user defined number of clusters. The compactness of the first c groups, for every α-value, are then calculated using the proposed fuzzy rough entropy (FRE) (defined in Section 3) and the granulation structures, corresponding to a particular α-value, which provide the lowest average FRE, are accepted. These are presented to a decision system S to extract the domain knowledge (explained in Section 2.3).

2.3. Introduce the concept of fuzzy rough sets to extract domain knowledge about data

Let the number of patterns in all the c-groups, obtained using the selected α-value, be denoted by r. These r patterns from c-groups, represented by $\{x_1, x_2, \ldots, x_r\}$, are then presented to a decision system $S = (U, \mathcal{A} \cup \{d\})$, where U represents the universe and $\mathcal{A}$ represents the attributes, say $\{a_1, a_2, \ldots, a_{3n}\}$. Here, each attribute is constructed by considering the corresponding dimension, from the $3n$-dimensional linguistic vectors (see Eq. (3)), for all the patterns. The decision attribute d is defined as X_k, where $k = 1, 2, \ldots, c$. The value of X_k, corresponding to a pattern, is assigned according to its group. Each X_k can be treated as a decision class. Each pattern x_i in U is classified by its decision classes. The fuzzy reflexive relation R_a, between any two patterns x and y in U, with respect to a quantitative attribute $a \in \mathcal{A}$, is defined as

$$R_a(x,y) = \begin{cases} max\left(min\left(\frac{a(y)-a(x)+\sigma_{a_{k_1}}}{\sigma_{a_{k_1}}}, \frac{a(x)-a(y)+\sigma_{a_{k_1}}}{\sigma_{a_{k_1}}}\right), 0\right), \\ \quad \text{if } a(x), a(y) \in R_d(X_{k_1}), \\ max\left(min\left(\frac{a(y)-a(x)+\sigma_{a_{k_2}}}{\sigma_{a_{k_2}}}, \frac{a(x)-a(y)+\sigma_{a_{k_2}}}{\sigma_{a_{k_2}}}\right), 0\right), \\ \quad \text{if } a(x) \in R_d(X_{k_1}), a(y) \in R_d(X_{k_2}), \text{ and } k_1 \neq k_2, \end{cases} \tag{8}$$

where $k_1 = 1, 2, \ldots, c$; $k_2 = 1, 2, \ldots, c$, and $\sigma_{a_{k_1}}$ and $\sigma_{a_{k_2}}$ represent the standard deviation of decision classes X_{k_1} and X_{k_2}, respectively. In Eq. (8), $a(x)$ and $a(y) \in R_d(X_{k_1})$ imply that the patterns x and y belong to decision class X_{k_1} with respect to a decision attribute $\{d\}$, where $a \in \{d\}$. On the other hand, $a(x) \in R_d(X_{k_1})$ and $a(y) \in R_d(X_{k_2})$ imply that the patterns x and y belong to two different decision classes, X_{k_1} and X_{k_2}, respectively.

When a qualitative attribute $a \in \{d\}$, then the relation R_d between any two patterns x and $y \in U$, with respect to the attribute 'a', is defined as follows:

Defining decision classes using fuzzy sets:

The decision system S contains c-decision classes of a decision attribute. Assume that the $3n$-dimensional vectors O_{kj} and V_{kj}, $j = 1, 2, \ldots, 3n$, are the mean and standard deviation, respectively, of the patterns belonging to the kth class in the given decision system. The weighted distance of a pattern $\overrightarrow{F_i}$, $i = 1, 2, \ldots, r$, from the kth decision class is defined as

$$Z_{ik} = \sqrt{\sum_{j=1}^{n} \left[\frac{F_{ij} - O_{kj}}{V_{kj}}\right]^2}, \quad \text{for } k = 1, 2, \ldots, c, \tag{9}$$

where F_{ij} is the value of the jth component of the ith pattern. Note that, when the value of a feature for all the patterns in a class is the same, then the standard deviation will be zero. In that case, we consider V_{kj} = 0.000001 (for the sake of computation) so that the weighting distance Z_{ik} becomes high and the membership value of the ith pattern, belonging to the kth class along that feature, becomes low. The membership value of the ith pattern in the kth class is defined as [24]

$$\mu_k(\overrightarrow{F}_i) = \frac{1}{1 + \left(\frac{Z_{ik}}{f_d}\right)^{f_e}}, \tag{10}$$

where f_e and f_d are fuzzifiers. It may be noted that when a pattern has different membership values then its decision attribute becomes quantitative. It can be shown in two different ways, namely,

(1) the membership values of all patterns in the kth class to its own class is defined as

$$D_{kk} = \mu_k(\overrightarrow{F}_i), \quad \text{if } k = l, \tag{11}$$

where $\mu_k(\overrightarrow{F}_i)$ represents the membership value of the ith pattern to the kth class, and

(2) the membership values of all patterns in the kth class to other classes is defined as

$$D_{kl} = 1, \quad \text{if } k \neq l \tag{12}$$

where k and $l = 1, 2, \ldots, c$. For any two patterns x and $y \in U$, with respect to an attribute $a \in \{d\}$, the fuzzy decision classes are defined as

$$R_a(x, y) = \begin{cases} D_{kk}, & \text{if } a(x) = a(y), \\ D_{kl}, & \text{otherwise.} \end{cases} \tag{13}$$

Here, D_{kk} represents the membership value of each pattern $\overrightarrow{F}_i$ belonging to the same class ($k = l$), and D_{kl} represents an integer value '1' for all the patterns from other than the kth class ($k \neq l$).

The lower and upper approximations of a fuzzy set $A \subseteq U$, with a reflexive relation R under the fuzzy logic connectives, Eqs. (5) and (7), are defined as [21],

$$(R \downarrow A)(y) = \inf_{x \in U} I(R(x, y), A(x)), \quad \text{and} \tag{14}$$

$$(R \uparrow A)(y) = \sup_{x \in U} T(R(x, y), A(x)), \tag{15}$$

respectively, for all y in U. For any $B \subseteq \mathcal{A}$, the fuzzy positive region can be defined, based on the B-indiscernibility relation R_B, for $x \in U$, as

$$POS_B(y) = \left(\bigcup_{x \in U} R_B \downarrow R_d x\right)(y), \tag{16}$$

for all y in U. The degree of dependency of γ, on the set of attributes $B \subseteq \mathcal{A}$, is defined as

$$\gamma_B = \frac{\sum_{x \in U} POS_B(x)}{|U|}, \tag{17}$$

where $|\cdot|$ denotes the cardinality of a set U, and the value of γ is $0 \leq \gamma \leq 1$.

Table 1
Data set.

U	a	b	c	d
1	−0.4	−0.3	−0.5	1
2	−0.4	0.2	−0.1	2
3	−0.3	−0.4	0.3	1
4	0.3	−0.3	0	2
5	0.2	−0.3	0	2
6	0.2	0	0	1

2.4. Incorporation of domain knowledge in SOM

In this section, we first describe how the decision table can be used to explain the concept of granulation by partition and fuzzy rough set approximations, based on a fuzzy reflexive relation. Based on this principle, knowledge about the data is extracted and incorporated into the self-organizing map (SOM). It is then used for competitive learning. The knowledge encoding procedure is as follows:

Knowledge encoding procedure:

Let us recall the aforesaid decision table $S = (U, \mathcal{A} \cup \{d\})$ with its set of conditional attributes, decision attributes, set of patterns, and labeled values of patterns corresponding to $3n$-dimensional conditional attributes. The following steps are applied to the decision table $S = (U, \mathcal{A} \cup \{d\})$ for extracting the domain knowledge about data.

Step 1. Generate a fuzzy reflexive relational matrix by using the fuzzy reflexive relation (see Eq. (8)) on all possible pairs of patterns and obtain additional granulation structures based on the relational matrix.

Step 2. Use a fuzzy reflexive relational matrix to compute the membership value (belonging to lower approximation, using Eq. (14)) of every pattern of a concept, for each conditional attribute with respect to decision classes (using Eq. (13)).

Step 3. Calculate the fuzzy positive region (using Eq. (16)) of every pattern for each conditional attribute.

Step 4. Calculate the degree of dependency (using Eq. (17)), of each conditional attribute, corresponding to patterns within the concept, with respect to each decision class. Assign the resulting dependency factors as initial weights between the input layer nodes and c-number of output layer nodes in SOM, where c represents the user defined number of clusters.

2.4.1. Example

Let us consider an example data set [20], and two granulation structures, shown in Table 1. Each conditional attribute (feature) in Table 1 is transformed into a 3-dimensional granular space using Eq. (3). Then the resulting decision table is shown in Table 2. We apply the concept of fuzzy rough sets based on the fuzzy reflexive relation on the example data set to determine the initial weights of the FRGSOM. The fuzzy membership values of the patterns (using Eqs. (11) and (12)) are presented under the decision columns D_{kk} and D_{kl} in Table 2. Two typical examples of fuzzy reflexive relational matrices, resulting from conditional attributes L_1 and M_1, are as follows:

$$R_{L_1}(x,y) = \begin{pmatrix} 1 & 0.938 & 0.184 & 1 & 0.194 & 0.194 \\ 0.938 & 1 & 0.246 & 0.938 & 0.256 & 0.256 \\ 0.184 & 0.246 & 1 & 0.194 & 1 & 1 \\ 1 & 0.938 & 0.184 & 1 & 0.194 & 0.194 \\ 0.184 & 0.246 & 1 & 0.194 & 1 & 1 \\ 0.184 & 0.246 & 1 & 0.194 & 1 & 1 \end{pmatrix}$$

$$R_{M_1}(x,y) = \begin{pmatrix} 1 & 0.702 & 0.793 & 1 & 0.908 & 0.793 \\ 0.702 & 1 & 0.908 & 0.702 & 0.611 & 0.908 \\ 0.793 & 0.908 & 1 & 0.793 & 0.702 & 1 \\ 1 & 0.702 & 0.793 & 1 & 0.908 & 0.793 \\ 0.908 & 0.611 & 0.702 & 0.908 & 1 & 0.702 \\ 0.793 & 0.908 & 1 & 0.793 & 0.702 & 1 \end{pmatrix}.$$

Similarly, the fuzzy reflexive relational matrices $R_{H_1}(x,y), R_{L_2}(x,y), R_{M_2}(x,y), R_{H_2}(x,y), R_{L_3}(x,y), R_{M_3}(x,y)$, and $R_{H_3}(x,y)$ can be determined for the remaining attributes. We then calculate a membership value of an pattern, belonging to the lower approximation, using the values in the decision columns D_{kk} and D_{kl} corresponding to the decision classes $X_0 = \{1, 3, 6\}$ and $X_1 = \{2, 4, 5\}$ in Table 2.

Table 2
Decision table.

U	L_1	M_1	H_1	L_2	M_2	H_2	L_3	M_3	H_3	d	D_{kk}	D_{kl}
1	0.87	0.39	0	0.92	0.92	0	0.12	0.34	0	1	0.25	1
3	0.5	0.69	0	0.38	0.73	0	0.12	0.87	0	1	0.29	1
6	0	0.60	0.87	0	0.81	0.12	0	0.87	0.92	1	0.21	1
2	0.87	0.39	0	0	0.26	0.12	0	0.98	0.38	2	0.21	1
4	0	0.30	0.5	0.92	0.92	0	0	0.87	0.92	2	0.32	1
5	0	0.60	0.87	0.92	0.92	0	0	0.87	0.92	2	0.30	1

The membership value of pattern 1, belonging to the lower approximation (using Eq. (14)), with respect to the decision class $X_0 = \{1, 3, 6\}$ is

$$\begin{aligned}(R_{L_1} \downarrow R_d x)(1) &= \inf_{x \in \mathbf{U}} I\{R_{L_1}(x, 1), R_d(x, 1)\},\\ &= min\{I(1, 0.251714), I(0.938, 0.297584), I(0.184, 0.219859), I(1, 1), I(0.194, 1), I(0.194, 1)\},\\ &= min\{0.251714, 0.359490, 1, 1, 1, 1\},\\ &= 0.251714.\end{aligned}$$

For the remaining patterns, the membership values belonging to the lower approximation are
$(R_{L_1} \downarrow R_d x)(3) = 0.297584$, $(R_{L_1} \downarrow R_d x)(6) = 0.219859$, $(R_{L_1} \downarrow R_d x)(2) = 0.251714$, $(R_{L_1} \downarrow R_d x)(4) = 0.219859$, and $(R_{L_1} \downarrow R_d x)(5) = 0.219859$.
The membership value of pattern 1, belonging to the lower approximation (using Eq. (14)), with respect to the decision class $X_1 = \{2, 4, 5\}$, is

$$\begin{aligned}(R_{L_1} \downarrow R_d x)(1) &= \inf_{x \in \mathbf{U}} I\{R_{L_1}(x, 1), R_d(x, 1)\},\\ &= min\{I(1, 1), I(0.938, 1), I(0.184, 1), I(1, 0.210388), I(0.194, 0.321112), I(0.194, 0.300009)\},\\ &= min\{1, 1, 1, 0.210388, 1, 1\},\\ &= 0.210388.\end{aligned}$$

Similarly, for the remaining patterns, the membership values belonging to the lower approximation are $(R_{L_1} \downarrow R_d x)(3) = 0.271536$, $(R_{L_1} \downarrow R_d x)(6) = 0.300009$, $(R_{L_1} \downarrow R_d x)(2) = 0.210388$, $(R_{L_1} \downarrow R_d x)(4) = 0.300009$, and $(R_{L_1} \downarrow R_d x)(5) = 0.300009$.
Hence, the positive regions (using Eq. (16)) of patterns in the concept $\{1, 3, 6\}$, with respect to decision class $X_0 = \{1, 3, 6\}$, are defined as follows:

$$\begin{aligned}(R_{L_1} \downarrow R_d x)(1) &= max\{0.251714, 0.210388\},\\ &= 0.251714.\end{aligned}$$

For the remaining patterns, the positive regions are
$(R_{L_1} \downarrow R_d x)(3) = 0.297584$, $(R_{L_1} \downarrow R_d x)(6) = 0.300009$.
Similarly, the positive regions of patterns in the concept, $\{2, 4, 5\}$, with respect to decision class $X_1 = \{2, 4, 5\}$, are
$(R_{L_1} \downarrow R_d x)(2) = 0.251714$, $(R_{L_1} \downarrow R_d x)(4) = 0.300009$, and $(R_{L_1} \downarrow R_d x)(5) = 0.300009$.
The dependency degree of the attribute L_1 with respect to decision class $x_0 = \{1, 3, 6\}$ is

$$\begin{aligned}\gamma_{\{L_1\}}(x_0) &= (0.251714 + 0.297584 + 0.300009)/3,\\ &= 0.283102.\end{aligned}$$

The dependency degree of the attribute L_1 with respect to decision class $x_1 = \{2, 4, 5\}$ is

$$\begin{aligned}\gamma_{\{L_1\}}(x_1) &= (0.251714 + 0.300009 + 0.300009)/3,\\ &= 0.283911.\end{aligned}$$

The dependency degrees for the remaining attributes with respect to each decision class are as follows:

$$\begin{array}{lll}\gamma_{\{M_1\}}(x_0) = 0.314289, & & \gamma_{\{M_1\}}(x_1) = 0.298190,\\ \gamma_{\{H_1\}}(x_0) = 0.267812, & & \gamma_{\{H_1\}}(x_1) = 0.290945,\\ \gamma_{\{L_2\}}(x_0) = 0.303748, & & \gamma_{\{L_2\}}(x_1) = 0.273292,\\ \gamma_{\{M_2\}}(x_0) = 0.389971, & & \gamma_{\{M_2\}}(x_1) = 0.456389,\\ \gamma_{\{H_2\}}(x_0) = 0.273292, & & \gamma_{\{H_2\}}(x_1) = 0.273292,\\ \gamma_{\{L_3\}}(x_0) = 0.732064, & & \gamma_{\{L_3\}}(x_1) = 0.219859,\\ \gamma_{\{M_3\}}(x_0) = 0.475454, & & \gamma_{\{M_3\}}(x_1) = 0.308170,\\ \gamma_{\{H_3\}}(x_0) = 0.766670, & \text{and} & \gamma_{\{H_3\}}(x_1) = 303823.\end{array}$$

Let us now define the initial structure of FRGSOM for the above said example data set in Table 2. The data has nine input features (conditional attributes), so the number of input layer nodes of the FRGSOM is set to nine. The number of output layer nodes of the FRGSOM is set to two as it is assumed that there are two granulation structures in this data. The aforesaid dependency degrees, corresponding to the decision class X_0, are used as initial connection weights between the nine input layer nodes to the first node in the SOM's output layer. Similarly, the dependency degrees, corresponding to the decision class X_1, are used as initial connection weights between the nine input nodes to the second output node. The network is then trained through a competitive learning of SOM for clustering of the input data, as explained in the next section.

2.5. Train the FRGSOM and cluster the data

In this section, first we provide a brief description of the conventional self-organizing map (SOM) [13], and then we explain the training process of the proposed FRGSOM.

Let $x = x(t) \in R^n$ denote a sequence of input vectors and $\{w_{ij}(t), i = 1, 2, \ldots, N; j = 1, 2, \ldots, n\}$ denote a set of network parameters (initial connection weights), where t is the time coordinate, N is the number of nodes in the output layer, and n represents the dimension of the input vector as well as the number of nodes in the input layer. Initially, the network parameters in SOM are chosen as small random numbers. At each successive instant of time, t, an input vector $x_j(t)$ is randomly presented to the network. The Euclidean distance, d_i, between the input vector, $x_j(t)$, and weight vector, $w_{ij}(t)$, is computed as

$$d_i = \|x_j(t) - w_{ij}(t)\|^2. \tag{18}$$

The winning neuron obtained in the output layer, denoted by a, is determined by

$$a = argmin\{d_i\}, \quad i = 1, 2, \ldots, N. \tag{19}$$

The nodes in the output layer are arranged in a two-dimensional lattice. The neighborhood set, say N_a, around the winning neuron a, is defined in [13] as

$$N_a(t) = \exp\left(\frac{-\|r_a - r_c\|^2}{2\sigma(t)^2}\right), \tag{20}$$

where r_c and r_c represent the coordinates of winning node a and a node c within the neighborhood N_a, respectively. Here, $\sigma(t)$ is the width of the neighborhood set and is decreased with every iteration. The value of σ is chosen as in [13]. Now, the weights of neurons within the neighborhood set, N_a, are updated and neurons outside N_a are left intact. The updated weight of any neuron is defined as

$$w_{ij}(t+1) = w_{ij}(t) + N_{ai}(t)\alpha(t)(x_j(t) - w_{ij}(t)), \quad j = 1, 2, \ldots, n, \tag{21}$$

where α is a learning parameter, chosen between 0 and 1, and i represents the number of neurons within the neighborhood set N_a. Eq. (21) updates the weights of the winning neuron and its neighborhood neurons. The updated weights are more likely to become similar to the input patterns, which are presented to the network during training.

In the proposed FRGSOM, the input data is first transformed into 3-dimensional granular space using Eq. (3). During the training process of FRGSOM, instead of choosing the initial connection weights in SOM as small random numbers they are determined using the concept of fuzzy rough sets, explained in Section 2.4. The FRGSOM is then trained through the competitive learning process using Eq. (21). After completion of the competitive learning, FRGSOM is able to partition the granular input data into the groups/clusters (granulation structures) in a topological order. Here, the input data is partitioned by FRGSOM in a topological order in the sense that the weight updates of the neighborhood neurons, of a winning neuron, cause the whole output space to become approximately ordered [13].

3. Fuzzy rough entropy measure

The performance of clustering methods is evaluated using a newly defined entropy measure along with some of the existing ones and the results are reported. Before defining the proposed entropy measure, we explain the concept behind it. Let us consider, as an example, three clusters, say, C_1, C_2 and C_3. Let p_1, p_2 and p_3 denote the number of patterns belonging to C_1, C_2 and C_3, respectively. It may be noted that the data used for evaluation of the clusters, based on fuzzy rough entropy measure, is defined in terms of membership values using Eq. (2), where the parameters in Eq. (2) are considered corresponding to a linguistic term *medium* to reduce computational complexity.

Let X_i denote the ith set, with all the patterns, corresponding to the cluster C_i, $i = 1, 2$ and 3. That is, for $i = 1$, $X_1 = p_1$. The entropy measure for a cluster C_i is defined based on the roughness values of the set X_i, which is as follows:

Table 3
An example decision table for a set X_1 corresponding to a cluster C_1.

Patterns	Conditional Attributes corresponding to a linguistic term *medium*	Decision Attribute	Fuzzy decision classes	Fuzzy decision classes
(u_1)	$(\mathcal{A})$	(d)	(E_{kk})	(E_{kr})
p_1	a_1	class 1	E_{11}, for $k = 1$	E_{12}, for $k = 1, r = 2$
p_2, p_3	a_2, …, a_n	class 2	E_{22}, for $k = 2$	E_{21}, for $k = 2, r = 1$

3.1. Roughness of a set X_1 in a universe U

Let $S_i = (u_i, \mathcal{A} \cup \{d\})$ be a decision system corresponding to a cluster C_i, $i = 1, 2$ and 3. For $i = 1$, $S_1 = (u_1, \mathcal{A} \cup \{d\})$ represents the decision system for the set X_1, where $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ represents the conditional attributes, and d $(d \notin \mathcal{A})$ represents a decision attribute. Here, universe $u_1 = p_1 \cup p_2 \cup p_3$. The patterns, p_1, are labeled with an integer value "1", representing the decision class 1, and all other patterns, $p_2 \cup p_3$ are labeled with an integer value "2", representing the decision class 2. The methodology, say Procedure 1, of defining roughness of the set X_1, using the concept of fuzzy rough sets, is as follows:

Procedure 1:

(S1) For a quantitative attribute $a \in \mathcal{A}$, we calculate the fuzzy reflexive relation using Eq. (8).
(S2) For a qualitative decision attribute $a \in \{d\}$, we define a fuzzy way of decision classes for the patterns p_1 and $p_2 \cup p_3$.
(S3) Let the n-dimensional vectors O_{kj} and V_{kj}, $j = 1, 2, \ldots, n$, denote the mean and standard deviation of the data for the kth class of the decision system S_1. The weighted distance of a pattern $\overrightarrow{F_i}$ from the kth class is defined by Eq. (9), where $k = 1$ and 2 (decision class 1 and decision class 2). The membership values of the ith pattern to the kth class is defined by Eq. (10).
(S4) The values of the patterns corresponding to the decision classes are defined in terms of average membership values. Average membership values are defined in two ways, namely, (1) by computing the average of the membership values over all the patterns in the kth class to its own class ($k = 1$), and assigning it to each pattern, $\overrightarrow{F}_i$, in the kth decision class ($k = 1$), and (ii) by computing the average of the membership values over all the patterns in the kth class ($k = 1$) to the other class ($k = 2$), and assigning it to each pattern $\overrightarrow{F}_i$ in the other decision class ($k = 2$). So the average membership value of all the patterns in the kth class to its own class is defined as

$$E_{kk} = \frac{\sum_{i=1}^{m_k} \mu_k(\overrightarrow{F}_i)}{|m_k|}, \quad \text{if } k = r, \tag{22}$$

where r represents the total number of classes.

The average membership values of all the patterns in the kth class ($k = 1$) to the other decision class (say, $k = 2$) are defined as

$$E_{kr} = \frac{\sum_{i=1}^{m_k} \mu_r(\overrightarrow{F}_i)}{|m_k|}, \quad \text{if } k \neq r, \tag{23}$$

where $|m_k|$ indicates the number of patterns in the kth class. For a qualitative attribute 'a' $\in \{d\}$, the fuzzy decision classes are defined as

$$R_a(x, y) = \begin{cases} E_{kk}, & \text{if a(x) = a(y)}, \\ E_{kr}, & \text{otherwise}, \end{cases} \tag{24}$$

for all x and y in u_1.

A detailed description of defining the decision classes, in a fuzzy way, of the data can be found in [17]. A decision table S_1 along with fuzzy decision classes, for a set X_1 corresponding to a cluster C_1, is shown in Table 3, as an example.

Let $x_0 = \{p_1\}$ and $x_1 = \{p_2 \cup p_3\}$ denote two subsets of the universe u_1. For each conditional attribute (feature) $a \in \mathcal{A}$, we now compute the membership values of the patterns in the subset x_0, for belonging to the lower and the upper approximations of X_1, using Eqs. (14) and (15), respectively. Thereafter, for each conditional attribute $a \in \mathcal{A}$, we calculate the sum of weighted membership values of all the patterns in a subset x_0 in two ways:

(i) by multiplying the membership value of a pattern y to a subset x_0, with its membership value, for belonging to a lower approximation of X_1. For $x \in u_1$, it is denoted by LS, and is defined by

$$LS = \sum_{y \in x_0} m(y)(R \downarrow Ax)(y), \tag{25}$$

where $m(y)$ represents the membership value of a pattern y to a subset x_0, and $(R \downarrow Ax)(y)$ represents its membership value, for belonging to a lower approximation of X_1, corresponding to a conditional attribute $a \in \mathcal{A}$.

(ii) by multiplying the membership value of a pattern y in a subset x_0, with the membership value, for belonging to a upper approximation of a pattern y in a subset x_0. For $x \in u_1$, it is denoted by US, and is defined as

$$US = \sum_{y \in x_0} m(y)(R \uparrow Ax)(y). \tag{26}$$

For a conditional attribute $a \in \mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, the LS and US then become LS_{a_i} and US_{a_i} for $i = 1, 2, \ldots, n$. Therefore, the roughness of the set X_1 corresponding to the cluster C_1 is defined as

$$R(X_1) = 1 - \frac{\sum_{i=1}^{n} LS_{a_i}}{\sum_{i=1}^{n} US_{a_i}}. \tag{27}$$

Here, $R(X_1)$ quantifies the uncertainty in terms of roughness in the set X_1, corresponding to a cluster C_1.

The fuzzy rough entropy (FRE) of a cluster C_1, based on the roughness measure defined in Eq. (27) of the set X_1, is defined as

$$FRE(C_1) = -R(X_1) log_e(R(X_1)/e). \tag{28}$$

For the remaining clusters C_i, $i = 2$ and 3, we apply the same procedure for defining the FRE. Therefore, the fuzzy rough entropy (FRE) of a cluster C_i is defined as

$$FRE(C_i) = -R(X_i) log_e(R(X_i)/e) \tag{29}$$

where $i = 1, 2, \ldots, c$, and the average fuzzy rough entropy is defined as

$$FRE = \frac{1}{c}\left(\sum_{i=1}^{c} FRE(C_i)\right). \tag{30}$$

The fuzzy rough entropy measure, FRE(C_i), $i = 1, 2, \ldots, c$, satisfies the following properties:

1. Nonnegativity: FRE(C_i) ≥ 0 *iff* $R(X_i) \geq 0$.
2. Continuity: For all the values of $R(X_i) \in [0, 1]$, FRE(C_i) is a continuous function of $R(X_i)$.
3. Sharpness: The value of FRE(C_i) is zero when $(R \downarrow A)(y) = (R \uparrow A)(y)$, for $y \in X_i$.
4. Maximality and normality: When lower approximation of the set, X_i, is zero then the roughness value of the set is equal to 1. This implies that the entropy measure FRE(C_i) attains the maximum value of unity.
5. Resolution: For any $C_i^* \leq C_i \Rightarrow$ FRE(C_i^*) $\leq$ FRE(C_i).
6. Monotonicity: FRE(C_i) is a monotonic increasing function of $R(X_i)$.

As FRE(C_i), $i = 1, 2, \ldots, c$ satisfies all the above mentioned properties, the average fuzzy rough entropy (FRE) also satisfies them. A lower value of FRE for a cluster indicates that the cluster is good (in terms of compactness). It may noted that a symmetry property is not discussed here because we do not consider the compliment of the set X_i, while defining the entropy measure of each cluster.

4. Time complexity

In this section, we discuss about the time complexity of FRGSOM involving 3-dimensional linguistic vectors, Algorithms 1 and 2, fuzzy decision classes, initial connection weights and SOM. The time complexity of FRE and FRGSOM with FRE is also provided.

Time complexity of 3n*-dimensional linguistic data:*

Considering the procedure for transforming an n-dimensional data into a 3n-dimensional linguistic data (see Sections 2.1, 2.1.1 and 2.1.2), the total time complexity is $O(ns + n + nl + n + nh + n + s(3n))$, where $ns + n$, $nl + n$, and $nh + n$ refer to the time complexities for defining the centers and scaling factors of all the features corresponding to the linguistic terms *medium*, *low* and *high*, respectively and $s(3n)$ is the complexity for transforming n-features into 3n-dimensions of s patterns. Here, s, l and h represent the total number of patterns along three feature axes, corresponding to linguistic terms *medium low* and *high*, respectively. Therefore, the asymptotic time complexity is $O(s(3n))$ as $s(3n)$ is greater than each of the remaining terms. Note that s is also the total number of patterns.

Time complexity of Algorithm 1:

The time complexity of Algorithm 1 is $O(s^2(3n + 3n))$ where s is the total number of patterns and $3n$ is the number of features. Here, the complexity involved with the two consecutive innermost *for loops* (see steps 3 to 12 and 13 to 15 in Algorithm 1 in Section 2.2) is $O(3n + 3n)$ and the complexity $O(s^2)$ is related with the two outermost nested *for* loops (see steps 1 and 2 in Algorithm 1). Hence, the asymptotic time complexity of Algorithm 1 is $O(s^2(3n))$.

Time complexity of Algorithm 2:

The time complexity of Algorithm 2 is $O(s^2 + pp_c(3n))$, where s is the number of patterns, p is the number of clusters, p_c is the number of patterns within the cth cluster, and $3n$ is the number of features. Here, the complexity involved with two nested *for* loops (see steps 2 and 6 in Algorithm 2 in Section 2.2) is $O(s^2)$ and the complexity $O(pp_c(3n))$ is related with three nested *for* loops (see steps 14, 15 and 17 in Algorithm 2 in Section 2.2).

Time complexity of fuzzy decision classes:

In computing fuzzy decision classes, considering the weighted distances, the membership values of the patterns in the classes, and the membership values of all patterns to the decision classes, the time complexity is $O(cr_c(3n)+cr_c+cr)$, where c is the number of decision classes, r_c is the number of patterns in the cth class, r is the number of patterns in all the decision classes, and $3n$ is the number of features. Hence, the asymptotic time complexity is $O(cr_c(3n) + cr)$, where $cr_c(3n) > cr_c$.

Time complexity of initial connection weights:

The time complexity in computing initial connection weights, determined by using the concept of fuzzy rough sets shown in Section 2.4, is described as follows:

The time complexity in computing the fuzzy reflexive relational matrices of sizes $r \times r$ (see Step 1 in Section 2.4) corresponding to all the conditional attributes $(3n)$ is $O((3n)r^2)$, where, r is the total number of patterns in all the classes. In computing membership values of patterns for belonging to lower approximations (see Step 2 in Section 2.4), we have to compute the fuzzy reflexive relational matrices corresponding to all the conditional attributes $(3n)$ and the fuzzy decision classes (c). So, the time complexity in computing membership values is $O((3n)cr^2)$. The time complexity in computing positive membership values (see Step 3 in Section 2.4) is $O((3n)rc)$ and in computing dependency factors of all the conditional attributes $(3n)$, with respect to all the decision classes (c) (see Step 4 in Section 2.4), is $O(cr_c(3n))$, where, r_c is the number of patterns within the cth class. Hence, the time complexity of initial connection weights is $O((3n)r^2+(3n)cr^2+(3n)rc+cr_c(3n))$. Therefore, the asymptotic time complexity of initial connection weights is $O((3n)cr^2)$, where $r^2 > r > r_c$.

Time complexity of SOM:

During training of SOM, for each pattern, the time complexity in computing the distance between the input vector and weight vector (see Eq. (18) in Section 2.5) is $O(Nn)$, where n represents the number of nodes in the SOM's input layer and N represents the number of nodes in the SOM's output layer. The time complexity in computing the winning neuron for each pattern in the output layer (see Eq. (19) in Section 2.5) is $O(N)$. The time complexity in updating the connection weights (see Eq. (21) in Section 2.5) for each pattern is $O(Nn)$. Hence, the time complexity of SOM is $O(ts(Nn + N + Nn))$, where, t is the number of iterations, and s is the number of patterns. Therefore, the asymptotic time complexity is $O(tsNn)$. In FRGSOM, the number of nodes in the SOM's input layer is set corresponding to a $3n$-dimensional linguistic vector. So, the asymptotic time complexity of SOM, incorporated in FRGSOM, is $O(tsN(3n))$.

Time complexity of FRGSOM:

The time complexity of FRGSOM will involve all the asymptotic time complexities discussed so far. Hence, the worst case time complexity of FRGSOM is $O(s(3n) + s^2(3n) + s^2 + pp_c(3n) + cr_c(3n) + cr + (3n)cr^2 + tsN(3n))$ and the asymptotic time is $O(s^2(3n) + (3n)cr^2 + tsN(3n))$, where $s^2 > s \geq pp_c$ and $r > r_c$.

4.1. Time complexity of fuzzy rough entropy (FRE)

In Fuzzy Rough Entropy (FRE) (see Section 3), the time complexity for defining the sets $X_i, i = 1, 2, \ldots, c$, corresponding to the clusters $C_i, i = 1, 2, \ldots, c$, is $O(c^2p_{1c}p_{2c}n)$, where c represents the number of clusters, p_{1c} is the number of patterns within the cth cluster, p_{2c} is the number of patterns belonging to all other clusters (other than the cth cluster) and n is the number of features.

Time complexity of fuzzy reflexive relational matrices:

The time complexity in computing the pairwise fuzzy reflexive relational matrices of sizes $s \times s$ (see S1 in Section 3), corresponding to all the conditional attributes (n) and all the clusters (c), is $O(cns^2)$, where $s = p_{1c} + p_{2c}$.

Time complexity of fuzzy decision classes:

For all the clusters, the time complexity in computing 2 fuzzy decision classes (see S3 in Section 3), involving weighted distances (see Eq. (9)) of patterns p_{1c} and p_{2c} (having n features), is $O(c(2(p_{1c} + p_{2c})n))$. In computing the membership values of the patterns to the 2 decision classes (see S4 in Section 3), the time complexity is also $O(c(2(p_{1c} + p_{2c})))$. Now, the time complexity in computing the average of the membership values and assigning the average value to the patterns corresponding to the 2 fuzzy decision classes (see S4 in Section 3), is $O(c(2(p_{1c} + p_{2c}) + 2(p_{1c} + p_{2c})))$. Hence, the worst case time complexity in computing fuzzy decision classes is $O(c(2(p_{1c}+p_{2c})n+2(p_{1c}+p_{2c})+2(p_{1c}+p_{2c})+2(p_{1c}+p_{2c})))$ and the asymptotic time complexity is $O(c(2(p_{1c} + p_{2c})n))$, where $(2(p_{1c} + p_{2c})n) > (2(p_{1c} + p_{2c}))$.

Time complexity of membership values of patterns for belonging to lower and upper approximations:

The aforesaid fuzzy reflexive relational matrices and the fuzzy decision classes will be required in computing membership values of the patterns p_{1c} for belonging to lower and upper approximations. Considering the fuzzy reflexive relational matrices of the sizes $p_{1c} \times s$ and the fuzzy decision classes of patterns p_{1c}, for all the conditional attributes (n) and all the clusters (c), the time complexity in computing the membership values of patterns for belonging to lower and upper approximations is $O(cnp_{1c}s)$, where p_{1c} is the number of patterns in a set x_0.

Time complexity of roughness of sets (clusters):

The time complexity of multiplying the actual membership values of the patterns of a set x_0, corresponding to a single attribute and a cluster, with their membership values for belonging to lower approximations, and then taking the sum over all of them (see Eq. (25) in Section 3) is $O(p_{1c})$. So for all the attributes it becomes $O(np_{1c})$. This is also the same for upper approximations (see Eq. (26) in Section 3). Hence, the time complexity in computing roughness of the sets, which involves the ratio of lower and upper approximations corresponding to the clusters, is $O(cnp_{1c})$.

Time complexity of FRE:

The time complexity of average FRE will involve all the complexities mentioned in Section 4.1. Therefore, the worst case time complexity of average FRE is $O(c^2p_{1c}p_{2c}n) + cns^2 + c(2((p_{1c} + p_{2c})n)) + c(n(p_{1c}s)) + cnp_{1c}$ and the asymptotic time complexity of FRE is $O(cns^2)$, where $s^2 > p_{1c} + p_{2c}$, $s^2 > s$, and $(cns^2) > (c^2p_{1c}p_{2c}n)$.

Time complexity of FRGSOM with FRE:

The asymptotic time complexity of FRGSOM, when used along with FRE, is $O(s^2(3n) + (3n)cr^2 + tsN(3n) + cns^2)$.

5. Experimental results

The clustering algorithms like SOM, FSOM, Rough SOM and FRGSOM are implemented in the C language and all the programs are executed on the Linux operating system installed in a HP computer. The computer is configured with Intel Core i7 CPU 880 @ 3.07GHZ processor and 16 GB RAM. All the clustering algorithms are tested on three different types of real-life data sets, namely, Telugu vowel [24], medical [25], and gene expression microarray data sets like Cell Cycle [27], Yeast Complex [26] and [28], and All Yeast [28].

5.1. Description of data sets

Here, we provide a brief description of the data sets, used in this investigation.

(i) *Telugu vowel data:*

The Telugu vowel data [24] deals with 871 Indian Telugu vowel sounds. It consists of 871 patterns, and each of these patterns is described by 3 features and belongs to one of the 6 vowel classes. These classes are denoted by δ, a, i, u, e and o. Fig. 1 shows the projection of the 871 data points, for each class, in F_1–F_2 feature plane. It can also be observed from Fig. 1 that there are overlapping class boundaries.

(ii) *Medical data set:*

The medical data set [25] deals with various Hepatobiliary disorders of 536 patient cases, and it consists 9 features and 4 classes. Fig. 2 shows that the projection of the 536 patterns with the first three consecutive feature values, for each class, in the F_1–F_2 feature plane.

(iii) *Gene expression microarray data sets:*

For microarray gene expression analysis, data sets like Cell Cycle [27], Yeast Complex [26,28], and All Yeast [28], are chosen. The genes in these data sets belong to Saccharomyces cerevisiae and are classified into 16, 16 and 18 groups, respectively, according to the functional annotations of the Munich Information for Protein Sequences (MIPS) [29] database. Table 4 shows the name of the data sets, number of genes in each data set, the number of time points (attributes), and the number of top level functional categories (classes) for a particular data set. Microarray data sets are often with missing gene expression values due to experimental problems. In this investigation, for Cell-Cycle data, out of 653 genes, 19 genes with missing gene expression values are first eliminated from the data set. Thereafter, the remaining 634 genes, with all expression values, are used in our experiment. Similarly, for All Yeast data, out of 6221 genes, 6072 genes, with all expression values, are used in our experiment. The Yeast Complex complex data has no missing gene expression values.

5.2. Results on Telugu vowel data

The Telugu vowel data is first transformed into a $3n$-dimensional ($n = 3$) granular space using Eq. (3). If F_{i1}, F_{i2} and F_{i3} represent the 1st, 2nd and 3rd features of the ith pattern F_i, then the fuzzy granules of the features (in terms of *low*, *medium* and *high*) are quantified as

$$\begin{aligned}
\overrightarrow{F}_{i1} &\equiv \left\{\mu_{low}(\overrightarrow{F}_{i1}), \mu_{medium}(\overrightarrow{F}_{i1}), \mu_{high}(\overrightarrow{F}_{i1})\right\},\\
\overrightarrow{F}_{i2} &\equiv \left\{\mu_{low}(\overrightarrow{F}_{i2}), \mu_{medium}(\overrightarrow{F}_{i2}), \mu_{high}(\overrightarrow{F}_{i2})\right\}, \quad \text{and}\\
\overrightarrow{F}_{i3} &\equiv \left\{\mu_{low}(\overrightarrow{F}_{i3}), \mu_{medium}(\overrightarrow{F}_{i3}), \mu_{high}(\overrightarrow{F}_{i3})\right\}.
\end{aligned}$$

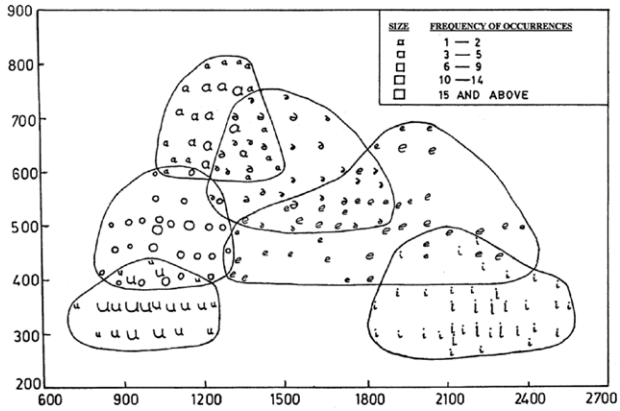

Fig. 1. Vowel data in the F_1–F_2 plane.

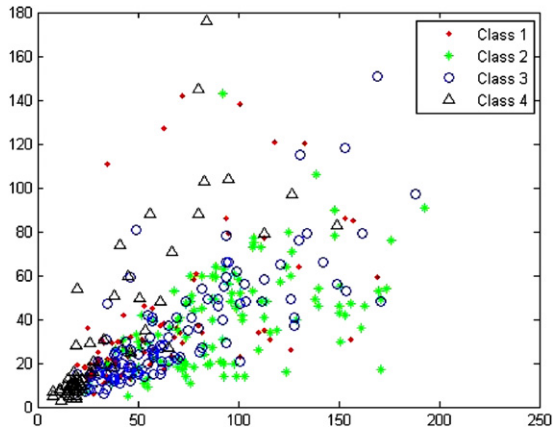

Fig. 2. Medical data in the F_1–F_2 plane.

Table 4
Summary for different microarray data sets.

Dataset	No. of genes	No. of time points	Classes
Cell Cycle	634	93	16
Yeast Complex	979	79	16
All Yeast	6072	80	18

The transformed data is then used to find a similarity matrix using Algorithm 1. Algorithm 2 (see Section 2.2.1) generates a different number of groups for different values of α (0.2, 0.25, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8) between 0 to 1. For example, 11 groups were found for $\alpha = 0.35$. These groups are then arranged in a descending order according to their size. The numbers of data points in these groups were 238, 211, 181, 94, 87, 31, 12, 8, 6, 2 and 1. Let $c = 6$ be the user defined number of clusters. We choose the top 6 groups, based on their size, among all 11 groups and the FRE values for 6 groups and the average FRE value are calculated. Similarly, we do the same process for the remaining values of α. Here, the average FRE of the first 6 groups is seen to be minimum for $\alpha = 0.35$. These 6 groups are most compact and hence presented to a decision system S. The knowledge about data, in S, is then extracted using the fuzzy rough concepts and incorporated in FRGSOM. The values of parameters, f_d and f_e, in Eq. (10), were considered to be 1.

During learning of the FRGSOM, 9 neurons are considered for the input layer as there are 3 input features in the vowel data and each feature has 3 dimensions. In the output layer of the SOM, we have considered 6 neurons corresponding to the 6 clusters. The initial connection weights, from the nodes in the input layer to the 6 nodes in SOM's output layer, are determined by dependency factors. The resultant network is then trained with competitive learning (see Section 2.5). After the completion of the learning process, the number of samples, obtained at each of the 6 output nodes of SOM, is shown in Table 5. For example, 100 data points are obtained at node 1 and 204 data points are obtained at node 6, in the SOM's output layer. The description of these clusters, in the form of a confusion matrix, is shown in Table 6.

Table 5
Alignment of patterns at winning nodes in the output layer of FRGSOM for Telugu vowel data.

$1^{(100)}$	$2^{(171)}$	$3^{(191)}$
$4^{(62)}$	$5^{(143)}$	$6^{(204)}$

Table 6
Confusion matrix corresponding to data points at the winning nodes of the FRGSOM for Telugu vowel data.

	δ	a	i	u	e	o
*cluster*1	43	11	2	3	41	0
*cluster*2	16	66	0	27	9	53
*cluster*3	0	0	145	0	46	0
*cluster*4	0	0	0	45	0	17
*cluster*5	10	0	25	0	106	2
*cluster*6	3	12	0	76	5	108

Table 7
Alignment of patterns at winning nodes in the output layer of SOM with random weights within −0.5 to 0.5, for Telugu vowel data.

$1^{(104)}$	$2^{(178)}$	$3^{(154)}$
$4^{(233)}$	$5^{(156)}$	$6^{(46)}$

Table 8
Confusion matrix corresponding to data points at the winning nodes of the SOM random weights within −0.5 to 0.5, for Telugu vowel data.

	δ	a	i	u	e	o
*cluster*1	41	14	2	1	45	1
*cluster*2	18	62	0	26	12	60
*cluster*3	0	0	110	0	44	0
*cluster*4	1	6	0	121	0	105
*cluster*5	4	0	57	0	94	1
*cluster*6	8	7	3	3	12	13

From the results in Table 6, obtained with FRGSOM using fuzzy rough sets, the sum of the diagonal elements is found to be 513. The data points belonging to different clusters and the sum of the diagonal elements are seen to corroborate well with the actual class description (see Fig. 1).

5.2.1. Comparison of FRGSOM with related algorithms

The performance of the FRGSOM is compared with the related algorithms, like (SOM) [13] and fuzzy self-organizing map (FSOM). The initial connection weights of SOM are chosen as random numbers within −0.5 to 0.5. In FSOM, an n-dimensional input vector is transformed into a $3n$-dimensional linguistic input vector corresponding to the linguistic terms *low*, *medium* and *high* (see Sections 2.1, 2.1.1 and 2.1.2). The number nodes in the SOM's input layer of FSOM is set equal to a $3n$-dimensional linguistic input vector and in the SOM's output layer, the number of nodes is set equal to the number of classes. The initial connection weights FSOM are chosen as random numbers within −0.5 to 0.5.

Time complexity of FSOM:

The time complexity of FSOM is $O(s(3n) + tsN(3n))$, where s is the number of patterns and $3n$ is the number of features, t is the number of iterations and N is the number of nodes in the SOM's output layer. Here, $O(s(3n))$ represents the time complexity for transforming an n-dimensional input vector into a $3n$-dimensional linguistic vector and $tsN(3n)$ refers to the time complexity of SOM, with the number of nodes in the input layer equal to the $3n$-dimensional linguistic vector. Therefore, the asymptotic time complexity of FSOM is $O(tsN(3n))$.

Tables 7 and 8 show the performance of SOM. The results of FSOM are shown in Tables 9 and 10.

The results in Tables 8 and 10 show that the SOM and FSOM are clustering the data points into 6 groups and the sum of the diagonal elements is 441 and 502, respectively.

Table 9
Alignment of patterns at winning nodes in the SOM's output layer of FSOM with random weights within −0.5 to 0.5, for Telugu vowel data.

$1^{(106)}$	$2^{(176)}$	$3^{(198)}$
$4^{(63)}$	$5^{(137)}$	$6^{(191)}$

Table 10
Confusion matrix corresponding to data points at the winning nodes of the FSOM with random weights within −0.5 to 0.5, for Telugu vowel data.

	δ	a	i	u	e	o
*cluster*1	46	11	2	7	40	0
*cluster*2	13	66	0	25	9	63
*cluster*3	0	0	146	0	52	0
*cluster*4	0	0	0	45	0	18
*cluster*5	11	0	24	0	101	1
*cluster*6	2	12	0	74	5	98

Let us now analyze the cluster wise results of FRGSOM, as compared to SOM and FSOM.

Cluster 1:

It can be observed from Fig. 1 that cluster 1, corresponding to class δ, has overlapping boundaries with the classes a, e and o. This information is also preserved and reflected in our results for cluster 1 using FRGSOM, where, out of 100 data points (see node 1 in Table 5), 43, 11 and 41 data points belong to the classes δ, a and e (see Table 6), respectively. In contrast, using SOM, out of 104 data points at node 1 (see Table 7), 41, 14, 45 and 1 data points belong to the classes δ, a, e and o, respectively, i.e., only 41 points belong to class δ. Using FSOM, out of 106 data points at node 1 (see Table 9), 46, 11 and 40 data points belong to the classes δ, a and e, respectively (see cluster 1 in Table 10), i.e., only 46 samples correspond to class δ. Although the overlapping class information is reflected in the results for all the methods, the number of data points (100, 104 and 106, respectively) allotted in cluster 1 for all the methods are higher than the actual number of points (72) and FRGSOM is moderately better than related methods. The results for SOM and FSOM show that class δ does not have overlapping information with class o, which is in contrast with the actual class information. Moreover, in SOM, the actual number of points (diagonal entry) belonging to class δ is less than those points belonging to class e, whereas, for a confusion matrix, the diagonal entry should be the dominant entry in each row.

Cluster 6:

Now consider cluster 6, where class o has overlapping boundaries with the classes δ, a, u and e (see Fig. 2). This information is supported by FRGSOM (see Table 6), where, out of 204 data points at node 6 (see Table 5), 3, 12, 76, 5, and 108 data points belong to the classes δ, a, u, e, and o, respectively. In contrast, while using SOM, 46 data points are obtained at node 6 (see Table 7), and 8, 7, 3, 12, and 13 data points belong to the classes δ, a, u, e, and o, respectively (see Table 7). From the results of FSOM, we can observe that 191 data points are obtained at node 6 (see Table 9), and 2, 12, 74, 5, and 98 data points belong to the classes δ, a, u, e, and o. Here, FRGSOM performs better than SOM and FSOM in terms of the total number of data points and actual data points. Moreover, this shows the effectiveness of incorporating fuzzy rough sets in extracting domain knowledge. The results of SOM also show that the class o has an overlapping boundary with the class i, which is not correct. Similar conclusions on the FRGSOM, SOM and FSOM can also be made, based on results for the remaining clusters.

5.2.2. Evaluation of clusters

In this section, we evaluate clusters, obtained by using SOM, FSOM and FRGSOM, based on the proposed fuzzy rough entropy (FRE) for $c = 4$, 6 and 8. In addition, comparisons are provided using the β-index [31] and Davies–Bouldin index (DB-index) [30]. A higher value of β-index indicates that the clustering solutions are compact whereas it is the opposite for the FRE and DB-index. For a given c and a clustering method, all indices are computed on the resulting clustering solution and the results are reported in Table 11.

From Table 11, for $c = 4$, 6 and 8, we find that the values of β-index and DB-index for FRGSOM are higher and lower, respectively, than those of FSOM and SOM. The FRE values for FRGSOM are also lower than those of FSOM and SOM. This signifies that the proposed FRGSOM performs the best in terms of all the indices. We can also observe that, for FRGSOM, FSOM and SOM, the β-index and DB-index for $c = 4$ are higher and lower, respectively, than those values for $c = 6$ and 8. This means $c = 4$ is the best choice for Telugu vowel data according to the β-index and DB-index. In contrast, for all the clustering methods, FRE values are seen to be the lowest for $c = 6$. Note that Telugu vowel data has 6 overlapping classes (shown in Fig. 1) and this information is therefore truly reflected by FRE.

Table 11
β-index, DB-index and FRE values of clustering methods for Telugu vowel data.

Algorithm	No. of clusters $c = 4$			No. of clusters $c = 6$			No. of clusters $c = 8$		
	β-index	DB index	FRE	β-index	DB index	FRE	β-index	DB index	FRE
FRGSOM	1.3045	1.1366	0.1688	1.1541	1.4687	0.1651	1.1116	1.6006	0.2163
FSOM	1.2032	1.1670	0.3342	1.0753	1.5256	0.3228	0.9827	1.7704	0.4220
SOM	0.9488	3.5593	0.5681	0.9317	1.5330	0.4452	0.9206	1.8450	0.4508

Table 12
Comparison of CPU time in seconds for FRGSOM, FSOM and SOM for Telugu vowel data.

Algorithm	4 clusters	6 clusters	8 clusters
FRGSOM	3.0490	3.074	3.164
FSOM	0.5070	0.6520	1.8360
SOM	0.4450	0.6410	1.5660

Table 13
Comparison of *FRE* values of SOM, FSOM and FRGSOM for $c = 6$.

Clusters	SOM	FSOM	FRGSOM
1	0.3950	0.6340	0.1818
2	0.5479	0.3563	0.3300
3	0.2250	0.0350	0.0308
4	0.7609	0.1847	0.0858
5	0.2754	0.5478	0.2922
6	0.4673	0.1790	0.0981
Avg. FRE	0.4452	0.3228	0.1698

Table 14
Alignment of patterns at winning nodes in the SOM's output layer of FRGSOM for medical data.

$1^{(141)}$	$2^{(143)}$
$3^{(124)}$	$4^{(128)}$

Table 12 provides CPU time in seconds for the FRGSOM, FSOM and SOM, corresponding to results of the clusters shown in Table 11, for Telugu vowel data. It is evident that the CPU time for FRGSOM, which has the highest time complexity, for $c = 4$, 5 and 6 is higher than FSOM and SOM. However, the performance of FRGSOM shown in Table 11 is better than the FSOM and SOM.

As a typical example, now we compare the values of FRE for clusters obtained by SOM, FSOM and FRGSOM for $c = 6$. The results are shown in Table 13. The FRE values for cluster 1, obtained by SOM and FSOM, are 0.3950 and 0.6340, respectively. In contrast, the same for cluster 1, obtained by FRGSOM, is 0.1818, and it indicates that cluster 1 for FRGSOM is more compact than SOM and FSOM. Similar results are observed for the remaining clusters, except cluster 5, where SOM performs the best and FRGSOM is the second best. As per the average FRE, the proposed FRGSOM is therefore seen to be the best.

5.3. Results on medical data

For medical data [25], first, c is chosen as 4 as the actual number of clusters is four and the actual number of points in each cluster (from 1 to 4) is 116, 178, 124 and 118, respectively. The number of nodes in the SOM's input layer is set to 27 in the fuzzy rough granular self-organizing map (FRGSOM). The number of nodes in the SOM's output layer is set to 4 and these are arranged in a 2×2 grid. The dependency factors for medical data are determined by using fuzzy rough sets in a similar way, as explained for Telugu vowel data in Section 5.2.

FRGSOM partitions the data into four clusters (see Table 14). The confusion matrix corresponding to these clusters is shown in Table 15 and the number of data points in each cluster is 141, 143, 124 and 128. In contrast, the number of data points in each cluster for SOM and FSOM, which are also set to partition the data into four clusters, is 169, 86, 112, 169 (see Table 16) and 131, 126, 167, 112 (see Table 18), respectively. The confusion matrix corresponding to the clusters of SOM

Table 15
Confusion matrix corresponding to data points at the winning nodes of the FRGSOM for medical data.

	class1	class2	class3	class4
*cluster*1	49	46	23	23
*cluster*2	19	56	37	31
*cluster*3	29	36	40	19
*cluster*4	19	40	24	45

Table 16
Alignment of patterns at winning nodes in the output layer of SOM for medical data.

$1^{(169)}$	$2^{(86)}$
$3^{(112)}$	$4^{(169)}$

Table 17
Confusion matrix corresponding to data points at the winning nodes in the output layer of the SOM for medical data.

	class1	class2	class3	class4
*cluster*1	50	33	39	47
*cluster*2	9	44	27	6
*cluster*3	12	67	22	11
*cluster*4	45	34	36	54

Table 18
Alignment of patterns at winning nodes output layer of FSOM, for medical data.

$1^{(131)}$	$2^{(126)}$
$3^{(167)}$	$4^{(112)}$

Table 19
Confusion matrix corresponding to data points at the winning nodes in the output layer of FSOM, for medical data.

	class1	class2	class3	class4
*cluster*1	48	39	27	17
*cluster*2	19	46	25	36
*cluster*3	24	57	57	29
*cluster*4	25	36	15	36

and FSOM is shown in Tables 17 and 19, respectively. From the results in the confusion matrix we can observe that the overlapping information is preserved by all the clustering algorithms. The number of data points in cluster 2, obtained by SOM and FSOM, is 86 and 126, respectively. In contrast, this is 143 for FRGSOM and is much closer to the actual number of points. Additionally, for SOM, in cluster 3, the diagonal entry is not a dominant entry in that row. Similarly, comparison between FRGSOM and the related clustering algorithms can be made from the remaining part of the experimental results and it can be concluded that FRGSOM performs better for medical data.

5.3.1. Evaluation of clusters

Here, we evaluate clusters (for $c = 2, 4$ and 6) produced by SOM, FSOM and the proposed FRGSOM, based on fuzzy rough entropy (FRE), β-index and DB index, and the results are depicted in Table 20.

From Table 20, it is clear that the value of β-index for FRGSOM is higher than the values of FSOM and SOM for $c = 2, 4$ and 6. The DB index and FRE values are also the lowest for FRGSOM for all the values of c. That means the proposed methods perform the best in terms of all three indices. Further, FRE, DB-index and β-index result in best performance when $c = 4, 6$

Table 20
β-index, DB-index and FRE values of clustering methods for medical data.

Algorithm	No. of clusters $c = 2$			No. of clusters $c = 4$			No. of clusters $c = 6$		
	β-index	DB index	FRE	β-index	DB index	FRE	β-index	DB index	FRE
FRGSOM	0.4840	13.6805	0.2837	0.2683	17.1123	0.2385	0.1934	11.2395	0.3007
FSOM	0.4801	24.0973	0.3699	0.2525	20.0444	0.3682	0.1648	13.7342	0.4182
SOM	0.4744	29.6839	0.3815	0.2167	27.9271	0.4228	0.1213	14.2989	0.4928

Table 21
Comparison of CPU time in seconds for FRGSOM, FSOM and SOM for medical data.

Algorithm	2 clusters	4 clusters	6 clusters
FRGSOM	1.1930	1.5180	2.315
FSOM	0.8160	1.0210	1.2900
SOM	0.5180	0.7680	1.0430

Table 22
Comparison of fuzzy rough entropy (FRE) values for all the algorithms.

Clusters	SOM	FSOM	FRGSOM
1	0.5940	0.5790	0.3454
2	0.4284	0.4168	0.1773
3	0.2211	0.0444	0.0988
4	0.4474	0.4326	0.3327
Avg. FRE	0.4228	0.3682	0.2385

and 2 respectively. Considering the actual number of clusters in the medical data, which is four, it can be concluded that it is only FRE which truly reflects the fact.

Table 21 provides CPU time in seconds for FRGSOM, FSOM and SOM, corresponding to the clustering solutions in terms of indices in Table 20, for medical data. It is clear that the CPU time for the FRGSOM for all the clusters is higher than the FSOM and the SOM. However, the performance of the FRGSOM, shown in Table 20, is better than the SOM and the FSOM.

Now, we compare the values of FRE for clusters obtained by SOM, FSOM and FRGSOM for $c = 4$, as a typical example. The results are presented in Table 22. By observing FRE values of the clusters and the average FRE of them from Table 22, we can say that, FRE values for FRGSOM are the lowest and it performs better than FSOM and SOM.

5.4. Results on microarray data sets

The performance of the proposed FRGSOM is also compared with SOM and FSOM, using gene expression microarray data sets, like Cell Cycle, Yeast Complex and All Yeast. After clustering the genes, using the aforementioned algorithms, one or several functional categories are assigned to each cluster by calculating the P-values for different functional categories in Munich Information for Protein Sequences (MIPS) [29]. Using hypergeometric distribution, the probability (P-value) of observing at least m genes from a functional category within a cluster of size n is given by

$$P = 1 - \sum_{i=0}^{m-1} \frac{\binom{f}{i}\binom{N-f}{n-i}}{\binom{N}{n}} \tag{31}$$

where f is the total number of genes within a functional category and N is the total number of genes within the genome. A lower P-value for a functional category within a cluster indicates that the functionally related genes are grouped in a better way by the clustering method. Experiments are performed for $c = 16$, 16 and 18 for Cell Cycle, Yeast Complex and All Yeast data sets, respectively. All data sets are initially transformed into a 3-dimensional granular space, using Eq. (3). During the learning of FRGSOM, the number of nodes in the SOM's input layer is set to 279, 237, and 240 and the number of nodes in the SOM's output layer is set to 16, 16 and 18 for Cell Cycle, Yeast Complex and All Yeast data sets, respectively. After completion of competitive learning of FRGSOM, microarray data sets are clustered corresponding to the number of user defined clusters. The results of Cell Cycle, Yeast Complex and All Yeast data sets are presented in Tables 23–25, respectively. For each data set, the cluster numbers, the number of genes in each cluster, the name of the functional category (obtained from the most

Table 23
Results for Cell Cycle Microarray Data using FRGSOM.

Cluster no.	No. of genes within cluster	MIPS functional category	Category related genes within cluster	No. of genes within genome	*P*-value
1	102	Phosphate metabolism	21	418	$8.05e^{-06}$
		Modification by phosphorylation, dephosphorylation, autophosphorylation	18	186	$2.29e^{-09}$
		Cytoskeleton/structural proteins	22	252	$2.20e^{-10}$
		Fungal and other eukaryotic Cell type differentiation	27	452	$7.67e^{-09}$
2	87	Modification by phosphorylation, dephosphorylation, autophosphorylation	8	186	$9.62e^{-04}$
		Microtubule cytoskeleton	9	47	$1.44e^{-09}$
		Fungal and other eukaryotic cell type differentiation	16	452	$2.39e^{-05}$
3	65	Phosphate metabolism	12	418	$2.57e^{-04}$
		Cell fate	9	273	$6.32e^{-04}$
		Fungal and other eukaryotic cell type differentiation	13	452	$1.32e^{-04}$
4	49	Modification by phosphorylation, dephosphorylation, autophosphorylation	8	186	$9.59e^{-05}$
		Cell fate	11	273	$7.19e^{-06}$
		Cytoskeleton/structural proteins	11	252	$3.31e^{-06}$
		Budding, cell polarity and filament formation	13	313	$6.21e^{-07}$
5	47	DNA damage response	5	77	$3.08e^{-04}$
		Cytoskeleton/structural proteins	8	252	$6.60e^{-04}$
		Development of asco-, basidio- or zygospore	9	167	$4.49e^{-06}$
6	36	Kinase inhibitor	2	14	$4.34e^{-03}$
		Small GTPase mediated signal transduction	5	58	$5.21e^{-05}$
		Budding, cell polarity and filament formation	8	313	$1.49e^{-03}$
7	33	Protein binding	8	392	$1.33e^{-03}$
		Microtubule cytoskeleton	4	47	$1.33e^{-04}$
8	33	Phosphate metabolism	9	418	$2.58e^{-04}$
		Modification by phosphorylation, dephosphorylation, autophosphorylation	8	186	$4.48e^{-06}$
		Fungal and other eukaryotic Cell type differentiation	12	452	$1.90e^{-06}$
9	30	Kinase inhibitor	2	14	$2.17e^{-03}$
		Pheromone response, mating-type determination	7	189	$3.33e^{-05}$
		Fungal and other eukaryotic cell type differentiation	11	452	$6.78e^{-06}$
10	29	Mating (fertilization)	3	69	$4.46e^{-03}$
		Cytoskeleton/structural proteins	9	252	$1.95e^{-06}$
		Budding, cell polarity and filament formation	7	313	$6.22e^{-04}$
11	28	Protein/peptide degradation	5	256	$4.63e^{-03}$
		Enzymatic activity regulation	7	180	$9.05e^{-06}$
		Cell aging	3	28	$2.32e^{-04}$
12	27	Phosphate metabolism	7	418	$8.15e^{-04}$
		Modification by phosphorylation, dephosphorylation, autophosphorylation	7	186	$4.73e^{-06}$
		Microtubule cytoskeleton	4	47	$2.88e^{-05}$
		Modification by acetylation, deacetylation	3	69	$1.80e^{-03}$
13	20	Enzymatic activity regulation	4	180	$3.47e^{-03}$
		Cytoskeleton/structural proteins	5	252	$1.67e^{-03}$
14	18	Vesicular cellular export	2	33	$3.63e^{-03}$
		Exocytosis	2	33	$3.63e^{-03}$

Table 23
(Continued.)

Cluster no.	No. of genes within cluster	MIPS functional category	Category related genes within cluster	No. of genes within genome	*P*-value
		Cell growth/morphogenesis	5	238	$2.60e^{-04}$
15	18	Cytoskeleton/structural proteins	4	252	$3.43e^{-03}$
		Mitotic cell cycle	3	165	$8.29e^{-03}$
16	12	Development of asco-, basidio- or zygospore	3	167	$6.41e^{-04}$

significant *P*-value using Eq. (31)), the actual number of genes within the category, the actual number of genes within the genome, and the functional category related *P*-values are also shown in the tables.

Using FRGSOM on Cell Cycle data, we found that every cluster shows functional enrichment in cell-cycle, mitotic cell cycle, cell cycle control and mitotic cell cycle as the data is itself related with cell cycle. The lowest and highest *P*-values involved in these functional categories are $2.38e^{-76}$ and $2.31e^{-03}$, respectively. The functional enrichment in other categories, other than Cell Cycle, are shown in Table 23. From Table 23, we can observe that 15 out of 16 clusters have more than 15 genes and show functional enrichment in more than one category. As a typical example, cluster 1 have 102 genes and shows functional enrichment in phosphate metabolism, modification by phosphorylation dephosphorylation, autophosphorylation, cytoskeleton/structural proteins, and fungal and other eukaryotic cell type differentiation category with *P*-values $8.05e^{-06}$, $2.29e^{-09}$, $2.20e^{-10}$ and $7.67e^{-09}$, respectively. The functional enrichment results for SOM and FSOM are not provided here but, we present a comparison among different clustering methods in terms of *NP* value, obtained from functional enrichment of clustering solutions. The *NP* is defined as

$$NP = \prod_{i=1}^{n} (1 - P_i) \tag{32}$$

where P_i represents the most significant *P*-value (lowest one) associated with the *i*th cluster, and *n* represents the total number of clusters obtained using any clustering method. Here, $(1 - P_i)$ gives the probability of *not* observing the related functional enrichment in the *i*th cluster and *NP* represents the probability of *not* observing any functional enrichment in all the clusters, found by any clustering method. So, a higher *NP* value for a clustering method indicates that the genes have a better functional relationship in the clustering solutions. Using Cell Cycle data, the values of *NP* for FRGSOM, SOM and FSOM are found to be 0.99, 0.94, and 0.95, respectively. The experimental results for Yeast Complex and All Yeast microarray data sets, using FRGSOM, are presented in Tables 24 and 25, respectively. The *NP* values for Yeast Complex and All Yeast microarray data sets are found to be 0.98 and 0.99 for FRGSOM, 0.96 and 0.97 for FSOM, and 0.96 and 0.95 for SOM. For all of the gene expression data sets, the *NP* values for FRGSOM are found to be greater than the related methods and indicate that FRGSOM groups the functionally related genes in a better fashion.

5.5. Evaluation of clusters

In this section, clustering solutions of the Cell-Cycle, Yeast Complex and All Yeast microarray data sets are also evaluated in terms of β-index, DB-index and FRE. The values of indices are depicted in Table 26. It can be seen from Table 26 that the values of DB-index and FRE are smallest for FRGSOM as compared to FSOM and SOM for all the gene expression microarray data sets. The results signify that the proposed FRGSOM clusters microarray data sets and groups the functionally related genes in a better fashion than the other two related clustering methods. The β-index is also seen to corroborate with FRE and DB, except for the Yeast Complex data.

Table 27 provides CPU time in seconds for FRGSOM, FSOM and SOM, corresponding to clustering solutions in Table 26, for gene expression data sets. It can be observed that FRGSOM takes more time than FSOM and SOM for all the data sets. However, from Table 26, the performance of FRGSOM is seen to be better than the FSOM and SOM.

5.6. Comparison of FRGSOM with Rough SOM

So far we have demonstrated the performance of the proposed method (FRGSOM) with SOM and FSOM in detail. In a part of the experiment, we also compared the performance of FRGSOM with RSOM (Rough SOM) [4]. The methodology of Rough SOM is described in [4] as follows:

Step 1. An *n*-dimensional input data is transformed into a $3n$-dimensional linguistic data, using a π-membership function with centers and scaling factors, corresponding to linguistic terms *low*, *medium* and *high*.
Step 2. Attributes which have membership values $\geq$ threshold value (0.5) are represented with 1 and others are represented with 0, in order to make a binary valued data.
Step 3. The most representative pattern, i.e., a pattern which is repeated a maximum number of times, is selected from the resultant binary valued data to serve as an object. Let there be *m* sets of objects, $O_1, O_2, \ldots, O_m$, in the attribute valued table such that $n_{k_1} > n_{k_2} >, \ldots, n_{k_m}$, where $card(O_i) = n_{k_i}$ and $i = 1, 2, \ldots, m$.

Table 24
Results for Yeast Complex microarray data using FRGSOM.

Cluster no.	No. of genes within cluster	MIPS functional category	Category related genes within cluster	No. of genes within genome	*P*-value
1	108	Protein synthesis	96	480	$1.51e^{-104}$
		Ribosome biogenesis	96	480	$1.51e^{-104}$
		Ribosomal proteins	91	246	$1.15e^{-123}$
2	92	Transcription	52	1077	$6.75e^{-13}$
		RNA synthesis	37	634	$3.02e^{-11}$
		mRNA synthesis	35	576	$4.05e^{-11}$
3	87	Transcription	44	1077	$1.57e^{-10}$
		RNA synthesis	26	634	$3.30e^{-06}$
		mRNA synthesis	25	576	$1.92e^{-06}$
4	84	Protein/peptide degradation	28	256	$1.07e^{-19}$
		Cytoplasmic and nuclear protein degradation	27	188	$3.70e^{-22}$
		Proteasomal degradation	27	128	$6.98e^{-27}$
5	83	Transcription	38	1077	$2.14e^{-10}$
		RNA synthesis	20	634	$8.86e^{-05}$
		mRNA synthesis	16	576	$6.06e^{-03}$
6	66	Protein synthesis	26	480	$1.70e^{-11}$
		Ribosome biogenesis	20	480	$3.19e^{-10}$
		Ribosomal proteins	20	246	$4.63e^{-12}$
7	66	Transcription	27	1077	$4.64e^{-6}$
		RNA synthesis	19	634	$1.83e^{-05}$
		mRNA synthesis	17	576	$6.98e^{-05}$
8	62	Transcription	21	1077	$1.41e^{-03}$
		RNA synthesis	16	634	$4.10e^{-04}$
		rRNA synthesis	6	576	$3.49e^{-05}$
9	56	Transcription	24	1077	$1.19e^{-05}$
		RNA synthesis	16	634	$3.78e^{-05}$
		mRNA synthesis	15	576	$1.72e^{-04}$
10	56	Energy	35	367	$7.78e^{-31}$
		Electron transport and membrane -associated energy conservation	16	61	$1.46e^{-40}$
		Respiration	15	138	$3.66e^{-28}$
11	46	Cell cycle	33	367	$7.78e^{-31}$
		Mitotic cell cycle and cell cycle control	28	61	$1.46e^{-40}$
		Mitotic cell cycle	15	138	$3.66e^{-28}$
12	44	Protein synthesis	14	480	$1.84e^{-06}$
		Ribosome biogenesis	11	310	$4.85e^{-06}$
		Ribosomal proteins	11	246	$4.94e^{-07}$
13	42	General transcription activities	9	235	$1.19e^{-05}$
		Transcription initiation	6	46	$3.45e^{-07}$
		Protein with binding function or cofactor requirement	17	1049	$9.49e^{-05}$
14	36	Protein synthesis	18	480	$1.39e^{-13}$
		Ribosome biogenesis	11	310	$5.42e^{-09}$
		Ribosomal proteins	12	246	$3.78e^{-10}$
15	35	Cytoskeleton/structural proteins	5	252	$3.65e^{-03}$
		Microtubule cytoskeleton	3	47	$9.17e^{-03}$
		Nucleus	4	149	$3.30e^{-03}$
16	16	DNA synthesis and replication	10	138	$3.68e^{-08}$
		Ori recognition and priming complex formation	5	25	$1.10e^{-08}$
		Extension/polymerization activity	5	37	$8.79e^{-08}$

Table 25
Results for All Yeast Microarray Data using FRGSOM.

Cluster no.	No. of genes within cluster	MIPS functional category	Category related genes within cluster	No. of genes within genome	*P*-value
1	540	Transcription	237	1077	$4.86e^{-13}$
		RNA synthesis	144	634	$6.73e^{-09}$
		mRNA synthesis	139	576	$1.66e^{-10}$
2	537	Metabolism of derivatives of dehydroquinic and chorismic acid	6	13	$6.34e^{-04}$
		RNA synthesis	4	5	$3.52e^{-04}$
		Actin cytoskeleton	18	96	$3.06e^{-03}$
3	500	Ribosome biogenesis	42	310	$1.68e^{-05}$
		Ribosomal proteins	38	246	$1.91e^{-06}$
		mitochondrion	41	171	$7.04e^{-13}$
4	495	Transcriptional control	57	495	$9.31e^{-05}$
		Protein fate (fold., mod., dest.)	122	1154	$3.47e^{-07}$
		Protein modification	67	616	$1.27e^{-04}$
5	490	Cell cycle and DNA processing	95	1012	$2.07e^{-07}$
		DNA processing	54	520	$9.24e^{-06}$
		Proteasomal degradation	24	128	$1.87e^{-07}$
6	461	Modification by acetylation, deacetylation	11	69	$1.21e^{-03}$
		Cellular transport, transport facilities and transport routes	77	1038	$2.47e^{-03}$
7	450	Protein modification	4	616	$1.36e^{-03}$
		O-directed glycosylation, deglycosylation	4	15	$4.23e^{-03}$
8	367	Cellular sensing and response to external stimulus	23	284	$8.48e^{-04}$
		Chemoperception and response	21	240	$5.23e^{-04}$
		Pheromone response, mating-type determination, sex-specific proteins	20	189	$5.43e^{-05}$
9	343	C-compound and carbohdt. metabolism	48	505	$1.26e^{-08}$
		Energy	52	367	$3.44e^{-16}$
		Electron transport and membrane -associated energy conservation	20	61	$9.88e^{-14}$
10	290	Protein processing (proteolytic)	13	89	$1.83e^{-05}$
		Cyto. and nuclear prot. degrad.	17	188	$4.96e^{-04}$
		Proteasomal degradation	16	128	$1.53e^{-05}$
11	264	Metabolism of the aspartate family	10	66	$1.50e^{-04}$
		Translation initiation	7	40	$4.73e^{-04}$
		Metabolism of cyclic nucleotides	4	17	$2.66e^{-03}$
12	238	Protein synthesis	115	480	$6.16e^{-76}$
		Ribosome biogenesis	108	310	$5.05e^{-90}$
		Ribosomal proteins	93	246	$1.05ee^{-79}$
13	210	Metabolism	73	1514	$4.05e^{-05}$
		Cellular signaling	16	199	$1.17e^{-04}$
		Enzyme mediated signal transduction	13	133	$2.86e^{-04}$
14	205	RNA processing	39	436	$1.82e^{-10}$
		rRNA processing	36	205	$6.39e^{-19}$
		Protein synthesis	45	480	$9.27e^{-13}$
15	195	Cell type differentiation	43	452	$5.95e^{-13}$
		Fungal/microorganismic cell type differentiation	43	452	$5.95e^{-13}$
		Fungal and other eukaryotic cell type differentiation	43	452	$5.95e^{-13}$
16	188	Cell cycle and DNA processing	74	1012	$4.72e^{-17}$
		Cell cycle	67	653	$2.75e^{-23}$
		Mitotic cell cycle and cell cycle control	36	447	$3.67e^{-09}$

Continued on next page

Table 25
(Continued.)

Cluster no.	No. of genes within cluster	MIPS functional category	Category related genes within cluster	No. of genes within genome	*P*-value
17	156	Energy	20	367	$1.29e^{-03}$
		Respiration	13	136	$5.61e^{-05}$
		Aerobic respiration	9	77	$1.58e^{-04}$
18	143	Cell cycle and DNA processing	49	1012	$2.00e^{-08}$
		Cell cycle	38	653	$1.16e^{-08}$
		Mitotic cell cycle and cell cycle control	27	447	$1.21e^{-06}$

Table 26
β-index, DB-index and FRE values of clustering methods for gene expression data sets.

Algorithm	Cell Cycle			Yeast Complex			All Yeast		
	No. of clusters $c = 16$			No. of clusters $c = 16$			No. of clusters $c = 18$		
	β-index	DB index	FRE	β-index	DB	FRE	β-index	DB	FRE
FRGSOM	0.0663	13.2738	0.1278	0.0643	20.9827	0.1319	0.0561	41.7863	0.0364
FSOM	0.0631	14.6150	0.1864	0.0622	21.8147	0.1461	0.0548	42.3979	0.0564
SOM	0.0641	16.8528	0.2097	0.0648	24.7178	0.1525	0.0556	56.1796	0.0604

Table 27
Comparison of CPU time in seconds for the FRGSOM, FSOM and SOM for gene expression data sets.

Algorithm	Cell Cycle	Yeast Complex	All Yeast
	16 clusters	16 clusters	18 clusters
FRGSOM	10.8330	16.6825	555.3600
FSOM	9.5170	13.5470	182.4230
SOM	5.8220	12.9230	176.351

Step 4. m objects, selected from m different sets, are presented to the attribute value table of size $m \times 3n$, where m represents the number of objects, and $3n$ represents the size of the $3n$-dimensional binary valued attributes.

Step 5. Let $n'_{k_1}, n'_{k_2}, \ldots, n'_{k_m}$ denote the distant elements (cardinalities) among the $n_{k_1}, n_{k_2}, \ldots, n_{k_m}$ such that $n'_{k_1} > n'_{k_2} > \cdots > n'_{k_m}$.

Step 6. A heuristic threshold, Tr, is defined as

$$Tr = \left\lceil \frac{\sum_{i=1}^{m} \frac{1}{n'_{k_i} - n'_{k_{(i+1)}}}}{TH} \right\rceil \tag{33}$$

where TH is a threshold chosen within 0 to 1. Now, the objects which have the cardinalities less than the threshold value Tr are removed from the attribute valued table. The resulting attribute valued table is called a reduced attribute valued table or information system, say S.

Step 7. Attribute reducts are generated from information system S based on the methodology explained in [4]. Let $U = \{x_1, \ldots, x_p\}$ and $A = \{a_1, a_2, \ldots, a_{3n}\}$ be the set of objects and the set of binary valued attributes, respectively, appearing in S. A discernibility matrix, c_{ij} of S, of size $p \times p$, is defined as

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j), \quad \text{for } i \text{ and } j = 1, 2, \ldots, p, \tag{34}$$

where $a(x_i)$ and $a(x_j)$ represent the attribute values of the patterns x_i and x_j, respectively. For each object $x_i \in U$, the discernibility function $f_S^{x_i}(a_1, \ldots, a_{3n})$ is defined as

$$f_S^{x_i}(a_1, \ldots, a_{3n}) = \bigwedge \left\{ \bigvee (c_{ij}) : 1 \leq j < i \leq p, c_{ij} \neq \emptyset \right\}, \tag{35}$$

where $\bigvee (c_{ij})$ is the disjunction of all variables of a such that $a \in c_{ij}$.

In Rough SOM, the number of nodes in the input layer is determined corresponding to 3-dimensional features, and in the SOM output layer, it is defined based on the number of reducts. Here, every node in the output layer is set corresponding to a reduct. The initial connection weights, between nodes of the input layer and the output layer of Rough SOM, are defined with large random numbers for the attributes appearing in the reducts and with small random numbers for the attributes which did not appear in the reducts.

Table 28
Comparison of FRGSOM with Rough SOM, in terms of β-index, DB index, proposed fuzzy rough entropy (FRE) and f_c, for Telugu vowel and Medical Data. The results are shown for 6 clusters as Rough SOM provides 6 clusters for both the data sets.

Data set	Algorithm	Iterations	β-index	DB index	FRE	f_c	CPU time in s
Telugu vowel data	FRGSOM	80	1.1541	1.4687	0.1690	513	3.0740
	Rough SOM	132	0.9911	1.6644	0.1699	464	2.1177
Medical data	FRGSOM	60	0.1934	11.2395	0.2287	–	1.5180
	Rough SOM	111	0.1632	12.6948	0.2619	–	1.4340

Time complexity of Rough SOM:

Now, we discuss the time complexity of Rough SOM. The time complexity in computing 3-dimensional linguistic data (see Step 1 Section 5.6) is $O(s(3n))$, where s is the number of patterns and $3n$ is the number of features. In computing a binary valued data (see Step 2 in Section 5.6), the time complexity is $O(s(3n))$. The time complexity in finding the most representative patterns (see Step 3 in Section 5.6), in sorting cardinalities of the most representative patterns of sets (m) in increasing order (see Step 3 in Section 5.6), in computing the attribute valued table (see Step 4 in Section 5.6) and a heuristic threshold (see Step 6 in Section 5.6) is $O(s^2(3n))$, $O(mlogm)$, $O(m(3n))$ and $O(m-1)$, respectively. The time complexity of discernibility functions by using a reduced attribute valued table, considering disjunction of attributes (say clauses), and disjunction of prime implicants (reducts), is $O(p(3n)+p^2(3n)+p(d_pa+a^2)$, where p is the number of patterns in the reduced attribute valued table, d_p is the total number of clauses determined corresponding to a discernibility function $f_S^{x_p}$ and a is the number of attributes. Hence, the total time complexity in computing reducts is $O(s(3n)+s(3n)+s^2(3n)+mlogm+m(3n)+(m-1)+p(3n)+p^2(3n)+p(d_pa+a^2))$. Therefore, the asymptotic time complexity is $O(s^2 3n+p(d_pa+a^2))$, where $s^2 > s > m > p$, and $s^2 > p^2$.

The time complexity in computing the initial connection weights of Rough SOM is $O(r(3n))$, where r is the number of reducts and $3n$ is the number of features. Therefore, the time complexity of Rough SOM is $O(s^2(3n)+p(d_p(a)+(a)^2)+r(3n)+tsN(3n))$, where the term $O(tsN(3n))$ is the time complexity of SOM. Here, $3n$ represents the number nodes in the SOM's input layer, which is set equal to a $3n$-dimensional attributes.

Table 28 show the performance of FRGSOM and Rough SOM in terms of β-index, DB-index and fuzzy rough entropy (FRE). For both the methods, the results are shown on Telugu vowel data and medical data for 6 clusters. The value corresponding to f_c in Table 28 represents the sum of diagonal elements in the confusion matrix for Telugu vowel data.

From the results in Table 28, we observe that FRGSOM performs better than Rough SOM for Telugu vowel data in terms of β-index and DB-index, whereas, FRE values of FRGSOM and Rough SOM are seen to be almost the same. The f_c values for Telugu vowel data indicate that, the total number of points, correctly classified by FRGSOM is much higher than that of RSOM (e.g., 513 vs. 464). Similar results are also found for medical data where FRGSOM performed better than the Rough SOM in terms of all three indices. It can also be observed that the CPU time for FRGSOM is higher than Rough SOM for both the data sets. Note that the f_c is not computed for medical data as the actual number of clusters is 4 in contrast to 6 clusters, obtained using FRGSOM and Rough SOM.

We obtain 6, 12 and 14 reducts for Cell-Cycle, Yeast Complex and All Yeast data sets, respectively, by using Rough SOM where each reduct corresponds to one cluster. Hence, for the purpose of comparison the performance of FRGSOM, FSOM and SOM is also tested on the same numbers of clusters (6, 12 and 14) for those data sets. The performances of all the clustering methods in terms of β-index, DB-index and fuzzy rough entropy (FRE) for different data sets are shown in Tables 29–31. From the results given in Table 29, it can be observed that the β-index for FRGSOM is higher than that of RSOM, FSOM, SOM, and the DB index and FRE values for FRGSOM are less than that of RSOM, FSOM and SOM. It is also observed that the CPU time for FRGSOM, which has the highest time complexity, is higher than all other methods. It is less for RSOM as compared to FSOM as FSOM converges slower than RSOM, where the initial connection weights are defined from the rough rules instead of random values chosen within -0.5 to 0.5. However, the performance of FRGSOM in terms of indices is better than the RSOM, FSOM and SOM for Cell Cycle data although it has the highest time complexity. Similar observations can also be made from the results shown in Tables 30 and 31 for Yeast Complex and All Yeast data sets, respectively.

The salient differences between the FRGSOM and the Rough SOM are as follows:

1. In Rough SOM, the network is mainly modeled by the integration of the fuzzy sets and rough sets with SOM. In contrast, the proposed FRGSOM is developed by the integration of fuzzy set theory and fuzzy rough sets with SOM.
2. In Rough SOM, rough set theory is used to extract attribute reducts from data and further each reduct is used to generate a set of rules. In contrast, fuzzy rough set theory is used in FRGSOM to generate rules in terms of dependency factors.
3. The number of nodes, in the output layer of Rough SOM, is based on the attribute reducts, whereas the number of nodes in the output layer of the FRGSOM is set according to the class information of the data.
4. The initial connection weights, between nodes of the input and output layers of Rough SOM, are defined with large random numbers for attributes which appeared in the rules, and for the attributes that did not appear in the rules, are defined with small random numbers. In contrast, no such attribute reducts are generated using fuzzy rough sets for

Table 29
β-index, DB-index and FRE values of clustering methods for Cell Cycle data.

Algorithm	Cell Cycle			CPU time in s
	No. of clusters $c = 6$			
	β-index	DB index	FRE	
FRGSOM	0.1988	8.5903	0.1270	9.9282
RSOM	0.1845	10.5134	0.1302	8.4490
FSOM	0.1711	18.6188	0.1472	9.0630
SOM	0.1660	18.0185	0.1983	7.7760

Table 30
β-index, DB-index and FRE values of clustering methods for Yeast Complex data.

Algorithm	Yeast Complex			CPU time in s
	No. of clusters $c = 12$			
	β-index	DB index	FRE	
FRGSOM	0.1308	6.1511	0.1263	18.6330
RSOM	0.1144	6.2012	0.1470	15.8870
FSOM	0.1150	6.9359	0.1482	16.2860
SOM	0.1092	8.4631	0.1501	12.5630

Table 31
β-index, DB-index and FRE values of clustering methods for All Yeast data.

Algorithm	All Yeast			CPU time in s
	No. of clusters $c = 14$			
	β-index	DB index	FRE	
FRGSOM	0.0702	41.6047	0.0420	591.2700
RSOM	0.0678	46.6423	0.0443	300.0153
FSOM	0.0665	43.1818	0.0553	307.3360
SOM	0.0709	47.8397	0.0594	250.3710

defining the initial connection weights in the FRGSOM. Attribute dependency factors, determined using the concept of fuzzy rough sets, are used in FRGSOM as initial connection weights.

6. Conclusion

In this study, we proposed a fuzzy rough granular neural network (FRGSOM) by integrating the concept of fuzzy rough sets with SOM, to predict the underlying clusters in a data set and to handle the uncertainty that comes from patterns of the overlapping regions. Fuzzy rough sets, based on a fuzzy reflexive relation, are used to extract the domain knowledge in the form of dependency factors of conditional attributes. The dependency factors are encoded as the initial weights of the network, and the input vector of the network is defined in terms of fuzzy granules *low, medium* and *high*. The superiority of the FRGSOM, as compared to SOM, FSOM, and RSOM, is extensively demonstrated on Telugu vowel, medical, and microarray gene expression data sets, with dimensions varying from 3 to 93.

A new fuzzy rough entropy measure (FRE) is also proposed using the concept of fuzzy rough sets, based on a fuzzy reflexive relation. The uncertainty arising from overlapping regions is quantified based on this measure. The lower the value of entropy of a cluster, the higher is its compactness. The FRE is seen to reflect well the actual number of clusters for a data set as compared to the β-value and DB-index. Apart from showing the superiority of FRGSOM, the investigation demonstrates a way of integrating two different facets of natural computing, namely, granular computing, and self-organization in a soft computing framework.

Acknowledgments

Support of the Dept. of Science and Technology, Govt. of India to the Center for Soft Computing Research through its IRHPA scheme is acknowledged. The work was done when S.K. Pal was a J.C. Bose Fellow of the Govt. of India. The authors would also like to thank the anonymous reviewers for their suggestions in improving the quality of the research.

References

[1] L. Kari, G. Rozenberg, The many facets of natural computing, Communications of the ACM 51 (10) (2008) 72–83.
[2] L.A. Zadeh, Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems 90 (1997) 111–127.
[3] S.K. Pal, L. Polkowski, A. Skowron (Eds.), Rough-Neural Computing: Techniques for Computing with Words, Vol. 543–553, Springer-Verlag, Germany, 2004.
[4] S.K. Pal, B. Dasgupta, P. Mitra, Rough self-organizing map, Applied Intelligence 21 (2004) 289–299.
[5] S. Mitra, S.K. Pal, Self-organizing neural network as a fuzzy classifier, IEEE Transactions on Systems, Man and Cybernetics 24 (1994) 385–399.
[6] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer, Netherlands, 1991.
[7] Z. Pawlak, A. Skowron, Rough sets: some extensions, Information Sciences 177 (2007) 28–40.
[8] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences 177 (2007) 3–27.
[9] S.K. Pal, A. Skowron (Eds.), Rough-Fuzzy Hybridization: A New Trend in Decision Making, Springer Verlag, Singapore, 1999.
[10] P. Maji, S.K. Pal, Rough-Fuzzy Pattern Recognition: Applications in Bioinformatics and Medical Imaging, John Wiley & Sons, New Jersey, 2011.
[11] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, International Journal of General Systems 17 (1990) 91–209.
[12] J.P. Herbert, J.T. Yao, A granular computing frame work for self-organizing maps, Neurocomputing 72 (2009) 2865–2872.
[13] T. Kohonen, Self-organizing maps, Proceedings of the IEEE 78 (1990) 1464–1480.
[14] P. Lingras, M. Hogo, M. Snorek, Interval set clustering of web users using modified Kohonen self-organizing maps based on the properties of rough sets, Web Intelligence and Agent Systems 46 (1999) 105–119.
[15] L.A. Zadeh, From computing with numbers to computing with words from manipulation of measurements to manipulation of perceptions, IEEE Transactions on Circuits and Systems 46 (1999) 105–119.
[16] S.K. Pal, P.K. Pramanik, Fuzzy measures in determining seed points in clustering, Pattern Recognition Letters 4 (1986) 159–164.
[17] A. Ganivada, S. Dutta, S.K. Pal, Fuzzy rough granular neural networks, fuzzy granules, and classification, Theoretical Computer Science 412 (2011) 5834–5853.
[18] A. Ganivada, S.S. Ray, S.K. Pal, Fuzzy rough granular self-organizing map, in: Proceedings of the Sixth International Conference on Rough Sets and Knowledge Technology, Canada, 2011, pp. 659–668.
[19] G.J. Klir, T. Folger, Fuzzy Sets, Uncertainty and Information, Prentice Hall, New Jersey, 1988.
[20] R. Jensen, Q. Shen, New approaches to fuzzy-rough feature selection, IEEE Transactions on Fuzzy Systems 17 (2009) 824–838.
[21] C. Cornelis, R. Jensen, G. Hurtado, D. Slezak, Attribute selection with fuzzy decision reducts, Information Sciences 180 (2010) 209–224.
[22] D. Sen, S.K. Pal, Generalized rough sets, entropy, and image ambiguity measures, IEEE Transactions on Systems, Man, and Cybernetics, Part B 39 (2009) 117–128.
[23] J.-S. Mi, Y. Leung, H.-Y. Zhao, T. Feng, Generalized fuzzy rough sets determined by a triangular norm, Information Sciences 178 (2008) 3203–3213.
[24] S.K. Pal, D. Dutta Majumder, Fuzzy sets and decision making approaches in vowel and speaker recognition, IEEE Transactions on Systems, Man, and Cybernetics 7 (1977) 625–629.
[25] Y. Hayashi, Neural expert system using fuzzy teaching input and its application to medical diagnosis, Information Sciences Applications 1 (1994) 47–58. http://www.broad.mit.edu/cancer/pub/miGCM.
[26] Z. Bar-Joseph, D.K. Gifford, T.S. Jaakkola, Fast optimal leaf ordering for hierarchical clustering, Bioinformatics 17 (2001) 22–29.
[27] S.S. Ray, S. Bandyopadhyay, S.K. Pal, Dynamic range based distance measure for microarray expressions and a fast gene ordering algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part-B 37 (2007) 742–749.
[28] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, in: Proceedings of National Academy of Sciences, USA, 1998, pp. 4863–14868.
[29] Munich Information for Protein Sequences, [Online]. Available: http://mips.helmholtz-muenchen.de/proj/funcatDB/.
[30] D.L. Davies, D. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis Machine Intelligence PAMI-1 (1979) 224–227.
[31] S.K. Pal, A. Ghosh, B. Uma Shankar, Segmentation of remotely sensed images with fuzzy thresholding, and quantitative evaluation, International Journal of Remote Sensing 21 (2000) 2269–2300.