ELSEVIER

Neurocomputing 48 (2002) 863–877

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Grey self-organizing feature maps

Yi-Chung Hu[a], Ruey-Shun Chen[a], Yen-Tseng Hsu[b], Gwo-Hshiung Tzeng[c,*]

[a] *Institute of Information Management, National Chiao Tung University, Hsinchu, Taiwan, ROC*

[b] *Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, ROC*

[c] *Institute of Management of Technology, National Chiao Tung University, Hsinchu, Taiwan, ROC*

Received 10 September 2000; accepted 13 August 2001

Abstract

In each training iteration of the self-organizing feature maps (SOFM), the adjustable output nodes can be determined by the neighborhood size of the winning node. However, it seems that the SOFM ignores some important information, which is the relationships that actually exist between the input training data and each adjustable output node, in the learning rule. By viewing input data and each adjustable node as a reference sequence and a comparative sequence, respectively, the grey relations between these sequences can be seen. This paper thus incorporates the grey relational coefficient into the learning rule of the SOFM, and a grey clustering method, namely the GSOFM, is proposed. From the simulation results, we can see that the best result of the proposed method applied for analysis of the iris data outperforms those of other known unsupervised neural network models. Furthermore, the proposed method can effectively solve the traveling salesman problem. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Self-organizing feature maps; Grey relation; Grey clustering; Traveling salesman problem

1. Introduction

Kohonen originally proposed the self-organizing feature maps (SOFM) learning algorithm in 1984 [19], and since then it has served as a powerful tool for a

* Corresponding author. Fax: +886-3-5753926.
E-mail address: ghtzeng@cc.nctu.edu.tw (G.-H. Tzeng).

PII: S0925-2312(01)00677-4

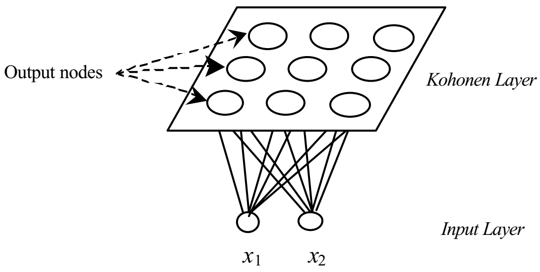

Fig. 1. Basic model of SOFM.

variety of applications, including problem solving for pattern recognition and image processing. The SOFM can map the distribution of input data with any number of dimensions to a one- or two-dimensional feature map graph, preserving the statistical properties of the data distribution [16,17,3]. Furthermore, each output node of the SOFM is restricted to a smaller distance around the cluster center in the cluster analysis [3]. Significantly, this paper demonstrates that the problem-solving capability of the SOFM can be enhanced by incorporating grey relations, previously proposed by Deng [10], into the SOFM.

We show the basic model of SOFM in Fig. 1, indicating that there are two layers in the model: one is the Kohonen layer, consisting of multiple output nodes with one- or two-dimensions; and the other is the input layer. Both layers are fully connected and each connection is given an adjustable weight. Let the number of the output nodes to be m, the number of the input nodes be n, and $\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_{in})(1 \le i \le m)$ be the connection weight vector corresponding to the output node i. Thus, $\mathbf{w}_i$ can be viewed as the center of the cluster i.

Whenever new input training data $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is presented to the network during the training phase of the SOFM, the output value of the output node i can be obtained by computing the square of the Euclidean distance denoted by o_i between $\mathbf{x}$ and $\mathbf{w}_i$, as:

$$o_i = (d_i)^2 = \|\mathbf{x} - \mathbf{w}_i\|^2 = \sum_{j=1}^{n}(x_j - w_{ij})^2, \quad 1 \le i \le m. \tag{1}$$

If the node i^* satisfies Eq. (2) then it is the winner.

$$(d_{i^*})^2 = \min_i o_i, \quad 1 \le i \le m. \tag{2}$$

Adjustable output nodes including the winning node i^* and its neighbor nodes are determined by the neighborhood size of the winning node i^*, which can be denoted by Λ_{i^*}. Subsequently, connection weights of the adjustable nodes are all updated. The learning rule of the SOFM is as follows [19,16,17]:

$$\Delta w_{ij} = \eta(x_j - w_{ij}), \quad i \in \Lambda_{i^*}, \quad 1 \le j \le n, \tag{3}$$

where η is the learning rate. To achieve a better convergence, η and Λ_{i^*} should be decreased gradually with learning time [17,22]. After sufficient training time, the SOFM can map the distribution of input data with any dimensions to the Kohonen feature maps.

By inspecting Eq. (3), we can see that the movable amount is determined only by the learning rate and the difference between x_j and w_{ij}. However, it seems that the SOFM ignores some important information, which is the relationships that actually exist between the input training data and each adjustable output node, in the learning rule. Indeed, there exist distinct relationships between any two subsystems in the real world [10,24], although we do not know exactly what these relationships are. Grey theory, as proposed by Deng [10], can perform grey relational analysis for these subsystems by dealing with finite and incomplete output data series obtained from these subsystems [15]. Given one reference sequence, for example $\mathbf{x}$, and some comparative sequences, for example $\mathbf{w}_i (1 \leq i \leq m)$, we can easily obtain the grey relation between each corresponding data in these sequences by viewing the reference sequence as a desired goal. Therefore, we consider that the learning rule should take into account the grey relation which actually exists between w_{ij} and x_j. Such a significant relation is called the grey relational coefficient (GRC). The connection weight w_{ij} can thus acquire more movement if there exists a larger GRC between w_{ij} and x_j. This paper incorporates the GRC into the learning rule of the SOFM, and we refer to this novel combination as grey self-organizing feature maps (GSOFM), which can thus be viewed as a grey clustering method. This is the main difference between the original SOFM and the GSOFM.

In the following sections, we first review concepts of the GRC and describe how to compute the GRC between x_j and w_{ij} in Section 2. In Section 3, we describe in detail the GSOFM learning algorithm. To show the problem-solving capability of the GSOFM, in Section 4, the performances are examined by computer simulations on two representative problems: one is the classification problems, including the iris data proposed by Fisher [11], the appendicitis data and the wine recognition data; the second is the traveling salesman problems (TSP). In the first simulation, we compare the best result of the GSOFM with that of the SOFM in each problem. Moreover, the best result of the GSOFM with respect to the iris data is compared with other known unsupervised neural networks models. For applying the neural network with unsupervised learning to classification problems, the summarized results can demonstrate the effectiveness and feasibility of the GSOFM. In the latter simulation, we first briefly introduce the TSP. Since it seems that the learning algorithm introduced in Section 3 simplifies the mechanism for lateral feedback, we incorporate the neighborhood function into the learning rule. A complete learning algorithm of the GSOFM for solving the TSP is described in Section 4.2. Subsequently, we apply the proposed method on the TSPLIB problem set proposed by Reinelt [23] to show the effectiveness of the GSOFM.

2. Grey relational coefficient (GRC)

Grey relational analysis is a method that can find the relationships between one major sequence and the other sequences in a given system [14]. Given the reference sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and the comparative sequences $\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_{in})$ $(1 \le i \le m)$ with the normalized form, the GRC ξ_{ij} between x_j and $w_{ij} (1 \le j \le n)$ can be computed as [24,15,14,8]

$$\xi_{ij} = \frac{\Delta_{\min} + \rho \Delta_{\max}}{\Delta_{ij} + \rho \Delta_{\max}}, \tag{4}$$

where ρ is the discriminative coefficient $(0 \le \rho \le 1)$, and usually $\rho = 0.5$ [14,8]; and

$$\Delta_{\min} = \min_i \min_j |x_j - w_{ij}|, \quad 1 \le i \le m, \quad 1 \le j \le n, \tag{5}$$

$$\Delta_{\max} = \max_i \max_j |x_j - w_{ij}|, \quad 1 \le i \le m, \quad 1 \le j \le n, \tag{6}$$

$$\Delta_{ij} = |x_j - w_{ij}|, \tag{7}$$

where $|\cdot|$ denotes the absolute value. Clearly, ξ_{ij} is between zero to one. Moreover, ξ_{ij} approaches one if Δ_{ij} is near $\Delta_{\min}$. That is, the larger degree of relationship that exists between x_j and w_{ij}, the more movement should be acquired for moving w_{ij} toward x_j. Thus ξ_{ij} is incorporated into the learning rule of the SOFM. We should note that the appropriate value of ρ is actually dependent on individual applications.

Unlike correlation analysis, which only stresses the relationship between any two random variables, grey relational analysis tries to find the relationships between one reference sequence and other comparative sequences by viewing the reference sequence as a desired goal that each comparative sequence expects to attain. In the following section, the learning algorithm for the grey self-organizing feature maps is introduced.

3. Grey self-organizing feature maps (GSOFM)

The learning algorithm of the GSOFM is categorized as unsupervised learning, that is we need not know the desired output of each training data during the training phase. Before training, we usually normalize all the input data and weight vectors [17]. Similar to the SOFM, the training phase in GSOFM is typically composed of the ordering phase and the convergence phase [17,22]. Initially, η should be chosen close to 1.0. Moreover, Λ_i should cover all output nodes. During the ordering phase, η will gradually decrease but not below 0.1. Λ_i will also decrease slowly, as depicted in Fig. 2 [17,22] where t_1 and t_2 are the number of iterations and $0 < t_1 < t_2$. At the end of this phase, both η and Λ_i will achieve much smaller values, and they continue to decrease during the convergence phase. In principle,

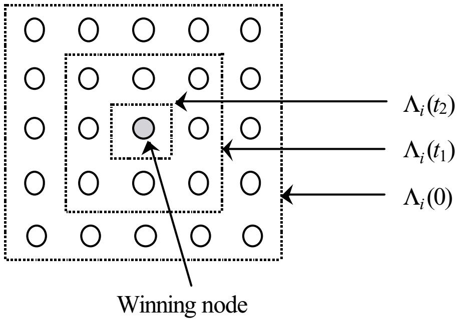

Fig. 2. Neighborhood size of the winning node gradually decrease with time.

η will not be decreased below a given value, say 0.05, and Λ_i will decrease to only cover itself during the training phase. It should be noted that both η and $\Lambda_i(1 \leq i \leq m)$ are decreased at the end of each iteration or each complete pass (i.e. each training data has been presented to the network).

As we have stated in the previous section, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_{in})(1 \leq i \leq m)$ are the reference sequence and the comparative sequences, respectively. Note that, the value of m serving as the number of clusters must be specified before the training task is performed. Significantly, the learning rule of the GSOFM is as follows:

$$\Delta w_{ij} = \eta(\xi_{ij})^k(x_j - w_{ij}), \quad i \in \Lambda_{i^*}, \quad 1 \leq j \leq n, \tag{8}$$

where k is a pre-specified positive real number, and ξ_{ij} is the GRC between x_j and w_{ij}. This implies that if ξ_{ij} is much smaller, then the value of $(\xi_{ij})^k$ will approach zero when k is a larger value. On the other hand, ξ_{ij} will be dampened by a much larger value of k. Significantly, the connection weight w_{ij} could acquire a large amount of movement if there exists a larger GRC between w_{ij} and x_j. We describe the learning algorithm of the GSOFM as the following.

Algorithm : Grey self-organizing feature maps learning algorithm

Input: A given set of training data.

Output: The center of each cluster.

Method:

Step 1: Initialize connection weights and parameters

a. Initialize weights corresponding to each output node with random small values;
b. Initialize $\eta(0)$ and the number of neighbor nodes $\Lambda_i(0)$ of node i: $\eta(0)$ should approach 1.0, and $\Lambda_i(0)$ should cover all output nodes, $1 \leq i \leq m$;
c. Set $t = 1$, where t is an iteration counter.

Step 2: Present input training data $\mathbf{x}(t)$

Step 3: Calculate the output value $o_i(t)$ of each output node i

$$o_i(t) = (d_i(t))^2 = \sum_{j=1}^{n} (w_{ij}(t) - x_j(t))^2, \quad 1 \le i \le m.$$

Step 4: Determine the winning node i^*
The node i^* is the winner if

$$(d_{i^*}(t))^2 = \min_i o_i(t), \quad 1 \le i \le m.$$

Step 5: Adjust the winning nodes i^* and its neighbor nodes
a. The neighbor nodes around the winning node i^* can be determined by $\Lambda_{i^*}(t)$;
b. The learning rule based on $\xi_{ij}(t)$ can be given as Eq. (8)

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)[\xi_{ij}(t)]^k [x_j(t) - w_{ij}(t)], \quad i \in \Lambda_{i^*}(t), \quad 1 \le j \le n,$$

where k is a pre-specified positive real number, and $\xi_{ij}(t)$ is the GRC between $x_j(t)$ and $w_{ij}(t)$. If each training data is presented to the network, then go to Step 6; otherwise go to Step 2.

Step 6: Shrink the learning rate $\eta(t)$ and the neighborhood size $\Lambda_i(t)$
$\eta(t)$ and $\Lambda_i(t)$ may shrink gradually with linear or exponential time, where $1 \le i \le m$.

Step 7: Convergence test
If the winning node of each input data is not changed then stop. Otherwise, set $t+1$ to t and go to Step 2.

To achieve the convergence, empirically, many thousands of iterations for the GSOFM are necessary. We can see that the learning rule of the GSOFM is not determined only by the learning rate and the difference of $w_{ij}(t)$ and $x_j(t)$. To show its effectiveness, we apply the GSOFM for two representative problems: one is the classification problem, including the iris data proposed by Fisher [11], the appendicitis data and the wine recognition data; the second is the TSP. Simulations with specified parameter specifications are described in Section 4.

4. Simulations

To examine the performance of the GSOFM, we first employ it to obtain classification rates on the well-known data including the iris data, the appendicitis data and the wine recognition data. Subsequently, we show that the GSOFM can effectively solve the TSP in comparison with other known neural network models. All programs coded by Delphi version 5.0 were executed by a personal computer with Pentium III-500 CPU. It should be noted that we stress the feasibility and the problem-solving capability of the GSOFM, rather than providing formal methods to find general parameter specifications that can optimize problems.

Table 1
The best result (96.00%) obtained by the GSOFM for the iris data with various k versus ρ

ρ	k	ρ	k
0.02	6.30, 6.38	0.14	8.83, 9.12
0.03	5.23	0.16	9.44, 9.45, 9.58, 9.64, 9.67
0.07	7.78, 8.01, 8.42	0.17	8.13, 9.51
0.08	7.71	0.21	8.06
0.09	5.41, 7.76, 8.84, 8.87	0.23	6.62
0.10	7.98, 8.51, 9.18, 9.40	0.34	7.65
0.11	9.52	0.39	8.90
0.13	8.47, 9.09, 9.23, 9.24		

4.1. Performance for classification problems

We compare the best result of the GSOFM with that of the SOFM for each problem. Moreover, the best result of the GSOFM with respect to the iris data is compared with those of other known unsupervised neural networks models. Good parameter specifications for suggesting the GSOFM to obtain the best result can be found through the following sections.

4.1.1. The iris data

The iris data consists of three classes (class 1: iris setosa; class 2: iris versicolor; class 3: iris virginica) and each class consists of fifty data with four dimensions. Moreover, class 2 overlaps class 3.

The Kohonen layer is implemented by a one-dimensional array. Initial parameter specification including m, η and Λ_i is described as follows:

$$m = 3,$$

$$\eta(0) = 1.0,$$

$$\Lambda_i(0) = 2, \quad 1 \le i \le m,$$

During the training phase, η is gradually decreased by a much smaller and fixed amount (i.e., 0.005) at the end of each iteration. Actually, η will not be decreased below a given value (i.e., 0.05). Similar to Fig. 2, Λ_i is gradually decreased after each of the 100 iterations are executed. In decreasing both η and Λ_i, we follow the principles described in Section 3. We examine the performance by the misclassified number through various k versus ρ (i.e., $0 \le k \le 10, 0.0 \le \rho \le 1.0$), and the best result that the GSOFM can attain is 96.00% (i.e., misclassified number is 6). Simulation results are summarized in Table 1. From this table, we can see that the best result can be obtained for $\rho < 0.4$ by carefully tuning parameters.

Next, we compare the best result of the GSOFM with that of other known neural network models that have been applied on the iris data. These models include the generalized learning vector quantization (GLVQ) [20], the unsupervised fuzzy competitive learning (UFCL) [21], the soft competition scheme (SCS) [6], and the descending fuzzy learning vector quantization (↓FLVQ) [6]. The fuzzy c-means

Table 2
Compare best result of the GSOFM with those of other known unsupervised neural networks

GSOFM (%)	SOFM (%)	LVQ (%)	GLVQ (%)	↓FLVQ (%)	UFCL (%)	SCS (%)	FCM (%)
96.00	88.00	89.33	88.67	88.67	90.00	89.33	91.33

(FCM) [21,7] is also taken into account. From Table 2, we can see that the best result of the GSOFM is superior to those of other unsupervised neural network models.

4.1.2. The appendicitis data

The appendicitis data consists of 106 cases classified into two classes with seven attributes. Initial parameter specification including m, η and Λ_i is described as follows:

$$m = 2,$$

$$\eta(0) = 1.0,$$

$$\Lambda_i(0) = 1, \quad 1 \leq i \leq m,$$

The method for decreasing both η and Λ_i are the same as that used in Section 4.1.1. By carefully tuning values of k and ρ (i.e., $0 \leq k \leq 10, 0.0 \leq \rho \leq 1.0$), the best result that the GSOFM can attain is 86.79% (i.e., misclassified number is 14). We also find that the best result is obtained only when $\rho = 0.07$. On the other hand, the best result for the SOFM is 78.30% (i.e., misclassified number is 23), clearly worse than that of the GSOFM.

4.1.3. The wine recognition data

The wine recognition data, which are the results of a chemical analysis of three types of wines, consists of 178 cases classified into three classes with 13 continuous attributes. Initial parameter specifications, including m, η and Λ_i, and the corresponding decreasing method are used as those described in Section 4.1.2. Using the SOFM, we find the classification result is 92.13% (i.e., misclassified number is 14). By carefully tuning values of k and ρ (i.e., $0 \leq k \leq 10, 0.0 \leq \rho \leq 1.0$), the best result of the GSOFM is 96.63% (i.e., misclassified number is 6). We also find that the best result is obtained only when $\rho = 0.02$. From the viewpoint of the best classification capability, the GSOFM again outperforms the SFOFM.

From the simulation results, we can see that the best classification capability of the SOFM could be enhanced by incorporating grey relations into the learning rule. For applying the neural networks to classification problems, simulation results can thus demonstrate the effectiveness and the feasibility of the GSOFM.

4.2. Performance for the traveling salesman problem (TSP)

The TSP can be stated as follows: "Given N cities, find the shortest path for a salesman so that he can visit all the cities exactly once" [9]. TSP is a combinatorial optimization problem and is known to be NP-complete [3]. In addition to the first successful neural model proposed by Hopfield and Tank [13], other neural network models for solving the TSP have been proposed. Some approaches have been well surveyed and simulated by Aras et al. [3], for example, the guilty net (GN) by Burke and Damany [5], Angeniol et al.'s method (AVL) by Angeniol et al. [2], the KNIES-TSP (KL), and the KNIES-TSP-Global (KG) by Aras et al. [3].

On the other hand, the SOFM can also be used to solve the TSP with various number of output nodes through trial and error. The SOFM could give us a near optimal solution [9]. However, the quality of the solution depends on the number of output nodes. If we do not find an acceptable path after sufficiently long time, then the path is not useful and extra output nodes are added. Since more than one node can be attracted to the same city, it is actually best to have more nodes than cities [12]. The number is usually 2, 3 or 4 times N. The number of output nodes of the GSOFM are thus experimentally set to be three times of the number of cities.

In this section, we employ the GSOFM to solve the TSP to determine its effectiveness. However, it seems that poor results are obtained if we apply the learning algorithm presented in Section 3 to solve the TSP, since it simplifies the lateral feedback mechanism [22]. Thus, it is necessary to incorporate the neighborhood function Ω_{i^*}, which is a type of Gaussian function, around the winning node i^* as Eq. (9):

$$\Omega_{i^*} = \exp(-d_{ii^*}/\sigma), \quad i \in \Lambda_{i^*}, \quad 1 \le i \le m \tag{9}$$

into the learning rule. While, d_{ii^*} is the cardinal distance [1] measured along the ring between the nodes i and i^*:

$$d_{ii^*} = \min\{|i^* - i|, m - |i^* - i|\}, \tag{10}$$

where $|\cdot|$ denotes the absolute value, and i and i^* are actually the labels of the winning node and the node i, respectively. As for σ, it is called the "gain parameter" [3,1], reflecting the scope of the neighborhood [3] and it is decreased at the end of each complete pass by Eq. (11) [1]

$$\sigma(t+1) = \varepsilon\sigma(t), \tag{11}$$

where $0 \le \varepsilon \le 1$. A detailed learning algorithm of the GSOFM for solving the TSP is described as follows.

Algorithm : GSOFM learning algorithm for solving the TSP

Input: Given N cities.

Output: Find the shortest path for a salesman so that he can visit all the cities exactly once.

Method:

Step 1: Initialize connection weights and parameters

a. Initialize weights corresponding to each output node with random small value;

b. Let $\eta(0)=1.0$, and $\Lambda_i(0)=3(m-1)$, where $1 \le i \le m$, i.e., the total number of output nodes is $3m$;
c. Randomize the order of cities and label cities $1,\ldots,N$. The variable r indexes the order of city and set $r=1$, where $1 \le r \le N$. In addition, we assign the label i to the node i, where $1 \le i \le m$;
d. Set $t=1$, where t is an iteration counter.

Step 2: Present the rth city $\mathbf{x}^{(r)}(t)$

Step 3: Calculate the output value $o_i(t)$ of each output node i

$$o_i(t)=(d_i(t))^2=\sum_{j=1}^{n}(w_{ij}(t)-x_j^{(r)}(t))^2, \quad 1 \le i \le m$$

Step 4: Determine the winning node i^*
The node i^* is the winner if

$$(d_{i^*}(t))^2=\min_i o_i(t), \quad 1 \le i \le m$$

Step 5: Adjust the winning nodes i^* and its neighbor nodes
a. The neighbor nodes around the winning node i^* can be determined by $\Lambda_{i^*}(t)$;
b. The learning rule based on $\xi_{ij}(t)$ can be given as

$$w_{ij}(t+1)=w_{ij}(t)+\eta(t)[\xi_{ij}(t)]^k\Omega_{i^*}(t)[x_j^{(r)}(t)-w_{ij}(t)],$$
$$i \in \Lambda_{i^*}(t), \quad 1 \le j \le n \tag{12}$$

where k is a pre-specified positive real number, and $\xi_{ij}(t)$ is the GRC between $x_j^r(t)$ and $w_{ij}(t)$

Step 6: Increment the value of r
If r equals to N, then
a. Shrink the gain parameter $\sigma(t)$ as Eq. (11);
b. shrink the learning rate $\eta(t)$ and the neighborhood's size $\Lambda_i(t)$;
c. Set $t+1$ to t.
Go to Step 7. Otherwise, set $r+1$ to r and go to Step 2.

Step 7: Convergence test
Checking whether or not locations of output nodes are within an acceptable distance of cities. If yes then stop. Otherwise, set $t+1$ and 1 to t and r, respectively, and go to Step 2.

During the training phase, η is gradually decreased by a much smaller and fixed amount (i.e., 0.0005) at the end of each iteration. Actually, η will not be decreased below a given value (i.e., 0.05). We employ the same methods described in Section 4.1.1 to gradually decrease Λ_i during the training phase. For simplicity, we set $\rho=0.5$, which is commonly used in other applications [14], and set $k=1$. Therefore, the initial values of σ and ε are two tunable variables that can determine whether or not the GSOFM can find a high quality solution in convergence. Using

Table 3
Eight problems selected from the TSPLIB

Problems	Number of cities	Optimal length
bier127	127	118282
eil51	51	426
eil76	76	538
eil101	101	629
pr107	107	44303
pr136	136	96772
rd100	100	7910
st70	70	675

Table 4
Compare the best result of the GSOGM with those of the SOFM, GN, AVL, KL and KG

Problems	Optimal lenght	GSOFM ($\sigma(0), \varepsilon$)	SOFM	GN	AVL	KL	KG
bier127	118282	121181.3 (55,0.92)	122211.7	155163.2	122673.9	121548.7	121923.7
eil51	426	437.71 (36,0.91)	443.9	470.7	443.5	438.2	438.2
eil76	538	562.41 (139,0.74)	571.2	614.3	571.3	564.8	567.5
eil101	629	658.04 (85,0.85)	688.7	771.9	671.4	658.3	664.4
pr107	44303	44483.46 (147,0.87)	44504.3	80481.3	45096.4	44628.3	44491.1
pr136	96772	98956.42 (41,0.78)	103878.0	135887.7	103442.3	101156.8	101752.4
rd100	7910	8143.72 (72,0.69)	8137.9	8731.2	8265.8	8075.7	8117.4
st70	675	692.06 (21, 0.52)	692.8	755.7	693.3	685.2	690.7

eight problems shown in Table 3 from the TSPLIB proposed by Reinelt [23], we compare the best solution by the GSOFM with those of other neural network models, including the SOFM, the GN, AVL method, the KNIES-TSP (KL) and the KNIES-TSP-Global (KG), all as reported in [3]. Note that cities in each selected problem are spread on the two-dimensional Euclidean space, while the Euclidean norm is used to compute the distance among any two cities. Simulation results are summarized in Table 4.

In Table 4, the first column shows the testing problem sets that are used in our simulation. The second column shows the known optimal length for these problem sets. The real numbers in the third to eighth column show the best solutions through various parameter specifications obtained by using GSOFM, SOFM, GN, AVL, KL and KG, respectively. In the third column, the numbers in parentheses are parameter specifications (i.e., $\sigma(0)$ and ε) for which the best results can be obtained by the GSOFM.

We find that the GSOFM outperforms the other neural network models for bier127, eil51, eil76, eil101, pr107 and pr136. In the case of st70, the result of the GSOFM is slightly inferior compared with those of KL and KG. Furthermore, to show the relative deviations from the optimal length, we summarize the results in Table 5. From this table, we may thus conclude that the GSOFM outperforms the

Table 5
Deviation from the optimal length of the various algorithms

Problems	GSOFM	SOFM	GN	AVL	KL	KG
bier127	2.45	3.32	31.18	3.71	2.76	3.08
eil51	2.75	4.20	10.49	4.11	2.86	2.86
eil76	4.54	6.17	14.18	6.19	4.98	5.48
eil101	4.62	9.49	22.72	6.74	4.65	5.63
pr107	0.41	0.45	81.66	1.79	0.73	0.42
pr136	2.26	7.34	40.42	6.89	4.53	5.15
rd100	2.95	2.88	10.38	4.49	2.09	2.62
st70	2.53	2.64	11.96	2.71	1.51	2.33
Average	2.81	4.56	27.87	4.58	3.01	3.45

other neural network models. Actually, Tables 3 and 4 show the feasibility and effectiveness of the GSOFM for solving the TSP.

5. Discussions and future works

Since the original SOFM in the training phase ignores important information, which is the relationships that actually exist between the input training data and each adjustable output node, we thus incorporate the grey relational coefficients into the learning rule of the SOFM, namely the GSOFM. The GSOFM can be viewed as a grey clustering method. To show the problem-solving capability of the GSOFM, the performances are examined by complete computer simulations for two representative problems: one is the classification problems, including the iris data proposed by Fisher [11], the appendicitis data and the wine recognition data; and the other is the TSP, selecting from the TSPLIB problem set proposed by Reinelt [23].

In the classification problems, we find that the best result from the GSOFM outperforms that of the SOFM in each problem. Moreover, the best result from the GSOFM with respect to the iris data is compared with those of other known unsupervised neural networks models. Although criteria in selecting a method for classification problems are subjective and dependent on applications, accuracy is always the primary goal [25]. For applying the unsupervised neural networks on classification problems, simulation results thus demonstrate the effectiveness and the feasibility of the GSOFM.

As for the TSP, we selected some problem sets from the TSPLIB to test the performance of the GSOFM. In our simulations, the number of output nodes are experimentally set to be three times of the number of cities. It is furthermore possible to check if the quality of the solution depends on the number of output nodes, especially for the large size problems such as pcb442 and att532 in the TSPLIB. In addition, we can modify the GSOFM to dynamically add nodes in the Kohonen layer to obtain good quality during the training phase.

On the other hand, we are also very interested to use the GSOFM to solve problems encountered in other fields. For example, we may design a framework to integrate the GSOFM with the fuzzy query processing. Previously, Kamel et al. had proposed a clustering method for fuzzy query processing [18] from the viewpoint of enhancing the flexibility of the existing database systems. Kamel et al.'s works provide a good basis for the future integration. Also, the GSOFM could serve as a data mining tool. Previously, the SOFM has been a powerful tool for data mining that can help a business to analyze the characteristics of customers from transaction databases. Therefore, it is possible to apply the GSOFM for knowledge discovery. For example, a large bank could try to understand customers who currently have home equity loans to determine the best strategy for increasing its market share [4].

6. Conclusions

By applying the GSOFM on classification problems and on the TSP, we can see that simulation results demonstrate the effectiveness and feasibility of the GSOFM, and we will continue to study related topics. From the discussions and the future works mentioned above, we can see that it is worthwhile to measure the effectiveness and the feasibility applying the GSOFM to fuzzy query processing and data mining.

Acknowledgements

We are very grateful to the anonymous referees for their valuable comments and constructive suggestions.

References

[1] M. Abdolhamid, S. Samerkae, E. Takao, A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems, Int. Trans. Oper. Res. 6 (6) (1999) 591–606.
[2] B. Angeniol, C. Vaubois, J.Y. LeTexier Jr, Self-organizing feature maps and the traveling salesman problem, Neural Networks 1 (4) (1988) 289–293.
[3] N. Aras, B.J. Oommen, I.K. Altinel, The Kohonen network incorporating explicit statistics and its application to the traveling salesman problem, Neural Networks 12 (9) (1999) 1273–1284.
[4] M. Berry, G. Linoff, Data Mining Techniques: For Marketing, Sales, and Customer Support, Wiley, New York, 1997.
[5] L.I. Burke, P. Damany, The guilty net for the traveling salesman problem, Comput. Oper. Res. 19 (4) (1992) 255–265.
[6] J.C. Bezdek, N.R. Pal, Two soft relatives of learning vector quantization, Neural Networks 8 (5) (1995) 729–743.
[7] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York, 1981.
[8] C.S. Cheng, Y.T. Hsu, C.C. Wu, Grey neural networks, IEICE Transactions on Fundamentals Electronics, Commun. Comput. Sci. E81-A (11) (1998) 2433–2442.
[9] S.T. Clifford, W.C. Siu, New approach for solving the travelling salesman problem using self-organizing learning, in: Proceedings of the IEEE International Conference on Neural Networks, Vol. 5, 1995, pp. 2632–2635.
[10] J.L. Deng, Control problems of grey systems, Systems Control Lett. 1 (5) (1982) 288–294.

[11] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugenics 7 (2) (1936) 179–188.
[12] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the Theory of Neural Computation, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, MA, 1991.
[13] J.J. Hopfield, D.W. Tank, Neural computation of decisions in optimization problems, Biolog. Cybernet. 52 (1985) 141–152.
[14] Y.T. Hsu, C.M. Chen, A novel fuzzy logic system based on N-version programming, IEEE Trans. Fuzzy Systems 8 (2) (2000) 155–170.
[15] Y.P. Huang, C.H. Huang, Real-valued genetic algorithms for fuzzy grey prediction system, Fuzzy Sets Systems 87 (3) (1997) 265–276.
[16] T.L. Huntsberger, P. Ajjimarangsee, Parallel self-organizing feature maps for unsupervised pattern recognition, Int. J. of General Systems 16 (4) (1990) 357–372.
[17] J.S.R. Jang, C.T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice-Hall, Englewood Cliffs, NJ, 1997.
[18] M. Kamel, B. Hadfield, M. Ismail, Fuzzy query processing using clustering techniques, Inform. Process. Manage. 26 (2) (1990) 279–293.
[19] T. Kohonen, Self-Organizing Maps, Springer, Berlin, 1995.
[20] N.R. Pal, J.C. Bezdek, E.C.K. Tsao, Generalized clustering networks and Kohonen's self-organizing scheme, IEEE Trans. Neural Networks 4 (4) (1993) 549–557.
[21] N.R. Pal, J.C. Bezdek, R.J. Hathaway, Sequential competitive learning and the fuzzy c-means clustering algorithms, Neural Networks 9 (5) (1996) 787–796.
[22] A.S. Pandya, R.B. Macy, Pattern Recognition with Neural Networks in C++, CRC Press, Boca Raton, FL, 1996.
[23] G. Reinelt, TSPLIB-a traveling salesman problem library, ORSA J. Comput. 3 (4) (1991) 376–384.
[24] K.Q. Shi, G.W. Wu, Y.P. Hwang, Theory of Grey Information Relation, Chuan Hwa, Taiwan, 1994.
[25] S. M. Weiss, C. A. Kulikowski, Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems, Morgan Kaufmann, CA, 1991.

Yi-Chung Hu received the B.S. degree in information and computer engineering from the Chung Yuan Christian University, Chungli, Taiwan, and the M.S. degree in computer and information science from the National Chiao Tung University, Hsinchu, Taiwan, in 1991 and 1993, respectively. From October 1993 to January 2000, he worked as a research assistant in the Chung-Shan Institute of Science and Technology, Taoyan, Taiwan. He is currently working toward the Ph.D degree in information management at the National Chiao Tung University.

His primary research interests include soft computing, data mining, and multiple criteria decision making.

Ruey-Shun Chen received the Ph.D degree in computer science and information engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 1995, respectively.

He is currently an Associate Professor at the National Chiao Tung University. His research interests include genetic algorithms, reliability, performance evaluation, business networks, and distributed systems.

Yen-Tseng Hsu received the B.S. degree in electrical engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, the M.S. degree in electrical engineering from the National Tsing Hwa University, Hsinchu, Taiwan, and the Ph.D degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1985, 1987, and 1991, respectively.

He is currently a Professor at the National Taiwan University of Science and Technology. His research interests include fuzzy logic, neural networks, grey information theory, and VLSI/FPGA

Gwo-Hshiung Tzeng received the B.S. degree in business management from the Tatung University; Taipei, Taiwan, the M.S. degree in urban planning from Chung Hsing University; Taipei, Taiwan, and the Ph.D. degree in management science from Osaka University, Osaka, Japan, in 1967, 1971, and 1977, respectively.

He was a Research Associate at Argonne National Laboratory from July 1981 to January 1982, a Visiting Professor in the Department of Civil Engineering at the University of Maryland, College Park, from August 1989 to August 1990, a Visiting Professor in the Department of Engineering and Economic System and Operations Research (EESOR), Energy Modeling Forum at Stanford University, from August 1997 to August 1998, and a Professor at Chaio Tung University from 1981 to the present. He is a member of IEEE, IAEE, ISMCDM, and World Transport. He is currently a National Distinguished Chair Professor at the National Chaio Tung University. His research interests include multivariate analysis, routing and scheduling, multiple criteria decision making, and fuzzy theory.