International Journal of Architectural, Civil and Construction Sciences
ISSN: 2415-1734
Vol:2, No:6, 2008

Enhanced Clustering Analysis and Visualization Using Kohonen's Self-Organizing Feature Map Networks

Kasthurirangan Gopalakrishnan, Siddhartha Khaitan and Anshu Manik

***Abstract*—**Cluster analysis is the name given to a diverse collection of techniques that can be used to classify objects (e.g. individuals, quadrats, species etc). While Kohonen's Self-Organizing Feature Map (SOFM) or Self-Organizing Map (SOM) networks have been successfully applied as a classification tool to various problem domains, including speech recognition, image data compression, image or character recognition, robot control and medical diagnosis, its potential as a robust substitute for clustering analysis remains relatively unresearched. SOM networks combine competitive learning with dimensionality reduction by smoothing the clusters with respect to an *a priori* grid and provide a powerful tool for data visualization. In this paper, SOM is used for creating a toroidal mapping of two-dimensional lattice to perform cluster analysis on results of a chemical analysis of wines produced in the same region in Italy but derived from three different cultivators, referred to as the "wine recognition data" located in the University of California-Irvine database. The results are encouraging and it is believed that SOM would make an appealing and powerful decision-support system tool for clustering tasks and for data visualization.

***Keywords*—**Artificial Neural Networks, Cluster Analysis, Kohonen Maps, Wine Recognition.

I. Introduction

ARTIFICIAL Neural Networks (ANNs) have been found to be powerful and versatile computational tools for organizing and correlating information in ways that have proved useful for solving certain types of problems too complex, too poorly understood, or too resource-intensive to tackle using more-traditional computational methods [1]. In recent years, Artificial Neural Networks (ANNs) are increasingly being used to solve pavement engineering problems which deal with highly non-linear functional approximations [2]. The main advantages of ANN are nonlinearity, low influence of noise and ability to generalize solved problems [3] and they appear to be complementary to classical statistical approaches, such as principal component analysis, cluster analysis, linear discriminant analysis, etc. [3]. ANN can be used as an unsupervised technique to detect the latent structure of data, but likewise as a supervised technique to classify samples. In this sense, they have been used to classify vinegars [4]–[6], wines [8]–[10] and wine distillates [11].

Kohonen Self-Organizing Feature Maps (SOFMs) or Self-Organizing Maps (SOMs) are the special type of neural networks [12], which provides projection of multidimensional data into one-, two- or in special cases into three-dimensional space. It was designed specially for clustering, visualization and abstraction [13]. The fundamental of the SOM is the soft competition between the nodes in the output layer; not only one node (the winner) but also its neighbors are updated. The Kohonen network can learn to recognize clusters of data and relate similar classes to one another. SOM has only two layers: the input layer and the output layer. The input layer is one-dimensional, while the output layer consists of radial units typically organised in two dimensions (see Fig. 1 [13]).

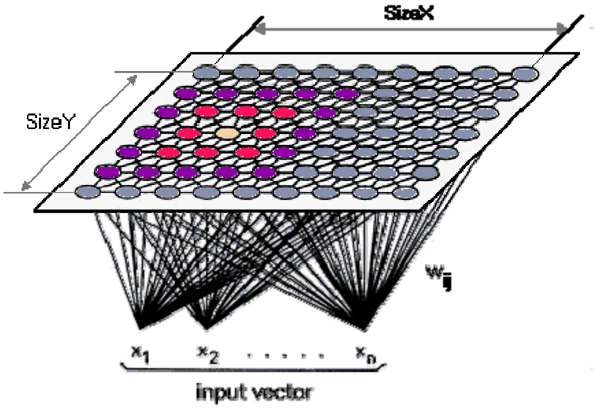

Fig. 1 Schematic of Kohonen's self-organizing map [13]

SOM adapt to the training data in a way such that a high-dimensional-input space is reflected to a 2-dimensional grid on the generated Kohonen-map [14]. By preserving the neighborhood on the topology of the map, structures of the input space can be discovered by exploring the feature map. If

Manuscript received March 9, 2007.
Dr. Kasthurirangan Gopalakrishnan is with the Department of Civil Engineering, Iowa State University, Ames, IA 50011, USA (e-mail: rangan@iastate.edu).
Mr. Siddhartha Khaitan is with the Department of Electrical Engineering, Iowa State University, Ames, IA 50011, USA (e-mail: skhaitan@iastate.edu)
Dr. Anshu Manik is with the Department of Civil Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA (e-mail: manik@uiuc.edu).

the property of topology preservation holds, then similar input data should correspond to data on the Kohonen lattice with the same close relationship.

As cluster analysis groups similar objects into adjunct subsets called clusters [15], cluster in the $\Re^n$ should therefore also appear in the lattice of lower dimensionality. At this point, the SOFM arranges clusters into different regions of the lattice. As the distances among the data points are evenly distributed, no clustering would be detected without previous knowledge of the original membership [16].

Cluster analysis is an exploratory data analysis tool for solving classification problems. The domination of statistical clustering methods in the field of complex data analysis has been very stable over the past. Many clustering algorithms have been developed in this domain, as described in man books and articles such as Hartigan [17], Späth [18], Andernberg [19], and McLachlan and Basford [20]. Two of the most commonly used clustering algorithms are the hierarchical clustering and the *k-means* method. The Kohonen map is, at heart, *k-means* clustering with the additional constraint that cluster centers be located on a regular grid (or some other topographic structure) and furthermore their location on the grid be monotonically related to pairwise proximity [21].

The SOM training process starts with a random set of radial centers in the output layer and during training they are adjusted to cluster the training data. The algorithm is based on adjusting the winning neuron (neuron with the output nearest to the input) to be more like the input case. The topological ordering property is achieved by adjusting weights of neighbourhood neurons to bring their output closer to the winning neuron's output and decreases with the topological distance. Usually the distance at which the changes take place decreases during the training process. Similarly, the learning rate is higher at the start and decreases during adaptation. When the network is trained, it can be used for visualization and for classification. The most similar samples are in the same cell or in neighbourhood cells. The weights show the reason for clustering and similarity of objects [3].

While Kohonen's SOM networks have been successfully applied as a classification tool to various problem domains, including speech recognition [22], image data compression [23], image or character recognition [24]-[25], robot control [26]-[27] and medical diagnosis [28], its potential as a robust substitute for clustering analysis remains relatively unresearched [29].

Cluster analysis is a technique for grouping subjects into clusters of similar elements. In cluster analysis, similar elements are identified by their attributes. Groups, or clusters are formed, that are homogeneous and different from other groups. The object of cluster analysis is to sort cases (people, things, events, etc) into groups, or clusters, so that the degree of association is strong between members of the same cluster and weak between members of different clusters. Each cluster thus describes, in terms of the data collected, the class to which its members belong; and this description may be abstracted through use from the particular to the general class or type.

SOM networks combine competitive learning with dimensionality reduction by smoothing the clusters with respect to an a priori grid and provide a powerful tool for data visualization [29]. In this paper, SOM is used to perform cluster analysis on results of a chemical analysis of wines produced in the same region in Italy but derived from three different cultivators, referred to as the "wine recognition data" [30].

II. Theory

The SOM defines a mapping from the input data space spanned by $x_1 \ldots x_n$ onto a one- or two-dimensional array of nodes. The mapping is performed in a way that the topological relationships in the *n*-dimensional input space are maintained when mapped to the SOM. In addition, the local density of data is also reflected by the map: areas of the input data space which are represented by more data are mapped to a larger area of the SOM [13].

Each node of the map is defined by a vector w_{ij} whose elements are adjusted during the training. The basic training algorithm is quite simple:

- select an object from the training set
- find the node which is closest to the selected data (i.e. the distance between w_{ij} and the training data is a minimum)
- adjust the weight vectors of the closest node and the nodes around it in a way that the w_{ij} move towards the training data
- repeat from step 1 for a fixed number of repetitions

The amount of adjustment in step 3 as well as the range of the neighborhood decreases during the training. This ensures that there are coarse adjustments in the first phase of the training, while fine tuning occurs during the end of the training [13].

The node with the minimum distance is the winner and adjusts its weights to be closer to the value of the input pattern. For the sake of illustration, Fig. 2 displays a 4 x 4 Kohonen layer and shows how the neighboring nodes are defined with a radial distance $r = 1$. In this study, Euclidean distance which is the most common way of measuring distance between vectors, is used.

Researchers have proposed a Gaussian type of neighborhood adaptation function, which decreases both in the spatial domain and the time domain [31]-[33]. Lo and Bavarian [33] have shown that an algorithm that uses the Gaussian type function will enforce ordering in the neighborhood set for every training iteration, yielding faster convergence. Lo and Bavarian [33] modified Kohonen's adaptation rule to include the amplitude of neighborhood adaptation $A_i(t)$ as follows:

$$w_i(t+1) = w_i(t) + \alpha(t)A_i(t)[w_i(t) - x_v] \; for \; i \in N_i(t), 0 \le r \le R,$$
$$w_i(t+1) = w_i(t) \quad otherwise. \tag{1}$$

Mitra and Pal [34] proposed a Gaussian type neighborhood adaptation function *h(t,r)* which decreases both in spatial and time domains:

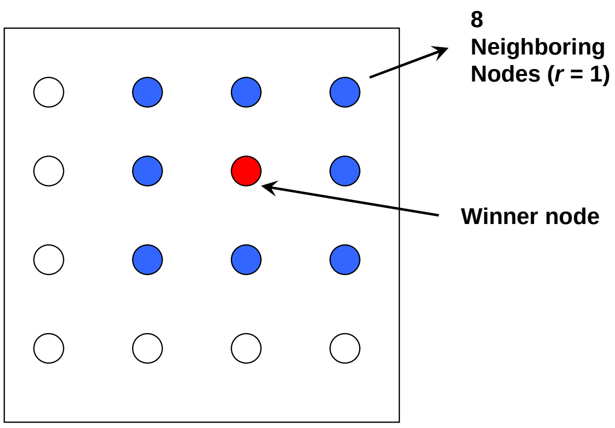

Fig. 2 A 4 × 4 Kohonen layer and definition of neighboring nodes (after [29])

$$h(t,r) = \frac{\alpha(1-rf)}{[1+(t/cdenom)^2]} \tag{2}$$

where r is the radial distance from the winner node i. Nodes within a radius R are considered for adaptation at time t. Parameter α determines the initial value of $|h|$. The parameter $f(0 < f < 1/r)$ determines the rate of decrease of $|h|$ in the spatial domain. In the time domain, t controls the value of $|h|$ whereas the parameter *cdenom* determines the rate of its decay [29].

In general, the farther a node is from the winner, the lower is the amplitude A_i and hence the lower is the update rate of the node's weight vector [29]. The network undergoes a self-organization process through a number of training cycles, starting with randomly chosen w_i's. The state of the network is denoted by the mean-squared-distance, *msd*, between the input vectors and the weight vectors of the nodes in the set N_i:

$$msd = \frac{1}{|trainset|} \sum_{x \in trainset} \left[\sum_{r=0}^{R} \left\{ \left(\frac{1}{|N_r|} \sum_{i \in N_r} \|x - m_i\|^2 \right) (1-rf) \right\} \right] \tag{3}$$

where *trainset* is the training set number and N_r is the number of the set of nodes that are distance r away from the winner node. Note that the nodes closer to the winner node contribute more to *msd*.

In a self-organized map, a few nodes may end up representing too much of the input data due to the effect of the initial random weight values assigned to them. In order to avoid this, DeSieno [35] proposed a "conscience" mechanism the purpose of which is to give each node in the Kohonen layer an opportunity to represent approximately equal information about the input data. The conscience mechanism adjusts the Euclidean distance between a node's weight vector and the input vector $||x_v - w_i||$ by a bias B_i. B_i is proportional to the difference between the node's winning frequency and the average winning frequency:

$$B_i = \gamma \left(\frac{1}{N} - F_i \right) \tag{4}$$

where F_i is the winning frequency of node i and is updated at every iteration of the training process. Initially, F_i is assigned the average value $1/N$; thus $B_i = 0$. The γ coefficient starts at a large value and decreases over time. The winning frequencies are updated as (where β is a small positive fraction):

$$for \; the \; winning \; node: F_{i,t+1} = F_{i,t} + \beta(1.0 - F_{i,t}),$$
$$for \; all \; other \; nodes: F_{i,t+1} = F_{i,t} + \beta(0.0 - F_{i,t}) \tag{5}$$

When the number of clusters desired is different from the number of nodes on the SOM output map, additional steps are required to analyze and group the points on the output map into the desired number of clusters [29]. Merkl and Rauber [36] proposed the Growing-Hierarchical Self-Organizing Map (GH–SOM), a neural network model based on Kohonen SOM network. GH–SOM can grow both in map size and into a three-dimensional tree structure to reflect any hierarchical structure hidden in the underlying data set. The input data are shown in increasingly 6ner levels of detail along the hierarchy defined by the tree structure. The advantage of their approach is the ability of the output to represent hierarchical structure, if exists, in the data set. However, the added growing capability of the network also increased the computational complexity of the algorithm hence is not as efficient as the original SOM network.

Kiang [29] extended the model by adding a separate clustering process that takes the output generated by the SOM network to arrive at the desired number of clusters. Moreover, although the tree structure map allows user to visualize if any hierarchical structure exists in the data set, it does not show the relationships among the border nodes that reside in neighboring groups [29]. Vesanto and Alhoniemi [37] also proposed a way to cluster output from SOM. The process starts with a visual inspection of the output map. Then, both agglomerative and partitive clustering algorithms were applied [29]. The focus of Vesanto and Alhoniemi's research is on comparing the computational efficiencies of different approaches whereas ours is to introduce the extended SOM network as an alternative clustering tool and compare the performance with those of other popular statistical clustering techniques [29].

To automate the segmentation process to complement the usage of the Kohonen SOM networks, Murtagh [38] proposed an agglomerative contiguity-constrained clustering method. The method groups the output from SOM based on a minimal distance criterion to merge the neighboring nodes together. The rationale is that the SOM networks will maintain the original topological relations; therefore the nodes that are closely located on the representational grid should have similar cluster centers [29].

Kiang [29] proposed and implemented a contiguity-constrained grouping algorithm based on a minimal variance criterion that is a better received approach known in traditional statistical clustering methods. In his method, Kiang [29] starts with each node in the map representing one group, and calculates the centroid of each group. Then he tries to merge two neighboring groups so the result of the merge will maintain the global minimal variance for that number of clusters. The merge process is repeated until a user speci6ed number of clusters has been derived or when only one cluster remains. The detailed process is described as follows [29]:

STEP 1: For each $node_i$, calculate the centroid (C_i) of nodei as:

$$C_i = \frac{1}{node_i} \sum_{x \in node_i} \vec{x} \tag{6}$$

where $|node_i|$ is the number of input vectors associated with the node.

STEP 2: Assign a group number (G_k) to each $node_i$ if $|node_i| > 0$, and update the corresponding centroid value G_k.

STEP 3: Calculate the overall variance of the map:

(a) Sum the square distance between input vector x and the group centroid C_k for all x in G_k. Calculate for every group k

$$V_k = \sum \|x - C_k\|^2, \; x \in G_k \tag{7}$$

(b) Total the variances from all groups. This will give the global variance of the map:

$$V_{Total} = \sum V_k \tag{8}$$

STEP 4: For each pair of neighboring groups, calculate the total variance of the map if the two groups were merged. Merge the two groups that result in the minimum global variance.

(a) Calculate the new centroid for G_{pq} if G_p and G_q were merged:

$$G_{pq} = \frac{\left(|G_p|\vec{G}_p + |G_q|\vec{G}_q\right)}{|G_p| + |G_q|} \tag{9}$$

(b) Calculate the new variance if G_p and G_q were merged (modified from Murtagh [38]):

$$V_{pq} = \sum \|x - C_{new}\| \; for \; all \, x, \; x \in G_p \; or \; x \in G_q \tag{10}$$

(c) Calculate the new global variance for merging G_p and G_q:

$$V_{pqTotal} = V_{Total} + V_{pq} - V_p - V_q \tag{11}$$

(d) Calculate the $V_{pqTotal}$ for every pair of p and q on the map. For each iteration, groups p and q must be within a fixed radial distance on the grid. Start with radial distance = 1, hence for each node there are eight neighboring nodes within that distance (see Fig. 2). Increase the radial distance by one each time if there is no neighboring group within current radial distance for all groups k. Finally, merge the two groups that result in global minimal variance.

(e) Update V_{Total} and the group number and group centroid of the two newly merged groups.

STEP 5: Repeat STEP 4 until only one cluster or the pre-specified number of clusters has been reached.

III. Problem

The problem in this study is to train a Kohonen map to classify and identify chemical analysis data. A Kohonen SOM was employed which seeks to transform an incoming signal pattern of arbitrary dimension into one- or two-dimensional discrete map, and performs this transformation adaptively in a topological ordered fashion. The architecture generally consists of a two-dimensional array of linear units (neurons), where each neuron receives the same set of input (sensory) signals ($\mathbf{X}^k \in \Re^n$).

The data for this project are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars (3 classes). Kohonen SOM was used to study the latent structure of the data and classify samples according to the geographical origin of the wines.

The analysis determined the quantities of 13 constituents ($\Re^{13}$) found in each of the three types of wines. These features are:

1) Alcohol
2) Malic acid
3) Ash
4) Alkalinity of ash
5) Magnesium
6) Total phenols

7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10) Color intensity
11) Hue
12) OD280/OD315 of diluted wines, and
13) Proline.

The wine recognition data is located in the UC-Irvine database in */pub/machine-learning-databasees/wine*. It is recognized that there are only 178 data vectors which is too small to produce a highly reliable result. All attributes are continuous and have no missing value. It is also suggested that the attributes be standardized for classifiers that are not scale invariant. All 13 attributes were normalized to have a standard deviation of 1.0. Figs. 3 and 4 display the 178 data vectors and their values in 13-dimensional space before and after normalization, respectively.

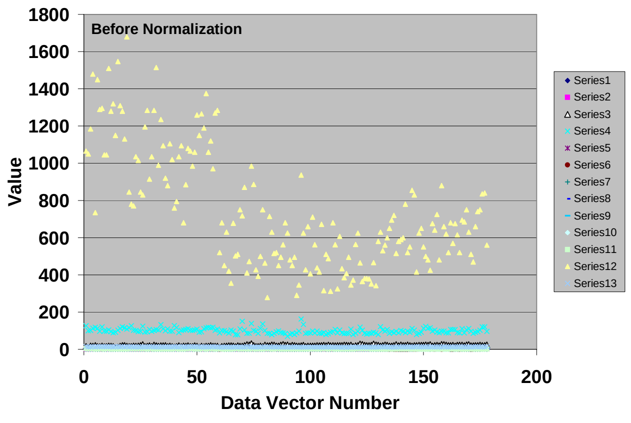

Fig. 3 Wine recognition data before normalization

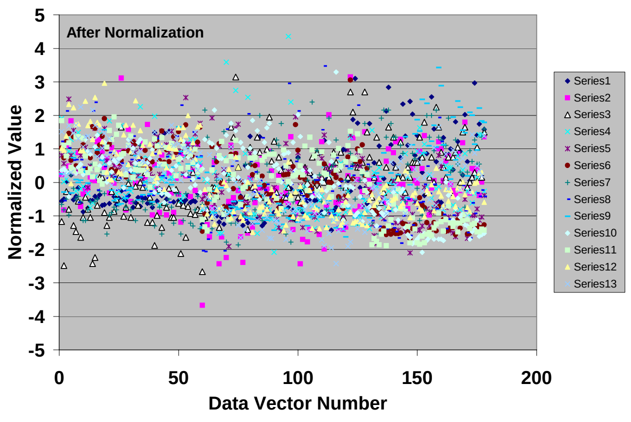

Fig. 4 Wine recognition data after normalization

IV. Methodology

The *Visual Basic* (VB6.0) programming language was used in solving this problem. A two-dimensional square lattice (array) of dimension *5x5* was used for this case. The wine data suggests that the means and variances of the attributes (features) are wildly different. This will cause the distance measures computed between such vectors to overemphasize the features with large means and/or variances. Also, no specific information about the relative importance of various features is known. Therefore, it was decided to give equal emphasis to all the features and as a result, each feature was reduced to zero mean and unity variance (i.e. column-wise).

Each unit (neuron) in the Kohonen map (array) is characterized by an n-dimensional weight vector. In this case, n = 13. The j-th unit weight vector W_j is sometimes viewed as a position vector that defines a "virtual position" for unit j in $\Re^n$. Thus, changes in W_j can be interpreted as "movements" of unit j.

The first step in forming a Kohonen map involves the initialization of the weight vectors of all units to random values. The only restriction here is that the W_j (t = 0) be different for j = 1, 2, …N, where N is the number of neurons in the lattice. Here, N = 5x5 = 25. It is desirable to keep the initial weights distributed inside a small interval. For this problem, the weights were initialized randomly in the range *(-0.1, +0.1)*.

Once the codebook (weight) vectors are initialized, a sample X_i from the input distribution is drawn. In this case, instead of selecting the input data vectors randomly, the same order of presentation, as present in the data file was used at every epoch.

The best-matching (winning) neuron j^* for the input data vector X_i is determined by using the minimum-distance Euclidean criterion:

$$j^* = \arg\min_j \| X_i - W_j \|, \quad j = 1,2,\ldots,N \tag{12}$$

Thus, the winner node is the node with the weight vector closest (in a Euclidean distance sense) to the input vector. The synaptic weight vectors of all neurons were adjusted using the following update formula:

$$W_j(t+1) = \begin{Bmatrix} W_j(t) + \eta(t)\left[X_i - W_j(t)\right], & j \in \Lambda_{j^*}(t) \\ W_j(t), & otherwise \end{Bmatrix} \tag{13}$$

where $\Lambda_{j^*}(t)$ is the neighborhood function around the winning neuron j^* at time t.

Self-organization learning consists of adaptively modifying the synaptic weights of a network of locally interacting units in response to input excitations and in accordance with a learning rule until a final useful configuration develops. The local interaction of units implies that the changes in the behavior of a unit only (directly) affect the behavior of its immediate neighborhood. Careful consideration has to be given to the neighborhood function for topological ordering of the weight vectors W_j to take place. A *square neighborhood* *function* was used for this small map size (5x5). With a square neighborhood function, a neighborhood "radius" of one includes the winning neuron plus the eight neighbors. A neighborhood radius of zero includes just the winning neuron.

Also, a *toroidal* mapping of the two-dimensional array was produced by making the vertical edges coincide and the horizontal edges touch one other. This defines the topological adjacency of neurons in a better manner and also eliminates the "edge effects". Thus, a corner-most neuron will have the same number of weight updates as that of any other neuron in the network at a given time.

V. Results

The neighborhood function usually begins such that it includes all neurons in the network and then gradually shrinks with time. For this problem, with 5x5 Kohonen map, an initial neighborhood radius of 2 was used which was then decreased in steps. The rate at which it should be decreased could be obtained by studying the Sum of the Squared Error (SSE) over an epoch. After a few simulation runs, it was decided to reduce the neighborhood radius in the following way: reduce the NBRHD_RADIUS (*r*) from 2 to 1 after 50 epochs and reduce the NBRHD_RADIUS from 1 to 0 after 100 epochs.

It is normally suggested that the learning-rate parameter $\eta(t)$ used to update the synaptic weight vector $W_j(t)$ should be time-varying. However, for sake of simplicity, the learning-rate parameter η was set to a constant value of ***0.01***. This value provided a good learning curve which was neither too steep nor oscillatory.

The above procedure starting from sampling to updating of weights is repeated until convergence is achieved. In this case, convergence was defined as reaching approximate stability of the Sum of the Squared Error (SSE) over successive passes (epochs) through the data set:

$$[SSE]_{EPOCH} = \sum_{i=1}^{178} \left\| X_i - W_{j^*} \right\|^2 \tag{14}$$

where X_i is one of the 13-dimensional input vectors and W_{j^*} is the weight vector of the winning node j^* for X_i. By trial and error, it was found that the SSE reached an approximate stability around 250 epochs. Therefore, the number of epochs used to reach convergence is ***250***. Fig. 5 displays the monotonically decreasing trend of the SSE with respect to the number of epochs, which is a prominent feature of the Kohonen map algorithm. The points at which the curve shifts (i.e. at 50 epochs and 100 epochs) are the points at which the neighborhood radius was decreased.

After the training converged, the probability, $P(C_j \mid Node_k)$ at each node was calculated using the desired class information provided in the input data file. The term, $P(C_j \mid Node_k)$ means that given a node k, what is the probability that it "represents" class C_j. In this case, there are 3 distinct classes. Therefore, $P(C_1 \mid Node_k) + P(C_2 \mid Node_k) + P(C_3 \mid Node_k)$ should equal 1 assuming that $Node_k$ is used at all. The final results are shown in the form of a color-coded bar graph in Fig. 6. Fig. 7 displays the final K-map diagram labeled with

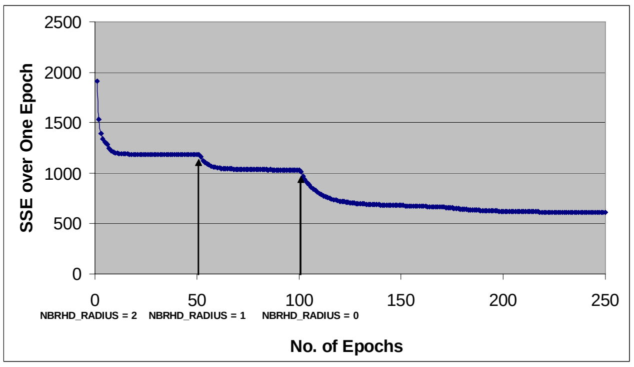

Fig. 5 Learning curve (5 x 5 map)

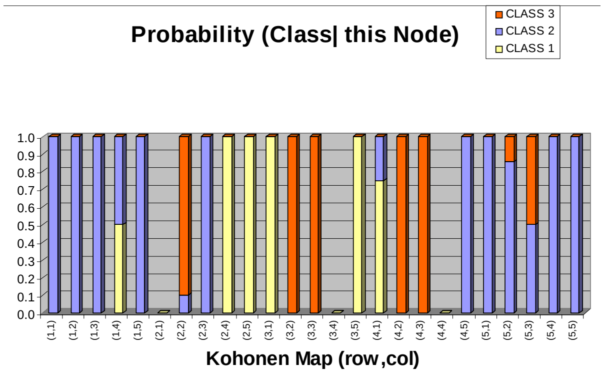

Fig. 6 Color-coded bar chart depicting the final K-map after calibration (5 x 5 map)

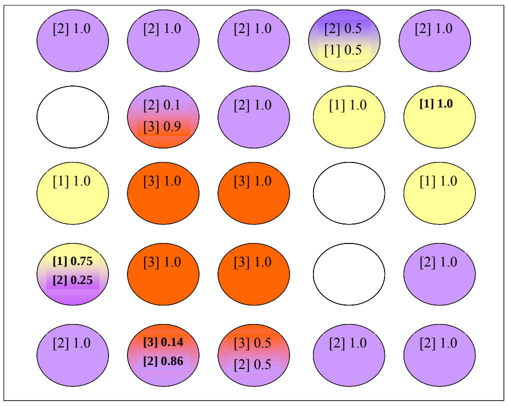

Fig. 7 Final K-map diagram labeled with the probability values

the probability of the various classes that map to the individual nodes. The summary of main parameters used in solving the problem is stated in Table 1. It can be seen from Fig. 5 that the classes 1 and 2 are predominantly represented by the edge rows and edge columns whereas class 3 is represented by the interior nodes.

TABLE I
SUMMARY OF MAIN PARAMETERS (5 X 5 MAP)

Parameter	Property
Kohonen map (toroid)	Dimension: ***5x5***
$\mathbf{X}_i$ (reduced input data vector)	Dimension: ***13***
$\mathbf{W}_j$ ($t = 0$) [initial weight vector]	Random values in the Interval: ***(-0.1,+0.1)***
Neighborhood Function	***Square*** (Radius: ***2, 1, 0***)
η (learning-rate paramater)	***0.01***

TABLE II
SUMMARY OF MAIN PARAMETERS (6 X 6 MAP)

Parameter	Property
Kohonen map (toroid)	Dimension: ***5x5***
$\mathbf{X}_i$ (reduced input data vector)	Dimension: ***13***
$\mathbf{W}_j$ ($t = 0$) [initial weight vector]	Random values in the Interval: ***(-0.1,+0.1)***
Neighborhood Function	***Square*** (Radius: ***2, 1, 0***)
η (learning-rate paramater)	***0.01***

To investigate further, a 6 x 6 Kohonen network was used to study the wine recognition (chemical analysis) data. In Fig. 8, the monotonically decreasing trend of the SSE with respect to the number of epochs is displayed for the 6 x 6 network. Fig. 9 displays the color-coded bar chart depicting the final K-map after calibration for the 6 x 6 network. The summary of main parameters used in the 6 x 6 network is displayed in Table 2.

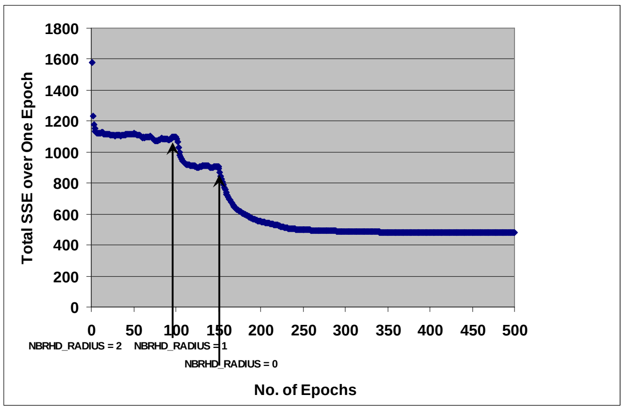

Fig. 8 Learning curve (6 x 6 map)

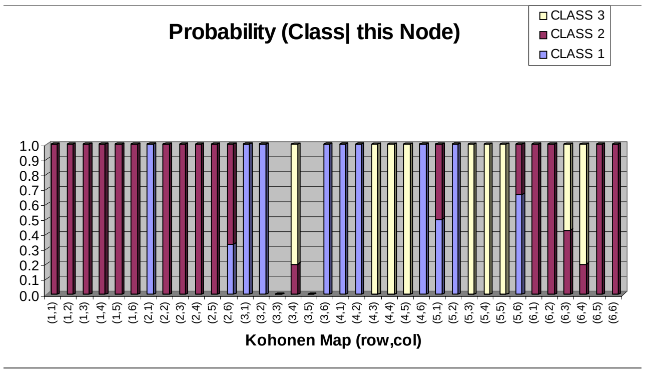

Fig. 9 Color-coded bar chart depicting the final K-map after calibration (6 x 6 map)

REFERENCES

[1] TR Circular, "Use of Artificial Neural Networks in Geomechanical and Pavement Systems", *Transportation Research Circular No. E-C012*, TRB, Washington, DC, 1999.
[2] K. Gopalakrishnan, M.R. Thompson, and A. Manik, "Rapid Finite-Element Based Airport Pavement Moduli Solutions using Neural Networks", *Int J. Comp. Intelligence*, 3 (1), 2006, pp. 63-71.
[3] S. Frias, J.E. Conde, M.A. Rodriguez, V. Dohnal, and J.P. Perez-Trujillo, "Metallic content of wines from the Canary Islands (Spain). Application of artificial neural networks to the data analysis", *Nahrung/Food*, 46(5), 2002, pp. 370-375.
[4] SM.J. Benito, M.C. Ortiz, M. Sagrario, L. Sarabia, and M. Iniguez, *Analyst*, 1999, 124, pp. 547–552.
[5] M.C. Garcia-Parrilla, G.A. Gonzalez, F.J. Heredia, and A.M. Troncoso, *J. Agric. Food Chem.*, 1997, 45, pp. 3487–3492.
[6] M.C. Garcia-Parrilla, F.J. Heredia, A.M. Troncoso, *Food Res. Int.*, 1999, 32, pp. 433–440.
[7] M.C. Ortiz, A. Herrero, M.S. Sanchez, L. Sarabia, and M. Iniiguez, *Chemom. Intell. Lab. Syst.*, 1995, 28, pp. 273–285.
[8] S. Vlassides, J.G. Ferrier, D.E. Block, *Biotechnol. Bioeng.*, 2001, 73, pp. 55–68.
[9] E. Marengo, M. Aceto, V.J. Maurino, *Chromatogr. A*, 2001, 94, pp. 123–137.
[10] L.X. Sun, K. Danzer, G. Thiel, J. Fresenius, *Anal. Chem.*, 1997, 359, pp. 143–149.
[11] C.G. Raptis, C.I. Siettos, C.T. Kiranoudis, G.V. Bafas, *J. Food Eng.*, 2000, 46, pp. 267–275.
[12] T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps", *Biological Cybernetics*, Vol. 43, 1982, pp. 59–69.
[13] SDL Component Suite, "Kohonen Network - Background Information", http://www.lohninger.com/helpcsuite/kohonen_network_-_background_information.htm. Accessed online March 13, 2007.
[14] T. Kohonen, *Self Organization and Associative Memory*, Springer Verlag, Berlin, 1989.
[15] G. Deichsel and H.J. Trampisch, *Clusteranalyse und Diskriminanzanalyse*, Gustav Fischer Verlag, Stuttgart, New York, 1985.
[16] A. Ultsch and C. Vetter, "Self-Organizing-Feature-Maps versus Statistical Clustering Methods: A Benchmark", Research Report 0994, FG Neuroinformatik & Künstliche Intelligenz, University of Marburg, Denmark, 1995.
[17] .A. Hartigan, *Clustering Algorithms*, Wiley and Sons, New York, 1975.
[18] H. Späth, *Cluster Analysis Algorithms*, Chichester, UK, 1980.
[19] M.R. Andernberg, *Cluster Analysis for Applications*, New York, Academic Press, 1973.
[20] G.J. McLachlan and K.E. Basford, *Mixture Models*, New York: Marcel Dekker, Inc., 1988.

[21] Murtagh F. and Hernández-Pajares M. (1995). The Kohonen Self-Organizing Map Method: An Assesment. Journal of Classification, **12**, 165-190.

[22] L. Leinonen, T. Hiltunen, K. Torkkola, and J. Kangas, "Self-organized acoustic feature map in detection of fricative-vowel coarticulation", *J. Acoust. Soc. Am.*, 93 (6), 1993, pp. 3468–3474.

[23] C.N. Manikopoulos, "Finite state vector quantisation with neural network classification of states", *IEEE Proc.-F*, 140 (3), 1993, pp. 153–161.

[24] A.D. Bimbo, L. Landi, S. Santini, "Three-dimensional planar-faced object classification with Kohonen maps", *Opt. Eng.*, 32 (6), 1993, pp. 1222–1234.

[25] M. Sabourin, A. Mitiche, "Modeling and classi6cation of shape using a Kohonen associative memory with selective multiresolution", *Neural Networks 6*, 1993, pp. 275–283.

[26] J.A. Walter, K.J. Schulten, "Implementation of self-organizing neural networks for visuo-motor control of an industrial robot", *IEEE Trans. Neural Networks 4 (1)*, 1993, pp. 86–95.

[27] H. Ritter, T. Martinetz, K. Schulten, "Topology-conserving maps for learning visuo-motorcoordination", *Neural Networks 2*, 1989, pp. 159–168.

[28] L. Vercauteren, G. Sieben, M. Praet, G. Otte, R. Vingerhoeds, L. Boullart, L. Calliauw, H. Roels, "The classi6cation of brain tumours by a topological map", in *Proc. of the International Neural Networks Conference*, Paris, 1990, pp. 387–391.

[29] M. Y. Kiang (2001). "Extending the Kohonen self-organizing map networks for clustering analysis", Computational Statistics & Data Analysis, Vol. 38, pp. 161-180.

[30] UCI Machine Learning Repository, Wine recognition data, ftp://ftp.ics.uci.edu/pub/machine-learning-databases/wine/, Apr 16th, 2005.

[31] M. Cottrell and J.C. Fort, "A stochastic model of retinotopy: a self-organizing process", *Biol. Cybern.*, 53, 1986, pp. 405–411.

[32] H. Ritter and K. Schulten, "On the stationary state of Kohonen's self-organizing sensory mapping", *Biol. Cybern.*, 54, 1986, pp. 99–106.

[33] Z.-P. Lo and B. Bavarian, "On the rate of convergence in topology preserving neural networks", *Biol. Cybern.*, 65, 1991, pp. 55–63.

[34] S. Mitra and S.K. Pal, "Self-organizing neural network as a fuzzy classifier", *IEEE Trans. Systems, Man, Cybernetics*, 24 (3), 1994, pp. 385–399.

[35] D. DeSieno, "Adding a conscience to competitive learning", in *Proc. of the International Conference on Neural Networks*, Vol. I, IEEE Press, 1988, New York, pp. 117–124.

[36] D. Merkl and A. Rauber, "Uncovering the hierarchical structure of text archives by using an unsupervised neural network with adaptive architecture", *PADKK*, LNAI 1805, 2000, pp. 384–395.

[37] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map", *IEEE Trans. Neural Networks*, 11 (3), 2000, pp. 586–600.

[38] F. Murtagh, "Interpreting the Kohonen self-organizing feature map using contiguity-constrained clustering", *Pattern Recognition Lett.*, 16, 1995, pp. 399–408.