CLUSTERING ANALYSIS USING A SELF-ORGANIZED NETWORK INSPIRED BY IMMUNE ALGORITHM

Rahmat Widyanto[1], Benyamin Kusumoputro[2] and Kaoru Hirota[1]

[1] Department of Computational Intelligence and Systems Science
Interdisciplinary Graduate School of Sciences and Engineering, Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan
email: {widyanto, hirota}@hrt.dis.titech.ac.jp
[2] Faculty of Computer Sciences, University of Indonesia
Depok Campus, Indonesia
email: kusumo@cs.ui.ac.id

ABSTRACT

An automatic construction of neurons in neural network inspired by immune algorithm is proposed. The new network is combined with the contiguity-constrained method to perform clustering analysis. The applicability of this technique is tested with two widely referenced machine-learning cases. The experiment shows that the new technique achieved 99.33% and 100% correctness for Iris plant data and Wine recognition data respectively, better than other popular clustering methods.

Keywords: Data Mining, Clustering Analysis, Self-Organized Network, Immune Algorithm

Makalah diterima [27 Agustus 2002]. Revisi akhir [15 Oktober 2002].

1. INTRODUCTION

The problem of data mining and knowledge discovery has become increasingly important in recent years. Clustering, in data mining, is useful to discover distribution patterns in underlying data. Clustering is used when there is no class to be predicted but rather when the instances are to be divided into natural groups [1]. Clustering analysis is a technique for grouping subjects into clusters of similar elements. In clustering analysis, similar elements are identified by their attributes. Groups that are homogeneous and different from other groups are formed.

There are many conventional clustering analysis techniques, most of them can be categorized into two classes. The first one is non-hierarchical clustering method, i.e., cover coefficient-based clustering [2] and k-means algorithm, which are based on an iterative manner. The second is hierarchical clustering analysis method like Single link [3], Minimum spanning tree [4], Complete link [5], and Group average [6].

Neural network is used in clustering analysis to help reducing dimensionality of input data as input vectors associated with corresponding neurons [7]. Kohonen Self-Organizing Map (SOM) neural networks [8,9] is frequently used for this task. The output of SOM networks, however, does not automatically provide groupings of the points in the SOM output. Therefore, additional step is required to analyze and group the points of the SOM output into the desired number of clusters. Kiang [10] proposes a contiguity-constrained method based on minimal variance criterion as an extension of SOM networks for clustering analysis. Since in SOM networks the number of neurons should be decided manually, however, careless decision of this number consequences in poor clustering results.

Immune algorithm is an alternative algorithm for data visualization due to its mechanism to automatically create cluster distribution as B-cell being cloned and mutated. Timmis [11] proposes an ARB (Artificial Recognition Ball) concept to replace B-cell network. This ARB construction in immune algorithm inspires us for an automatic neuron construction mechanism in neural network.

In this paper, an automatic construction of neurons in neural network inspired by construction of B-cell in immune algorithm is proposed. The new neural network is called SONIA (Self-Organized Network inspired by Immune Algorithm). This network has been applied successfully for food quality prediction [12], but its applicability for clustering analysis is still a challenge. Here, SONIA network is combined with the contiguity-constrained method and applied to clustering analysis problem. The new technique is tested using two widely used machine learning cases, i.e., Iris plant data and Wine recognition data. Performance of the new clustering technique is compared with combination of SOM and the contiguity-constrained method.

Performances of other popular statistical clustering methods, i.e., k-means clustering, Ward's and MODECLUS are also compared.

Section 2 presents the concept of SOM networks and the contiguity-constrained method for clustering analysis. Section 3 described the proposed SONIA network and its automatic neuron construction mechanism. In section 4, the results of experiments are explained including performance comparison of the proposed technique and other clustering methods.

2. SOM AND THE CONTIGUITY-CONSTRAINED METHOD

Kiang [10] proposed a contiguity-constrained method as an extension of SOM networks for clustering analysis. Here, an explanation of SOM networks and the contiguity-constrained method are given.

2.1. SOM Networks

Kohonen SOM Networks [8],[9] is one of a wide variety of neural network algorithms that have seen a steady increase in usage over the last decade. SOM uses network neurons that model the data that are presented to the network. This is achieved by each neuron in the network competing for a particular data item with the links between neurons not being removed but stretched to allow the neurons to move to the required point in the data space.

This results in a detailed map of the data, showing regions in the data that are similar, as similar data items will be clustered around the same neuron. Relationships between neurons are maintained by links that stretch to allow the neurons to move toward data items. This process results in a map that is representative of the data presented to the SOM, showing areas of similarity and relative relationships between those areas. The output of SOM, however, does not automatically provide groupings of the points on the map.

2.2. Contiguity-Constrained Method

Additional step is required to analyze and group the points on SOM output map into the desired number of clusters. To solve the above problem, Kiang [10] proposed an agglomerative contiguity-constrained method based on minimal variance criterion. This approach recursively merge groups from the Kohonen SOM output until a desired number of clusters is reached. The detailed process of this method is divided into two steps, i.e., initialization and merging process as follows:

Initialization

1. For each $neuron_i$, calculate the centroid (C_i) of $neuron_i$ as

$$C_i = \frac{1}{t_i} \sum_{y \in neuron_i} \vec{y}. \quad (1)$$

where t_i is the number of input vectors y associated with the $neuron_i$.

2. Assign a group number (G_j) to each $neuron_i$ if $t_i > 0$, and update the corresponding centroid value C_j.

3. Calculate the overall variance of the map:
   a. Sum the square distance between input vector y and the group centroid C_j for all y in G_j. Calculate for every group j

$$V_j = \sum \| y - C_j \|^2, y \in G_j. \quad (2)$$

   b. Total the variances from all groups. This results the global variances of the map:

$$V_{Total} = \sum V_j. \quad (3)$$

Merging Process

Repeat this merging process until pre-specified number of clusters has been reached. For each pair of neighboring groups, calculate the total variance of the map if the two groups were merged. Merged the two groups that result in the minimum global variance.

1. Calculate the new centroid for G_{ef} if G_e and G_f were merged

$$G_{ef} = \left(| G_e | \vec{G}_e + | G_f | \vec{G}_f \right) / | G_e | + | G_f |. \quad (4)$$

2. Calculate the new variance if G_e and G_f were merged:

$$V_{ef} = \sum \| y - C_{new} \| \; \forall \, y, y \in G_e \text{ or } y \in G_f. \quad (5)$$

3. Calculate the new global variance for merging G_e and G_f.

$$V_{efTotal} = V_{Total} + V_{ef} - V_e - V_f. \quad (6)$$

4. Calculate the $V_{efTotal}$ for every pair of e and f on the map.

5. Update V_{Total}, the group number and group centroid of the two newly merged groups.

3. SONIA NEURAL NETWORK

SONIA (Self-Organized Network inspired by Immune Algorithm) neural network is developed based on Kohonen Self-Organizing Network [8,9]. Here, a method for automatic construction of neurons inspired by Immune Algorithm is proposed.

3.1. Introduction

Immune algorithm is an alternative algorithm for data visualization [11] due to its mechanism to automatically create cluster distribution as B-cell being cloned and mutated. On the other hand, Kohonen SOM networks have a weakness that number of neurons must be set manually. Frequently, careless decision on neuron number results in poor neurons created. In order to overcome the problem, a method for automatic creation of neurons in hidden layer inspired by immune algorithm is proposed.

3.2. Immune Metaphor

Immune system protects our bodies from infectious agents called antigen, such us viruses, bacteria, fungi and other parasites. Antigen phenomena are similar to training data in computing concept. Immune system has also phenomena called first and second immune responses that are similar to training and testing phases in neural network. In first immune response, the number of B-cell will be increased through mutation and cloning due to presentation of a new antigen set. Once number of B-cell is fixed due to mutation and cloning, the system is ready to perform the second immune system in which antigens will be matched to the most suitable B-cell network.

Timmis [2] proposes an ARB (Artificial Recognition Ball) concept to replace B-cell network, where a clustering-like mechanism rather than network is used to define similar B-cells. This ARB construction in immune algorithm inspires a mechanism for automatic construction of neurons in neural network.

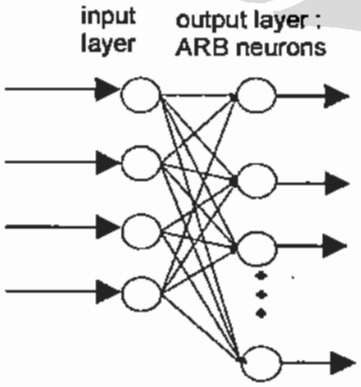

Fig. 1. Architecture of SONIA neural network: input layer and output layer where ARB neurons located.

3.3. Architecture

SONIA neural network has two layers (Fig. 1). The first layer is input layer. The second layer is the output layer in which the nodes are automatically constructed. These constructed nodes are called ARB neurons.

3.4. Neuron Construction Procedure

This section introduces a neuron construction procedure inspired by Immune Algorithm. Input pattern of the network is defined as antigen vectors that represent the characteristics of an observed data set, the data are normalized between 0 and 1 as:

$$y(k) = [y_1(k), y_2(k), y_3(k), \ldots, y_s(k)], \quad (6)$$

where s is data dimension and k is time instance. For each ARB neuron, the following three values are memorized: number of B cells (t_a), representative vector (x_a), and stimulation level (sl_a), where a is an index. In initialization process, an ARB neuron is created with $t_a = 0$, x_a is taken arbitrarily from antigen vectors, and $sl_a = 1$.

Repeat the below procedure starting from the first antigen vector $y(1)$ to the last $y(p)$, until all antigen vectors have found their corresponding ARB neuron (p is number of antigen vectors).

1. Find distances of antigen and all representative vectors.
   For $a = 1$ to m (where m is number of ARB neurons) do
   $$d_a(k) = |y(k) - x_a(k)| . \quad (7)$$
2. Find the closest distance
   $$c = \arg a \min d_a(k) . \quad (8)$$
3. Check if $d_c(k) \le R \cdot sl_c(k)$ (where $0 < R \le 1$).
   If $d_c(k) \le R \cdot sl_c(k)$ (ARB neuron found) update the followings ($0 < h < 1$):
   $$t_c(k+1) = t_c(k) + 1 \quad (9)$$
   $$x_c(k+1) = x_c(k) + h.d_c(k) \quad (10)$$
   If $d_c(k) > R.\ sl_c(k)$ (ARB neuron not found) repeat to find the next closest distance in step 2. If there is no next closest distance do the followings:

a. Prune unlearned neurons
Check if there are neurons that not learn in which the number of B-cells is equal to zero. If exist delete the neurons.
b. Adopt the current antigen
Create a new ARB neuron with $t_a = 0$, x_a is the current antigen vector, and $sl_a = 1$.
c. Create a mutated ARB neuron
Construct a new ARB to create a diverse vector with $t_a = 0$, x_a is generated randomly, and $sl_a = 1$.

3.5. Complexity Analysis

In the above procedure, creation of new neurons is guaranteed to be finite since a new neuron is created only if an antigen vector could not find its corresponding neuron. At maximum the number of neurons created is p, where p is number of antigen vectors. The p neurons are possibly created when the antigen vector cannot find its corresponding neuron, and the p neurons are also possibly created as mutated neurons. However, the maximum number of neuron created is p because of unlearned neuron pruning. In normal condition the number of neuron created is less than p, because similar antigens are located to the same ARB neuron. Complexity of this procedure in the worst condition is $\Omega(p) = 1/2p^2$ and in the best condition is $O(p) = p$. In the worst condition, all antigen vectors cannot find their corresponding neuron, so for every antigen vector a new neuron is created. In the best condition all neurons can find their corresponding neuron, so there is no need to create new neurons.

3.6. Testing Phase

After the network learned using the above procedure, it is ready for testing phase. In testing phase, antigen vectors are classified using Kohonen Self-Organized Map (SOM) algorithm [8,9] to the closest ARB neuron. All the antigen vectors are inserted to the network. Finally each ARB neuron has input vectors associated with it. Then, the contiguity-constrained method is ready to apply.

4. CLUSTERING EXPERIMENTS

In the experiment, SONIA network is combined with the contiguity-constrained method called *extended SONIA*. The output of SONIA network's testing phase become an input for the contiguity-constrained method. Therefore, the pre-specified number of clusters is reached.

The performance of above technique is compared with combination of SOM networks and the contiguity-constrained method called *extended SOM*. Performances of other popular statistical clustering methods, i.e., k-means clustering, Ward's and MODECLUS are also compared.

The extended SONIA is developed using Matlab 6.1 under PC with Pentium III 600 Mhz and 64 MB memory environment on Microsoft Windows 2000 operating system. For SOM networks, we simply used Matlab toolbox on the same environment. Performance of other clustering methods refer to Kiang's works [10], which are simulated by SAS statistical software.

The applicability of extended SONIA is tested with two widely referenced machine learning cases, i.e., Iris plant data and Wine recognition data.

4.1. The Iris Plant Data

The database was created by Fisher in 1936 and has been widely used in subsequent research in pattern classification [13]. The data set contains three classes of 50 instances each, where each class refers to a type of Iris plant. The three classes: *Iris sentosa*, *Iris versicolor*, and *Iris virginica*. There are four numeric attributes and no missing value. The four attributes are: sepal length in cm., sepal width in cm., petal length in cm., and petal width in cm. (cm. = centimeters). Table I is a brief statistical analysis of the sample attributes.

TABLE I
STATISTICAL ANALYSIS OF IRIS PLANT DATA

Attribute	Min	Max	Mean	Standard Deviation
Sepal length	4.3	7.9	5.84	0.83
Sepal width	2.0	4.4	3.05	0.43
Petal length	1.0	6.9	3.76	1.76
Petal width	0.1	2.5	1.20	0.76

Statistical analysis of the Iris plants sample attributes (in centimeter).

Stimulation parameter R of SONIA network is set to be 0.1. Meanwhile for SOM networks, number of neurons is set to be 74 since this number results in highest correctness percentage. To compare with SONIA network, the SOM networks used is a one-dimensional network. The epoch number of SOM networks is set to be 200. Matlab's SOM neural networks toolbox is used to simulate the SOM networks. Results of other popular statistical clustering methods refers to Kiang's paper [10], which are simulated by SAS statistical software.

Table 2 shows the correctness percentage for Iris plant data. Extended SONIA reached the highest correctness percentage of 99.3%. Meanwhile, extended SOM reached 92% and k-means analysis, Ward's, and MODECLUS got 88,67% of correctness percentage. The

total time needed for extended SONIA is 73.45 seconds, far faster than extended SOM that is 253.82 seconds.

TABLE II
CORRECTNESS PERCENTAGE FOR IRIS PLANT DATA

Method	Rate of Correctness
Extended SONIA	99.3 %
Extended SOM	92 %
k-means analysis	88.67 %
Ward`s	88.67 %
MODECLUS	88.67 %

Experiment results for Iris plant data. Extended SONIA shows the highest correctness percentage comparing with other clustering techniques.

4.2. The Wine Recognition Data

The wine recognition data contains three classes and there are 59, 71, and 48 instances in each class, respectively. These data are the results of a chemical analysis of wines produced in the same region in Italy but derived from three different cultivators. The analysis determined the quantities of 13 attributes found in each of the three types of wines [4]. The 13 attributes are: Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Non-flavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, and Proline. All attributes are continuous and have no missing value.

TABLE III
CORRECTNESS PERCENTAGE FOR WINE RECOGNITION DATA

Method	Rate of Correctness
Extended SONIA	100 %
Extended SOM	97.33 %
k-means analysis	85.96 %
Ward`s	97.75 %
MODECLUS	88.67 %

Experiment results for Wine recognition data. Extended SONIA shows the highest correctness percentage comparing with other clustering techniques.

Stimulation parameter R of SONIA network is set to be 0.4. Meanwhile for SOM networks, number of neurons is set to be 60 since this number results in highest correctness percentage. To compare with SONIA network, the SOM networks used is a one-dimensional network. The epoch number of SOM networks is set to be 200. Matlab`s SOM neural networks toolbox is used to simulate the SOM networks. Results of other popular statistical clustering methods refers to Kiang`s paper [4], which are simulated by SAS statistical software.

Table 3 shows the correctness percentage for Wine recognition data. Extended SONIA reached the highest correctness percentage of 100%. Meanwhile, extended SOM, k-means analysis, Ward`s and MODECLUS got 97.33%, 85.96%, 97.75% and 88.67% of correctness percentage, respectively. The total time needed for extended SONIA is 402.5 seconds, faster than extended SOM that is 439.95 seconds.

5. CONCLUSION

A method for automatic creation of neurons in neural network inspired by immune algorithm is proposed. The new network is called SONIA network. SONIA network is combined with the contiguity-constrained method called *extended SONIA* to perform clustering analysis. Experiments on two widely referenced machine learning cases, i.e., Iris plant data and Wine recognition data shows the extended SONIA outperformed other clustering analysis techniques. The time needed by extended SONIA is also faster than extended SOM.

Extended SONIA provides better neurons constructed comparing to the combination of SOM networks and the contiguity-constrained method called *extended SOM*. Therefore, in extended SONIA input vectors are better associated with their corresponding neuron rather than in extended SOM. Thus, the contiguity-constrained method is easier to group the neurons into the desired number of clusters.

The mutation mechanism of SONIA network helps to find the appropriate neuron weight. The appropriate neuron weight is needed to construct neurons that lead to a better association with their corresponding input vectors. An inappropriate neuron weight created by the network will be pruned. But, further investigation is needed to see the impact of this mutation mechanism to the quality of neurons. An improvement of mutation mechanism is considered for the future work.

Further research is also needed to see relation between clustering results and number of neuron setting for SOM networks, as well as stimulation level parameter of SONIA network. However, our preliminary investigation shows that changing the stimulation parameters of SONIA network results in more stable clustering results comparing to SOM networks.

REFERENCES

[1] I. H. Witten and E. Frank, *Data mining: practical machine learning tools* (San Francisco: Morgan Kaufmann, 1999).

[2] F. Can, and E. A. Ozkarahan, Concepts of the cover coefficient-based clustering methodology, *Special*

Research Group on Information Retrieval (ACM Press, 1985).
[3] R. Sibson, SLINK: an optimal efficient algorithm for the single-link cluster method, *Computer Journal, 16*(1), 1973, 30-34.
[4] F. J. Sohlf, Single-link clustering algorithms, in P. R. Krishnaiah, and J. N. Kanal (Ed.), *Classification, pattern recognition, and reduction of dimensionality* (Amsterdam: North Holland, 1982) 267-843.
[5] W. B. Frakes, and R. Baeza-Yates, *Information retrieval: data structures and algorithms* (Prentice-hall, 1992).
[6] E. M. Voorhees, Implementing agglomerative hierarchic clustering algorithms for use in document retrieval, *Information processing & management, 22*(6), 1986, 465-476.
[7] L. Smith, *An introduction to neural networks, http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html*
[8] T. Kohonen, *Self-organizing and associative memory* (Berlin: Springer, 1989).
[9] J. Kangas and T. Kohonen, Developments and applications of the self-organized map and related algorithms, *Mathematics and Computers in simulation, 41*(1-2), 1996, 3-12.
[10] M. Y. Kiang, Extending the Kohonen self-organizing map network for clustering analysis, *Computational Statistics & Data Analysis, 38*(2), 2001, 161-180.
[11] J. I. Timmis, Artificial immune systems: a novel data analysis technique inspired by the immune network theory, *PhD. Dissertation* (Aberystwyth: University of Wales, 2001).
[12] R. Widyanto, Megawati, Y. Takama, and K. Hirota, A time-temperature-based food quality prediction using a self-organized network inspired by immune algorithm (accepted), *International Conference on Soft Computing and Intelligent Systems*, Tsukuba, Japan, 2002.
[13] R. O. Duda, and P. E. Hart, *Pattern classification and scene analysis* (New York: Wiley, 1973).