Chapter 14

Fuzzy Rough Granular Neural Networks for Pattern Analysis

Avatharam Ganivada\textsuperscript{a}, Shubhra Sankar Ray\textsuperscript{a, b} and Sankar K. Pal\textsuperscript{a, b}

\textsuperscript{a}Center for Soft Computing Research
Indian Statistical Institute, India

\textsuperscript{b}Machine Intelligence Unit
Indian Statistical Institute, India
sankar@isical.ac.in

Granular computing is a computational paradigm in which a granule represents a structure of patterns evolved by performing operations on the individual patterns. Two granular neural networks are described for performing the pattern analysis tasks like classification and clustering. The granular neural networks are designed by integrating fuzzy sets and fuzzy rough sets with artificial neural networks in a soft computing paradigm. The fuzzy rough granular neural network (FRGNN) for classification is based on multi-layer neural networks. On the other hand, the fuzzy rough granular self-organizing map (FRGSOM) for clustering is based on self-organizing map. While the FRGNN uses the concepts of fuzzy rough set for defining its initial connection weights in supervised mode, the FRGSOM, as its same implies, exploits the same in unsupervised manner. Further, the input vector of FRGNN & FRGSOM and the target vector of FRGNN are determined using the concepts of fuzzy sets. The performance of FRGNN and FRGSOM is compared with some of the related methods using several life data sets.

14.1. Introduction

Natural Computing is a consortium of different methods and theories that are emerged from natural phenomena such as brain modeling, self-organization, self-repetition, self-evaluation, self-reproduction, group behaviour, Darwinian survival, granulation and perception. Based on the tasks abstracted from these phenomena, various computing
paradigms/technologies like artificial neural networks, fuzzy logic, rough logic, evolutionary algorithms, fractal geometry, DNA computing, quantum computing and granular computing are developed. They take insanitation from the nature for their computation, and are used for solving real life problems. For example, granulation abstracted from natural phenomena possesses perception based information representation which is an inherent characteristic of human thinking and reasoning process. One can refer to [1] and [2] for using fuzzy information granulation as a way to natural computing.

Granular computing (GrC) provides an information processing framework [3–6] where computation and operations are performed on information granules. Information granules are formulated by abstracting the common properties of patterns in data such as similarity, proximity and equality. GrC is based on the realization that precision is sometimes expensive and not very meaningful in modeling and controlling complex systems [7]. When a problem involves incomplete, uncertain, and vague information, one may find it convenient to consider granules, instead of the distinct elements, for its handling. Accordingly, granular computing became an effective framework for the design and implementation of efficient and intelligent information processing systems for various real life decision-making applications. The said framework may be modeled in soft computing using fuzzy sets, rough sets, artificial neural networks and their integrations, among other theories.

Information granulation using fuzzy sets can be found in [8] and [9]. Rough set [10–12] computes rough information granules induced by crisp equivalence classes. The crisp equivalence class represents a clump of objects obtained by grouping them with equal values (indiscernibility relation) corresponding to a feature. These are used to find lower and upper approximations which are exploited in approximating a set. As crisp equivalence classes are essential to rough sets, fuzzy equivalence granules, represented by fuzzy similarity relations, are important to fuzzy rough sets. In fuzzy rough set, the set is characterized by lower and upper approximations [13, 14]. The patterns in the set posses membership values belonging to the approximations computed using fuzzy equivalence granules and fuzzy decision classes. The fuzzy decision classes incorporate the average of memberships of patterns in each class. The memberships for belonging to the decision classes are computed by finding the weighted distances from all patterns in the class to its mean, and normalizing the distances with the help of fuzzyfiers. The use fuzzy information granulation is to efficiently
handle uncertainty arising in the overlapping classes and incompleteness in the class information.

Several researchers have incorporated the information granulation into artificial neural networks to develop different granular neural networks [15, 16]. Methods [17, 18] for classification are based on the integration of fuzzy information granulation and artificial neural networks. A fuzzy rule based neural network for linguistic data fusion is designed in [17]. In [18], an interval valued fuzzy membership function is used to design a granular network for classification where the network is trained using interval valued evolutionary algorithm. Different rule based neural networks for classification using rough sets are developed in [19–21]. The rules are generated using the concepts of rough sets. An interval set representation of clusters of web users of three educational web sites are evolved in [22] using self-organizing map, where the lower and upper approximations are computed at the nodes in the output layer of the self-organizing map. The granular hierarchical self-organizing map [23] uses a bidirectional update propagation method to modify its network parameters. A fuzzy self-organizing map is designed in [24] as a fuzzy classifier using the concept of fuzzy sets. The fuzzy rough rule based neural networks for pattern recognition tasks are designed in [25]- [26]. The input of the networks [25, 27] is defined in terms of 3D input vector using fuzzy set. The network [26] uses the normalized data as its input. The dependency factors of the features (attributes) defined using fuzzy rough sets are encoded into the networks as the initial connection weights.

This article describes two different granular neural networks, developed using fuzzy sets and fuzzy rough sets, for classification and clustering tasks. The network in [25] is developed for classification task and is based on a granular input vector and network parameters. While the fuzzy set defines the input vector of the network in terms of 3D granular vector, the fuzzy rough set determines dependency factors of the features and these are used as the network parameters (initial connection weights). The network consists of input, hidden and output layers and is referred as FRGNN. The number of nodes in the input layer of the network is set equal to 3D features (granular input vector), corresponding to 3 components low, medium and high. The number of nodes in the hidden and output layers are set equal to the number of classes. The connection weights between the input & hidden layers and the hidden & output layers are initialized with the dependency factors of the 3D features. A gradient decent method is used for training the network. During training the weights are modified. The
weights generate a non linear decision boundary by which the overlapping patterns are efficiently classified.

The network is designed in [27] for identifying underlying cluster structures in the data and is based on a granular input vector, initial connection weights and the conventional self-organizing map (SOM). In the process of designing the network, fuzzy sets and fuzzy rough sets are employed. While fuzzy sets are used to develop linguistic input vectors representing fuzzy information granules, fuzzy rough sets are used to extract the crude domain knowledge about data in terms of dependency factors. The dependency factor for every feature using fuzzy rough set is defined in unsupervised manner. The initial connection weights are incorporated into the SOM. The initial knowledge based network having the granular inputs is called fuzzy rough granular self organizing map (FRGSOM) and 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 output layer of the FRGSOM in a topological order. The initial connection weights of FRGSOM enable it to cope with uncertainty arising in overlapping regions between cluster boundaries.

A brief description of fuzzy rough granular neural network for classification is provided in Section 14.2. The formation of fuzzy rough granular self organizing map for clustering is discussed in Section 14.3. This is followed by experimental results section. The last section provides discussions and conclusions of this investigation.

14.2. Architecture of Fuzzy Rough Granular Neural Network for Classification

The fuzzy rough granular neural network (FRGNN) [25] for classification is designed by integrating the concepts of fuzzy sets, fuzzy rough sets and neural networks. While fuzzy sets are used for defining input vector and target vector of the network, the fuzzy rough sets are utilized for determining initial connection weights of the network. While the input vector is represented in terms of granular form, the target vector is expressed in terms of membership value and zeros corresponding to a pattern using fuzzy sets. The network is trained using gradient decent method of multi-layer perception.
14.2.1. **Back Propagation Algorithm for Multi-layered Neural Network**

For updating the initial connection weights of the fuzzy rough granular neural network, the back propagation algorithm of multi-layer perceptron is used. The algorithm is as follows:

**Input:**

- **D:** Data set with training patterns in the granular form and their associated target vectors in terms of membership values and zeros (see Section 14.2.3.1 for details).
- **η:** Learning rate
- **α:** Momentum term
- **b_l = b:** Bias term which is kept constant at each node \((l)\) in the hidden and output layers

**Method:**

1. Assign initial weights, in terms of dependency factors, between the nodes (units) in all the layers of the network;
2. While the network is trained for all training patterns for a particular number of iterations {
   **Forward propagation:**
   3. For each unit \(j\) of the input layer, the output, say \(x_j\), of an input unit, say, \(I_j\) is
   \[
   x_j = I_j;
   \]
   4. For each unit \(l\) of the hidden or output layers, compute the output \(o_l\) of each unit \(l\) with respect to the previous layer, \(j\), as
   \[
   o_l = \sum_j w_{lj} x_j + b;
   \]
   5. Apply logistic activation function to compute the output of \(l\)th node in the hidden or output layers
   \[
   \phi(x_l) = \frac{1}{1 + e^{-x_l}};
   \]
   **Back propagation:**

(6) For each unit in the output layer, say $k$, compute the error using
\[
\text{Error}_k = \phi(x_k)(1 - \phi(x_k))(T_k - \phi(x_k)),
\]
where $T_k$ denotes the target value for $k$th unit in the output layer.

(7) Compute the error for $l$th node in the hidden layer by using the error obtained at $k$th node in the output layer
\[
\gamma_l = \phi(x_l)(1 - \phi(x_l)) \sum_k \text{Error}_k w_{kl};
\]

(8) Update the weight $w_{lj}$ in the network using
\[
\Delta w_{lj} = (\eta)x_j \gamma_l; \\
\Delta w_{lj}(t) = (\eta)x_j \gamma_l + (\alpha)\Delta w_{lj}(t - 1);
\]

(9) Update bias $b$ in the network using
\[
\Delta b = (\eta)\gamma_l; \\
b(t) = \Delta b(t - 1);
\]

/* end while loop for step 2*/

Output:
Classified patterns

Here the momentum term $\alpha$ is used to escape the local minima in the weight space. The value of bias $b$, chosen within 0 to 1, is kept constant at each node of the hidden and output layers of the network and $t$ denotes the number of iterations. For every iteration, the training data is presented to the network for updating the weight parameters $w_{lj}$ and bias $b$. The resulting trained network is used for classifying the test patterns.

14.2.2. Input Vector Representation in terms of 3D Granular Vector

A pattern $F$ in $A \subseteq U$ is assigned a grade of membership value, denoted by $\mu_A(F)$, to a fuzzy set $A$, using a membership function as
\[
A = \{(\mu_A(F), F), F \in U, \mu_A(F) \in [0, 1]\}, \forall F \in R^n.
\]
The $\pi$ membership function is defined as \begin{equation}
\pi(F, C, \lambda) = \begin{cases} 
2(1 - \frac{\|F - C\|_2}{\lambda})^2, & \text{for } \frac{1}{2} \leq \|F - C\|_2 \leq \lambda, \\
1 - 2(\frac{\|F - C\|_2}{\lambda})^2, & \text{for } 0 \leq \|F - C\|_2 \leq \frac{1}{2}, \\
0, & \text{otherwise}
\end{cases}
\end{equation}
where $\lambda > 0$ is a scaling factor (radius) of the $\pi$ function with $C$ as a central point and $\| \cdot \|_2$ denotes the Euclidian norm. An $n$-dimensional $i$th pattern $F_i$ is represented by 3$n$-dimensional linguistic vector \[16\] as
\[
\overrightarrow{F}_i = [\mu_{\text{low}}(F_i_1)(\overrightarrow{F}_i), \mu_{\text{medium}}(F_i_1)(\overrightarrow{F}_i), \mu_{\text{high}}(F_i_1)(\overrightarrow{F}_i), \ldots, \mu_{\text{high}}(F_{in})(\overrightarrow{F}_i)].
\]
where $\mu_{\text{low}}, \mu_{\text{medium}}$ and $\mu_{\text{high}}$ indicate the membership values corresponding to the linguistic terms $\text{low}, \text{medium}$ or $\text{high}$ along each feature axis. The selection for the values of $C$ and $\lambda$ of $\pi$ functions is explained in the following section.

14.2.2.1. Choice of Parameters of $\pi$ Functions
Let $\{F_{ij}\}$ for $i = 1, 2, \ldots, s$; $j = 1, 2, \ldots, n$; denote a set of $s$ patterns for data with $n$ features, and $F_{j_{\text{min}}}$ and $F_{j_{\text{max}}}$ denote the minimum and maximum values along the $j$th feature considering all the $s$ patterns. The average of all the $s$ patterns along the $j$th feature, $F_j$, is initially computed in a way similar to \[25\] and is considered as the center of the linguistic term $\text{medium}$ along that feature, denoted by $r_{mj}$, as in Fig. 14.1. Then, the average values (along the $j$th feature $F_j$) of the patterns in the ranges $[F_{j_{\text{min}}}, r_{mj})$ and $(r_{mj}, F_{j_{\text{max}}}]$, corresponding to the feature $F_j$, are defined as the means of the linguistic terms $\text{low}$ and $\text{high}$, and denoted.

![Fig. 14.1. Parameters of linguistic terms, low, medium and high.](image-url)
by \( r_{l_j} \) and \( r_{h_j} \), respectively. Again, using the same ranges we find the minimum and maximum for each of the ranges. While for the first range (low) the minimum and maximum are \( F_{jmin} \) and \( r_{m_j} \), respectively, for the second range (high) these are \( r_{m_j} \) and \( F_{jmax} \). Then the center \( C \) and the corresponding scaling factor \( \lambda \) for linguistic terms low, medium and high along the \( j \)th feature \( F_j \) are defined as follows [25]:

\[
C_{medium_j} = r_{m_j},
\]
\[
p_1 = C_{medium_j} - \frac{F_{jmax} - F_{jmin}}{2},
\]
\[
q_1 = C_{medium_j} + \frac{F_{jmax} - F_{jmin}}{2},
\]
\[
\lambda_{m_j} = q_1 - p_1,
\]
\[
\lambda_{medium} = \sum_{j=1}^{n} \lambda_{m_j}.
\]

\((14.4)\)

\[
C_{low_j} = r_{l_j},
\]
\[
p_2 = C_{low_j} - \frac{r_{m_j} - F_{jmin}}{2},
\]
\[
q_2 = C_{low_j} + \frac{r_{m_j} - F_{jmin}}{2},
\]
\[
\lambda_{l_j} = q_2 - p_2,
\]
\[
\lambda_{low} = \sum_{j=1}^{n} \lambda_{l_j}.
\]

\((14.5)\)

\[
C_{high_j} = r_{h_j},
\]
\[
p_3 = C_{high_j} - \frac{F_{jmax} - r_{m_j}}{2},
\]
\[
q_3 = C_{high_j} + \frac{F_{jmax} - r_{m_j}}{2},
\]
\[
\lambda_{h_j} = q_3 - p_3,
\]
\[
\lambda_{high} = \sum_{j=1}^{n} \lambda_{h_j}.
\]

\((14.6)\)

Figure 14.1 depicts the parameters, center and corresponding scaling factor, for linguistic terms low, medium or high along the feature \( F_j \). Figure 14.2 shows a pattern \( \vec{F}_i \) having membership values to the linguistic properties low, medium or high. For one of these properties, the pattern would have strong membership value.

14.2.3. Class Memberships as Output Vectors

The membership of the \( i \)th pattern to \( k \)th class is calculated in [28] using

\[
\mu_k(\vec{F}_i) = \frac{1}{1 + (Z_{ik})^{f_e}}.
\]

where \( Z_{ik} \) is the weighted distance, and \( f_d \) & \( f_e \) are the denominational & exponential fuzzy generators, respectively, controlling membership of the
patterns in the class. The weighted distance $Z_{ik}$ is computed \[16\] using

$$Z_{ik} = \sum_{j=1}^{n} \sum_{g=1}^{3} \frac{1}{3} (\mu_g(F_{ij}) - \mu_g(O_{kj}))^2,$$  

for $k = 1, 2, \ldots, c$, \hspace{1cm} (14.8)

where $\mu_g(F_{ij})$ represents the membership values of $j$th feature, corresponding to the linguistic terms low, medium and high, of $i$th pattern, $O_{kj}$ is the mean of the $k$th class, and $c$ is the number of classes.

14.2.3.1. Applying Membership Value to Target Vector

The target vector corresponding to $i$th pattern in a class contains one non zero membership value for one output node, representing that class, and zeros for the remaining output nodes, representing the remaining classes. That is, for the $i$th training pattern from the $k$th class, the target vector for $k$th output node is defined as \[25\]

$$d_k = \begin{cases} 
\mu_{INT}(F_i), & \text{if } i\text{th pattern is from } k\text{th class denoting } k\text{th output node,} \\
0, & \text{otherwise.}
\end{cases}$$

\hspace{1cm} (14.9)

14.2.4. Determining Initial Connection Weights of Network using Fuzzy Rough Sets

In fuzzy rough set theory, a similarity between two objects in $U$ is modeled by a fuzzy tolerance relation $R$, that is,

$$R(x, x) = 1 \quad \text{(reflexive),}$$

$$R(x, y) = R(y, x) \quad \text{(symmetry), and}$$

$$T(R(x, y)R(y, z)) \leq R(x, z) \quad \text{($T$-transitivity),}$$
for all $x$, $y$ and $z$ in $U$. The fuzzy similarity relations are commonly considered to measure the approximate equality of objects. Given a $t$-norm, the relation $R$ is called a fuzzy reflexive relation when the relation $R$ does satisfy the properties of symmetry and $T$-transitivity.

14.2.4.1. Defining Fuzzy Reflexive Relations Corresponding to Features

Let $S = (U, A \cup \{D\})$ denote a decision system. The 3D features, say $\{f_1, f_2, \ldots, f_n\}$, are represented by $A$. If there are $c$ classes then the belongingness of a pattern to each of the classes can be represented by $\{D\} = \{X_k, k = 1, 2, \ldots, c\}$ where $X_k$ is a decision feature containing membership values for all the patterns in class $k$. For every feature $f$, the fuzzy reflexive relation $R$ between any two objects $x$ and $y$ is defined as [27]

$$
R_f(x, y) = \begin{cases} 
\max \left\{ \min \left( \frac{f(y) - f(x) + \sigma_{f_{k_1}}}{\sigma_{f_{k_1}}}, \frac{f(x) - f(y) + \sigma_{f_{k_2}}}{\sigma_{f_{k_2}}} \right), 0 \right\}, \\
\text{if } f(x) \text{ and } f(y) \in R_D(X_{k_1}) \\
\text{(i.e., both patterns } x \text{ and } y \text{ belong to one particular decision class)}, \\
\max \left\{ \min \left( \frac{f(y) - f(x) + \sigma_{f_{k_2}}}{\sigma_{f_{k_2}}}, \frac{f(x) - f(y) + \sigma_{f_{k_1}}}{\sigma_{f_{k_1}}} \right), 0 \right\}, \\
\text{if } f(x) \in R_D(X_{k_1}), f(y) \in R_D(X_{k_2}) \\
\text{(i.e., both patterns } x \text{ and } y \text{ belong to two different decision classes)} \\
\text{and } k_1 \neq k_2,
\end{cases}
$$

(14.10)

where $\sigma_{f_{k_1}}$ and $\sigma_{f_{k_2}}$ represent the standard deviation of the patterns in the classes $k_1$ and $k_2$, corresponding to feature $f$ with respect to the decision classes $R_D(X_{k_1})$ and $R_D(X_{k_2})$, respectively, $k_1$ and $k_2 = 1, 2, \ldots, c$.

14.2.4.2. Defining Fuzzy Decision Classes and Rough Representation

Suppose the $i$th pattern $\overrightarrow{F}_i$ (see Section 14.2.2) is represented with $\overrightarrow{x}_i$, and $O_{kj}$ and $V_{kj}$, $j = 1, 2, \ldots, 3n$, denote the mean and the standard deviation, respectively, of the patterns belonging to $k$th class. The weighted distance $Z_{ik}$ of a pattern $\overrightarrow{x}_i$, $i = 1, 2, \ldots, s$, from the mean of the $k$th class is
defined as [28]
\[ Z_{ik} = \sqrt{\sum_{j=1}^{3n} \left| \frac{x_{ij} - O_{kj}}{V_{kj}} \right|^2}, \]  
for \( k = 1, 2, \ldots, c \), \hspace{1cm} (14.11)
where \( x_{ij} \) is the \( j \)th feature (3D feature) of \( i \)th pattern. Here \( s \) is the total number of patterns. The membership value of the \( i \)th pattern in the \( k \)th class, \( \mu_k(\vec{x}_i) \), is computed using Eq. 14.7. The decision features representing fuzzy decision classes are defined as follows [25]:

i) Compute the average of the membership values of all the patterns in the \( k \)th class to its own class and replace the individual membership with that average. The average membership of patterns in the \( k \)th decision class is defined as
\[ D_{kk} = \frac{\sum_{m_k=1}^{m_k} \mu_k(\vec{x}_i)}{|m_k|}, \]  
if \( k = l \), \hspace{1cm} (14.12)
where \( |m_k| \) indicates the number of patterns in the \( k \)th decision class, and \( k \) and \( l = 1, 2, \ldots, c \). ii) Calculate the average of the membership values of all the patterns in the \( k \)th class to the other classes and replace the membership values with the corresponding average values. The average membership values for the patterns from their own class to other classes are computed as
\[ D_{kl} = \frac{\sum_{m_k=1}^{m_k} \mu_l(\vec{x}_i)}{|m_k|}, \]  
if \( k \neq l \), \hspace{1cm} (14.13)

The fuzzy decision classes are defined as [25]
\[ R_f(x, y) = \begin{cases} 
D_{kk}, & \text{if } f(x) = f(y), \\
D_{kl}, & \text{otherwise},
\end{cases} \]  
for all \( x, y \) in \( U \). Here \( D_{kk} \) corresponds to an average membership value of all the patterns that belong to the same class (\( k = l \)), and \( D_{kl} \) corresponds to the average membership values of all the patterns from classes other than \( k \) (\( k \neq l \)). The lower and upper approximations of a set \( A \subseteq U \) are defined as [29]
\[ (R_f \downarrow R_D)(y) = \inf_{x \in U} I(R_f(x, y), R_D(x)) \]  
and \[ (R_f \uparrow R_D)(y) = \sup_{x \in U} T(R_f(x, y), R_D(x)), \]  
for all \( y \) in \( U \). For \( B \subseteq A \), the fuzzy positive region based on fuzzy \( B \)-indiscernibility relation is defined as
\[ POS_B(y) = (R_B \downarrow R_D)(y), \]  
for all \( y \in U \). The dependency value for every feature in the set \( B \subseteq A \), denoted by \( \gamma_B \), is defined as
\[ \gamma_B = \frac{\sum_{x \in U} POS_B(x)}{|U|}, \]  
where \( |\cdot| \) denotes the cardinality of \( U \), and \( 0 \leq \gamma_B \leq 1 \).
14.2.4.3. Knowledge Encoding Procedure

We compute \(c\) decision tables \(S_l = (U_l, A \cup \{D\})\), \(l = 1, 2, \ldots, c\) using the training data. Each of the decision tables consists of patterns from each of the \(c\) classes where the patterns are added sequentially in the tables. Each pattern in \(S_l\) has \(3n\)-dimensional features. The decision table \(S_l\) satisfies the following three conditions:

\[
\begin{align*}
\text{i)} & \quad U_l \neq \emptyset, \\
\text{ii)} & \quad \bigcup_{l=1}^{c} U_l = U, \quad \text{and} \\
\text{iii)} & \quad \bigcap_{l=1}^{c} U_l = \emptyset. 
\end{align*}
\]

(14.19)

The number of patterns in \(U_l\) belongs to the \(c\) classes, say \(U_{d(e_k)}, k = 1, 2, \ldots, c\). The size of the \(S_l\), \(l = 1, 2, \ldots, c\) is dependent on the available number of objects from all the classes. Based on the \(c\) decision tables, the initial connection weights of the network are determined using fuzzy rough sets. We explain the procedure for determining the initial connection weights, based on the decision table \(S_1\) when \(l = 1\), as an example. The procedure is as follows.

**S1:** Find fuzzy reflexive relation matrix using Eq. 14.10 for every feature. Here each row in the matrix represents fuzzy equivalence granule.

**S2:** Compute fuzzy decision classes using Eq. 14.14.

**S3:** Calculate the lower approximations of patterns in a set (class) using Eq. 14.15 for each feature, based on S1 and S2.

**S4:** Calculate the dependency value using Eq. 14.18 for each feature. Initialize the resulting values as initial connection weights between the input nodes and one hidden node as follows:

Let \(\gamma_{i}, i = 1, 2, \ldots, n\); denote the dependency degree of \(i\)th feature \((a_i)\) in the decision table \(S_1\). The weight between one hidden node, representing one decision table \((S_1)\), and \(i\)th input node is denoted by \(w_{1i}\). Therefore, the connection weight between \(i\)th input node and the hidden node \((w_{1i})\), represented with \(\gamma_{i}^{S_1}\), is defined as

\[
\gamma_{i}^{S_1} = \frac{\sum_{x \in U} POS(x)}{|U|}. 
\]

(14.20)

**S5:** The connection weights between the hidden node and the nodes in the output layer are initialized as follows: Let \(\beta_{k}^{S_1}\) denote the average of dependency factors of all the features with respect to the decision class \(k\). The connection weight between the hidden node and \(k\)th output node is denoted by \(w_{k1}, k = 1, 2, \ldots, c\). Therefore, the weight \(w_{k1}\) is
initialized with $\beta^k_{S_l}$ as
\[
\beta^k_{S_l} = \frac{\sum_{i=1}^{n} \gamma^1_{i}}{|n|}, \quad (14.21)
\]
where $\gamma^1_{i}$ is defined as
\[
\gamma^1_{i} = \sum_{x \in U_{de_k}} \frac{POS_{i}(x)}{|U_{de_k}|}, \quad k = 1, 2, \ldots, c. \quad (14.22)
\]

Similarly, we define the dependency degrees, $\gamma^S_{l}$ and $\beta^k_{S_l}$ using the decision tables $S_l$, $l = 2, \ldots, c$. The weights between $i$th input node & $l$th hidden node of the network and $l$th hidden node & $k$th output node are initialized with $\gamma^S_{l}$ and $\beta^k_{S_l}$, respectively. The connection weights of the network are refined using the gradient decent method (back propagation algorithm of MLP).

**14.3. Fuzzy Rough Granular Self-Organizing Map**

Here, the methodology of fuzzy rough granular self-organizing map (FRG-SOM) for clustering is described. The input vector of FRGSOM, in terms of 3D granular input vector corresponding to linguistic terms *low* medium and *high*, is defined using Eq. 14.3. The granular input data is used to generate granulation structures on basis of $\alpha$-cut. The method involving $\alpha$-cut for generating the granulation structures is developed as follows [27].

**Method:** Let $x_{1j}$ and $x_{2j}$, $j = 1, 2, \ldots, 3n$; denote two patterns with $3n$ dimensional features. The similarity value between the patterns, $x_{1j}$ and $x_{2j}$, using fuzzy implication operator and $t$-norm, is calculated as

\[
S_1) \ I_{1j} \leftarrow \min(1 - x_{1j} + x_{2j}, 1), \ I_{2j} \leftarrow \min(1 - x_{2j} + x_{1j}, 1). \ /\ast \ \text{use fuzzy implication} \ /\ast
\]

\[
S_2) \ T_{j} \leftarrow \max(I_{1j} + I_{2j} - 1, 0). \ /\ast \ \text{use} \ t\text{-norm} \ /\ast
\]

\[
S_3) \ m_{12} \leftarrow \min\{T_{j}\}.
\]

The similarity values for all possible pairs of patterns are computed and a similarity matrix of size $s \times s$ is constructed, where $s$ is the total number of patterns in the data. An $\alpha$-value, chosen within 0 to 1, is applied on the matrix and $p$ number of granules are generated. The process of determining the granules by using $\alpha$-value on the matrix is described in [27]. We arrange the $p$ groups according to their size in decreasing order, where the size is the number of patterns in the group. The top $c$ groups, out of the $p$ groups, are selected. The patterns in $c + 1$ to $p$ groups are added into the top $c$-groups.
according to their closeness to the cluster centers of the top \(c\) groups. The resultant \(c\) groups are employed in defining the initial connection weights of FRGSOM.

### 14.3.1. Determining Initial Connection Weights of FRGSOM

The \(c\)-groups labeled with the numbers as its crisp decision classes are presented to a decision table \(S = (U, \mathcal{A} \cup \{D\})\). Here, \(\mathcal{A} = \{f_1, f_2, \ldots, f_{3n}\}\) characterizes \(3n\) dimensional features and \(\{D\}\) represents decision classes. Based on the decision table \(S\), the initial connection weights of FRGSOM using fuzzy rough sets are determined as follows [27].

1. **S1)** Calculate fuzzy reflexive relational matrix using Eq. 14.10 for every feature \(f_j, j = 1, 2, \ldots, 3n\).

2. **S2)** Compute fuzzy decision classes corresponding to crisp decision classes in two steps.
   - a) The membership values of all the patterns in the \(k\)th class to its own class is defined as
     \[
     D_{kk} = \mu_k(\vec{x}_i), \quad \text{if} \quad k = l,
     \]  
     (14.23)
     where \(\mu_k(\vec{x}_i)\) represents the membership value of the \(i\)th pattern to the \(k\)th class calculated using Eq. 14.7, and
   - b) the membership values of all patterns in the \(k\)th class to other classes is defined as
     \[
     D_{kl} = 1, \quad \text{if} \quad k \neq l,
     \]  
     (14.24)
     where \(k\) and \(l = 1, 2, \ldots, c\). For any two patterns \(x\) and \(y \in U\), with respect to a feature \(f \in \{D\}\), the fuzzy decision classes are specified as
     \[
     R_{D}(x, y) = \begin{cases} 
     D_{kk}, & \text{if } f(x) = f(y), \\
     D_{kl}, & \text{otherwise.}
     \end{cases}
     \]  
     (14.25)

3. **S3)** Compute lower approximations of each set (granule) using Eq. 14.15 for every feature \(f_j, j = 1, 2, \ldots, 3n\).

4. **S4)** Calculate the dependency value using Eq. 14.18 for every feature \(f_j, j = 1, 2, \ldots, 3n\); with respect to each decision class. Assign the resulting dependency factors as initial weights, say \(w_{kj}, j = 1, 2, \ldots, 3n; k = 1, 2, \ldots, c\) between \(3n\) nodes in the input layer and \(c\) nodes in output layer of FRGSOM. Here \(c\) represents the number of clusters.
14.3.2. Training of FRGSOM

The FRGSOM is trained using the competitive learning of the conventional SOM [30]. The procedure for training FRGSOM is as follows.

S1) Present an input vector, \( x_j(t) \), \( j = 1, 2, \ldots, 3n \); at the nodes in the input layer of FRGSOM for a particular number of iteration \( t \).

S2) At \( t \)th iteration, compute the Euclidian distance between the input vector, \( x_j(t) \), and weight vector \( w_{kj}(t) \) at the \( k \)th output node using

\[
L_k = \| x_j(t) - w_{kj}(t) \|^2. \tag{14.26}
\]

S3) Find the winner neuron \( p \) using

\[
p = \underset{k = 1, 2, \ldots, c}{\text{argmin}} \{ L_k \}. \tag{14.27}
\]

S4) Find neighborhood neurons, \( N_p \), around the winner neuron \( p \) using the Gaussian neighborhood function as

\[
N_p(t) = \exp\left( -\frac{\sum_{k}^p}{2\sigma(t)^2} \right), \tag{14.28}
\]

where \( \sum_{p}^k(t) \) is the distance between the position of winning neuron \( p \) and the position of a neuron \( k \), lying in the neighborhood of \( N_p \), in two dimensional array. The width of the neighborhood set \( \sigma \) at iteration \( t \) is defined as

\[
\sigma(t) = \sigma_0 \exp\left( -\frac{t}{\tau_1} \right). \tag{14.29}
\]

Here the value of \( \sigma \) is decreased with every value of iteration \( t \) and \( \tau_1 \) is a constant chosen as in [30].

S5) Update the connection weights of the neighborhood neurons \( N_p \) using

\[
w_{kj}(t+1) = \begin{cases} w_{kj}(t) + \alpha(t) N_p(t)(x_j(t) - w_{kj}(t)), & \text{if } k \in N_p(t), \\ w_{kj}(t), & \text{else}. \end{cases} \tag{14.30}
\]

The learning parameter \( \alpha \) in Eq. 14.30 is defined as [30]

\[
\alpha(t) = \alpha_0 \exp\left( -\frac{t}{\tau_2} \right). \tag{14.31}
\]

where \( \alpha_0 \) is chosen between 0 and 1 and a time constant \( \tau_2 \) is set equal to the total number of iterations.

The training of FRGSOM is performed for a fixed number of iterations and the connection weights are updated. The updated weights are more likely to become similar to the input patterns, which are presented to the network during training. After completion of the competitive learning, FRGSOM partitions the 3D input data into clusters (granulation structures) which are then arranged in a topological order [30].
14.4. Experimental Results

The methods of FRGNN and FRGSOM for classification and clustering, respectively, are implemented in the C language and 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. The details of data sets are discussed as follows.

Artificial data: This artificial data [31] contains 250 patterns, two features and five classes. There are fifty patterns in each of the classes. Fig. 14.3 shows the two dimensional plot of the overlapping patterns in the five groups.

Haberman’s Survival Data: The data set is based on a study conducted between 1958 and 1970 at the University of Chicago’s Billings Hospital on the survival of patients who had undergone surgery for breast cancer. There are 306 patients (patterns) with 3 features for two categories, the patients who survived upto 5 years and the patients who died within 5 years after surgery. The age of the patient at the time of operation, patient’s year of operation and the number of positive axillary lymph nodes detected are the values of features. Fig. 14.4 depicts the two dimensional plot of the overlapping patterns in the two categories. The data is downloaded from https://archive.ics.uci.edu/ml/datasets/Haberman’s+Survival.

Synthetic Pattern1 data: This data [32] consists of 880 patterns with two features. Fig. 14.5 represents the scatter plot of the data for three non convex classes which are linearly non separable in 2D plane.
Classification Results

The fuzzy rough granular neural network (FRGNN) is used for classifying the real life data sets. During training the network, 5 folds cross validation designed with randomized sampling is used. Out of 5 folds, 4 folds of data, selected randomly from each of the classes, are considered as training set and the remaining one fold is used as test set. The classification process is repeated for 5 times considering 4 folds of training set and 1 fold of test set and the average of the classification accuracies is reported.

Before providing the classification results of test set, the configuration of
FRGNN based on granular input features, knowledge extraction about the data and fuzzy target vector is explained as follows. For defining granular input vector, the data is transformed into 3D granular feature space using Eq. 14.3. The values of center and scaling factor of \( \pi \) membership function along each of the 3D granular features, low, medium and high, are chosen using Eq. 14.6. The procedure for choosing these values is described in Section 14.2.2.1. The fuzzy output vector, in terms of membership value and zeros, for every input pattern is defined using Eq. 14.9 (see Section 14.2.3.1).

**Knowledge extraction procedure:** Here, we explain the knowledge extracting procedure of FRGNN for synthetic pattern1 data as an example. A 3D granular training set is presented to a decision table \( S = (U, \mathcal{A} \cup \{d\}) \). It is then divided into 3 decision tables \( S_l = (U_l, \mathcal{A} \cup \{d\}), l = 1, 2 \) and 3 as there are 3 number of classes in the data. Based on the decision table \( S_l \), we use the procedure in Steps 1-5, (see Section 14.2.4.3) for extracting the knowledge about the training set. The resultant knowledge is encoded into the network as its initial connection weights. For example, the initial weights between input nodes \& hidden nodes and hidden nodes \& output nodes of FRGNN for the synthetic pattern1 data are shown in Table 14.1.

<table>
<thead>
<tr>
<th>Input to Hidden Layer( (w_{jl}) )</th>
<th>Hidden to Output Layer( (w_{lk}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.295 0.294 0.279</td>
<td>0.291 0.294 0.290</td>
</tr>
<tr>
<td>0.288 0.293 0.278</td>
<td>0.072 0.083 0.104</td>
</tr>
<tr>
<td>0.288 0.294 0.279</td>
<td>0.283 0.286 0.284</td>
</tr>
<tr>
<td>0.288 0.293 0.279</td>
<td></td>
</tr>
<tr>
<td>0.288 0.288 0.305</td>
<td></td>
</tr>
<tr>
<td>0.294 0.287 0.278</td>
<td></td>
</tr>
</tbody>
</table>

**Performance of FRGNN:** The average classification results of 5 folds for the three data sets are compared with the related fuzzy multilayer perceptron (FMLP), K-nearest neighbor (KNN) and Naive Bayes method. The FRGNN becomes fuzzy multilayer perceptron (FMLP) when the initial weights are chosen randomly within 0 to 1. The number of nodes in the input layer for both FRGNN and FMLP is chosen equal to the three dimensional granular input vector. The number of nodes in the output layer is equal to the number of classes in the data. The number of hidden nodes
for FRGNN is chosen equal to the number of decision tables (see Section 14.2.4.3). For FMLP, the number of hidden nodes is varied from 1 to 10 in steps of 1 and the number for which the FMLP provides the highest accuracy is selected. The connection weights of FRGNN are initialized by the dependency factors of features and, the random numbers within 0 to 1 are encoded into FMLP as its initial weights. For a particular number of iterations \( t \), we perform the training process for different values of learning rate \( \eta \) & momentum term \( \alpha \) within 0 and 1 by varying them in steps of 0.1. The value of \( t \) is chosen by varying it within 100 to 1500 in steps of 100. Hence, the appropriate values of the parameters \( \eta, \alpha \) and \( t \) are obtained by trial and error process. The value of ‘K’ in KNN is chosen as the square root of the number of patterns in the training set. The values of parameters for Naive Bayes method are not specified as the method automatically chose the parameter values.

The results of FRGNN for all the data sets, as compared to FMLP, KNN and Naive Bayes method, are shown in Table 14.2. For all the data sets, the value of \( t \) is set equal to 1500 for FRGNN and FMLP. As an example, for one fold of training set and test set, the values of parameters \( \alpha \) and \( \eta \) for FRGNN and FMLP are shown in the last column of the table. The number of nodes in the hidden layer of FMLP and the value of K in KNN are provided in parenthesis of the table.

It is clear from Table 14.2 that the performance of FRGNN, in terms of average classification accuracy (99.60%), is superior to FMLP (99.20%), KNN (97.60%) and Naive Bayes (97.20%). Similarly observation can be also made for Haberman’s survival data. The results indicate that the FRGNN generalizes the patterns arising in the overlapping regions (see Figs. 14.3 and 14.4) better than the remaining algorithms.

Although the synthetic pattern1 data has concave and linearly non separable classes (see Fig. 14.5), FRGNN has achieved the best average classification accuracy. It can be stated from the results that the FRGNN is an efficient method for handling the data with linearly non separable and concave classes as well as overlapping pattern classes.

We also compared the squared errors of FRGNN and FMLP for all the five folds of test set for all the data sets. Figure 14.6 shows the variation of the error values with the increasing number of iterations for the FRGNN and FMLP for Haberman’s survival data.

The errors correspond to that fold of test set for which the best accuracy is achieved. It is clear from the figure that the error decreases with increasing number of iterations. However the error curve for FRGNN is
<table>
<thead>
<tr>
<th>Data</th>
<th>Algorithm</th>
<th>No. of folds</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>fold1</td>
<td>fold2</td>
</tr>
<tr>
<td>Artificial</td>
<td>FRGNN</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>FMLP(6)</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>100.00</td>
<td>96.00</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>100.00</td>
<td>98.00</td>
</tr>
<tr>
<td>Haberman's Survival</td>
<td>FRGNN</td>
<td>67.21</td>
<td>78.69</td>
</tr>
<tr>
<td></td>
<td>FMLP(2)</td>
<td>68.85</td>
<td>67.21</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>68.85</td>
<td>72.13</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>73.77</td>
<td>65.57</td>
</tr>
<tr>
<td>Synthetic Pattern1</td>
<td>FRGNN</td>
<td>83.91</td>
<td>88.51</td>
</tr>
<tr>
<td></td>
<td>FMLP(3)</td>
<td>85.63</td>
<td>90.23</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>85.63</td>
<td>85.63</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>74.21</td>
<td>74.13</td>
</tr>
</tbody>
</table>
Fig. 14.6. Comparison of squared errors of FRGNN and FMLP for Haberman’s survival data.

much lower than that of FMLP, when the number of iterations is within 300 to 1500. The initial connection weights of FRGNN enables it to carry out the learning process better than that those of FMLP by reducing the search space for solution. Hence, the FRGNN achieves the best accuracy.

14.4.2. Clustering Results

The fuzzy rough granular self-organizing map (FRG-SOM) is used to perform clustering on three real life data sets, artificial, Haberman’s survival and synthetic pattern1. The FRGGSOM consists of the input layer and output layer. The input layer contains the number of nodes equal to the number of features in the data. In the output layer, the number of nodes is set equal to \( c \), representing the user defined number of clusters. The initial connection weights of FRG-SOM is defined using dependency values of features. The FRG-SOM is trained using the competitive learning of self-organizing map (see Section 14.3.2).

The results of FRG-SOM is compared with self-organizing map [30], rough possibilistic \( c \)-means (RPCM) [33] and \( c \)-medoids, in terms of cluster evaluation measures like \( \beta \)-index, DB-index and Dunn-index and fuzzy rough entropy (FRE). The results are provided in Table 14.3. For FRG-SOM and SOM, the values of iteration \( t \) and learning parameter \( \alpha_0 \) are provided in the last column of the table. While the value of threshold \( Tr \)
& possibilistic constant $w$ are provided for RPCM, for \textit{c}-medoids the value of $c$ is shown in the last column of the table. In RPCM, $w$ effects the possibilistic membership of the patterns in a cluster. The value of $c$ is chosen as the same for SOM and FRGSOM.

Table 14.3. Clustering results of all the algorithms, in terms of $\beta$-index, DB-index, Dunn-index and FRE, for all the data sets.

<table>
<thead>
<tr>
<th>Data</th>
<th>Algorithm</th>
<th>$\beta$-index</th>
<th>DB-index</th>
<th>Dunn-index</th>
<th>FRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial</td>
<td>FRGSOM</td>
<td>3.964</td>
<td>1.177</td>
<td>1.183</td>
<td>0.124</td>
</tr>
<tr>
<td></td>
<td>SOM</td>
<td>1.551</td>
<td>4.282</td>
<td>0.336</td>
<td>0.147</td>
</tr>
<tr>
<td></td>
<td>RPCM</td>
<td>5.249</td>
<td>1.225</td>
<td>1.000</td>
<td>0.136</td>
</tr>
<tr>
<td></td>
<td>\textit{c}-medoids</td>
<td>2.665</td>
<td>1.828</td>
<td>0.563</td>
<td>0.161</td>
</tr>
<tr>
<td>Haberman’s</td>
<td>FRGSOM</td>
<td>1.695</td>
<td>1.175</td>
<td>1.658</td>
<td>0.238</td>
</tr>
<tr>
<td>Survival</td>
<td>SOM</td>
<td>1.575</td>
<td>1.065</td>
<td>1.520</td>
<td>0.249</td>
</tr>
<tr>
<td></td>
<td>RPCM</td>
<td>1.693</td>
<td>1.129</td>
<td>1.464</td>
<td>0.280</td>
</tr>
<tr>
<td></td>
<td>\textit{c}-medoids</td>
<td>1.594</td>
<td>1.123</td>
<td>1.435</td>
<td>0.296</td>
</tr>
<tr>
<td>Synthetic</td>
<td>FRGSOM</td>
<td>2.124</td>
<td>2.029</td>
<td>0.639</td>
<td>0.180</td>
</tr>
<tr>
<td>Pattern1</td>
<td>SOM</td>
<td>3.444</td>
<td>1.838</td>
<td>0.811</td>
<td>0.208</td>
</tr>
<tr>
<td></td>
<td>RPCM</td>
<td>2.727</td>
<td>1.025</td>
<td>0.768</td>
<td>0.221</td>
</tr>
<tr>
<td></td>
<td>\textit{c}-medoids</td>
<td>2.740</td>
<td>0.902</td>
<td>0.940</td>
<td>0.177</td>
</tr>
<tr>
<td></td>
<td>$\alpha_0 = 0.09, \ t = 500$</td>
<td>$\alpha_0 = 0.85, \ t = 1600$</td>
<td>$\alpha_0 = 0.00001, \ t = 1500$</td>
<td>$\alpha_0 = 0.01, \ t = 2000$</td>
<td>$\alpha_0 = 0.000092, \ t = 70$</td>
</tr>
<tr>
<td></td>
<td>$Tr = 0.05, \ w = 0.35$</td>
<td>$c = 5$</td>
<td>$Tr = 0.05, \ w = 0.05$</td>
<td>$c = 2$</td>
<td>$Tr = 0.085, \ w = 0.07$</td>
</tr>
</tbody>
</table>

It is evident from Table 14.3 that the results of artificial data for FRGSOM is better than SOM, RPCM & \textit{c}-medoids in terms of DB-index, Dunn-index & FRE except $\beta$-index, where the FRGSOM is the second best and the RPCM is the best. Note that lower values of FRE & DB-index and higher values of $\beta$-index & Dunn-index indicate that the output clusters are better.

For Haberman’s survival data, the FRGSOM achieves the best results in terms of $\beta$-index, Dunn-index & FRE except DB-index where the FRGSOM is the third best.

For synthetic pattern1 data, it can be seen from Table 14.3 that the performance of FRGSOM is inferior to the remaining algorithms using the most of the clustering measures. The reason for inferior performance of FRGSOM in this case is that synthetic pattern1 data does not have overlapping class boundaries that are in non convex shape. Further, the performance of FRGSOM, which is useful for overlapping patterns, is inferior to SOM for all the indices as the former uses 3D granular input, instead of actual input like latter one. It can be concluded from these results that the FRGSOM efficiently partitions the data whenever there are overlapping patterns.
14.5. Conclusion

The present investigation describes an integration of fuzzy sets and fuzzy rough sets with neural networks in granular computing framework and also demonstrates the effectiveness of the hybrid methods on different types of data. Two methods, fuzzy rough granular neural network (FRGNN) and granular self-organizing map (FRGSOM) are described for performing classification and clustering, respectively. The FRGNN is formulated by incorporating 3D granular input vector, fuzzy initial connection weights and target vector (defined in terms membership values) into multi-layer perceptron. The learning process of FRGNN is performed using the gradient decent method. The FRGNN is compared with fuzzy MLP (FMLP), Naive Bayes and KNN where the first two methods are known for handling overlapping classes and the last method is efficient for handling non convex & non linear classes by generating piecewise linear boundaries. The classification accuracies of FRGNN are found to be better than the remaining algorithms. As the FRGNN incorporates the fuzzy information granulation at the input level & initial connection weights level and the membership values at the output level, it achieves the best classification results. Hence, it can be stated that the FRGNN is an effective algorithm for classifying any data having linearly non separable classes, non convex class shape and overlapping regions between class boundaries.

The performance of FRGSOM is compared with rough possibilistic \( c \)-means (RPCM), self-organizing map (SOM) and \( c \)-medoids. RPCM is a recent method for clustering data with overlapping patterns, SOM is a classical clustering technique that maps \( n \) dimensional input space into two dimensional array such that the output clusters are arranged in a topological order (organization of the output clusters corresponds to the neighborhood relations between them) and \( c \)-medoids is also a classical method for clustering data with convex class shape. In most of the cases for the data sets with overlapping patterns, FRGSOM is seen to be superior to others in terms of \( \beta \)-index, DB-index, Dunn-index and fuzzy rough entropy (FRE).

References


[22] P. Lingras, M. Hogh, and M. Shnay, Interval set clustering of web users using
modified kohonen self-organizing maps based on the properties of rough sets, 

[23] J. P. Herbert and J. T. Yao, A granular computing frame work for self-


fuzzy granules and classification, *Theoretical Computer Science*. 412(42),

neural network for unsupervised feature selection, *Neural Networks*. 48,

map and fuzzy rough entropy, *Theoretical Computer Science*. 466, 37–63
(2012).


[31] S. Bandyopadhyay and U. Maulik, Nonparametric genetic clustering; com-
parison of validity, *IEEE Transactions on Systems, Man, and Cybernetics

sification: Performance evaluation for certain nonconvex decision regions,

[33] P. Maji and S. K. Pal, Rough set based generalized fuzzy c-means algo-
rithm and quantitative indices, *IEEE Transactions on Systems, Man, and