# Hybrid Self Organizing Map for Overlapping Clusters

M.N.M. Sap[1], Ehsan Mohebi[2]
[1,2]Faculty of Computer Science and Information Systems, University Technology Malaysia
[1]mohdnoor@utm.my
[2]mehsan3@siswa.utm.my

**Abstract.** The Kohonen self organizing map is an excellent tool in exploratory phase of data mining and pattern recognition. The SOM is a popular tool that maps high dimensional space into a small number of dimensions by placing similar elements close together, forming clusters. Recently researchers found that to capture the uncertainty involved in cluster analysis, it is not necessary to have crisp boundaries in some clustering operations. In this paper to overcome the uncertainty, a two-level clustering algorithm based on SOM which employs the rough set theory is proposed. The two-level stage Rough SOM (first using SOM to produce the prototypes that are then clustered in the second stage) is found to perform well and more accurate compared with the proposed crisp clustering method (Incremental SOM) and reduces the errors.

**Keywords:** Clustering, Rough set, SOM, Uncertainty, Incremental.

## 1 Introduction

The self organizing map (SOM) proposed by Kohonen [1], has been widely used in industrial applications such as pattern recognition, biological modeling, data compression, signal processing and data mining [2]. It is an unsupervised and nonparametric neural network approach. The success of the SOM algorithm lies in its simplicity that makes it easy to understand, simulate and be used in many applications. The basic SOM consists of neurons usually arranged in a two-dimensional structure such that there are neighborhood relations among the neurons. After completion of training, each neuron is attached to a feature vector of the same dimension as input space. By assigning each input vector to the neuron with nearest feature vectors, the SOM is able to divide the input space into regions (clusters) with common nearest feature vectors. This process can be considered as performing vector quantization (VQ) [3].

Clustering algorithms attempt to organize unlabeled input vectors into clusters such that points within the cluster are more similar to each other than vectors belonging to different clusters [4]. The clustering methods are of five types: hierarchical clustering, partitioning clustering, density-based clustering, grid-based clustering and model-based clustering [5]. The rough set theory employs two upper and lower thresholds in the clustering process which result in a rough clusters appearance. This technique also could be defined in incremental order i.e. the number of clusters is not predefined by users.

In this paper, a new two-level clustering algorithm is proposed. The idea is that the first level is to train the data by the SOM neural network and the clustering at the second level is a rough set based incremental clustering approach [6], which will be applied on the output of SOM and requires only a single neurons scan. The optimal number of clusters can be found by rough set theory which groups the given neurons into a set of overlapping clusters.

This paper is organized as following; in section 2 the basics of SOM algorithm are outlined. The basic of incremental clustering and rough set based approach are described in section 3. In section 4 the proposed algorithm is presented. Section 5 is dedicated to experiment results and section 6 provides brief conclusion and future works.

## 2 Self Organizing Map

Competitive learning is an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, sets of samples in a specific domain of the input space. A division of neural nodes emerges in the network to represent different patterns of the inputs after training.

The division is enforced by competition among the neurons: when an input $x$ arrives, the neuron that is best able to represent it wins the competition and is allowed to learn it even better. If there exist an ordering between the neurons, i.e. the neurons are located on a discrete lattice, the competitive learning algorithm can be generalized. Not only the winning neuron but also its neighboring neurons on the lattice are allowed to learn, the whole effect is that the final map becomes an ordered map in the input space. This is the essence of the SOM algorithm. The SOM consist of $m$ neurons located on a regular low-dimensional grid, usually one or two dimensional. The lattice of the grid is either hexagonal or rectangular.

The basic SOM algorithm is iterative. Each neuron $i$ has a $d$ -dimensional feature vector $w_i = [w_{i1},...,w_{id}]$. At each training step $t$, a sample data vector $x(t)$ is randomly chosen for the training set. Distance between $x(t)$ and all feature vectors are computed. The winning neuron, denoted by $c$, is the neuron with the feature vector closest to $x(t)$:

$$c = \arg\min_i \|x(t) - w_i\|, \qquad i \in \{1,...,m\} \tag{1}$$

A set of neighboring nodes of the winning node is denoted as $N_c$. We define $h_{ic}(t)$ as the neighborhood kernel function around the winning neuron $c$ at time $t$. The neighborhood kernel function is a non-increasing function of time and of the distance of neuron $i$ from the winning neuron $c$. The kernel can be taken as a Gaussian function:

$$h_{ic}(t) = e^{-\frac{\|Pos_i - Pos_c\|^2}{2\sigma(t)^2}} \tag{2}$$

where $Pos_i$ is the coordinates of neuron $i$ on the output grid and $\sigma(t)$ is kernel width. The weight update rule in the sequential SOM algorithm can be written as:

$$w_i(t+1) = \begin{cases} w_i(t) + \varepsilon(t) h_{ic}(t)\big(x(t) - w_i(t)\big) \forall i \in N_c \\ w_i(t) \qquad\qquad ow \end{cases} \tag{3}$$

Both learning rate $\varepsilon(t)$ and neighborhood $\sigma(t)$ decrease monotonically with time. During training, the SOM behaves like a flexible net that folds onto a cloud formed by training data. Because of the neighborhood relations, neighboring neurons are pulled to the same direction, and thus feature vectors of neighboring neurons resemble each other. There are many variants of the SOM [7], [8]. However, these variants are not considered in this paper because the proposed algorithm is based on SOM, but not a new variant of SOM.

## 3 Rough set Incremental Clustering

### 3.1 Incremental Clustering

Incremental clustering [9] is based on the assumption that it is possible to consider data points one at a time and assign them to existing clusters. Thus, a new data item is assigned to a cluster without looking at previously seen patterns. Hence the algorithm scales well with size of data set.

It employs a user-specified threshold and one of the patterns as the starting leader (cluster's leader). At any step, the algorithm assigns the current pattern to the most similar cluster (if the distance between pattern and the cluster's leader is less or equal than threshold) or the pattern itself may get added as a new leader if its similarity with the current set of leaders does not qualify it to get added to any of the existing clusters. The set of leaders found acts as the prototype set representing the clusters and is used for further decision making.

An incremental clustering algorithm for dynamic information processing was presented in [10]. The motivation behind this work is that, in dynamic databases, items might get added and deleted over time. These changes should be reflected in the partition generated without significantly affecting the current clusters. This algorithm was used to cluster incrementally a database of 12,684 documents.

The quality of a conventional clustering scheme is determined using within-group-error [11] $\Delta$ given by:

$$\Delta = \sum_{i=1}^{m} \sum_{u_h,u_k \in C_i} distance(u_h,u_k) \qquad u_h,u_k \textit{ are objects in the same cluster } C_i. \tag{4}$$

## 3.2 Rough set Incremental Clustering

This algorithm is a soft clustering method employing rough set theory [12]. It groups the given data set into a set of overlapping clusters. Each cluster is represented by a *lower approximation* and an *upper approximation* $(\underline{A}(C),\overline{A}(C))$ for every cluster $C \subseteq U$. Here $U$ is a set of all objects under exploration. However, the lower and upper approximations of $C_i \in U$ are required to follow some of the basic rough set properties such as:

*(1)* $\emptyset \subseteq \underline{A}(C_i) \subseteq \overline{A}(C_i) \subseteq U$

*(2)* $\underline{A}(C_i) \cap \underline{A}(C_j) = \emptyset,\ i \neq j$

*(3)* $\underline{A}(C_i) \cap \overline{A}(C_j) = \emptyset,\ i \neq j$

*(4) If an object* $u_k \in U$ *is not part of any lower approximation, then it must belong to two or more upper approximations.*

Note that (1)-(4) are not independent. However enumerating them will be helpful in understanding the basic of rough set theory.

The lower approximation $\underline{A}(C)$ contains all the patterns that definitely belong to the cluster $C$ and the upper approximation $\overline{A}(C)$ permits overlap. Since the upper approximation permits overlaps, each set of data points that are shared by a group of clusters define *indiscernible* set. Thus, the ambiguity in assigning a pattern to a cluster is captured using the upper approximation. Employing rough set theory, the proposed clustering scheme generates soft clusters (clusters with permitted overlap in upper approximation) see figure 1. A high level description of a rough incremental algorithm is as following pseudo code [13].

```
Rough_Incremental (Data, upper_Thr, lower_Thr){
  Cluster_Leader = d1;
  While (there is unlabeled data){
   For (i = 2   to    N)
    If (distance(Cluster_Leader, di) <= lower_Thr)
     Put di in the lower approx of Cluster_Leader;
    Else If (distance(Cluster_Leader, di) <= upper_Thr)
     Put di in all existing clusters (j=1 to k)that
     distance(Cluster_Leaderj, di) <= upper_Thr ;
    Else
     Cluster_Leader = di; // new Cluster
  }//end of while
 }
```

For a rough set clustering scheme and given two objects $u_h, u_k \in U$ we have three distinct possibilities:

1. Both $u_k$ and $u_h$ are in the same lower approximation $\underline{A}(C)$.
2. Object $u_k$ is in lower approximation $\underline{A}(C)$ and $u_h$ is in the corresponding upper approximation $\overline{A}(C)$, and case 1 is not applicable.
3. Both $u_k$ and $u_h$ are in the same upper approximation $\overline{A}(C)$, and case 1 and 2 are not applicable.

For these possibilities, three types of equation (4) could be defined as following:

$$\Delta_1 = \sum_{i=1}^{m} \sum_{u_h, u_k \in \underline{A}(X_i)} distance(u_h, u_k) \tag{5}$$

$$\Delta_2 = \sum_{i=1}^{m} \sum_{u_h \in \underline{A}(X_i) \text{ and } u_k \in \overline{A}(X_i)} distance(u_h, u_k)$$

$$\Delta_3 = \sum_{i=1}^{m} \sum_{u_h, u_k \in \overline{A}(X_i)} distance(u_h, u_k)$$

The total error of rough set clustering will then be a weighted sum of these errors:

$$\Delta_{total} = w_1 \times \Delta_1 + w_2 \times \Delta_2 + w_3 \times \Delta_3 \qquad where \quad w_1 > w_2 > w_3. \tag{6}$$

Since $\Delta_1$ corresponds to situations where both objects definitely belong to the same cluster, the weight $w_1$ should have the highest value.

## 4 Rough set Clustering of the Self Organizing Map

In this paper rectangular grid is used for the SOM. Before training process begins, the input data will be normalized. This will prevent one attribute from overpowering in clustering criterion. The normalization of the new pattern $X_i = \{x_{i1}, ..., x_{id}\}$ for $i = 1,2,..., N$ is as following:

$$X_i = \frac{X_i}{\|X_i\|}. \tag{7}$$

Once the training phase of the SOM neural network completed, the output grid of neurons which is now stable to network iteration, will be clustered by applying rough set algorithm as described in the previous section. The similarity measure used for rough set clustering of neurons is Euclidean distance (the same used for training the SOM). In this proposed method (see figure 1) some neurons, those never mapped any data are excluded from being processed by rough set algorithm.

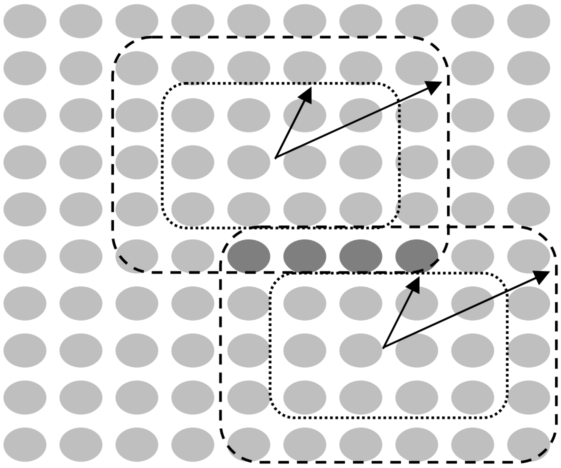


**Fig. 1.** Rough set incremental clustering of the self organizing map. The overlapped neurons have been highlighted (the overlapped data is mapped with these neurons).

From the rough set algorithm it can be observed that if two neurons are defined as indiscernible (those neurons in the upper approximation of two or more clusters), there is a certain level of similarity they have with respect to the clusters they belong to and that similarity relation has to be symmetric. Thus, the similarity measure must be symmetric.

According to the rough set clustering of SOM, overlapped neurons and respectively overlapped data (those data in the upper approximation) are detected. In the experiments, to calculate errors and uncertainty, the previous equations (5) and (6) will be applied to the results of SOM (clustered and overlapped data).

The aim of the proposed approach is making the rough set clustering of the SOM to be as precise as possible. Therefore, a precision measure needs to be used for evaluating the quality of the proposed approach. A possible precision measure can be defined as the following equation [12]:

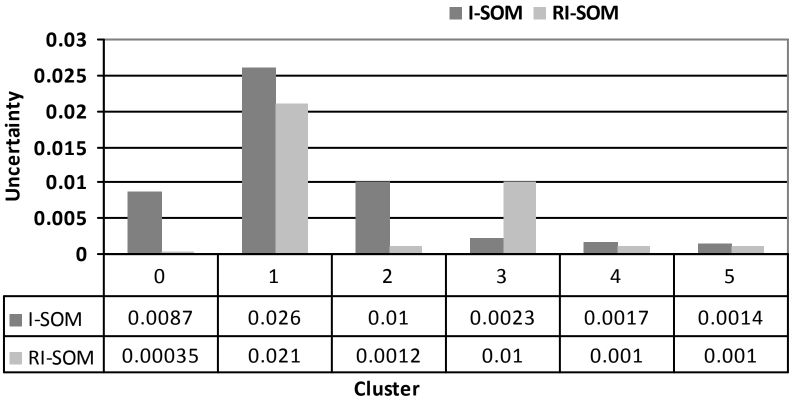
$$certainty = \frac{\text{Number of objects in lower approximation}}{\text{Total number of objects}} \tag{8}$$

## 5 Experimentation and Results

To demonstrate the effectiveness of the proposed clustering algorithm RI-SOM (Rough set Incremental clustering of the SOM), two phases of experiments has been done on two data sets, one artificial and one real-world data set.

The first phase of experiments presents the uncertainty that comes from the both data sets and in the second phase the errors has been generated. The results of RI-SOM are compared to I-SOM (Incremental clustering of SOM), which described in section 3.1. The input data are normalized such that the value of each datum in each dimension lies in $[0,1]$. For training, SOM $10 \times 10$ with 100 epochs on the input data is used.

The artificial data set has 569 data of 30 dimensions which is trained twice, once with I-SOM and once with RI-SOM. The generated uncertainty (figure 2) is gained by the equation (8). From the figure 2 it could be observed that the number of predicted clusters on the artificial data set is 5 and the uncertainty-level in clustering prediction of RI-SOM is satisfactory compared to I-SOM.



| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| ■ I-SOM | 0.0087 | 0.026 | 0.01 | 0.0023 | 0.0017 | 0.0014 |
| ▨ RI-SOM | 0.00035 | 0.021 | 0.0012 | 0.01 | 0.001 | 0.001 |

**Fig. 2.** Comparison of the generated uncertainty on the prepared artificial data set.

The second data set is Iris data set [14] has been widely used in pattern classification. It has 150 data points of four dimensions. The data are divided into three classes with 50 points each. The first class of Iris plant is linearly separable from the other two. The other two classes are overlapped to some extent. Figure 3 shows the certainty generated from epoch 100 to 500 by the equation (8). From the gained certainty it's obvious that the RI-SOM could efficiently detect the overlapped data that have been mapped by overlapped neurons (table 1).

**Table 1.** The certainty-level of I-SOM and RI-SOM on the Iris data set from epoch 100 to 500.

| *Epoch* | *100* | *200* | *300* | *400* | *500* |
|---|---|---|---|---|---|
| **I-SOM** | 33.33 | 65.23 | 76.01 | 89.47 | 92.01 |
| **RI-SOM** | 67.07 | 73.02 | 81.98 | 91.23 | **97.33** |

In the second phase, the same initialization for the SOM has been used. The errors that come from both data sets, according to the equations (5) and (6) have been generated by our proposed algorithms (table 2). The weighted sum equation (6) has been configured as following:

$$\sum_{i=1}^{3} w_i = 1 \qquad \textbf{(9)}$$
$$\textit{and for each } w_i \textit{ we have}:$$
$$w_i = \frac{1}{6} \times (4 - i).$$

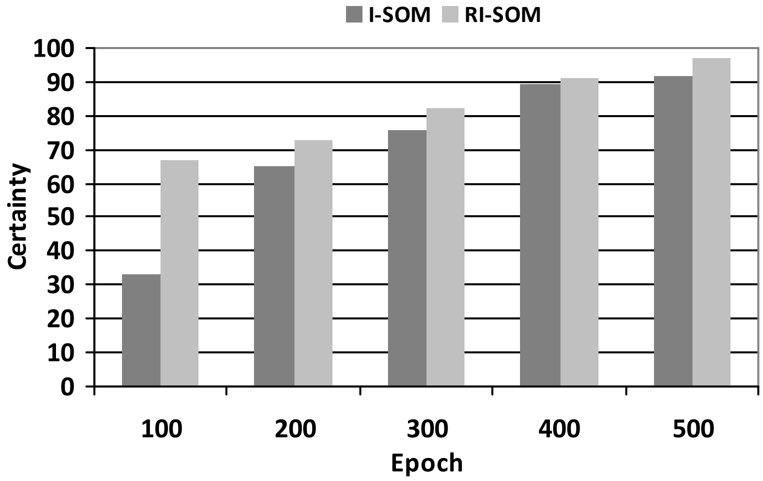


**Fig. 3.** Comparison of the certainty-level of RI-SOM and I-SOM on the Iris data set.

**Table 2.** Comparative generated errors of I-SOM and RI-SOM according to equations (5) and (6).

| | Method | $\Delta_1$ | $\Delta_2$ | $\Delta_3$ | $\Delta_{total}$ |
|---|---|---|---|---|---|
| Artificial Data set | **RI-SOM** | 0.6 | 0.88 | 0.04 | **1.4** |
| | **I-SOM** | | | | 1.8 |
| Iris Data set | **RI-SOM** | 1.05 | 0.85 | 0.043 | **1.94** |
| | **I-SOM** | | | | 2.8 |

## 6 Conclusion and Future Work

In this paper a two-level based clustering approach (RI-SOM), has been proposed to predict clusters of high dimensional data and to detect the uncertainty that comes from the overlapping data. The approach is based on the rough set theory that employs a soft clustering which can detects overlapped data from the data set and makes clustering as precise as possible. The results of the both phases indicate that RI-SOM is more accurate and generates fewer errors as compared to crisp clustering (I-SOM).

The proposed algorithm detects accurate overlapping clusters in clustering operations. As the future work, the overlapped data could be assigned correctly to true clusters they belong to, by assigning *fuzzy membership value* to the indiscernible set of data. Also a weight can be assigned to the data's dimension to improve the overall accuracy.

## Acknowledgments

This research was supported by the Research Management Centre, University Technology Malaysia (UTM) and the Malaysian Ministry of Science, Technology and Innovation (MOSTI) under vote number 79224.

## References

1. T. Kohonen.: Self-organized formation of topologically correct feature maps. Biol. Cybern. 43 59–69 (1982)
2. T. Kohonen.: Self-Organizing Maps. Springer, Berlin, Germany (1997)
3. R.M. Gray.: Vector quantization. IEEE Acoust. Speech, Signal Process. Mag. 1 (2) 4–29 (1984)
4. N.R. Pal, J.C. Bezdek, and E.C.K. Tsao.: Generalized clustering networks and Kohonen's self-organizing scheme. IEEE Trans. Neural Networks (4) 549–557 (1993)
5. J. Han, M. Kamber.: Data mining: concepts and techniques. Morgan Kaufman, San Francisco (2000)
6. S. Asharaf, M. Narasimha Murty, and S.K. Shevade.: Rough set based incremental clustering of interval data. Pattern Recognition Letters (27) 515-519 (2006)
7. Yan and Yaoguang.: Research and application of SOM neural network which based on kernel function. Proceeding of ICNN&B'05 (1) 509- 511 (2005)
8. M.N.M. SAP and Ehsan Mohebi.: Outlier Detection Methodologies: A Review. Journal of Information Technology, UTM, Vol. 20, Issue 1 87-105 (2008)
9. A.K. Jain, M.N. Murty, and P.J. Flynn.: Data Clustering: A Review. ACM Computing Surveys (31) (3) 264–323 (1999)
10. Can, F.: Incremental Clustering for dynamic information peocessing. ACM Trans. Inf. System (11) 2 143-164 (1993)
11. S.C. Sharma and A. Werner.: Improved method of grouping provincewide permanent traffic counters. Transaction Research Report 815, Washington D.C. 13-18 (1981)
12. Pawlak, Z.: Rough sets. Internat. J. Computer Inf. Sci. (11) 341–356 (1982)
13. Lingras, P.J., West, C.: Interval set clustering of web users with rough K-means. J. Intelligent Inf. Syst. (23) (1) 5–16 (2004)
14. UCI Machine Learning Repository, www.ics.uci.edu/mlearn/MLRepository.html