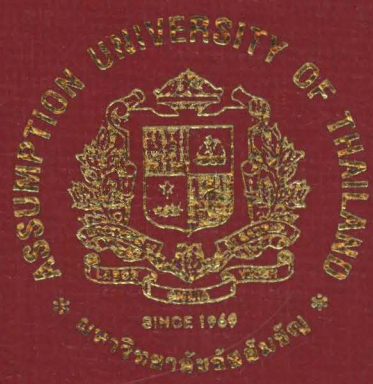


A Self-Growing Approach for Self-Organizing Map with Automatic Stopping Condition

By
Mr. Se Won Kim

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in Computer Science
Assumption University

March, 2013

A Self-Growing Approach for Self-Organizing Map with Automatic Stopping Condition

By

Mr. Se Won Kim

Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science Assumption University

March, 2013

The Faculty of Science and Technology

Master Thesis Approval

Thesis Title	A Self-Growing Approach for Self-Organizing Map with Automatic Stopping Condition
By	Mr. Se Won Kim
Thesis Advisor	Asst. Prof. Dr. Tang Van To
Academic Year	2/2012

The Department of Computer Science, Faculty of Science and Technology of Assumption University has approved this final report of the **twelve** credits course. **SC7000 Master Thesis**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Approval Committee:

(Asst. Prof. Dr. Tang Van To)
Advisor

(Dr. Songsak Channarukul)
Committee Member

(Asst.Prof. Dr. Thitipong Tanprasert)
Committee Member

(Asst.Prof.Dr. Pirawat Watanapongse)
Commission of Higher Education
University Affairs

Faculty Approval:

(Dr. Songsak Channarukul)
Program Director

(Asst. Prof. Dr. Jirapun Daengdej)
Dean

March 2013

TABLE OF CONTENTS

TABLE OF CONTENTS

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor Asst. Prof. Dr. Tang Van To for suggesting this interesting research topic and for his guidance, motivation and patience throughout the course of this thesis. Without his support and encouragement, this thesis would not have been possible.

I am also very grateful to Asst. Prof. Dr. Thitipong Tanprasert for his suggestions and support given to me in the process of preparing a conference paper and getting the paper accepted in the 5th International Conference on Knowledge and Smart Technologies.

I would like to thank the committee members, Asst. Prof. Dr. Thitipong Tanprasert, Dr. Songsak Channarukul, and Dr. Pirawat Watanapongse, Representative of the Higher Education Commission (HEC) for their valuable time spent on my thesis and insightful comments.

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

ABSTRACT

In this thesis, a model of self-growing and self-organizing feature map is proposed, which has been designed to alleviate the difficulty of pre-specifying an appropriate choice of the network topology, i.e., size and shape of the feature map, suitable for a given data set in the application of the Self-Organizing Map.

The proposed model progressively builds a feature map by incremental growing of the network during the training process in a way that maintains two-dimensional regular grid structure compatible with the Kohonen network and by gradual adaptation of the reference vectors through coordinated competitive learning dynamics of the Batch Map algorithm.

Experimental results based on iris data set and Italian olive oil data set show that the stopping criteria applied in the proposed model is effective in discovering an appropriate topology of the network suitable for the data manifold at hand while avoiding undergrowth and overgrowth, and that the proposed model is capable of growing the network dynamically during the training process to manifest a feature map of suitable size and shape for a given data set in significantly less time than the time taken by the standard SOM algorithm to produce a feature map of similar quality on a predetermined network structure.

CHAPTER 1
INTRODUCTION

1.1 Statement of the problem

Kohonen's Self-Organizing Map (SOM) [Teuvo, 2001], also known as Kohonen network, is a model of unsupervised artificial neural network [Tommy and Siu-Yeung, 2007] that is capable of capturing statistical relationships that exist in the principal components of a high-dimensional input data manifold and map them onto a low-dimensional regular structure, typically a two-dimensional regular grid for easier visualization of the topology of the data manifold.

The learning dynamics of the SOM can be described as a *coordinated competitive learning*. Upon repeated presentation of input data over a long period of training epochs, the reference vectors will adapt to the distribution of the input data vectors for which they have become winners forming a *quantized approximation* of the distribution of input data [Teuvo, 2001], and due to the coordinated updating of the reference vectors of nodes in the neighborhood of the winners, nodes that are near each other in the grid space will develop similar reference vectors, giving rise to the important property of *topology preservation* [Kimmo, 1996]. In other words, data samples that are near from one another in the input space will be mapped to nodes that are also close to one another in the space of the network grid. The overall effects thus achieved by the SOM algorithm can also be regarded as performing the combination of two concurrent tasks: topology preserving dimensionality reduction and vector quantization.

The SOM has been widely embraced as an effective tool for visualization of high-dimensional complex systems and data mining problems especially for classification and clustering tasks as evidenced by myriads of applications [Helge and Klaus, 1988; Teuvo et al., 1996] found in various fields of interests.

Nevertheless, practical applications of the SOM entail a time-consuming trial and error method of generating numerous feature maps of different sizes and shapes using varied learning parameters and evaluating the feature maps based on subjective criteria to determine a feature map that suits the intended purpose of the application. Experimental analyses [Hiong and Susan, 2005] suggest that variances in learning parameters have nominal effects on the learning dynamics of the SOM as long as they are made to decrease monotonically over the training period except for possible differences in the training times and there exist commonly recommended and adapted choices for the learning parameters [Tuevo, 2001].

No such general network configuration exists since the appropriate size and shape of the network grid to yield a good feature map depend on the statistical characteristics of the data set and therefore cannot be made into a generic parameter independent of the input data. Thus, *the requirement in the SOM algorithm to specify the size and shape of the required feature map prior to the training process imposes an undue difficulty to the experimenter who is applying the SOM to discover the unknown structure of the data manifold that exists in the data set.*

This requirement to pre-specify the topology of the network prior to the training process is one of the most criticized drawbacks of the standard SOM algorithm along with the long training epochs required by the algorithm. In this study, we investigate if it is feasible to grow the two-dimensional regular grid network structure of the Kohonen network during the training process in such a way as to

automatically generate a feature map of appropriate size and shape suitable for the data manifold of a given data set.

1.2 Research objectives, scope and methodology

The objectives of the research reported in this thesis are to examine methods of growing the feature maps incrementally during the training process in a way that maintains the structure of a two-dimensional rectangular grid in accordance with the standard Kohonen network, and to devise and *propose a model of self-growing and self-organizing map that can autonomously yield a feature map of appropriate size and shape suitable for the input data without requiring the size and the shape of the network to be specified prior to the training process.*

Such a model would alleviate the difficulty of predetermining suitable size and shape of the feature map in the application of the SOM while offering a structural compatibility to the standard Kohonen network so that visualization tools [Juha, 1999] designed for the feature maps produced by the standard SOM algorithm can be applied to the feature maps generated by the proposed model without any modification.

Since the topology of the feature maps adopted in most applications of the SOM is a two-dimensional rectangular uniform grid structure for easy visualization of the data manifold, the scope of the investigation on the design and assessment of the proposed model of the self-growing and self-organizing feature map is confined to those mechanisms that always maintain the network structure to a two-dimensional rectangular uniform grid.

Assessment of the proposed model is carried out by evaluating the qualities of the feature maps generated by the proposed model and by comparing them with the

qualities of the feature maps produced by the standard SOM algorithm using well known data sets in the field of machine learning and pattern recognition specifically chosen to represent varying complexity of the data manifolds they represent.

For each experiment using a particular pair of a training data set and a test data set, a feature map is first generated by the proposed model using the training data set. Then the standard SOM algorithm is applied to a network having the same size and shape as the feature map generated by the proposed model using the same training data set. Furthermore, to ascertain that the stopping condition used in the proposed model does not cause the problem of undergrowth and overgrowth, the standard SOM algorithm is applied to a network grid twice the size. Quality of the three feature maps produced by the proposed model and the standard SOM algorithm are then evaluated and compared using the test data set.

In the evaluation and relative comparison of the quality of the feature maps, we use two postulatory quality measures designed to measure how well the values of test data samples falling on a particular region of the input data space are represented by the reference vectors in the corresponding neighborhood of the feature map space, and an application-centric quality measure designed to quantify how well the generated feature maps identify the correct data classes of the test data samples.

1.3 Contributions

The results of the experiments strongly suggest that the proposed model is capable of growing the network dynamically during the training process to an appropriate topology suitable for a given input data set while avoiding undergrowth and overgrowth to yield a feature map of suitable size and shape for the input data manifold in significantly less training time than the time taken by the standard SOM

algorithm to produce a feature map of similar quality on a predetermined network structure.

The proposed model of self-growing and self-organizing batch map has been shown to be a useful and viable model of growing SOM that successfully addresses the difficulty in pre-specifying a suitable structure of the feature map prior to the training process in the application of the SOM. With the proposed model, it is no longer necessary to go through the time consuming process of generating numerous feature maps of different sizes and shapes applying different training epochs in search of a suitable feature map for a given data set.

We also believe that design criteria for the proposed model provides an important insight into the necessary and sufficient condition for stopping the process of growing and competitive learning in the design of growing models of the SOM and that the mean value of the weighted distortion errors in the neighborhood of the winner nodes used as the stopping condition in the proposed model presents a viable criteria that is guaranteed to converge while offering possibility of fine control when required.

1.4 Outline of thesis

The remainder of this thesis is organized as follows. Chapter 2 provides background information on the context of the thesis by reviewing the Kohonen's SOM in the framework of competitive learning models and analyzing the computational characteristics and the problems associated with the standard SOM algorithm. Review and analytical summaries of existing growing models of the SOM are provided at the end of chapter 2. Chapter 3 provides the detailed description of the proposed model of self-growing and self-organizing batch map starting with

motivations and design objectives of the proposed model followed by the details of the Batch Map [Teuvo, 1993; Teuvo, 2001] competitive learning algorithm, the stopping condition and the mechanism for the growth of the network. In Chapter 4, methodology and settings for the experiments used to evaluate and compare the performance of the proposed model against the standard Kohonen SOM algorithm are described. Chapter 4 begins with the description of the two data sets used in the experiments including the summary of their statistical characteristics and presents the outcomes of the experiments conducted along with the summaries of the findings from the results of the experiments. Finally, chapter 5 concludes the thesis with a summary of main objectives and results of the research including the contributions and suggestions for directions in the future research work for development of alternative growing models of the SOM.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

As are typical of unsupervised learning networks, the SOM operates on the basis of competitive learning, in which neurons in the network compete for the right to respond to a given input data [Laurene, 1993]. In this chapter, we provide fundamental background information about the context of the thesis by analyzing the learning dynamics of the SOM algorithm in the framework of a *coordinated competitive learning network* and by examining characteristics and problems associated with the standard SOM algorithm. We then establish foundations and grounds for our proposed model of self-growing and self-organizing batch SOM by reviewing the rationale behind growing variants of the SOM and summarizing different growth mechanisms applied in the existing models.

2.1 Competitive learning and the Self-Organizing Map

The SOM employs a form of competitive learning as its main device; consequently it shares some similarities with other competitive learning models. However, considerable differences exist in the underlying learning dynamics among different competitive learning models. In this section, we compare several models related to competitive learning using a common architectural notation and categorize the SOM as a competitive learning network in which Winner-Take-Most learning approach is used to bring about *vector quantization* of arbitrarily structured input data in a way that maintains some *topological order* that exist in the data manifold and the network structure of a fixed dimensionality is imposed to give rise to *dimensionality*

reduction from a high dimensional input space to the lower dimensional structure making it possible to get a low-dimensional representation of the data which may be useful for visualization purposes. Although some of the models reviewed in this section do not bear the notion of artificial neural network in the strict sense of architectural viewpoint, we describe them using a common framework of neural networks.

A competitive learning network consists of a set of K neural nodes $A = \{c_1, c_2, \ldots, c_K\}$. Associated with each node c is a *reference vector* w_c that represents the *receptive field center* in the input space. Each neuron assumes the responsibility of representing a subset of input data vectors acting as a prototype for them. An input data is encoded as an n-dimensional vector $x \in \Re^n$, and accordingly the reference vectors have the same dimensionality as the input vectors, i.e. $w \in \Re^n$. The n-dimensional input data are assumed to be generated either according to a continuous probability density function $p(x)$, or from a finite set of input data $D = \{x_1, x_2, \ldots, x_L\}$, where $L = | D |$.

The competitive learning process involves for each given input data vector x finding a node c whose reference vector w_c is most similar to x and adapting the reference vector w_c (and reference vectors of other nodes determined to be within the neighborhood of c in some competitive learning models like the SOM) to the values of x so that node c learns from the input data. The learning process is competitive in that nodes in the network compete for the right to learn from a given input data hence the process of identifying a node whose reference vector is most similar to a given input data vector is referred to as finding a *winner neuron* or *best matching node* in the SOM literature. Similarity among the data vectors and the reference vectors are commonly measured using the Euclidean metric. Accordingly, a winner node c for an

input data vector x is a node whose reference vector w_c has minimum distance to x in the data space among the reference vectors of all the nodes in the network.

2.1.1 Voronoi tessellation

A concept that is useful for illustrating goals of competitive learning is known as *Voronoi Tessellation* [Teuvo, 2001; Bernd, 1997; Franz, 1991]. Given a set of vectors $w_1, \ldots, w_K$ in $\Re^n$, the *Voronoi Region* V_i of a particular w_i is defined as the set of all points in $\Re^n$ for which w_i is the nearest vector, i.e.,

$$V_i = \{x \in \Re^n \mid i = \arg\min_{j \in \{1,\ldots,K\}} \| x - w_j \|\}. \quad (2.1)$$

The partition of $\Re^n$ formed by all Voronoi regions is called Voronoi tessellation.

The competitive learning process culminates in the partitioning of the input data space into a Voronoi tessellation where the constituent Voronoi regions are represented by the reference vectors of the nodes in the network. For each neural node in the network $c \in A$, the *Voronoi region* V_c of the node c is defined to consists of points for which the reference vector w_c is the nearest vector among all the reference vectors of the network, i.e.

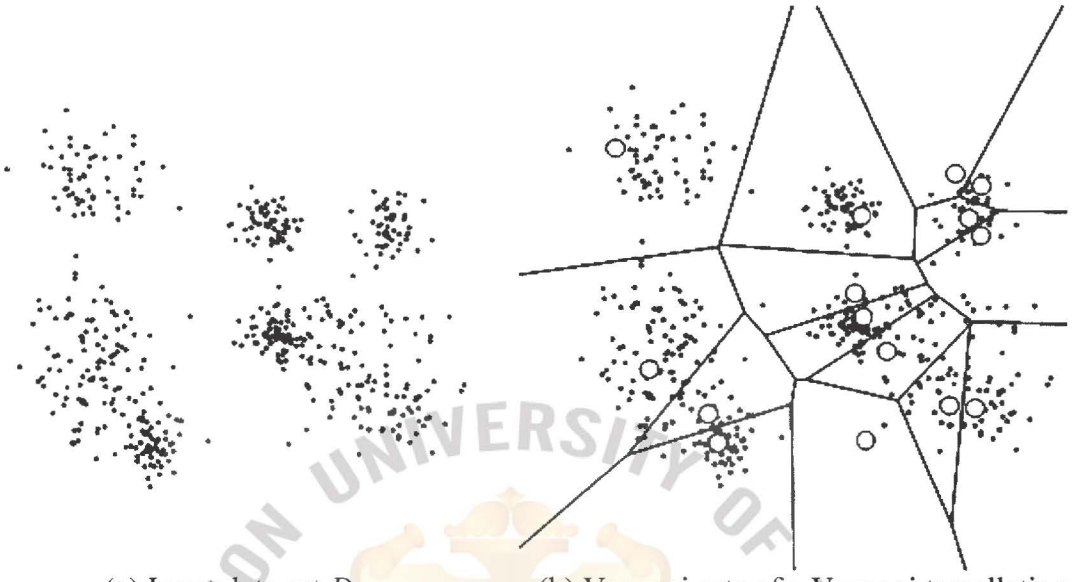
$$V_c = \{x \in \Re^n \mid c = \arg\min_{i \in A} \| x - w_i \|\}. \quad (2.2)$$

When the input to the network consists of a finite data set D, we define the subset V_c of D for which the neuron unit c is the winner as the *unit's Voronoi Set*:

$$V_c = \{x \in D \mid c = \arg\min_{i \in A} \| x - w_i \|\}. \quad (2.3)$$

Figure 2-1 illustrates a Voronoi tessellation partitioned by Voronoi sets for a finite number of data points in two-dimensional space. The input space is partitioned into regions bordered by lines such that each partition contains a reference vector that

is the nearest to any input vector within the same partition. Each Voronoi set contains the data points within the corresponding Voronoi receptive field.



(a) Input data set D (b) Voronoi sets of a Voronoi tessellation

Figure 2-1 Voronoi tessellation [Bernd, 1997]: An input data set D is shown on the left (a) and the partition of D into Voronoi sets for a particular set of reference vectors is shown on the right (b).

2.1.2 Objectives and classifications of competitive learning models

On the whole, the mainstay of the competitive learning is the formation of a Voronoi tessellation by adaptive updating of the reference vectors according to the distribution of input data vectors, but algorithmic details of how the reference vectors are updated vary significantly among different models of competitive learning. The specifics of the update mechanism employed by a particular competitive learning model is shaped by the intended purpose for the Voronoi tessellation to be formed hence the objectives of the competitive learning model.

Different competitive learning models can be classified into two main classes based on which reference vectors are updated for each input data vector, namely, *Winner-Take-All* learning and *Winner-Take-Most* learning. The Winner-Take-All learning models employ an update mechanism where the reference vector of the only

the single chosen winner is adapted for each given input data vector whereas in the Winner-Take-Most learning models, not only the winner node learns from an input data, but other nodes in the network may also learn from the input data by adapting their reference vectors to the input data vector.

One of the most common applications of the Winner-Take-All competitive learning is *vector quantization*, which is a classical signal-approximation method that usually forms a quantized approximation to the distribution of the input data vectors using a finite number of reference vectors (or more commonly called codebook vectors in the context of vector quantization). Once the codebook vectors are formed through the process of competitive learning, the approximation of a data vector x means finding the codebook vector w_c nearest to x in the input space [Teuvo, 2001]. Accordingly, vector quantization can be stated as the partitioning of the input space into a Voronoi tessellation whose constituent Voronoi sets are portrayed by the reference vectors as the codebooks, which typically represent the *centroid point* of the Voronoi set. In this context, the objective of the Winner-Take-Most competitive learning is the *minimization of expected quantization (or distortion) error*, i.e., to minimize the average of the differences between the reference vector of a Voronoi set and the data points falling on the receptive field of the Voronoi set. The quantization error is typically measured using the mean squared error, i.e., for a finite data set D and a competitive learning network A, the expected quantization error is

$$\mathrm{E}(D, A) = 1/|D| \cdot \sum_{c \in A} \sum_{x \in V_c} \| x - w_c \|^2 , \qquad (2.4)$$

where V_c is the Voronoi set of the unit c. Therefore, traditional vector quantization algorithms such as the k-means algorithm [James, 1967; Stuart, 1982] and the LBG (also known as generalized Lloyd) algorithm [Yoseph et al., 1980] can be regarded as the embodiment of the Winner-Take-All competitive learning. In fact, the k-means

algorithm and the LBG algorithm illustrate two alternative modes of updating the reference vectors in competitive learning: *online update mode* and *batch update mode*.

In the online update mode, which the *k*-means algorithm employs, the reference vectors are updated immediately after a winner node for an input data is determined and the process of determining the winner node and updating of the reference vectors are repeated for every input data vector. A complication with the online update mode is that the learning algorithm must decide the magnitude of change in the reference vector solely based on individual input data at a time without knowledge of other input data. In competitive learning models utilizing the online update mode, a parameter called *learning rate* α is used to control the extent to which the reference vectors are adapted toward the current input vector. If the network is allowed to learn continuously from the stream of input vectors, i.e. the learning rate α is kept constant, the network always stays adaptive; therefore there is no convergence. Typically, the learning rate is made to decrease monotonically over the period of the training process to stochastically stabilize the reference vectors.

The LBG algorithm can be considered an instance of the Winner-Take-All competitive learning network where batch mode of updating the reference vectors is employed to form a quantized approximation of the distribution of the data samples on the nodes of the network. In batch update mode, all data vectors from a finite set of input data, which is a necessary condition for the batch update mode, are first evaluated to determine the winner nodes for each of input data vectors, assigning the input data vectors to their corresponding winner nodes during which no updating of the reference vectors takes place. Only after the winner nodes of all the input data samples have been determined, the reference vectors of the nodes are updated to the centroid points of the input data samples for which they are the winners. Because the

reference vectors are updated to the centroid points of their corresponding Voronoi sets, the LBG algorithm is guaranteed to converge in a finite number of iterations to a local minimum of the distortion error and yields more stable asymptotic values for the reference values without resorting to the use of the monotonically decreasing learning rate α parameter.

A general problem associated with the Winner-Take-All learning models is their sensitive dependence on initialization values for the reference vectors [Bernd, 1997]. Inappropriate initialization may lead to certain neurons never becoming winner for any input data and, therefore keeping their initialization values indefinitely. More significantly, sensitive dependence on initialization causes different random initializations to produce very different results. The process of single node adaption may not be able to get the system out of the poor local minimum where it was started. One way to cope with the problem is to adapt the reference vectors of not only the winner but also those of some other nodes as advocated by Winner-Take-Most learning models, which in general decreases the dependency on initialization.

However, the hallmark of the Winner-Take-Most competitive learning lies with the notion of *topology mapping* [Bernd, 1995a]. By allowing a group of neural nodes correlated by a neighborhood relationship to learn from the same input data in a systematic way, the Winner-Take-Most competitive learning fosters a set of nodes defined to be within the neighborhood to develop similar values for their reference vectors, and the similarity relationships learned and encoded onto the reference vectors are either explicitly represented by the connectivity among the corresponding nodes or implicitly depicted onto the predefined network structure. Therefore, *the Winner-Take-Most competitive learning aims* to *capture the similarity relationships that exist in the input data manifold and map the topological structure of the data*

manifold onto the structure of the network in addition to the quantization of the input data space by the reference vectors. A necessary condition for the optimal mapping of the topology is that the topology of the employed network has to match the topology of the manifold of the input data represented [Thomas and Klaus, 1991]. This requires prior knowledge about the topological structure of the manifold which is not always available or might even be difficult to obtain if the topological structure is very heterogeneous, e.g., composed of subsets of different effective dimensions or disjunct and highly fractured.

Two different approaches to topology mapping can be observed among different Winner-Take-Most competitive learning models in the way the neighborhood relationship is defined and in the way the topological structure of the data manifold is represented on the network. In an attempt to capture and map the topology of the input data manifold more faithfully, competitive learning models like the *Neural Gas Network* [Thomas and Klaus, 1991] do not impose any constraints on the structure of the network and define the neighborhood of a winner node in the *input data space*, without being guided or restricted by the predefined structure of the network. In the Neural Gas Network, determination of the neural nodes whose reference vectors are to be updated to the values of an input data and the magnitude of changes in the reference vectors are based on the rank order of distances between the reference vectors and the input data vector in the input data space, not in the network space. To reflect the learned topology of the data manifold onto the network structure, the Neural Gas Network dynamically and *explicitly* maintains connections among the nodes of the network in a connectivity matrix as a topological map based on similarities among the reference vectors of the nodes. However, extracting the overall topological relationships captured by a *high dimensional arbitrary network structure*

may not be easy and is unsuitable for visualization even if the learning algorithm is capable of capturing a heterogeneous topological structure inherent in the data manifold.

In contrast, competitive learning models like the SOM employ a *fixed regular network structure*, usually a two-dimensional rectangular grid, to make the extraction and visualization of the captured topology of the data manifold simpler. In the Winner-Take-Most competitive learning models with a predefined network structure, determination of the neighborhood of a winner node and the magnitude of changes in the reference vectors of the nodes in the neighborhood are based on their distances from the winner node measured *in the space of the predefined network structure*. As the nodes in the neighborhood learn from the same set of data samples, data samples that are near from one another in the input data space will be mapped on to nodes that are also close to one another in the network space. Thus the topology of the input data manifold is implicitly encoded by the reference vectors onto the structure of the network without explicit maintenance of the connectivity among the nodes.

By constraining the network structure to a fixed dimensionality, the SOM aims to find a low-dimensional subspace of the input data space, containing most or all of the input data samples, in such a way that similarity relationships present in the principal components of the input data samples are reflected and mapped on to the fixed lower dimensional structure of the network. This objective known as *topology preserving dimensionality reduction* (or *feature mapping*) is one of the most distinguishing characteristics of the learning dynamics of the SOM, but the dimensionality reduction may cause the loss of some topological information when there is a mismatch between the topology inherent in the input data manifold and the topological structures restricted by the fixed dimensionality.

2.1.3 Self-Organizing Map as a Winner-Take-Most competitive learning network with a fixed network dimensionality

The SOM is a prime instance of a Winner-Take-Most competitive learning network with a fixed dimensionality. In typical applications of the SOM, the network is usually defined to be a two-dimensional rectangular grid with a *predetermined* number of neural units making up the nodes of the grid. Higher dimensional lattice structures are theoretically possible, but are not commonly used due to the difficulty in visualizing topological relationships captured by the learning algorithm.

Architecture of a typical two-dimensional SOM is illustrated in Figure 2-2, in which 36 neural units are arranged in a two-dimensional rectangular grid of 6 by 6. In this configuration, the input vectors are presented to the network through the input layer.

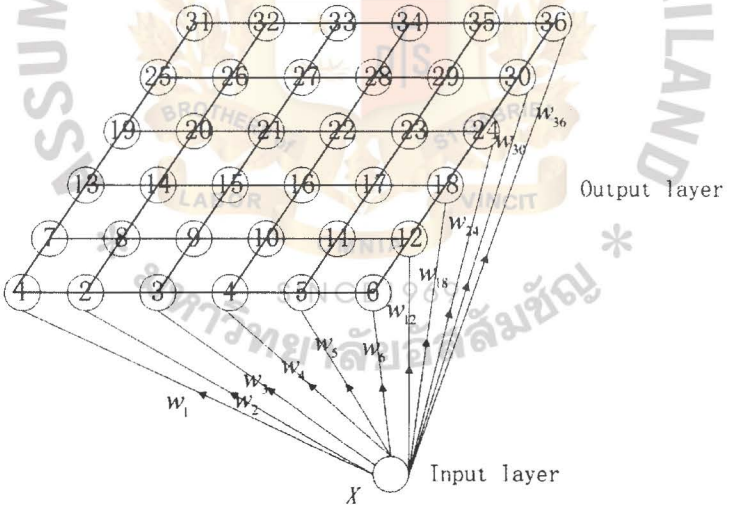


Figure 2-2 Architecture of 2-dimensional SOM [Tommy and Siu-Yeung, 2007]

The standard SOM algorithm employs the online update mode of Winner-Take-Most competitive learning. For each input vector, the SOM's learning algorithm repeats the following three major actions:

1. Determination of the winner node.
2. Determination of the nodes in the neighborhood of the winner.
3. Adaptation of reference vectors of the nodes in the neighborhood.

For a given input vector x, a winner node c is determined based on a similarity measure in the *input data space* with the similarity commonly measured in terms of Euclidean distance. In other words, the node whose reference vector w_c is nearest to the to the input vector x in the data space among all the reference vectors of the network is chosen as the winner, i.e.,

$$c = \arg\min_{i \in A} \| x - w_i \|. \tag{2.5}$$

However, the neighborhood area of the winner, thus the reference vectors to be updated, is determined based on the distance measure *in the pre-structured network space*. In other words, the decision as to whether a neural node i is determined to be in the neighborhood of the winner node c or not is made *based on the location of node i relative to the location of the winner node c in the network* without regards to the magnitudes of the reference vectors. This is in stark contrast to the network models without a fixed dimensionality like the Neural Gas Network, where the neighborhood area is determined solely based on the magnitudes of the reference vectors without regards to the spatial location of the nodes in the network. Consequently, the number of neurons whose reference vectors are adapted toward a given input vector depends on the size of the neighborhood area, or the *radius of the neighborhood* as referred to in the SOM literature.

In the standard SOM algorithm, the radius of the neighborhood is set initially large and made to decrease over time approaching zero as time increases for convergence to a stable topological structure [Teuvo, 2001], i.e., the radius of the neighborhood is defined as a monotonically decreasing function of time. This is

illustrated for a rectangular neighborhood in Figure 2-3, which shows the radius of the neighborhood at time t_1, t_2 and t_n for a winner unit c.

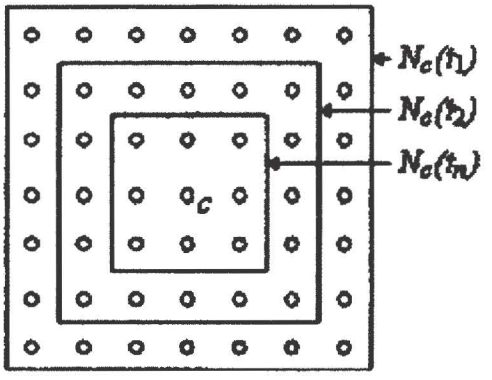


Figure 2-3 Topological neighborhood of a 2-dimensional SOM [Tuevo, 2001]: c is the winning neural node and $N_c(t_i)$ is a group of nodes considered as in the neighborhood of c at discrete time t_i.

The magnitude of adaptations for reference vectors of nodes in the neighborhood of the winner node depends on the spatial distance between a neighboring node and the winner node and on the *learning rate parameter* α, which is made to decrease monotonically over time. More specifically, the magnitude of the adaptations in the SOM can be defined as

$$\Delta w_i = \alpha(t) \cdot h_{c,i}(t) \cdot (x - w_i) \tag{2.6}$$

for a node i in the neighborhood, where $h_{c,i}(t)$ is the so called *neighborhood function* that determines the extent of adaptations for the node i based on its spatial distance from the winner node c.

There are two frequently cited choices for $h_{c,i}(t)$ in the SOM literature, one based on a set membership and the other based on the Gaussian function. In the simpler case of $h_{c,i}(t)$ being based on a set membership, which is computationally less demanding than the other alternative, all the units determined to be within the neighborhood of the winner node acquire the same magnitude of adaptations that are

decided by the learning rate $\alpha(t)$ and other units determined to be outside of the neighborhood are not adapted at all, i.e., $h_{c\cdot i}(t) = 1$ if $i \in N_c(t)$ and $h_{c\cdot i}(t) = 0$ if $i \notin N_c(t)$. Obviously, the winner node is always decided to be in the neighborhood for this purpose.

Instead of making all the neurons in the neighborhood gain the same amount of adaptations, the extent of adaptations can be made to depend on the spatial distance between a neighboring node and the winner node. When doing so, we must ensure that the extent of the adaptation is the highest for the winner node itself and decreases monotonically to zero with increasing spatial distance while the nodes at the same distance from the winner node are given the same extent regardless of their directional orientation. A widely adapted choice for $h_{c\cdot i}(t)$ satisfying the above conditions is the Gaussian function

$$h_{c\cdot i}(t) = e^{\left(-\frac{\|l_c - l_i\|^2}{2\sigma^2(t)}\right)}, \tag{2.7}$$

where l_c and l_i are the *location vectors* of nodes c and i, respectively and $\sigma(t)$ defines the radius of the neighborhood, which decreases monotonically over time as discussed earlier. It must be emphasized that $\| l_c - l_i \|$ is the spatial distance between the winner node and a neighboring node *in the network space*. Therefore, for a two-dimensional grid, $l_c \in \Re^2$ and $l_i \in \Re^2$ are the vectors representing the locations of the winner node and a neighboring node, respectively in the network grid.

An example of a 2-dimensional Gaussian neighborhood function with a neighborhood radius of 5 for a 50 by 50 neural grid is illustrated in Figure 2-4.

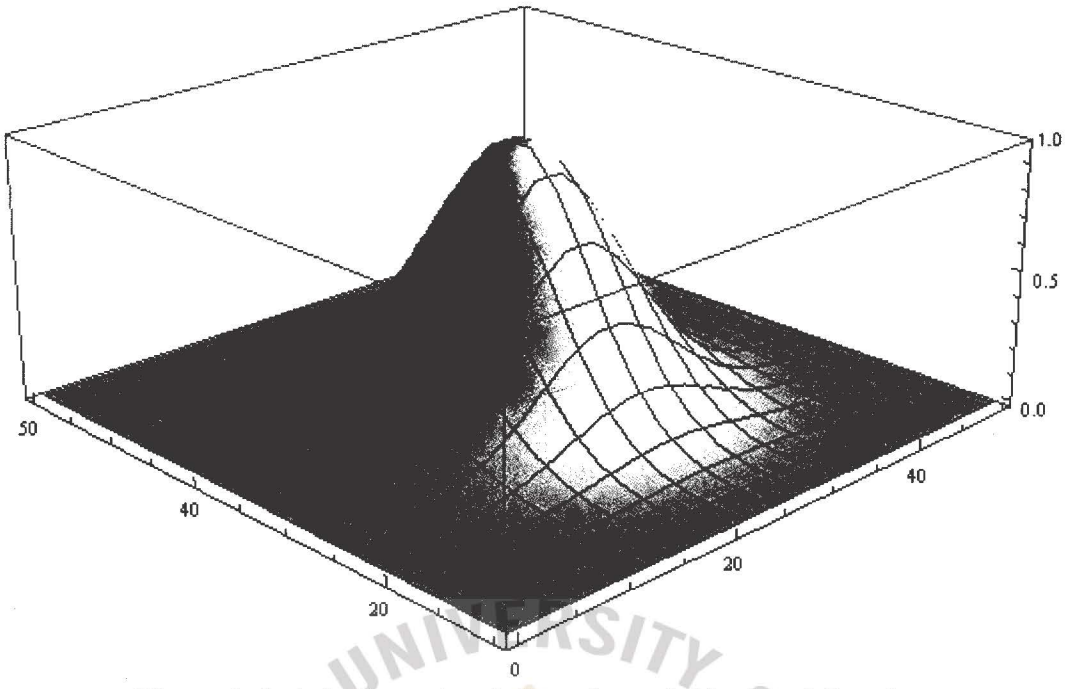


Figure 2-4: A 2-dimensional Gaussian neighborhood function

Since the reference vectors within the neighborhood of the winner are *coordinated* together to adapt toward given input vectors, neurons that are close in the grid up to a certain spatial distance will activate each other to learn something from the same input vector x. This will result in a local *relaxation* or *smoothing effect* on the reference vectors of neurons in this neighborhood, which in continued learning leads to *global ordering* [Teuvo, 2001]. As a consequence, *the topological relationships in the input data manifold is gradually encoded implicitly into the reference vectors of the neural nodes during the continued learning process without requiring them to be explicitly represented and maintained*, for example, in a connectivity matrix as is done in the Neural Gas Network.

Upon repeated presentation of input vectors, the reference vectors will adapt to the distribution of the input vectors for which they have been winners forming a quantized approximation to the distribution of the input data vectors, essentially performing vector quantization. Furthermore, due to the coordinated updating of reference vectors in the neighborhood, neurons that are adjacent in the network will

tend to develop similar values for their reference vectors. Overall effect of the SOM's learning algorithm is the formation of a Voronoi tessellation in which the constituent Voronoi sets are *ordered by topological relationships* inherent in the input data manifold. Thus the learning dynamics of the SOM can be described as a *coordinated competitive learning* performing the combination of two concurrent tasks: topology preserving dimensionality reduction and vector quantization.

The competitive learning algorithm of the standard SOM with a network structure of two-dimensional regular grid using the online update mode is summarized below:

1.1. Arrange K number of neurons in the set A as a two-dimensional rectangular grid.

1.2. Initialize their reference vectors with randomly selected vectors from the training data set.

1.3. Initialize the epoch counter $t = 0$.

1.4. Set the initial learning rate $\alpha(t_0)$

1.5. Set the initial radius of neighborhood $\sigma(t_0)$

2. For each and every input vector x in the training data set, do the following:

2.1. Determine the winner node: $c = \arg\min_{i \in A} \| x - w_i \|$

2.2. Adapt the reference vectors of the nodes in the neighborhood of the winner according to $\Delta w_i = \alpha(t) \cdot h_{c,i}(t) \cdot (x - w_i)$.

3.1. Increase the epoch counter: $t = t + 1$.

3.2. Monotonically decrease the learning rate $\alpha(t)$

3.3. Monotonically decrease the radius of neighborhood $\sigma(t)$

4. If $t < t_{max}$ (a predefined number of training epochs), continue with step 2.

2.2 Characteristics and problems of the standard SOM algorithm

In the previous section, we presented and reviewed the standard SOM algorithm in the context of a competitive learning network in comparison to other alternative competitive learning schemes in order to highlight the main principles that shape the core of the learning dynamics of the standard SOM algorithm and how it functions to form a nonlinear, ordered, smooth mapping of high-dimensional input data manifolds onto the elements of a regular, low-dimensional grid [Teuvo, 2001], called a *feature map.*

In this section, building upon the theoretical foundations presented in the previous section, we provide analytical summary of the main features and characteristics of the standard SOM algorithm and discuss some of the problems associated with it in order to lay the ground work for our proposed dynamically growing model of the SOM.

2.2.1 Features and characteristics of the standard SOM algorithm

The popularity of the SOM as an effective tool for exploratory data analysis and modeling of complex system processes [Teuvo et al., 1996], especially for those problems demanding classifications and clustering of complex data as in data mining and pattern recognition applications, can be attributed to its *capability to discover generalized categories from high-dimensional complex data and map them onto a low-dimensional regular grid structure for easy visualization.*

This capability can be explained in terms of vector quantization and topology preserving dimensionality reduction. As described in the previous section, not only does the SOM perform a quantized approximation to the distribution of the input data manifold using a finite number of reference vectors, but also do so in a way that

learns the topological relationships that exist in the data manifold and maps them onto a lower-dimensional network in an *orderly* fashion. The mapping tends to preserve the topological relationships of the data manifold and due to this order it manifests *categories* of input data and their relationships on the map. The reason we denote this process as the discovery of *generalized* categories is that the SOM may lose some topological relationships that exist in the high-dimensional data manifold and not able to preserve all topological information faithfully on the network structure due to the dimensionality reduction.

This loss of topological information, nonetheless, can be regarded as a feature rather than a flaw since in the analysis of high-dimensional arbitrarily structured data we may not be interested in all the glory of complex topological information but be concerned with only the most important topological relationships of the *principal components* of the input data manifold, hence the *generalized categories*. The whole process can also be regarded as the *compression* of information while preserving the most important topological relationships of the principal components from an information-theoretic view. However, it must be pointed out that when approximating the probability distribution of data manifold that has significant variations in more than two dimensions onto a two-dimensional network structure, the resulting map will expose *twists* and *discontinuities* in the topological order of the reference vectors. Whether the possible loss of the topological information due to the dimensionality reduction is seen as a feature or a flaw will depend on the design goals of a particular competitive learning model. In the case of the SOM, it is a feature intended for easier visualization of captured topology of the data manifold.

In order to better comprehend the capabilities and limitations of the standard SOM algorithm, it is also necessary to understand how the learning dynamics of the

standard SOM algorithm works to achieve the capabilities described. In the following, we provide an analysis of the iterative process of reference vector adaptations in the standard SOM algorithm as per its capabilities and the roles played by the two time dependent parameters, the learning rate $\alpha(t)$ and the radius of neighborhood $\sigma(t)$, in the process.

After initializing the reference vectors, the SOM algorithm iteratively repeats the process of finding a winner node, determining the neighborhood of the winner node and updating the reference vectors of the nodes in the neighborhood. The formation of a final feature map by means of the iterative updating of the reference vectors in the SOM can be seen as taking in two phases of the *ordering phase* when topological ordering of the reference vectors takes place followed by the *convergence phase* in which the fine-tuning of reference vectors takes place for statistical accuracy of the mapping.

The radius of the neighborhood at the start of the iterative process, $\sigma(t_0)$, is set to a large value, typically recommended to cover half the diameter of the network [Teuvo, 2001]. During the early iterative steps, say, the first 1,000 steps or so, the radius of the neighborhood is made to decrease monotonically over these early iterative steps to a unit value (radius of 1). Therefore, the number of nodes that are adapted towards a given input vector will be large in the beginning and will gradually reduce to include only the directly neighboring nodes at the ends of these early steps. Furthermore since the reference vectors within the neighborhood of the winner are interacting together to adapt towards given input vectors, neurons that are close in the network up to a certain spatial distance will excite each other to learn from the same input vector. The overall effect of the *lateral interaction* among the nodes in the neighborhood with the gradual shrinking of the radius is the *ordering of the reference*

vectors. Accordingly, this early iterative steps is referred to as the ordering phase. Throughout the ordering phase, the learning parameter $\alpha(t)$ should have reasonably high values (close to 1) to facilitate the lateral interaction or the coordinated learning process. The motive for the large radius at the very beginning of the ordering phase is to avoid the problem of the map not becoming globally ordered. If the neighborhood is too small to start with, various kinds of mosaic-like parcellations of the map are seen during the ordering stage, between which the ordering direction changes discontinuously [Teuvo, 2001].

After proper ordering of the reference vectors has taken place in the ordering phase, the reference vectors are finely adjusted during the convergence phase. Since the purpose of the convergence phase is the fine-tuning of the reference vectors for statistical accuracy of the quantization, the radius of the neighborhood $\sigma(t)$ is usually kept very small usually covering only the nearest nodes as set at the end of the ordering phase and including only the winner node itself towards the end of the convergence phase. The learning rate $\alpha(t)$ during the convergence phase should decrease monotonically to a very small value (e.g., of order of or less than .02) over a *long* period. Since the learning is a stochastic process, the final statistical accuracy of the mapping depends on the number of iterative steps in the convergence phase, which must be reasonably long; there is no way to circumvent this requirement [Teuvo, 2001]. Kohonen recommends that the number of iterative steps in the convergence phase be at least 500 times the number of the nodes in the network as a rule of thumb.

Hence the radius of the neighborhood $\sigma(t)$ during the ordering phase and the learning rate $\alpha(t)$ during the convergence phase are required to be monotonically decreasing functions of the training time for global ordering and convergence of the

reference vectors. However, the accuracy of the decay seems to have nominal effects on the learning dynamics of the SOM, and both linear and exponential decay functions [Teuvo, 2001] are widely used in the applications of the SOM algorithm. In fact, the effects of these parameters on the learning dynamics of the SOM algorithm have been so far analyzed experimentally only [Hiong and Susan, 2005] as no formal theory demonstrating the dynamics of the SOM algorithm has been achieved yet.

2.2.2 Problems associated with the standard SOM algorithm

Despite the wide acceptance as an effective tool for exploratory data analysis, the SOM is not without its critics. Common problems reported in the literature about the standard SOM algorithm are predominantly in the following areas:

- Possibility of loss of topological information in the learning process which could lead to the topological mismatch between the topology of the data manifold and the topology of the resultant map.
- The variances in the final feature map formed depending on the input sequence and the choice of learning parameters, e.g., $\alpha(t)$ and $\sigma(t)$, applied.
- Lack of standard quality measures for objective evaluation and comparison of feature maps generated.
- Difficulty in pre-determining the number of nodes and the shape of the network.

We already provided an analysis on the possible loss of topological information in the previous subsection where we reasoned that the loss of topological information during the learning process of the SOM is an inevitable consequence of the dimensionality reduction and that it should be considered a feature rather than a limitation in the light of its capability to extract topological relationships of the

principal components of the input data manifold and offer effortless visualizations of the *generalized categories* of information. Therefore, we will not pursue this matter any further in this subsection.

Our view on the remaining problems is that they are all related together in a logical sequence of consequences from a common root cause: *the absence of a formal theory describing how the SOM algorithm works to generate topologically correct feature maps*. Despite attempts by several researchers, a clear mathematical formalism which describes the objective function computed by the SOM process has not been achieved. It is not clear under what conditions the algorithm may be guaranteed to converge or whether the algorithm works by performing a stochastic gradient descent on some potential function, and problems of important practical interest, like the number and type of the algorithm's stationery states, convergence speed as a function of the algorithm's learning parameters and the avoidance of sub-optimal representations, are not solved [David et al., 1992].

The only formal proof achieved of topological ordering and convergence is for the simplest case in the formation of a topological map of a one-dimensional input space on a one-dimensional linear lattice (array of nodes) when the neighborhood function is a monotonically decreasing function of time. Initially Kohonen proved that his algorithm converges to a topologically ordered, stable state in the case of one-dimensional input and one-dimensional map by modeling the one-dimensional map as a Markov chain for a restricted case of neighborhood function defined as a unit step function [Teuvo, 2001]. He showed if the input is randomly selected, once the reference vectors become ordered, they never become disordered and the point density of the nodes will finally approximate that probability density of the input. Ritter quantified the relationship more exactly showing that given an input probability

density $p(x)$, the map's point density approaches $p^{2/3}(x)$ [Teuvo, 2001]. The proof was generalized to hold for any neighborhood function, e.g., the Gaussian function, which is monotonically decreasing in distance [David et al., 1992].

Unfortunately, the theoretical work on the one-dimensional map case is not immediately extendable to two-dimensional case, let alone higher-dimensional maps, because in two-dimensional case the vectors are not as clearly ordered as in the case of the linear array. Various attempts to apply the Markov methods to the analysis of the SOM process on the two-dimensional map have not yielded any conclusive general formal proof of convergence to a global topological order; the analyses have rather revealed that statistical fluctuations can occur in the reference vectors during convergence and that their characteristics depends on the time dependent learning parameters. In fact, it has been shown that the learning dynamics of the standard SOM algorithm cannot be described by a gradient descent on a *single global* energy function even for the one-dimensional map case [David et al., 1992].

Determining how the final state of the map depends on the time evolution of the neighborhood function $\sigma(t)$ and the learning rate function $\alpha(t)$ and being able to predict the final topological state of the two-dimensional map for a given input data based on a formal theory of the learning dynamics are the major yet most elusive goals of the theoretical analysis of the SOM. The lack of formalism in the description of the learning dynamics of the SOM algorithm has caused various difficulties in the practical applications of the SOM.

Since there is no guarantee of convergence to a globally optimal state, the learning algorithm of the SOM can generate possibly highly different feature maps for the same training data set depending on the initialization values for the reference vectors, choices of the learning parameters used and even on a particular sequence of

sample training data vectors applied, with each different feature map perhaps representing an instance of some local optima. These *incontrovertible variances in the resultant feature* maps have posed difficult problems in the applications of the SOM. The only remedy often suggested in the literature for the problem is to generate many different feature maps in order to get one final *good* map. This approach, however, brings forward another problem of the SOM: *the lack of the standard quality measures for objective evaluation and comparison of feature maps.*

As the learning dynamics of the standard SOM algorithm cannot be described by a gradient descent on a single global objective function, there is no single standard function with which to measure the quality of feature maps to evaluate how close a final feature map is to the global optimal state. Therefore, determining what constitutes a *good* feature map is a difficult undertaking in itself. Without a standard measure of qualities, generated feature maps cannot be evaluated and compared *objectively*, making the task of finding a good feature map rather subjective in its nature.

Although a few quantitative measures have been proposed, there exists no universally accepted standard measure for evaluating the quality of the feature maps, especially for comparing the quality of feature maps of different sizes. The measures proposed generally attempt to quantify the two main capabilities of the SOM's learning dynamics: accuracy of vector quantization and the level of topology preservation in the dimensionality reduction. The SOM's capability to approximate the probability distribution of the input data vectors is evaluated by means of the average quantization error, just as is done in other vector quantization methods. However, the quality of topology preservation remains a property that is not easy to define and even harder to measure quantitatively. There have been a small number of

suggested measures [Hans-Ulrich and Klaus, 1992; Thomas et al., 1994; Kimmo, 1996] for evaluating the quality of topology preservation but none has been widely embraced as a de facto standard. A survey of quality measures for the SOM and experimental comparisons of their sensitivity to the size of the network and size of the data can be found in [Georg, 2004].

The application of the SOM is further complicated by the fact that the standard SOM algorithm requires that the number of nodes in the network and the shape of the network, i.e., the topology of the network, be predetermined and specified prior to the training process. However, an appropriate choice of the network topology to yield a good feature map depends on the statistical characteristics of the data and cannot be made a generic parameter independent of the input data. Therefore, the requirement in the SOM algorithm to specify the size and the shape of the required feature map prior to the training process imposes an undue difficulty to the experimenter who is applying the SOM to discover the unknown structure of the data manifold, and it is one of the most criticized drawbacks in the application of the SOM. Moreover, evaluating and comparing qualities of feature maps of different sizes may not be straightforward due to the inherent characteristics of a particular quantitative measure used. For example, the average quantization error usually decreases with increasing number of nodes in the network because the data samples are distributed more sparsely on the map.

To recapitulate, the absence of formalism in the SOM's learning algorithm has caused inevitable practical problems that can only be dealt with through experimental approaches at present and this point is well reflected in Kohonen's own description [Teuvo and Timo, 2007] of the problem:

"*The mathematical theory of the SOM is very complicated and only the one-dimensional case has been analyzed completely. Apparently, the SOM belongs to the ill-posed problems in mathematics*".

Accordingly until a formal theory capable of precisely describing the learning dynamics of the SOM can be achieved, the effective application of the SOM has to rely on the time-consuming trial and error method of generating numerous feature maps while tuning the learning and network parameters in order to obtain a final good feature map. But then what constitutes a *good* feature map is not a question with clear-cut answers either. The result is that without a single global objective function to optimize, the exact mapping that is produced depends to some extent on how the experimenter tunes the learning and network parameters. Therefore, the feature map tends to reflect *some subjective quality* that the experimenter wants to see in the two-dimensional representation of the input data.

2.3 Growing models of the SOM

Few extensions to the SOM algorithm have been proposed that grow the network structure dynamically during the training process in an attempt to address the difficulty of predetermining the appropriate size and/or shape of the required feature map in the standard SOM algorithm. Alternative growing SOM models mainly differ in *when*, *where* and *how* new nodes are added and in the specification of the stopping condition for the growth; these heuristics on the growth mechanism are guided by the design goals and have important ramifications on the structure of the feature maps produced. In the following, we summarize the growth heuristics adopted by three prominent growing variants of the SOM.

Fritzke developed Growing Grid [Bernd, 1995b] to tackle the difficulty of predetermining a shape, i.e., aspect ratio of width and height, of the feature map suitable for the data at hand while maintaining the two-dimensional rectangular grid structure of the standard SOM by incrementally inserting rows or columns into the network grid until a required number of nodes, specified prior to the training process as the stopping condition for the growing process, is established. The training process consists of two distinct phases: a *growing phase* during which the network is incrementally grown followed by a *fine-tuning phase* in which no further changes are made to the network structure and the reference vectors are adjusted to improve statistical accuracy of the mapping.

The growing phase starts with an initial grid of 2 by 2 and the reference vectors are adapted in a similar fashion to the standard SOM algorithm except that the radius of neighborhood and the learning rate stay constant. After every $w \times h \times \lambda_g$ number of adaptation steps, where w and h are the width and height of the current network grid respectively and λ_g is the number of adaptation steps per node before a growth is initiated, a row or column is inserted around a node q that has been the winner most time during the period of adaptations. A direct neighbor f of q with the most different reference vector from q is identified and a new row (or column) between q and f is inserted depending on whether q and f share a common column (or row), and the reference vectors of the new nodes are interpolated from their direct neighbors.

When the number of nodes in the network exceeds the minimum required, the Growing Grid stops the growing process and proceeds to the fine-tuning phase, in which the radius of the neighborhood is kept the same as in the growing phase, but the learning rate is made to decrease monotonically over the fine-tuning phase. The total

number of adaptation steps applied is $w \times h \times \lambda_f$, where w and h are the width and the height of the final grid formed during the growing phase and λ_f is a user-defined parameter that specifies an average number of adaptation steps per node in the fine-tuning phase.

The Incremental Grid Growing (IGG) [Justine, 1993; Justine and Risto, 1995] abandons the rectangular grid structure of the standard SOM and takes on a two-dimensional *non-uniform* grid structure in an attempt to embed the cluster boundary information directly onto the grid structure. In a non-uniform grid structure, a node is not required to have all of its four direct neighbors and connections between direct neighbors may or may not exist. After initializing a 2 by 2 grid, the IGG repeats the following three steps until a desired number of nodes is achieved, which needs to be specified as a stopping condition for the training process:

1. Adapting the current grid to the input distribution by the application of standard SOM algorithm.
2. Adding new nodes to a *boundary node* with highest cumulative error compiled during step 1.
3. Examining reference vectors of every pair of directly neighboring nodes to determine whether a new connection should be established between the pair or the existing connection between the pair should be deleted.

A boundary node is defined as any node that has at least one directly neighboring position in the grid space not yet occupied. During the SOM learning process, whenever a boundary node is determined to be a winner, the square of the distance between its reference vector and the input vector is added to the error value of the node. After an application of a complete SOM algorithm, (between one and

three) new nodes are added and connected to the boundary node with the highest cumulative error, and their reference vectors are interpolated from their neighbors. Afterwards connectivity between the directly neighboring nodes is adjusted to reflect the changes in the reference vectors. For any pair of directly neighboring nodes which are currently unconnected together, if the difference between their reference vectors is below the *connect-threshold*, a new connection is established. Similarly, for any pair of connected neighboring nodes, if the difference between their reference vectors is greater than the *disconnect-threshold*, the existing connection is removed to reflect the fact that they have evolved into separate areas of the input space.

The Growing Self-Organizing Map (GSOM) [Damminda et al., 2000; Arthur et al., 2009] is another growing model of the SOM that strives to reflect the cluster boundary information onto the structure of the network and hence, shares some similar characteristics with the IGG. Just like the IGG, the GSOM adds new nodes to boundary nodes only, but does away with the elaborate connectivity maintenance scheme of the IGG. In the GSOM, connections to directly neighboring nodes, when the neighbors exist, are always kept established. The GSOM interweaves the growing steps and the learning steps more closely whereas the IGG applies the complete learning process of the standard SOM algorithm for each enlarged network.

After initializing a 2 by 2 grid, the GSOM goes through distinct phases of a *growing phase* followed by a *smoothing phase*, in which no new nodes are added and quantization errors are smoothed out. During the growing phase, each time a winner node is determined, a cumulative error for the node is maintained by adding the square of the distance between its reference vector and the input vector, and when the cumulative error is higher than a predefined parameter called Growth Threshold (GT), either between one and three new nodes are added to the winner if the winner is a

boundary node, or the error value is distributed to neighboring nodes if the winner is a non-boundary node. The distribution of the error value of a non-boundary winner to its neighbors is designed to give non-boundary nodes some indirect ability in initiating node growth by spreading out the error value and causing boundary nodes to increase their error values over time. The radius of neighborhood during the growing phase stays constant, but the learning rate is made to decrease exponentially while being reset to an initial value every time a growth is initiated. The growing phase stops when new node growth saturates, which can be identified by the low frequency of new node growth. Therefore, the parameter GT controls the stopping condition for the growth hence the size of the final feature map.

The three growing models of the SOM reviewed in this section demonstrate different approaches to growing the network during the training process and alternative ways of interweaving the competitive learning process and the growing process. However, they still require that stopping condition for the growth is specified in terms of the network size either directly or indirectly prior to the training process. For example, the Growing Grid and the IGG require the specification of minimum number of nodes in the network required before the growing process stops as a training parameter and the GSOM uses the Growth Threshold parameter to specify the stopping condition for the growth but it's unclear how to determine an appropriate value of the Growth Threshold for a given data set. Furthermore, the IGG and the GSOM abandon the uniform rectangular grid structure of the standard Kohonen network in an attempt to embed the cluster boundary information onto the structure of the feature map making their feature maps structurally incompatible to the feature maps generated by the standard SOM algorithm.

CHAPTER 3

PROPOSED MODEL OF SELF-GROWING AND SELF-ORGANIZING BATCH MAP

3.1 Motivations and design goals of the proposed model

Even though no formal theory describing how the learning dynamics of the SOM works to generate topologically correct feature maps has been established, the capability of the SOM to capture nonlinear statistical relationships that exist in the principal components of high-dimensional input data manifolds and map them onto a low-dimensional regular grid structure has been well demonstrated and its effectiveness as a tool for visualization of high-dimensional complex systems and data mining especially for classification and clustering tasks has been widely embraced as illustrated by myriads of applications of the SOM in various fields of interests [Teuvo et al., 1996; Helge and Klaus, 1988; Tuevo, 2001].

Nonetheless, all practical applications of the SOM entail the time-consuming trial and error method of experimentally generating numerous feature maps using varied learning parameters and different network configurations and evaluating the generated feature maps based on subjective criteria to determine a feature map that suits the intended purpose of the application, which is inevitable due to the absence of a clear mathematical formalism that describes the objective function computed by the learning process of the SOM.

The effects of learning parameters on the learning dynamics of the SOM and the quality of the resultant feature maps have been experimentally analyzed by several researchers [Hiong and Susan, 2005]. Although no formal proof has been achieved yet, various work suggest that different choices for the radius of the neighborhood and the

learning rate have nominal effects on the learning dynamics of the SOM as long as they are kept to decrease monotonically over the training time except for the possible impacts on the speed of the convergence. Neither are they judged to depend on the statistical characteristics of the training data set.

Various experimental analyses have made similar recommendations of setting the initial radius of the neighborhood to cover the half the diameter of the network and monotonically decreasing it to include only the direct neighbors while keeping the learning rate reasonably high to facilitate the lateral interaction during the ordering phase and then monotonically decreasing the learning rate over long period of convergence phase for fine-tuning of the reference vectors [Tuevo, 2001]. And these recommended learning parameters have been widely accepted as de facto choices in many applications of the SOM learning process.

However, not much has been investigated about how the learning process of the SOM and the quality of the resultant feature maps are affected by the configuration of the network despite the fact that the standard SOM algorithm requires the size and the shape of the network grid to be specified prior to the training process. We believe this is so because unlike the learning parameters the appropriate size and shape of the network to yield a good feature map depend on the statistical characteristics of the training data set and cannot be made into a generic parameter independent of the input data.

Therefore, the requirement in the standard SOM algorithm to specify the size and shape of the required feature map prior to the training process imposes an undue difficulty to the experimenter who is applying the SOM process to discover the unknown structure of the data manifold, and it is one of the most criticized drawbacks in the application of the SOM. Thus it is necessary to generate numerous feature maps

of different sizes and shapes until a satisfactory feature map based on the experimenter's subjective criteria is obtained.

Few extensions to the SOM have been suggested to make the network structure grow dynamically during the training process, but they do not completely address the difficulty of predetermining the size and shape of the required feature map because they still require the stopping condition for the growth be specified in terms of minimum network size. In addition, non-uniform grid network structures employed by some growing models make the resultant feature maps incompatible to existing visualization tools designed for the feature maps of the standard SOM algorithm.

The proposed model of self-growing and self-organizing batch map has been motivated by our desire to devise a growing model of the SOM that can autonomously grow the network during the training process while maintaining a two-dimensional rectangular grid structure to yield a feature map of appropriate topology suitable for a given data set without requiring the size of the network be specified prior to the training process. Such a model would alleviate the difficulty of predetermining suitable size and shape of the feature map in the application of the SOM while offering a structural compatibility to the standard Kohonen network so that visualization tools [Juha, 1999] designed for feature maps produced by the standard SOM algorithm can be applied to the feature maps generated by the proposed model without any modification. Therefore, the design of the proposed model has been guided by the following criteria and goals:

- The proposed model should grow the network dynamically during the training process in such a way that maintains the two-dimensional uniform regular grid structure consistent with the standard Kohonen network

- The proposed model should autonomously determine an appropriate stopping point for the growing and the learning process to yield a feature map of appropriate size and shape suitable for the data at hand while avoiding undergrowth and overgrowth.
- The final feature map produced by the proposed model should bear comparable qualities to the feature maps obtainable by the standard SOM algorithm in terms of its capability to capture statistical relationships that exist in the principal components of a high-dimensional input data manifold and represent them as a two-dimensional feature map.

3.2 The self-growing and self-organizing batch map

Fundamental to the design of any growing model of the SOM are the mechanisms used to grow the network structure during the training process. In particular, the following design questions on the growing mechanisms need to be addressed:

- When to stop the growing and the learning process
- Where to insert new nodes and how to initialize the reference vectors of newly added nodes
- How to interweave the growing process and the learning process

Obviously the design decisions on the growing mechanisms cannot be resolved based on a mathematical formalism due to the lack of a formal theory describing the learning dynamics of the SOM, and various heuristics are possible for the growing mechanisms. The design choices for the growing mechanisms form the quintessence of any growing model of the SOM and they must be carefully considered in accordance with the goals of the design. In this section, we present the

proposed model of self-growing and self-organizing batch map describing how the growing mechanisms adopted in the model is designed to accomplish the stated design criteria and goals.

The overall organization of main procedures in the training process of the proposed model is outlined in Fig. 3-1. The training process starts by creating an initial network arranged in a two-dimensional rectangular grid. Thereafter, the model repeatedly interweaves the process of the Winner-Take-Most competitive learning using a batch version of the SOM algorithm called the Batch Map [Teuvo, 1993; Teuvo, 2001] and the process of growing the network structure until a suitable feature map is deemed to have formed, which is signified by the stopping criteria.

For each newly generated network grid, either initially or by the growing process, the Batch Map algorithm is applied to adapt the reference vectors according to the distribution of the training samples, resulting in a formation of a feature map. Then the feature map produced by the Batch Map process is evaluated to determine if the stopping condition has been achieved. When the convergence to the stopping condition is achieved, the training process stops and the current feature map is output as the final feature map. Otherwise, the growing process is initiated to grow the current network and another cycle of the Batch Map process is applied to the enlarged network grid.

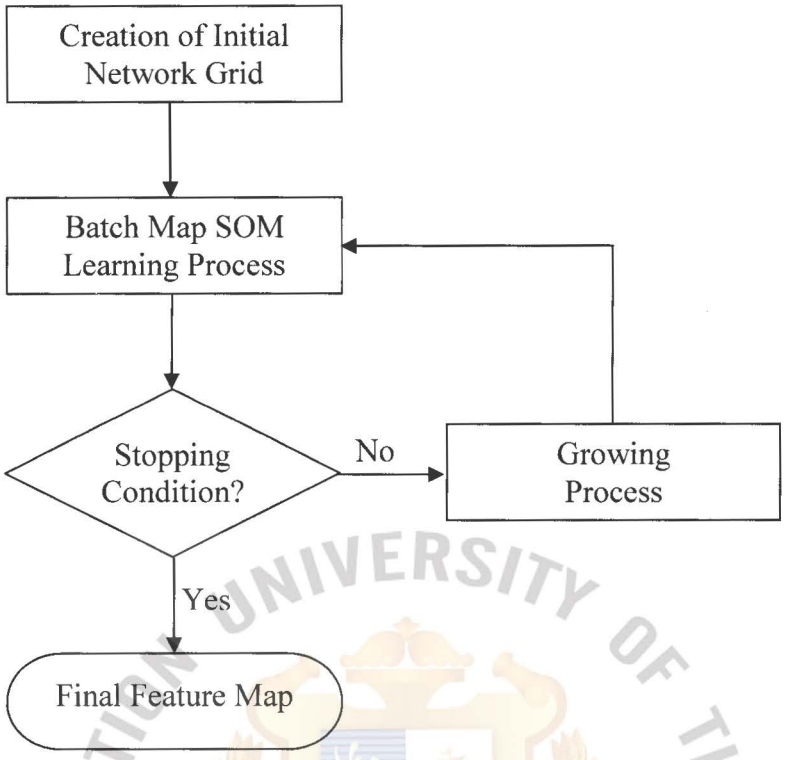


Figure 3-1 Outline of the training process of the proposed model

3.2.1 Initial network grid

The size of the initial network grid should not be too large or too small. If too large, the benefit of gradual learning through the incremental growing is not reaped. If too small, substantial amount of the learning and growing cycles will be initially wasted prolonging the overall training time. Furthermore, the sensitivity of the Batch Map algorithm to the initialization values [Jean-Claude et al., 2002] must be stabilized through adequate number of growing and learning cycles while ensuring sufficient numbers of well initialized reference vectors are present to begin with in the initial network grid. To achieve the desired effect, we make the values of about half of the training data samples from the training data set S to be represented in the initial network by creating a square grid with side length k of

$$k = \left\lfloor \sqrt{\frac{|S|}{2}} \right\rfloor, \tag{3.1}$$

and initializing the reference vectors of the nodes with the values of randomly selected samples from the training set S.

3.2.2 The batch map algorithm

Every time new nodes are inserted by the growing process, a new feature map is formed on the enlarged network grid by the Batch Map process, a batch version of the SOM algorithm that has been shown to achieve the same overall effect brought about by the online SOM algorithm computationally faster by updating the reference vectors with the net effect of all the training samples in batch mode [Teuvo, 1993; Teuvo, 2001]. For each training data vector x in the training set S, the Batch Map finds a winner node c whose reference vector w_c is closest to x amongst the nodes comprising the current network grid A, and then adds the training data vector x to the set V_c of the winner node c. Consequently, the training samples are partitioned into a Voronoi tessellation and the set V_c of a node c contains a collection of those training data samples for which the node c is the winner, i.e.

$$V_c = \{ x \in S \mid c = \arg\min_{c \in A} \| x - w_c \| \}. \tag{3.2}$$

The set V_c represents the Voronoi set of the node c and the centroid of the points in the set is the mean vector $\bar{y}_c$ of the data vectors in V_c. The Batch Map then updates the reference vectors with the average of *weighted mean vectors* of the nodes within their respective neighborhoods. Specifically, the reference vector w_i of a node $i \in A$ is updated to the value of

$$w_i = \frac{\sum_{j \in A} |V_j| h_{ji} \bar{y}_j}{\sum_{j \in A} |V_j| h_{ji}}, \tag{3.3}$$

where h_{ji} is a *neighborhood function* that controls how much the mean vector $\bar{y}_j$ of a neighbor node j contributes toward the value of the reference vector of the node i based on the spatial distance between the nodes j and i in the network grid. Thus the neighborhood function h_{ji} controls the degree of lateral interaction among the nodes in the network.

This way of batch updating the reference vectors to the average value of weighted mean vectors of the nodes within the neighborhoods is repeated while decreasing the radius of the neighborhood monotonically. When the neighborhood is reduced to include the node i only, the Batch Map operates in a similar fashion to the Linde-Buzo-Gray vector quantization algorithm [Jean-Claude et al., 2002] and the batch updating of the reference vectors is continued until the reference vectors converge. The proposed model adopts a Gaussian neighborhood function

$$h_{ji}(t) = \exp\left(-\frac{\| l_j - l_i \|^2}{2\sigma^2(t)}\right), \tag{3.4}$$

where $l_j \in \mathbb{N}^2$ and $l_i \in \mathbb{N}^2$ are the *location vectors* of the nodes j and i, respectively, in the rectangular grid space and $\sigma(t)$ is the radius of neighborhood, which is initially set to half the length of the longer side of the current network grid and made to decrease linearly.

The overall saving in the training time by the use of the Batch Map can be significant especially considering that the learning dynamics of the SOM needs to be applied each time the network grid is grown. Furthermore, because the Batch Map

updates the reference vectors with the net effect of processing all training samples, the learning dynamics and the resulting feature map are unaffected by the order of the training samples and yields more stable asymptotic values for the reference vectors without the use of the learning rate parameter to control the rate of reference vector adjustments.

3.2.3 Stopping Condition

Main design goal of the proposed model of self-growing and self-organizing batch map is to autonomously produce a feature map of appropriate size and shape suitable for the data at hand by incremental growing of the network during the training process without requiring the size and shape of the required feature map be specified prior to the training process in order to alleviate the difficulty of predetermining suitable size and shape of the feature map in the application of the SOM. Therefore, the stopping condition for the growing and learning process should not be defined in terms of the configuration of the network structure. Furthermore, the chosen criterion for the stopping condition must be eventually fulfilled to avoid growing the feature map indefinitely.

As the reference vectors are gradually adjusted to the distribution of the training samples by the incremental growing and the competitive learning dynamics of the Batch Map algorithm, not only do the reference vectors of the winners become closer to the values of the training samples for which they are winners, but also the reference vectors of the neighbors of the winners develop similar values. Furthermore, as the network grid is grown, the differences between the values of training data samples and the values of the reference vectors of the nodes in the neighborhood of the winner nodes of the training data samples will be reduced due to the increasing

number of available nodes serving as the prototypes for the training data samples and continued coordinated updating of the reference vectors.

The differences or the distortion errors in the neighborhood of the winner nodes may fluctuate during the early growing and learning cycles when the ordering of the reference vectors takes place, but once the network has grown large enough and the reference vectors have become ordered and stabilized through continued learning cycles, the changes in the distortion errors will be minimized. Therefore, the proposed model uses the convergence of the mean value of the weighted distortion errors in the immediate neighborhood of the winner nodes as the stopping condition for the growing and the learning process. For each training data sample, a weighted distortion error in the immediate neighborhood of the winner node is calculated by giving the mean squared error of the winner node the weight of 50% and taking the mean squared errors of the 8-neighbors for the remaining 50% weight. And the arithmetic mean over the distortion errors of all the training data samples is taken as the measure of stopping criteria. This measure we denote as the mean of weighted distortion error in immediate neighborhood of the winners (MNDE) is defined more formally as

$$MNDE = \frac{1}{|S|}\sum_{x \in S}\left(0.5 \| x - w_i \|^2 + \frac{0.5}{|D|}\sum_{j \in D} \| x - w_j \|^2\right), \tag{3.5}$$

where w_i is the reference vector of the winner node i and w_j is the reference vector of a neighbor node j in the set D defined as

$$D = \{j \in A \mid j \neq i \wedge \| l_j - l_i \| < 2\}. \tag{3.6}$$

The growing and the competitive learning process stops when

$$MNDE(t) < \theta \wedge | MNDE(t) - MNDE(t-1) | < \varepsilon, \tag{3.7}$$

where θ is a degree of error necessary to prevent a premature termination of the growing process that may lead to the problem of undergrowth, and ε is the threshold for the convergence of MNDE.

The value of θ indirectly controls the final size of the feature map and it must be chosen taking the dimension of the input data vectors into account because the dimension makes a significant impact on the distortion error. We have observed by experiments that the following function

$$\theta = 0.5 \times (1 - \exp(-0.02 \times K)), \tag{3.8}$$

where K is the dimension of the input data vectors, along with the value of $\varepsilon = 0.005$ is generally effective in avoiding the problems of undergrowth and overgrowth for data vectors that have been normalized to the values in the range of 0 and 1.

3.2.4 Growing Process

When it is determined that further growth is necessary, more nodes are added to the network and the Batch Map algorithm is applied to the enlarged network structure. There are numerous possibilities for the heuristic mechanisms used to add new nodes and grow the network structure, but the growing mechanism employed by the proposed model must ensure that the final feature map manifested has a two-dimensional rectangular grid structure consistent with the standard Kohonen network as per the design goals of the model. Therefore, the proposed model grows the network structure by inserting a whole row or column of nodes connecting them to their direct neighbors in order to maintain the two-dimensional rectangular grid structure at all times.

To help determine the location of the insertion, a counter is maintained for each node during the Batch Map learning process and whenever a node is found to be

within the radius of the neighborhood of a winner node, its counter is incremented. The node with the highest counter value at the end of the Batch Map process has had its reference vector adapted most frequently hence its vicinity represents the most *crowded* region of the network. Thus a row or column is inserted around a node $p \in A$ with the highest counter value in order to disperse the mappings of the data vectors to a wider area of the grid.

To decide the orientation of the insertion, direct neighbors N_p of p are examined, where the neighbors N_p is defined as

$$N_p = \{c \in A \mid \| l_c - l_p \| = 1 \}, \tag{3.9}$$

to identify a node $q \in N_p$ whose reference vector w_q satisfies the following condition:

$$\| w_q - w_p \| \geq \| w_c - w_p \|, \quad \forall c \in N_p. \tag{3.10}$$

The node q is a direct neighbor of p whose reference vector is the most different from the reference vector of p among the four directly neighboring nodes at the left, top, right and bottom of the node p. If p and q are directly neighboring nodes on a same row, then a column of new nodes (whose length is equal to the height of the current network grid) is inserted between the column positions of p and q. On the other hands, if p and q are on the same column, a new row (whose length is equal to the width of the current network grid) is inserted between the rows of p and q. The reasoning behind this heuristic is that q presumably points to a direction with more variance in the underlying data manifold, and thus a new row or column needs to be inserted between p and q to expand the grid toward the direction of q.

The reference vectors of the newly inserted nodes are interpolated from their two direct neighbors, i.e., left and right neighbors in the case of column insertion, and

top and bottom neighbors in the case of row insertion, by assigning the mean value of reference vectors of the two direct neighbors. Interpolation of the reference vectors of newly inserted nodes from those of the neighbors has the effect of inheriting the feature information learned in the previous batch SOM learning cycle and maintaining the topological relationship among the neighboring nodes.

CHAPTER 4
EXPERIMENTS AND RESULTS

A set of experiments are conducted to assess whether the proposed model of the self-growing and self-organizing batch map is capable of autonomously yielding a feature map of appropriate size and shape suitable for a given data set. The assessment is done by evaluating the qualities of the feature maps generated by the proposed model and by comparing them with the qualities of the feature maps produced by the standard SOM algorithm using two well known data sets.

In this chapter, the methodology adopted for the experiments, the quality measures used in the evaluation of the feature maps and the results of the experiments are presented.

4.1 Data sets used for experiments

One of the measures used to compare the qualities of the feature maps in our experiments is the ability of the feature maps to correctly identify the data class that test data belong to. Therefore, we employ data sets containing data samples that have been properly labeled with classes the data samples belong to in our experiments. It must be emphasized that the class information and labels are used only for the purpose of evaluation and comparison of the final feature maps and are never used in the formation of the feature maps during the SOM learning process.

The two data sets used in our experiments are commonly cited in research work in the field of machine learning and pattern recognition and have been selected to represent varying complexity of the data manifolds they represent. Descriptions and

characteristic details of the two data sets used in our experiments are summarized below.

4.1.1 Iris data set

Iris data set [Frank and Asuncion, 2010] is the most referred data set in the pattern recognition and machine learning literature according to the UCI machine learning repository. It is also used widely in various SOM research work.

The data set contains 3 data classes of 50 instances of each, where each class refers to a type of iris plant. One class is linearly separable from the other two classes, but the remaining two classes are not linearly separable from each other. Each data instance is described by four numeric attribute values that represent sepal length, sepal width, petal length and, petal width all in centimeters of an iris plant.

Iris data set is chosen as a representative of simple data manifolds in our experiments. Characteristics of the iris data set are summarized in Table 4-1.

Table 4-1 Characteristics and data distribution of iris data set

Total Number of Data Instances	150	
Number of Attributes	4	
Number of Classes	3	
Attribute Information		
Attribute Number	Description of Attribute	Value Type
1	sepal length in centimeters	Real
2	sepal width in centimeters	Real
3	Petal length in centimeters	Real
4	Petal width in centimeters	Real
Class Information and Data Distribution		
Class Number	Class Label	Number of samples in the class
1	Iris Setosa	50
2	Iris Versicolour	50
3	Iris Virginica	50

4.1.2 Italian Olive Oil Data Set

Italian olive oil data set [Jure and Johann, 1999] had been extensively investigated by various statistical and clustering methods and their results had been mainly published in the journals of chemometrics and analytical chemistry [Jure and Johann, 1999]. J. Zupan, M. Novic, X. Li and J. Gasteiger [Jure et al., 1994] used the data set to compare the classification abilities of back-propagation neural network method and Kohonen's SOM method and found that for the applied classification problem, Kohonen's SOM method was superior. Since then, this data set has featured frequently in the SOM literature.

The data set consists of analytical data of 572 instances of olive oil samples collected from 9 different regions of Italy. For each oil sample, a chemical analysis determined the percentage of eight different fatty acids. Because the proportion of some fatty acids may differ by two orders of magnitude, all values belonging to a given variable were normalized. These normalized values of percentage of the eight fatty acids form the attributes of the data set and the 9 regions of Italy from where the olive oil samples were collected represent the data classes of the sample data.

Italian olive oil data set is chosen as a representative of moderately complex data manifolds with larger number of data instances in our experiments. Characteristics of the Italian olive oil data set are summarized in Table 4-2.

Table 4-2 Characteristics and data distribution of Italian olive oil data set

Total Number of Data Instances	572	
Number of Attributes	8	
Number of Classes	9	
Attribute Information		
Attribute Number	Description of Attribute	Value Type
1	Palmitic fatty acid	Real
2	Palmitoleic fatty acid	Real
3	Stearic fatty acid	Real
4	Oleic fatty acid	Real
5	Linoleic fatty acid	Real
6	Arachidic fatty acid	Real
7	Linolenic fatty acid	Real
8	Eicosenoic fatty acid	Real
Class Information and Data Distribution		
Class Number	Class Label	Number of samples in the class
1	North Apulia	25
2	Calabria	56
3	South-Apulia	206
4	Sicily	36
5	Inner Sardinia	65
6	Coastal Sardinia	33
7	East Liguria	50
8	West Liguria	50
9	Umbria	51

4.2 Overall procedure for experiments

For each data set used to assess the performance of the proposed model, experiments are repeated three times using three different pairs of training data set and test data set constructed by a random selection process of the samples in the data set to account for possible variations in the distribution of the data samples in the training data set and the test data set.

For each experiment using a particular pair of the training data set and the test data set, a feature map is first generated by the proposed model using the training data set. The proposed model autonomously determines an appropriate size and shape of

the network for the given training data set by incremental growing and learning process and generates a feature map of certain size and shape. Then the standard SOM algorithm is applied to a network having the same size and shape as the feature map generated by the proposed model using the same training data set. Furthermore, to ascertain that the stopping condition used in the proposed model does not cause the problem of undergrowth leading to production of inferior feature maps, the standard SOM algorithm is applied to a network grid twice the size. Quality of the three feature maps produced by the proposed model and the standard SOM algorithm are then evaluated using the test data set and compared along with the overall training times taken. The overall steps for the experiments conducted are summarized in Figure 4-1.

To prevent any attribute from dominating the Euclidean distance and to constrain the values of the distortion error to a recognized range, the values of the data samples are first scaled to be in the range of 0.0 and 1.0 by min-max normalization and the normalized data samples are used to construct a pair of training data set and a test data set.

The training data set is made up of 90% of the data samples randomly selected from each class in the data set, and the remaining 10% are placed into the test data set. Additional test samples are included by replicating random 5% of the training data set into the test data set. The random selection process first identifies the number of unique class labels in the original data set and counts the number of data sample in each class. Then data samples from each class are randomly selected into a training data set and a test data set ensuring that required ratio of the data samples from each and every class are present in the training data set and the test data set. While at it, the selection process also transforms each unique class label into a unique class

identification number to make the class matching during the quality evaluation simpler.

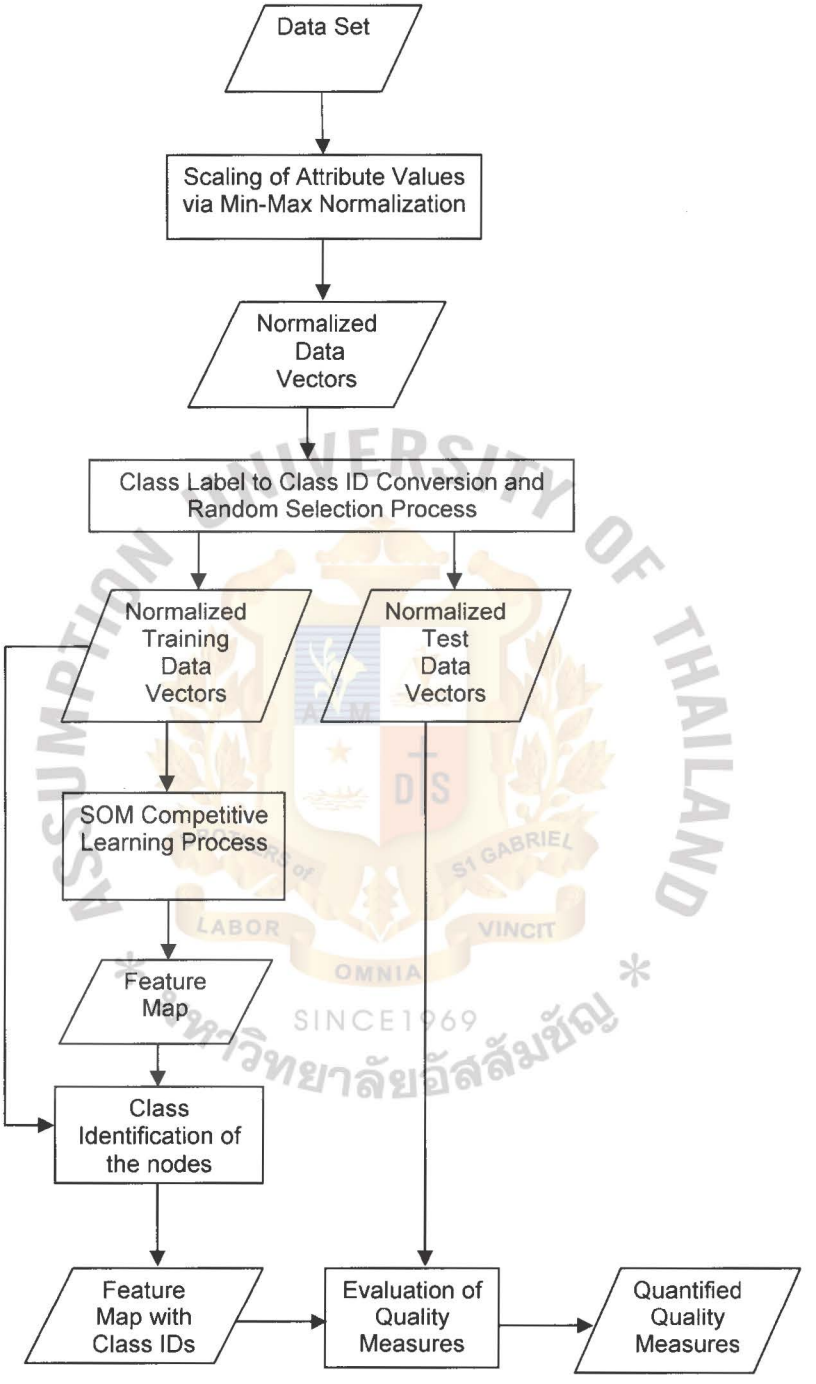


Figure 4-1 Summary of experiment procedure and processes

Then the feature maps are generated by the proposed model and standard SOM algorithm using the training data set and the qualities of the feature maps

generated are evaluated using the corresponding test data set. Two of the quality measures used in our experiments require associating each node of the feature map with a data class that its reference vector represents: one for measuring how well the feature maps correctly identify the classes of the test data samples and the other for measuring the relative similarity of neighboring clusters of data classes in feature maps. Therefore, prior to the evaluation of the quality measures, each node of the feature map is assigned the identification number of the data class category that the reference vector of the node is supposed to represent. To determine the class category of a node on the feature map, every data vector in the *training data set* is compared to the reference vector of the node in search of the training data vector which is closest to the reference vector. *The node is then assigned the class identification number of the training data vector most similar to the reference vector of the node* and this process is repeated for each and every node on the feature map.

4.3 Quality measures used for evaluation of feature maps

As discussed in the Chapter 2, there exists no universally accepted standard measure that can used for objective evaluation and comparison of quality of the feature maps, especially for feature maps of different sizes, due to the lack of mathematical formalism that describes an objective function computed by the SOM learning process. Since the intention of the experiments is to assess if the proposed model is capable of autonomously growing the network to an appropriate size and shape suitable for a given data and yield a feature map whose quality is comparable to the feature maps produced by the standard SOM algorithm, quality measures used for the purpose of the experiments need not be *absolute* indicators of quality. The quality

measures just need to be *relative* indicators of whether feature maps being evaluated bear comparable and similar qualities.

We use three quantified measures for evaluating and comparing the quality of the feature maps in our experiments. Two measures, mean of weighted quantization errors in immediate neighborhoods of the winners (MNQE) and similarity ratio of neighboring clusters (SRNC), are intended to provide postulatory measures of the quality while the rate of correct class identification (RCCI) is designed to indicate an application-centric measure of the quality.

The MNQE measures how well and close enough the values of test data samples falling on a particular Voronoi region of the input data space are represented by the reference vectors in the corresponding neighborhood of the feature map space by determining the weighted mean of differences between the values of the test data samples and the reference vectors in the neighborhood of the node whose reference vector is most similar to the test data vector. Thus the *MNQE quantifies the accuracy of vector quantization and the degree of local ordering of the reference vectors of a feature map*. The MNQE is calculated in a similar way to the MNDE, but for the test data set T:

$$MNQE = \frac{1}{|T|}\sum_{x \in T}\left(0.5 \| x - w_i \|^2 + \frac{0.5}{|D|}\sum_{j \in D} \| x - w_j \|^2\right), \tag{4.1}$$

where w_i is the reference vector of the winner node i and w_j is the reference vector of a neighbor node j in the set D defined as

$$D = \{j \in A \mid j \neq i \ \wedge \| l_j - l_i \| < 2\}. \tag{4.2}$$

The *SRNC*, on the other hand, *attempts to quantify the degree of global ordering of the reference vectors* by assessing the similarity in the neighboring

clusters of data classes portrayed by the feature maps. As the optimal topological state of a feature map is unknown, we compare the relative topological states of the feature maps produced by the proposed model and the standard SOM algorithm by comparing the neighborhood relationships that exist among the clusters of data classes represented by the reference vectors of the feature maps.

For each data class represented by the reference vector of a node on a feature map, a set of neighboring data classes are identified by examining the data classes of directly neighboring nodes and adding the data classes of the neighbors that are different from the data class of itself to the set of its neighboring data classes. This process of identifying neighboring data classes of every node in a feature map culminates in the construction of a list of neighboring data clusters for the data classes represented in the feature map projecting a global order of the topology of the reference vectors of the feature map.

Degree of similarity in the global order of the reference vectors in the feature maps produced by the proposed model and the standard SOM algorithm are then evaluated by comparing the lists of neighboring data clusters of the two feature maps and quantifying the ratio of similarity and averaging them over all the data classes. This measure of similarity ratio of neighboring clusters or SRNC is more specifically defined as

$$SRNC = \frac{1}{K}\sum_{i=1}^{K}\frac{|P_i \cap S_i|}{|P_i \cup S_i|}, \tag{4.3}$$

where K is the number of data classes in the training data samples, P_i is the set of neighboring data classes for the data class i in the feature map produced by the proposed model and S_i is the set of neighboring data classes for the same data class i in the feature map produced by the standard SOM algorithm.

Therefore, the combination of MNQE and SRNC represent hypothetical qualities of the feature maps indicating the level of accuracy of vector quantization and the degree of topology preservation in the transformation of a continuous high dimensional data manifold into a discrete two-dimensional feature map. However, these postulatory measures of quality may not reflect how well the feature maps serve the purpose of a particular application of the SOM in practice. Since the major applications of the SOM are predominantly in the field of data mining for clustering and classification tasks, we also evaluate the quality of the feature maps by quantifying how well the generated feature maps identify the correct classes of the data they belong to *as an application-centric quality* of the feature maps. This application-centric measure denoted as the RCCI is a percentage of the test data samples whose class labels match the class labels of their respective winner nodes and is thus defined as

$$RCCI = \frac{1}{|T|}\sum_{x \in T} cc(x), \tag{4.4}$$

where $cc(x) = 1$ if the class label of x matches the class label of the winner node for x and 0 otherwise.

It must be clarified that we use data sets that have been properly labeled with the data classes they belong to in our experiments only to evaluate the quality of the generated feature maps in terms of the RCCI and the SRNC. The class label information is not used during the training process in the formation of the feature maps. Therefore, the use of classified data sets in our experiments is a means for evaluating classification and clustering capabilities of the feature maps formed

through unsupervised learning process using the classified data sets, and should not be confused with the supervised learning models.

4.4 Experiment results

In this section, the results of the experiments using the iris data set and the Italian olive oil data set are presented summarizing the quality measures of the feature maps produced by the proposed model and the standard SOM algorithm along with their training times taken on a machine with a Core 2 CPU and 2 GB of memory.

The standard SOM model requires the size and the shape of the network grid to be defined prior to the training process. In evaluating and comparing the qualities of the feature maps, the standard SOM algorithm is applied to network grids having the same sizes and the shapes as the final feature maps generated by the proposed self-growing and self-organizing batch map model and the qualities of the resulting feature maps are evaluated and compared to those of corresponding feature maps generated by the proposed model. Furthermore, we also apply the standard SOM algorithm to a larger grid consisting of twice the number of the nodes to ascertain that the stopping condition used in the proposed model does not cause the problem of undergrowth leading to production of inferior feature maps.

All results produced by the standard SOM algorithm have been obtained by setting the initial radius of neighborhood to half the length of the longer dimension of the two-dimensional network grid and the initial learning rate to 0.9 and decreasing the radius of the neighborhood and the learning rate linearly over 2,000 epochs.

The feature maps and their quality measures presented in this section are obtained by running the same experiment using a particular pair of training data set

and test data set three times and taking the output of the run with the best quality measures out of the three runs in order to counter the possible effects on the learning dynamics that might be caused by the random selection of data vectors used to initialize the reference vectors of the initial network grid.

4.4.1 Results of experiments on the iris data set

150 data vectors in the iris data set are randomly selected to construct a pair of a training data set consisting of 135 data vectors and a test data set consisting of 21 data vectors. The random selection process ensures that equal number of instances of each data class is present in both the training set and the test set. Table 4-3 summarizes the data class label to class ID mapping and the distribution of the data vectors used for the experiments.

Table 4-3 Class ID mapping and data distribution used for experiments using iris data set

Data Class Label	Data Class ID	Number of Data Vector in the Class	Number of Training Vectors	Number of Test Vectors
Iris Setosa	1	50	45	7
Iris Versicolour	2	50	45	7
Iris Virginica	3	50	45	7

Tables 4-4 to 4-9 shows the results the three experiments on the iris data set using three random pairs of training data set and test data set. The MNQE and the RCCI of the feature maps produced by the proposed model and standard SOM algorithm along with their respective training times taken to produce the feature maps are summarized in Table 4-4, Table 4-6 and Table 4-8, respectively for the three experiments on the iris data set. Table 4-5, Table 4-7 and Table 4-9 summarizes the neighboring data clusters of each data class portrayed in the feature maps produced by the proposed model and the standard SOM algorithm and shows the SRNC between

the feature map produced by the proposed model and the feature map of the same size and shape produced by standard SOM algorithm and the SRNC between the feature map produced by the proposed model and the larger feature map produced by the standard SOM algorithm on a network having twice the number of nodes.

Table 4-4 MNQE, RCCI and training times for experiment No. 1 on iris data set

SOM Model	Feature Map Size	Training time	MNQE	RCCI
Proposed Model	26 by 28	99.578	0.0650	1.000
Standard SOM	26 by 28	294.953	0.0566	1.000
Standard SOM	38 by 38	592.329	0.0504	1.000

Table 4-5 Neighboring clusters and SRNC for experiment No. 1 on iris data set

Data Class ID	Neighboring clusters Proposed Model 26 by 28	Neighboring clusters Standard SOM 26 by 28	Similarity Rate with Proposed Model	Neighboring clusters Standard SOM 38 by 38	Similarity Rate with Proposed Model
1	2	2	1.00	2	1.0
2	1, 3	1, 3	1.00	1, 3	1.0
3	2	2	1.00	2	1.0
		SRNC of Proposed Model to Standard SOM 26 by 28	1.0	SRNC of Proposed Model to Standard SOM 38 by 38	1.0

Table 4-6 MNQE, RCCI and training times for experiment No. 2 on iris data set

SOM Model	Feature Map Size	Training time	MNQE	RCCI
Proposed Model	21 by 30	76.515	0.0937	0.952
Standard SOM	21 by 30	255.922	0.0874	0.952
Standard SOM	35 by 35	501.313	0.0799	0.952

Table 4-7 Neighboring clusters and SRNC for experiment No. 2 on iris data set

Data Class ID	Neighboring clusters Proposed Model 21 by 30	Neighboring clusters Standard SOM 21 by 30	Similarity Rate with Proposed Model	Neighboring clusters Standard SOM 35 by 35	Similarity Rate with Proposed Model
1	2	2	1.00	2	1.0
2	1, 3	1, 3	1.00	1, 3	1.0
3	2	2	1.00	2	1.0
		SRNC of Proposed Model to Standard SOM 21 by 30	1.0	SRNC of Proposed Model to Standard SOM 35 by 35	1.0

Table 4-8 MNQE, RCCI and training times for experiment No. 3 on iris data set

SOM Model	Feature Map Size	Training time	MNQE	RCCI
Proposed Model	25 by 30	111.250	0.0760	1.000
Standard SOM	25 by 30	305.360	0.0688	1.000
Standard SOM	38 by 38	593.032	0.0571	1.000

Table 4-9 Neighboring clusters and SRNC for experiment No. 3 on iris data set

Data Class ID	Neighboring clusters Proposed Model 25 by 30	Neighboring clusters Standard SOM 25 by 30	Similarity Rate with Proposed Model	Neighboring clusters Standard SOM 38 by 38	Similarity Rate with Proposed Model
1	2	2	1.00	2	1.0
2	1, 3	1, 3	1.00	1, 3	1.0
3	2	2	1.00	2	1.0
		SRNC of Proposed Model to Standard SOM 25 by 30	1.0	SRNC of Proposed Model to Standard SOM 38 by 38	1.0

The results of the three experiments using the iris data set clearly indicate the qualities of feature maps vary little between the ones generated by the proposed model and the corresponding feature maps produced by the standard SOM algorithm. The MNQE of the feature maps produced by the standard SOM algorithm is slightly

lower than that of the corresponding feature maps produced by the proposed model, but the difference (reduction in MNQE value of less than 0.01 on all cases) is not significant enough to indicate variance in the quality. The feature maps produced by the proposed model and the corresponding feature maps of the same size and shape produced by the standard SOM algorithm show perfect matches in the neighboring data clusters yielding SRNC value of 1.0 and all the feature maps manifest very high RCCI values with very little variance in all of the three experiments. Despite the close similarity in the quality of the feature maps produced, the proposed model takes noticeably less training time (about three times) to grow the network and yield a final feature map than the training time taken by the standard SOM algorithm to form a feature map on a given grid of the same size and shape.

Furthermore, the larger feature maps produced by the standard SOM algorithm on the network grids containing about twice more number of nodes do not show any significant improvements in quality. The MNQE is slightly reduced as expected due to the larger size of the network, but the average reductions about 0.015 in the values of MNQE come at the cost of about twofold increase in the training time. The RCCI of the larger feature maps do not improve on their smaller counterparts and SRNC value of 1.0 indicates that the global order of the topological structure manifest by the reference vectors of the feature map is very much consistent across all the feature maps. The results of the experiments conducted using the iris data set demonstrate that the stopping condition used in the proposed model is effective in determining an appropriate size of the network for a given data set to yield a feature map of comparable quality while avoiding undergrowth and unnecessary overgrowth of the network.

The final feature maps generated by the proposed model and the standard SOM algorithm for the three experiments using the iris data set are exhibited in Figure 4-2 to Figure 4-10 for visual references.

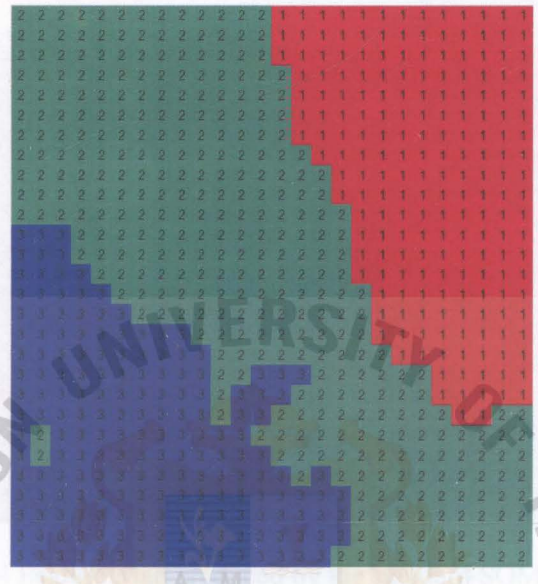


Figure 4-2 Feature Map produced by the proposed model for iris data set experiment No. 1 (26 by 28)

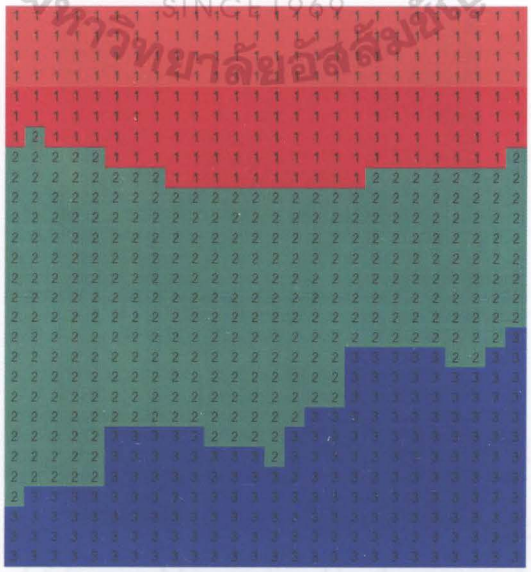


Figure 4-3 Feature Map produced by the standard SOM algorithm for iris data set experiment No. 1 (26 by 28)

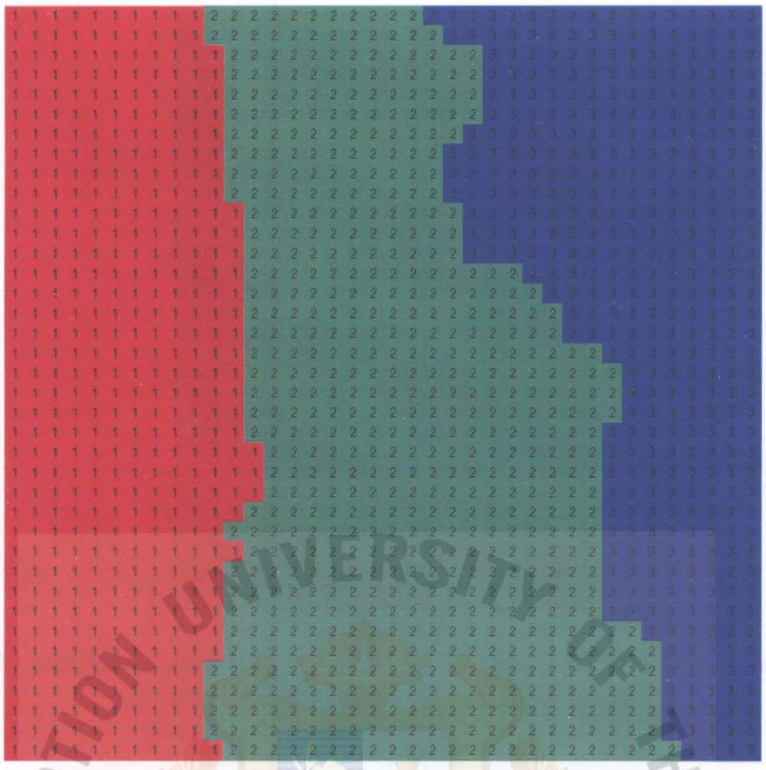


Figure 4-4 Feature Map produced by the standard SOM algorithm for iris data set experiment No. 1 (38 by 38)

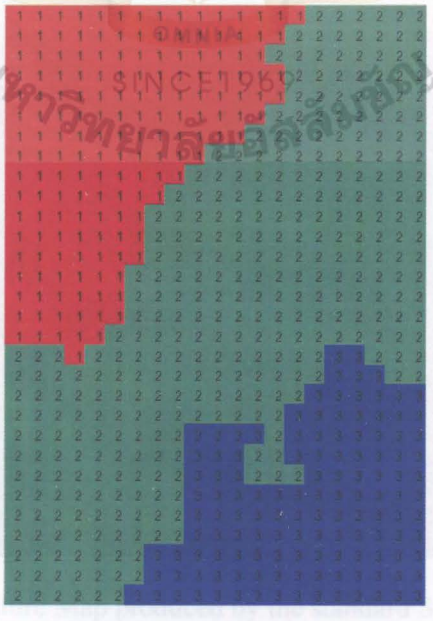


Figure 4-5 Feature Map produced by the proposed model for iris data set experiment No. 2 (21 by 30)

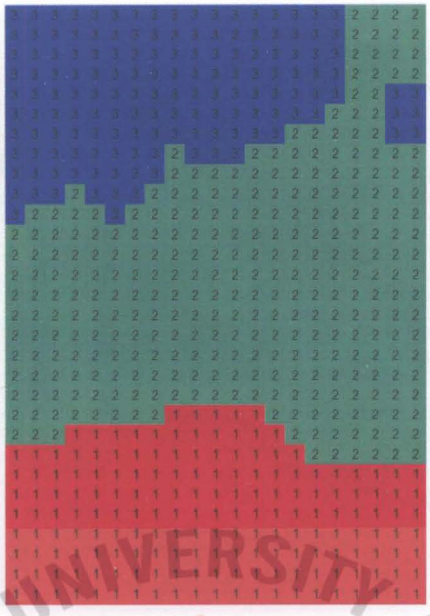


Figure 4-6 Feature Map produced by the standard SOM algorithm for iris data set experiment No. 2 (21 by 30)

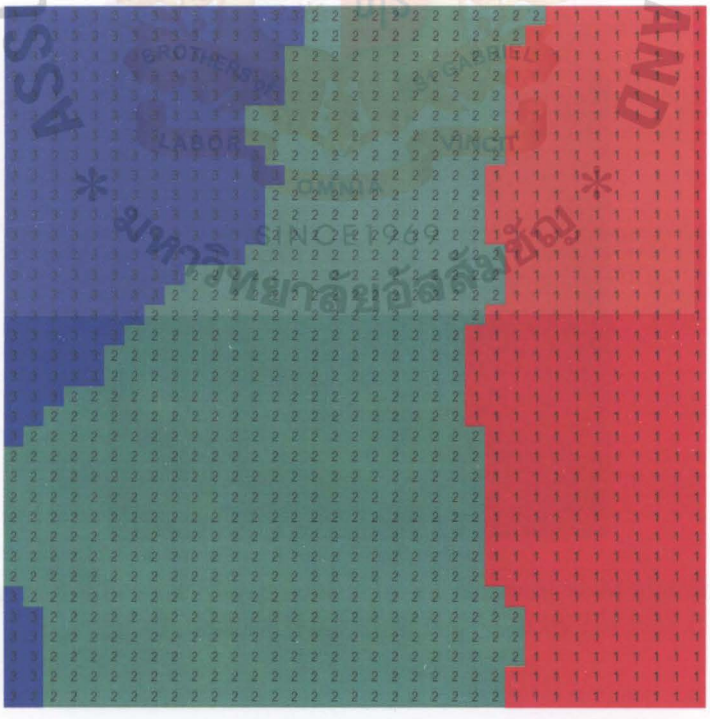


Figure 4-7 Feature Map produced by the standard SOM algorithm for iris data set experiment No. 2 (35 by 35)

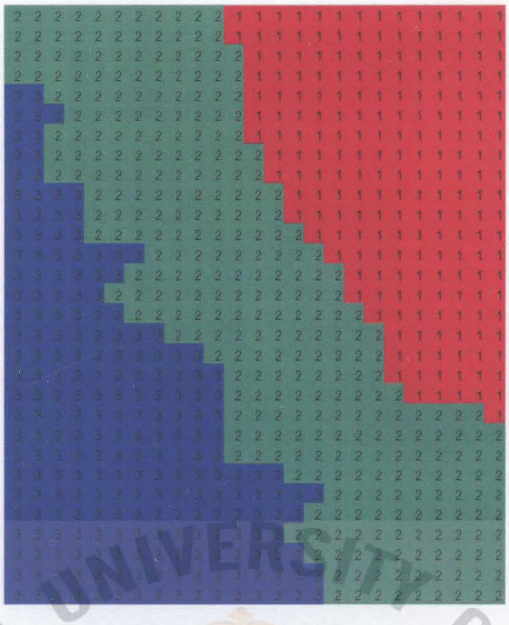


Figure 4-8 Feature Map produced by the proposed model
for iris data set experiment No. 3 (25 by 30)

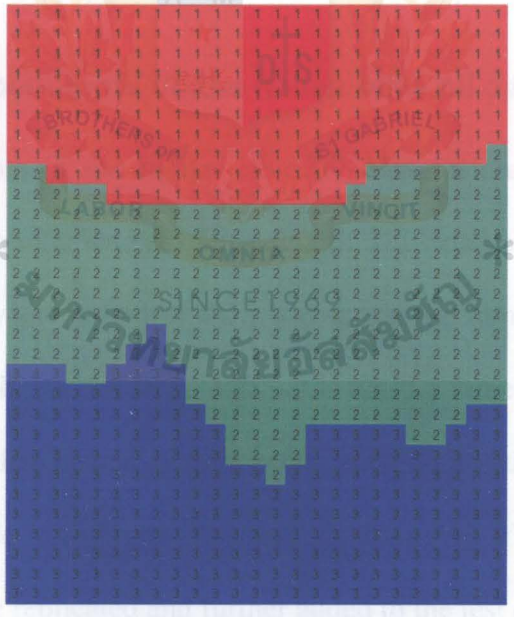


Figure 4-9 Feature Map produced by the standard SOM algorithm
for iris data set experiment No. 3 (25 by 30)

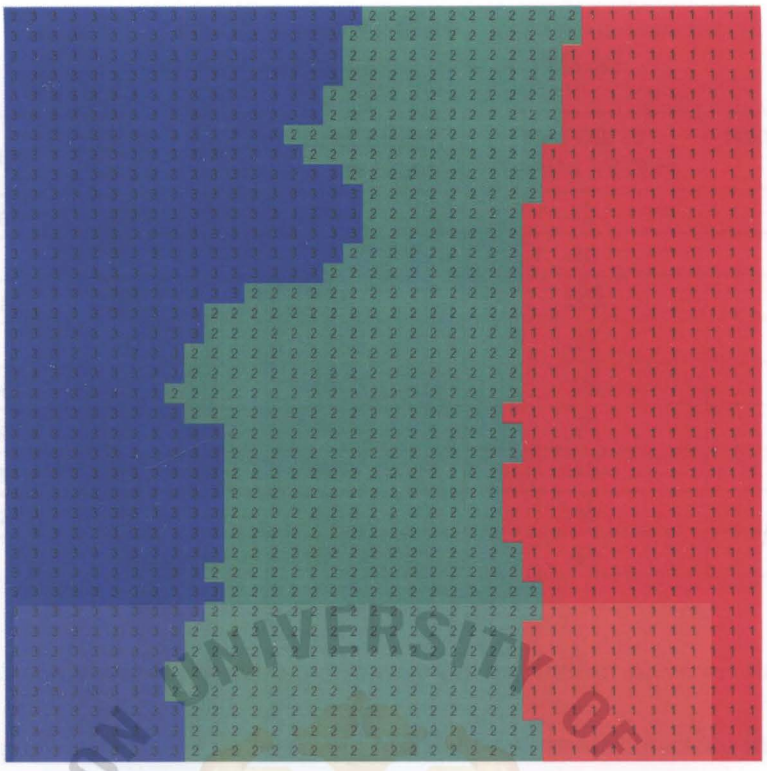


Figure 4.10 Feature Map produced by the standard SOM algorithm
for iris data set experiment No. 3 (38 by 38)

4.4.2 Results of experiments on the Italian olive oil data set

572 data vectors in the Italian olive oil data set are randomly selected to construct a pair of a training data set consisting of 511 data vectors and a test data set consisting of 83 data vectors. Unlike the iris data set, the number of data samples in each data class of the Italian olive oil data set is not the same. The random selection process takes 90% of data instances from each data class for the training data set and the remaining are made into the test data set. Random 5% of the data samples from the training data set are replicated and further added to the test data set.

Table 4-10 summarizes the data class label to class ID mapping and the distribution of the data samples used for the experiments using Italian olive oil data set.

Table 4-10 Class ID mapping and data distribution used for experiments using Italian olive oil data set

Data Class Label	Data Class ID	Number of Data Vector in the Class	Number of Training Vectors	Number of Test Vectors
North Apulia	1	25	22	4
Calabria	2	56	50	8
South Apulia	3	206	185	30
Sicily	4	36	32	5
Inner Sardinia	5	65	58	9
Coastal Sardinia	6	33	29	5
East Liguria	7	50	45	7
West Liguria	8	50	45	7
Umbria	9	51	45	8

The results of the three experiments on the Italian olive oil data set using three random pairs of training data set and test data set are summarized in Tables 4-11 to 4-16. Table 4-11, Table 4-13 and Table 4-15 respectively shows the MNQE and the RCCI of the feature maps produced by the proposed model and standard SOM algorithm along with their respective training times for the three experiments conducted using the Italian olive oil data set. Neighboring data clusters projected by the feature maps for the clusters of data classes represented by the reference vectors of the feature maps are shown in Table 4-12, Table 4-14, and Table 4-16 summarizing the SRNC between the feature map feature map generated by the proposed model and the feature map of the same size and shape produced by the standard SOM algorithm as well as the SRNC between the feature map generated by the proposed model and the larger feature map produced by the standard SOM algorithm on a network grid having twice the number of nodes.

Table 4-11 MNQE, RCCI and training times for experiment No. 1 on Italian olive oil data set

SOM Model	Feature Map Size	Training time	MNQE	RCCI
Proposed Model	40 by 42	2049.39	0.1249	1.000
Standard SOM	40 by 42	4542.06	0.1145	0.988
Standard SOM	58 by 58	9183.91	0.1016	0.988

Table 4-12 Neighboring clusters and SRNC for experiment No. 1 on Italian olive oil data set

Data Class ID	Neighboring clusters Proposed Model 40 by 42	Neighboring clusters Standard SOM 40 by 42	Similarity Rate with Proposed Model	Neighboring clusters Standard SOM 58 by 58	Similarity Rate with Proposed Model
1	2, 4, 7	2, 4, 7, 9	0.7500	2, 4, 7, 9	0.7500
2	1, 3, 4, 5, 7	1, 3, 4, 5, 7	1.0000	1, 3, 4, 7	0.8000
3	2, 4, 5, 6, 8	2, 4, 5, 6, 7	0.6667	2, 4, 5, 7	0.5000
4	1, 2, 3	1, 2, 3	1.0000	1, 2, 3	1.0000
5	2, 3, 6, 7, 8, 9	2, 3, 6, 7, 8	0.8333	3, 6, 7	0.5000
6	3, 5	3, 5	1.0000	5	0.5000
7	1, 2, 5, 8, 9	1, 2, 3, 5, 8, 9	0.8333	1, 2, 3, 5, 8, 9	0.8333
8	3, 5, 7	5, 7, 9	0.5000	7, 9	0.2500
9	5, 7	1, 7, 8	0.2500	1, 7, 8	0.2500
		SRNC of Proposed Model to Standard SOM 21 by 30	**0.7593**	SRNC of Proposed Model to Standard SOM 35 by 35	0.5982

Table 4-13 MNQE, RCCI and training times for experiment No. 2 on Italian olive oil data set

SOM Model	Feature Map Size	Training time	MNQE	RCCI
Proposed Model	45 by 35	1883.77	0.1345	0.988
Standard SOM	45 by 35	4267.27	0.1281	0.988
Standard SOM	56 by 56	8553.58	0.1116	0.988

Table 4-14 Neighboring clusters and SRNC for experiment No. 2 on Italian olive oil data set

Data Class ID	Neighboring clusters Proposed Model 45 by 35	Neighboring clusters Standard SOM 45 by 35	Similarity Rate with Proposed Model	Neighboring clusters Standard SOM 56 by 56	Similarity Rate with Proposed Model
1	2, 3, 4, 9	2, 3, 4, 9	1.0000	2, 3, 4, 9	1.0000
2	1, 3, 4, 7, 9	1, 3, 4, 5, 7, 9	0.8333	1, 3, 4, 5, 7	0.6667
3	1, 2, 4, 5, 6, 7, 9	1, 2, 4, 5, 6	0.7143	1, 2, 4, 5, 6, 7, 9	1.0000
4	1, 2, 3	1, 2, 3	1.0000	1, 2, 3	1.0000
5	3, 6, 7, 8	2, 3, 6, 7, 8	0.8000	2, 3, 6, 7, 8	0.8000
6	3, 5	3, 5	1.0000	3, 5	1.0000
7	2, 3, 5, 8, 9	2, 5, 8, 9	0.8000	2, 3, 5, 8, 9	1.0000
8	5, 7	5, 7	1.0000	5, 7	1.0000
9	1, 2, 3, 7	1, 2, 7	0.7500	1, 3, 7	0.7500
		SRNC of Proposed Model to Standard SOM 45 by 35	**0.8775**	SRNC of Proposed Model to Standard SOM 56 by 56	0.9130

Table 4-15 MNQE, RCCI and training times for experiment No. 3 on Italian olive oil data set

SOM Model	Feature Map Size	Training time	MNQE	RCCI
Proposed Model	43 by 36	1713.55	0.1252	1.000
Standard SOM	43 by 36	4185.95	0.1126	0.988
Standard SOM	57 by 57	8863.30	0.0991	0.976

Table 4-16 Neighboring clusters and SRNC for experiment No. 3 on Italian olive oil data set

Data Class ID	Neighboring clusters Proposed Model 43 by 36	Neighboring clusters Standard SOM 43 by 36	Similarity Rate with Proposed Model	Neighboring clusters Standard SOM 57 by 57	Similarity Rate with Proposed Model
1	2, 3, 4, 7, 9	2, 3, 4, 9	0.8000	2, 3, 4, 9	0.8000
2	1, 3, 4, 7	1, 3, 4, 7, 9	0.8000	1, 3, 4, 5, 7	0.8000
3	1, 2, 4, 5, 6, 7	1, 2, 4, 5, 6, 7	1.0000	1, 2, 4, 5, 6, 7, 9	1.0000
4	1, 2, 3, 7	1, 2, 3	0.7500	1, 2, 3	0.6000
5	3, 6, 7, 8	3, 6, 7, 8	1.0000	2, 3, 6, 7, 8	1.0000
6	3, 5	3, 5	1.0000	3, 5	1.0000
7	1, 2, 3, 4, 5, 8, 9	2, 3, 5, 8, 9	0.7143	2, 3, 5, 8, 9	0.7143
8	5, 7, 9	5, 7, 9	1.0000	5, 7	1.0000
9	1, 4, 7, 8	1, 2, 7, 8	0.8293	1, 3, 7	0.8333
		SRNC of Proposed Model to Standard SOM 45 by 35	**0.8516**	SRNC of Proposed Model to Standard SOM 56 by 56	0.8571

The results of the experiments comparing the qualities of the feature maps produced by the proposed model of self-growing and self-organizing batch map and those produced by the standard SOM algorithm on the Italian olive oil data set clearly indicate that the proposed model is capable of generating feature maps exhibiting similar qualities to those that can be produced by the standard SOM algorithm even for more complex data manifolds while offering significant savings in the overall training times.

The MNQE of the feature maps produced by the standard SOM algorithm is slightly lower than that of the corresponding feature maps produced by the proposed model, but the reduction (of less than 0.01 on average) in the values of the MNQE is not significant enough to indicate variance in the quality. Equally high RCCI values are exhibited by all the feature maps with the proposed model showing slightly higher values in experiment No. 1 and experiment No. 3. The SRNC values between the feature maps generated by the proposed model and the corresponding feature maps of the same size and shape produced by the standard SOM algorithm suggest that the feature maps portray similar topological orders in terms of neighboring data clusters.

Considering the fact that a mismatch of neighboring data cluster in one data class will also cause mismatches in the corresponding data classes lowering the similarity rates in multiple data classes, the SRNC values shown for the experiments on the Italian olive oil data set that is composed of 9 different data classes should be considered as indicating pretty high level of similarity in the overall topological orders of the feature maps. The SRNC values for experiment No. 2 and experiment No. 3 are especially high indicating high level of similarity in the overall global topological orders of the reference vectors. Although the SRNC value of 0.7593 for experiment No. 1 is lower in comparison, close observation reveals that the lower

SRNC value is mainly due to one or two mismatches in the neighboring clusters of the data class 8 and the data class 9 which have small number of neighboring data clusters (between 2 and 3 only) thus leading to very low similarity rates of 0.5 and 0.25.

The observation of the training times taken to generate the feature maps clearly indicates that the proposed model takes significantly less training time to grow the network and yield a final feature map than for the standard SOM algorithm to form a feature map on a given grid of the same size and shape. The savings in the training time achieved by the proposed model is noteworthy especially considering that actual wall time required for training more complex and larger data sets are generally a lot longer therefore offering larger reduction in training time in terms of wall time saved.

Quality of the larger feature maps produced by the standard SOM algorithm do not show any clear indication of improvements over the feature maps generated by the proposed model. As expected, the MNQE of the feature maps produced by the standard SOM algorithm on the larger network grids is slightly reduced due to the larger network size, but the small reductions in the MNQE are obtained at the cost of considerable increase in the training time especially in terms of wall time. The RCCI of the larger feature maps do not improve on their smaller counterparts and in experiment No. 3 on the Italian olive oil data set, the RCCI of the larger feature map is shown to be slightly worse, presumably caused by overfitting. The isolated case of reduced value of SRNC shown for the larger feature map in experiment no. 1 is presumably caused by the increase in the number of low similarity rates induced by mismatches in those data classes with small number of neighboring data clusters due to the larger network topology causing further mismatches from the smaller feature

map of proposed model. However, the SRNC between the feature maps generated by the proposed model and the larger feature map produced by the standard SOM algorithm show very similar or slightly improved values in experiment no. 2 and experiment no. 3 when compared to the SRNC values between feature maps of the same sizes indicating general consistency in the topological orders of the reference vectors.

The results of the experiments on the Italian olive oil data set affirm that the proposed model is effective in growing the network autonomously to an appropriate size and shape suitable for a given data manifold even for more complex and larger data sets and is capable of yielding feature maps of comparable qualities while avoiding undergrowth that may lead to production of inferior quality feature maps and overgrowth which prolongs the training time unnecessarily without any significant improvements in the quality of the feature maps.

The final feature maps generated by the proposed model and the standard SOM algorithm for the three experiments using the Italian olive oil data set are exhibited in Figure 4-11 to Figure 4-19 for visual references.

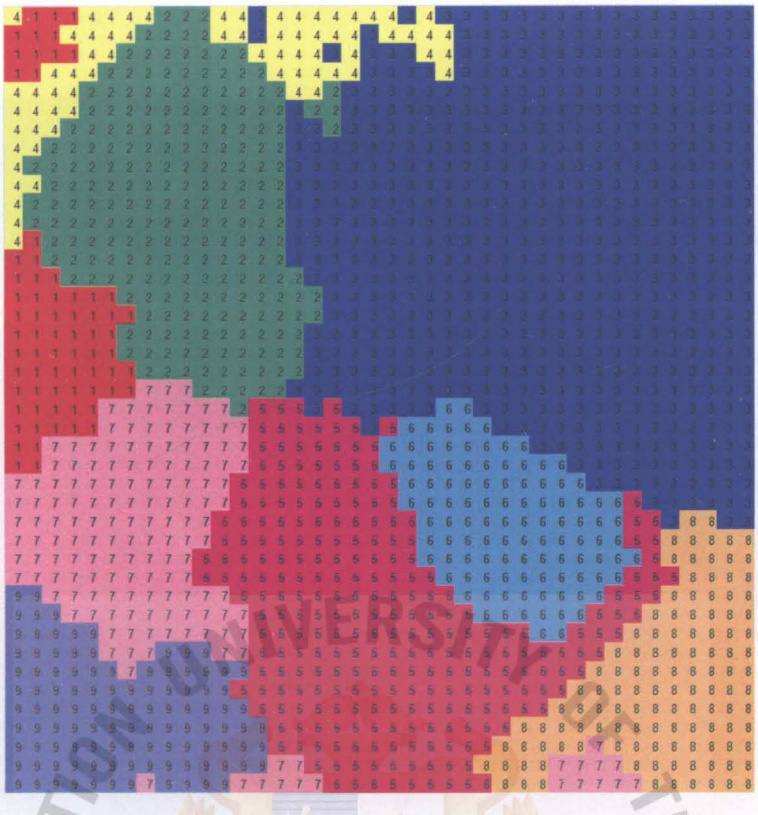


Figure 4-11 Feature Map produced by the proposed model for Italian olive oil data set experiment No. 1 (40 by 42)

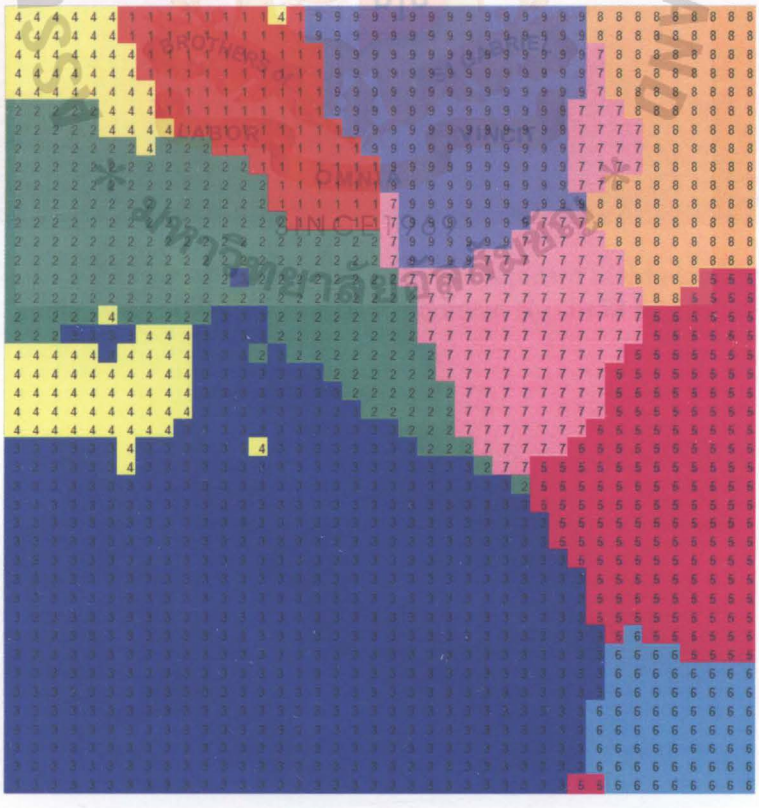


Figure 4-12 Feature Map produced by the standard SOM algorithm for Italian olive oil data set experiment No. 1 (40 by 42)

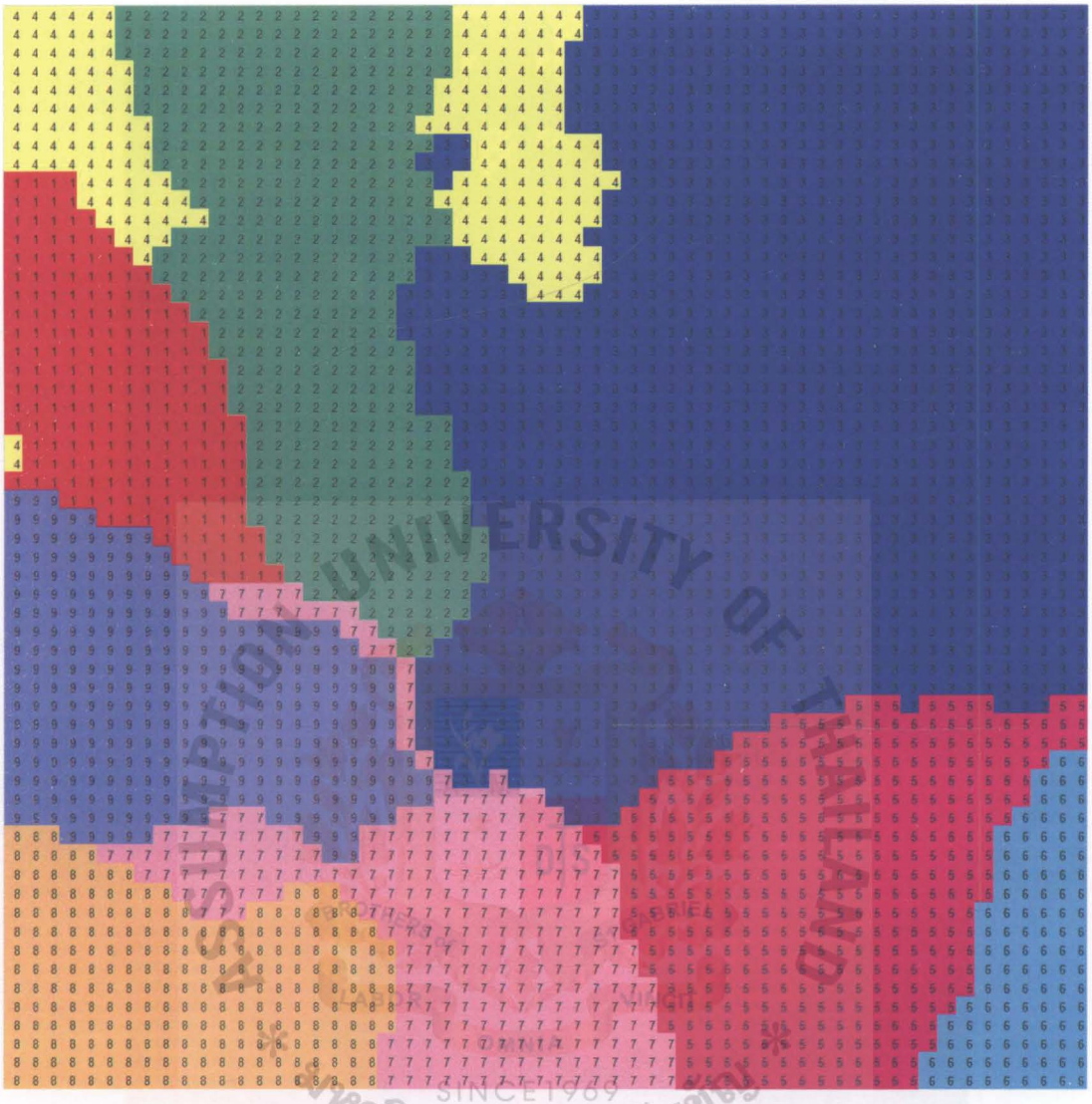


Figure 4-13 Feature Map produced by the standard SOM algorithm
for Italian olive oil data set experiment No. 1 (58 by 58)

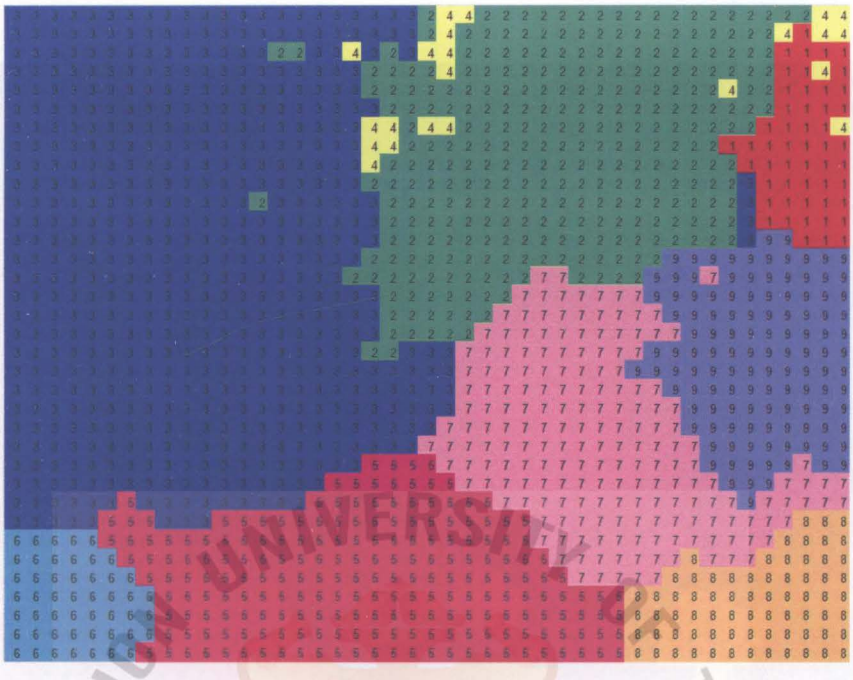


Figure 4-14 Feature Map produced by the proposed model
for Italian olive oil data set experiment No. 2 (45 by 35)

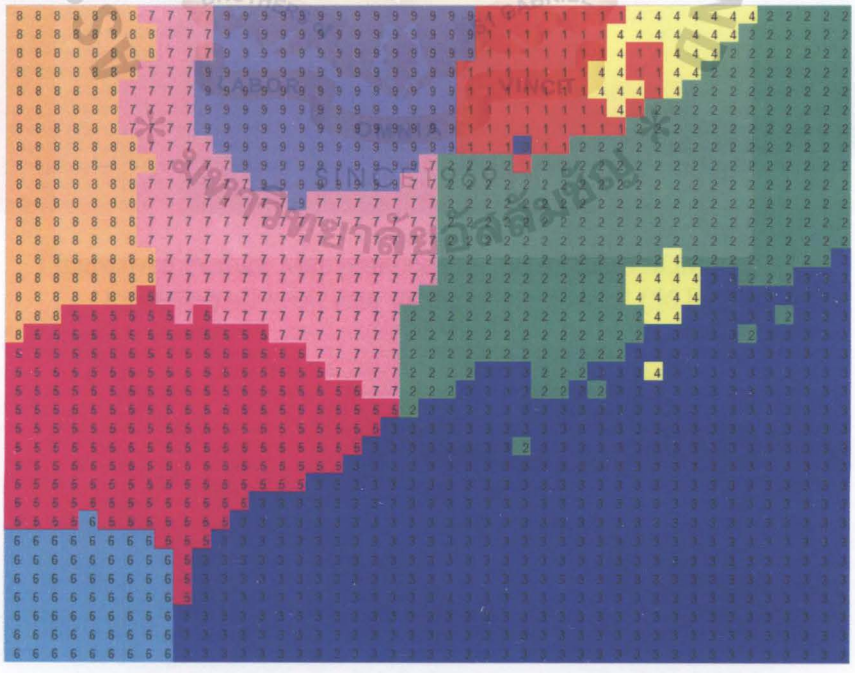


Figure 4-15 Feature Map produced by the standard SOM algorithm
for Italian olive oil data set experiment No. 2 (45 by 35)

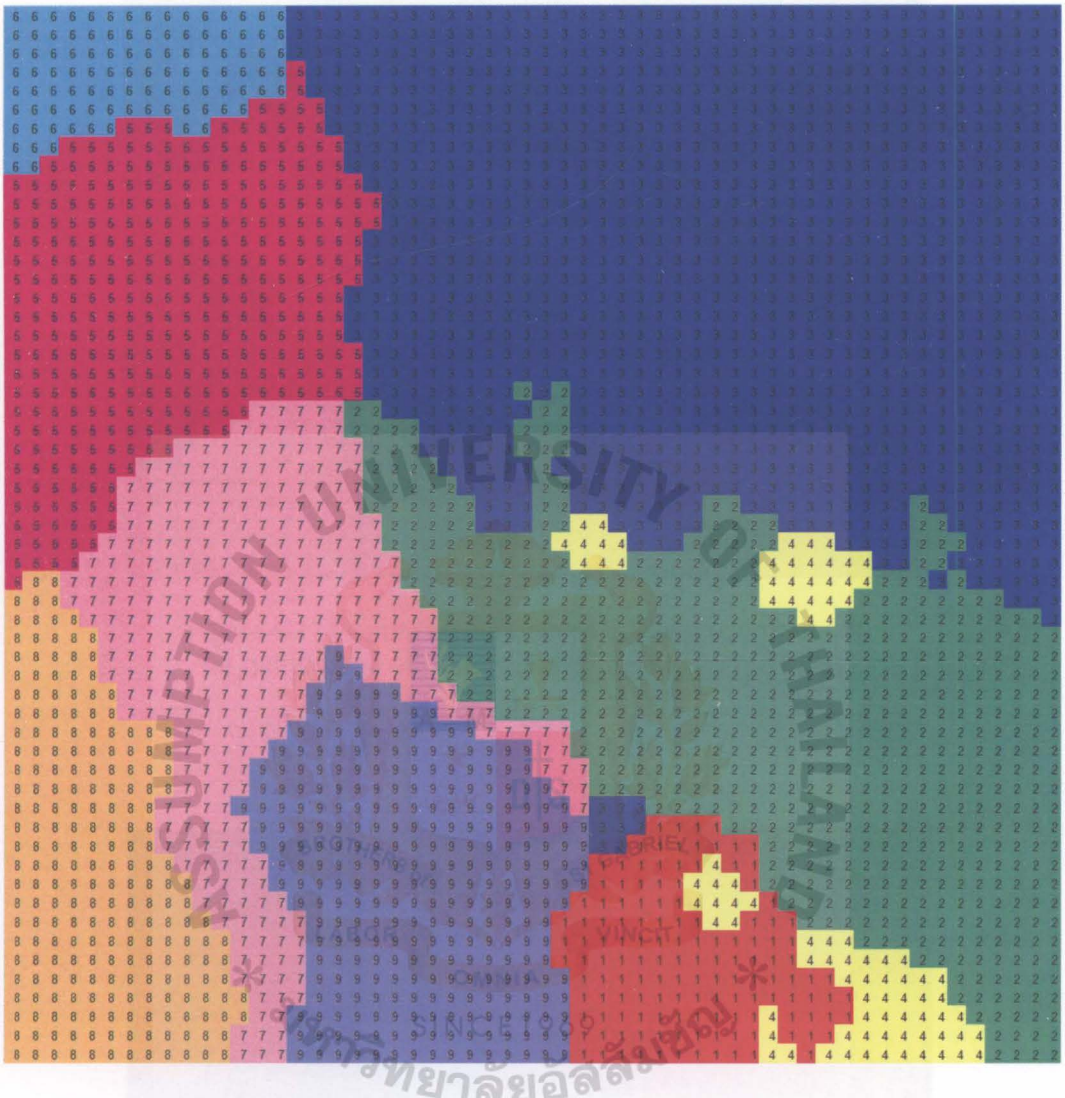


Figure 4-16 Feature Map produced by the standard SOM algorithm for Italian olive oil data set experiment No. 2 (56 by 56)

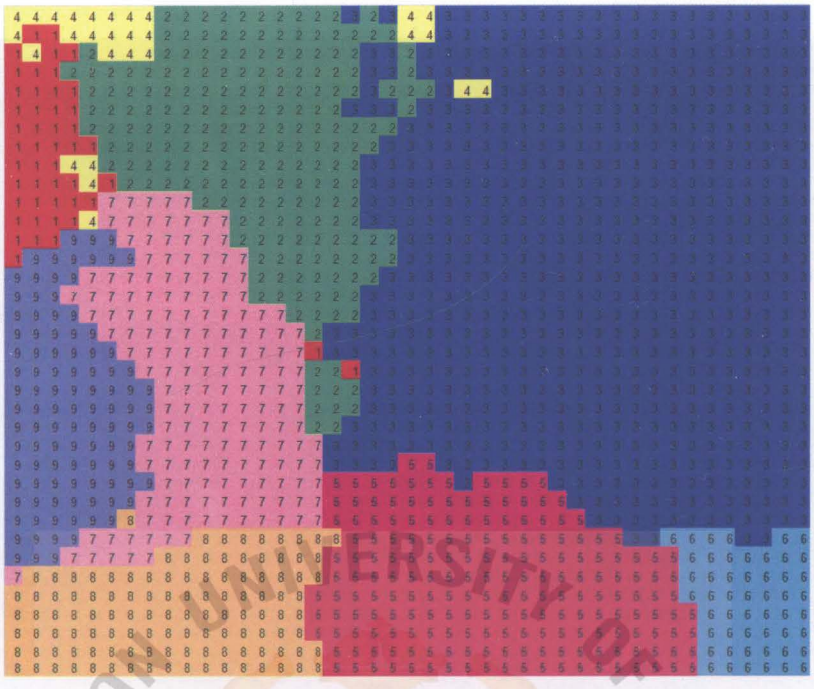


Figure 4-17 Feature Map produced by the proposed model
for Italian olive oil data set experiment No. 3 (43 by 36)

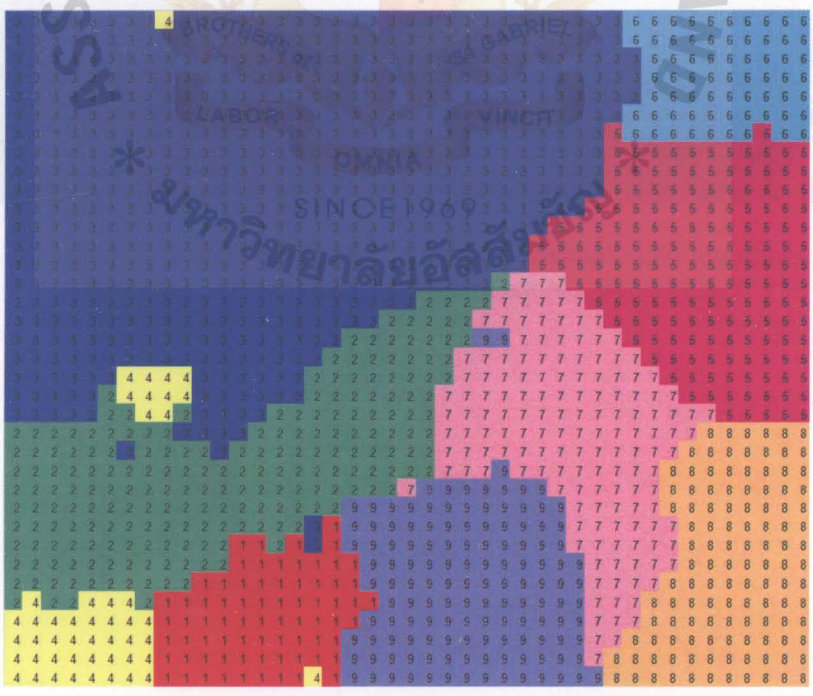


Figure 4-18 Feature Map produced by the standard SOM algorithm
for Italian olive oil data set experiment No. 3 (43 by 36)

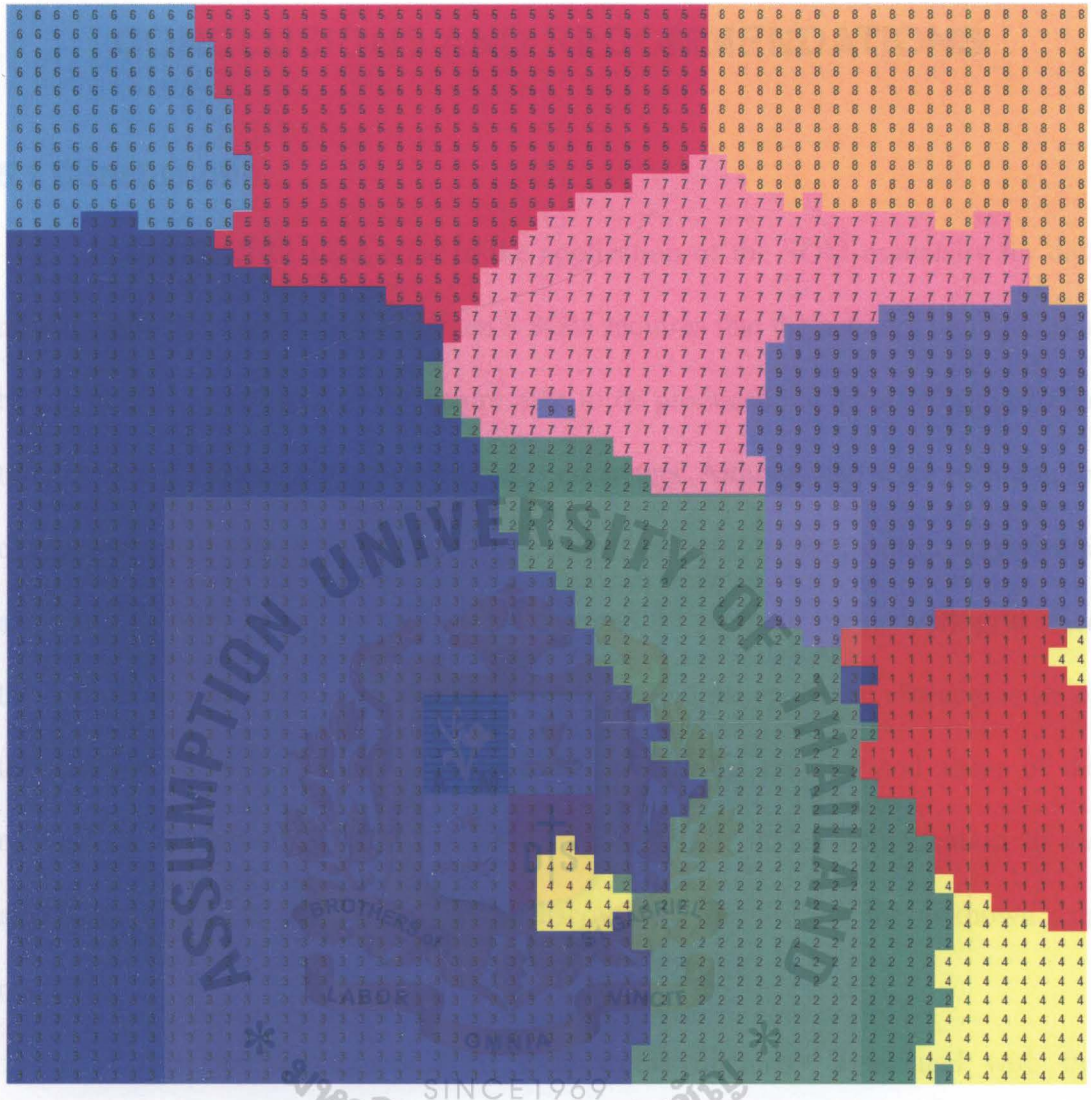


Figure 4.19 Feature Map produced by the standard SOM algorithm
for Italian olive oil data set experiment No. 3 (57 by 57)

CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

Pragmatic difficulties arise in the application of the SOM especially for data mining problems because the Kohonen's SOM requires that the network configuration, hence the size and shape of the feature map to be produced, be specified prior to the training process along with other learning parameters. Although there are common set of recommended values for the learning parameters based on numerous experimental analyses, the same cannot be said for the size and the shape of the network grid since the appropriate network configuration is dependent on the statistical characteristics of the data set. Therefore, the requirement to predetermine the size and shape of the network in the standard SOM algorithm represents an undue complication to the experimenter who is using the SOM to discover the structure of the unknown data manifold.

In this thesis, we sought to address the difficulty associated with the application of the SOM induced by the requirement to specify the size and the shape of the feature map prior to the training process in the standard SOM algorithm by investigating ways to grow the network grid incrementally during the training process until a feature map deemed suitable for a given data manifold have formed. The proposed model of self-growing and self-organizing map construct a feature map suitable for the input data by incremental growing of the network structure and gradual adaptation of the reference vectors by coordinated competitive learning dynamics of the Batch Map during the training process. It is a growing model of the SOM designed to automatically determine the size and shape of a feature map

appropriate for the data while maintaining the structural compatibility to the rectangular grid network of the standard Kohonen network. The application of the Batch Map is also particular to the proposed model and has allowed significant savings in the overall training time.

The experiments carried out using the iris data set and the Italian olive oil data set have demonstrated the capability of the proposed model to grow the network dynamically during the training process and generate a feature map of suitable size for a given data set in significantly less time than the time taken by the standard SOM algorithm to produce a feature map of similar quality on a predetermined network structure. The results of the experiments suggest that the incremental growing and learning mechanisms and the automatic stopping condition adapted in the proposed model is effective in discovering an appropriate network topology for the data at hand in order to yield a feature map of suitable size and shape for the given data manifold, while avoiding undergrowth and overgrowth of the network.

The model of self-growing and self-organizing map proposed and evaluated in this thesis has been shown to be a useful and viable model of the SOM that successfully addresses the difficulty of having to specify a suitable structure of the feature maps to be generated prior to the training process in the applications of the SOM. With the proposed model, it is no longer necessary to go through the time consuming process of generating numerous feature maps of different sizes and shapes applying different training epochs in search of a suitable feature map for a given data set.

The capability of the proposed model to capture nonlinear statistical relationships that exist in the principal components of high-dimensional input data manifolds and autonomously map them onto two-dimensional regular grid structures

of appropriate sizes combined with significant savings in the training time possible with the growing batch SOM model should prove valuable as an effective tool for the application of the SOM in data mining applications and visualization of high-dimensional complex systems and as a viable alternative model of growing SOM.

5.2 Suggestions for future work

Research work reported in this thesis demonstrates that the network can be adaptively grown during the training process to automatically discover a suitable two-dimensional representation of the topological structure of arbitrary high-dimensional data space by means of incremental growing and progressive learning. Although shown to be effective, the proposed model represents one of many possible alternative models of the growing SOM and much more work is required in exploring different mechanisms for the adaptive growing and learning process and for suitable criteria for the automatic stopping condition.

More research work could be directed at investigating different mechanisms for growing the network in terms of the location of insertions and the initialization methods for newly inserted nodes, and at evaluating whether and how the different growing mechanisms affect the quality of the final feature maps produced in order to better understand the possible impacts of the growing process on the overall learning dynamics of the SOM. Our own preliminary work in this direction involved assessing how inserting multiple rows and columns in each growth cycle affect the quality of the final feature maps in comparison to inserting a single row or column of nodes in each growth cycle. Although more experiments and analyses are required, initial findings seem to suggest that the difference in the quality of the feature maps produced by the two different insertion mechanisms is rather nominal.

A major avenue for further research is in the design and analysis of the automatic stopping criteria for growing models of the SOM. In fact, research in this area has been lacking so far, and much more comprehensive analytical and experimental studies of *necessary and sufficient conditions* for the automatic stopping criteria of growing SOM models are required. More analytical examination of the dynamics of the incremental growing and adaptive learning process is necessary to determine appropriate measures that are guaranteed to be fulfilled eventually to avoid indefinite growing and learning process, and experimental analyses are needed to indentify specific conditions that can lead to the formation of feature maps of suitable sizes and shapes without suffering from undergrowth and overgrowth. The automatic stopping criteria used in the proposed model based on the convergence of the MNDE value is shown to be an effective measure of automatic stopping condition and the analytical and experimental studies reported in this thesis should prove useful as the base of any further research in this direction.

One critical research area that can benefit not only the studies of growing SOM models, but also the SOM in general is the establishment of a universally accepted standard measure of quality for feature maps. Due to the lack of mathematical formalism in the theory of the SOM, it is doubtable that an absolute objective measure of quality can be constructed; however, it is conceivable that a hypothetical standard measure of quality is established and that the measure is used as the comparative basis for evaluation of the feature maps in the design and analysis of alternative SOM algorithms. More concerted research effort toward establishing a standard measure of quality should enable more objective evaluation and comparison of different SOM algorithms and help design SOM models that can grow feature maps in more objective and systematic manners. The MNQE used as a postulatory

measure of quality in the experimental analyses of the proposed model has a potential to be extended and developed as a possible standard measure of quality since it has been designed to quantify both the accuracy of vector quantization and degree of local ordering of the reference vectors. By increasing the radius of neighborhood for measuring the quantization errors in the vicinity of the winner nodes and by augmenting the quantization errors with a similarity measure among the neighboring nodes, the MNQE can possibly be extended to quantify both local and global ordering of the reference vectors as well as the overall accuracy of vector quantization. Investigation into how the MNQE can be extended as a unified measure of degree of topology preservation and accuracy of approximation for the SOM algorithms in the formation of the feature maps is certainly a potential area for our future research.

REFERENCES

[Arthur et al., 2009] Arthur L. Hsu, Isaam Saeed and Saman K. Halgamuge, Dynamic Self-Organizing Maps: Theory, Methods and Applications, Foundations on Computational Intelligence, Studies in Computational Intelligence Vol. 201 pages 363-379, Springer Berlin, 2009

[Bernd, 1995a] Bernd Fritzke, A Growing Neural Gas Network Learns Topologies, Advances in Neural Information Processing Systems Vol.7 pages 625-632, MIT Press, 1995

[Bernd, 1995b] Bernd Fritzke, Growing Grid – A Self-Organizing Network with Constant Neighborhood Rang and Adaptation Strength, Neural Processing Letters Vol. 2. No. 5, pages 9-13, 1995

[Bernd, 1997] Bernd Fritzke, Some Competitive Learning Methods, unpublished working paper, 1997

[Damminda et al., 2000] Damminda Alahakoon, Saman K. Halgamuge and Bala Srinivasan, Dynamic Self-Organizing Maps with Controlled Growth for Knowledge Discovery, IEEE Transactions on Neural Networks Vol. 11 No. 3 pages 601-614, 2000

[David et al., 1992] David E. Irwin, Klaus Obermayer, Klaus Schulten, Self-Organizing Maps: Ordering, Convergence Properties and Energy Functions, Biological Cybernetics Vol. 67 Issue 1 pages 47-55, Springer Berlin, 1992

[Frank and Asuncion, 2010] Frank, A. and Asuncion, A., UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010

[Franz, 1991] Franz Aurenhammer, Vornoi Diagrams – A Survey of a Fundament Geometric Data Strcuture, ACM Computing Surveys 23(3) pages 345-405, 1991

[Georg, 2004] Georg Pölzlbauer, Survey and Comparison of Quality Measures for Self-Organizing Maps, Proceeding of 5th Workshop on Data Analysis (WDA 2004) pages 67-82, 2004

[Hans-Ulrich and Klaus, 1992] Hans-Ulrich Bauer and Klaus R. Pawelzik, Quantifying the Neighborhood Preservation of Self-Organizing Feature Maps, IEEE Transactions on Neural Networks Vol. 3 Iss. 4 pages 570-579, 1992

[Helge and Klaus, 1988] Helge Ritter and Klaus Schulten, Kohonen's Self-organizing Maps: Exploring their Computational Capabilities, Proceedings of International Conference on Neural Networks Vol I, 1988, pages 109-116

[Hiong and Susan, 2005] Hiong Sen Tan and Susan E. George, Investigating Learning Parameters in a Standard 2-D SOM Model to Select Good Maps and Avoid Poor Ones, Lecture Notes in Computer Science Vol. 3339 pages 23-48, Springer Berlin, 2005

[James, 1967] James B. MacQueen, Some methods for classification and analysis of multivariate observations, Proceeding of the fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1 pages 281-297, University of California Press, 1967.

[Jean-Claude et al., 2002] Jean-Claude Fort, Patrick Letrémy, and Marice Cottrell, Advantages and drawbacks of the Batch Kohonen algorithm, Proceedings of 10th European Symposium on Artificial Neural Networks pages 223-230, 2002

[Juha, 1999] Juha Vesanto, SOM-based data visualization methods, Intelligent Data Analysis, Vol. 3 pages 259-126, 1999.

[Jure and Johann, 1999] Jure Zupan and Johann Gasteiger, Neural Networks in Chemistry and Drug Design: An Introduction, 2nd edition, Wiley-VCH, 1999

[Jure et al., 1994] Jure Zupan, Marjana Novic, Xinzhi Li, and Johann Gasteiger, Classification of Multicomponent Analytical Data of Olive Oils Using Different Neural Networks, Analytica Chimica Acta, Vol 292 pages 219-234, Elsevier

[Justine, 1993] Justine Blackmore, Incremental Grid Growing: Encoding High-Dimensional Structure into a Two-Dimensional Feature Map, Proceedings of the IEEE International Conference on Neural Networks pages 450-455, 1993

[Justine and Risto, 1995] Justine Blackmore and Risto Miikkulainen, Visualizing High-Dimensional Structure with the Incremental Grid Growing Neural Network, Proceedings of 12th Annual Conference on Machine Learning pages 55-63, 1995

[Kimmo, 1996] Kimmo Kiviluoto, Topology Preservation in Self-Organizing Maps, Proceedings of the Internal Conference on Neural Networks, IEEE Neural Network Council, 1996.

[Laurene, 1993] Laurene Fausett, Fundamentals of Neural Networks: Architectures, Algorithms and Applications, Prentice Hall, 1993

[Stuart, 1982] Least Squares Quantization in PCM, IEEE Transactions on Information Theory 28(2) pages 129-137, 1982

[Teuvo et al., 1996] Tuevo Kohonen, Olli Simula, Ari Visa, and Jari Kangas, Engineering Applications of the Self-Organizing Map, Proceedings of the IEEE 84(10) pages 1358-1384, 1996

[Teuvo, 1993] Teuvo Kohonon, Things You Haven't Heard about the Self-Organizing Map, Proceedings of the IEEE International Conference on Neural Networks Vol. 3 pages 1147 – 1156, 1993

[Teuvo, 2001] Tuevo Kohonen, Self-Organizing Maps, 3rd edition, Springer, 2001

[Teuvo and Timo, 2007] Teuvo Kohonen and Timo Honkela, Kohonen Network, Scholarpedia Vol 2. Issue 1 page: 1568, 2007

[Thomas and Klaus, 1991] Thomas Martinetz and Klaus Schulten, A "Neural-Gas" Network Learns Topologies, Artificial Neural Networks Vol. I pages 397-402, 1991

[Thomas et al., 1994] Thomas Villmann, Ralf Der, and Thomas M. Martinetz, A New Quantitative Measure of Topology Preservation in Kohonen's Feature Maps, Proceedings of the International Conference on Neural Networks, IEEE Neural Network Council, 1994.

[Tommy and Siu-Yeung, 2007] Tommy W S Chow and Siu-Yeung Cho, Neural Networks and Computing: Learning Algorithms and Applications, Imperial College Press, 2007

[Yoseph et al., 1980] Yoseph Linde, Andres Buzo, and Robert M. Gray, An Algorithm for Vector Quantizer Design, IEEE Transactions on Communications 28(1) pages 84-95, 1980

APPENDIX A

EVALUATION OF THE PARAMETER θ FOR THE AUTOMATIC STOPPING CONDITION

In this appendix, we discuss the premises behind our choice of the function used to determine an appropriate value of θ, the maximum tolerated MNDE value, in the specification of the automatic stopping condition for the proposed model, and present the results of the experiments conducted to validate our hypothesis assumed in the design of the function.

As elaborated in Chapter 3, a major goal of the proposed model is the automatic determination of stopping point for the growing and learning process to yield a feature map of appropriate size and shape suitable for the data at hand and the proposed model stops the training process when the value of the MNDE converges after the MNDE has become smaller than the maximum tolerated value θ. More specifically, the automatic stopping condition for the proposed model is defined as

$$MNDE(t) < \theta \wedge | MNDE(t) - MNDE(t-1) | < \varepsilon, \qquad \text{(A.1)}$$

The value of θ stipulates the degree of average distortion error in the direct neighborhood of the winners necessary to prevent premature termination of the growing and learning process which may lead the production of an inferior quality feature maps due to the undergrowth as well as to avoid the overgrowth which may prolong the training process unnecessarily without significant improvement in the quality of the feature maps.

An appropriate value of the parameter θ should depend on the dimensionality of the input data space since the distortion errors are measured in terms of the

Euclidean distance and the dimension of the data points significantly affect the Euclidean distance between the two data points. Note that the Euclidean distance between two data vectors p and q in a K-dimensional space is defined as

$$\|p-q\| = \sqrt{\sum_{i=1}^{K}(p_i - q_i)^2} \tag{A.2}$$

and the distortion errors are measured in terms of the square of the Euclidean distance

$$\|p-q\|^2 = \sum_{i=1}^{K}(p_i - q_i)^2 \tag{A.3}$$

to place progressively greater weight on points that are farther apart.

Thus, higher the dimensionality of the data vectors, more number of featural differences between the data vectors to be added in the calculation of the distortion errors. Hence, it would become increasingly more difficult for the MNDE to reach a small value as the number of the features present in the data vectors, or the dimensionality of the data space, increases. The main premise for the function used to determine a suitable value of θ in our proposed model is that the degree of tolerated distortion error should increase as the dimensionality of the data space increases but the magnitude of each additional degree of tolerance in the distortion error should decrease so that there is a limit to the overall degree of tolerated distortion error and the effect of each additional dimension is lessened as the dimensionality increases.

We model the desired value of the maximum tolerated MNDE value θ with the following logistic function of the dimensionality of the input data space

$$\theta = 0.5 \times (1 - \exp(-0.02 \times K)), \tag{A.4}$$

where K is the dimensionality of the input data vectors in order to give the maximum tolerated MNDE value θ of about 0.01 (or tolerance of 1%) when $K = 1$ and to allow for gradually diminishing additional tolerance of less than 1% for each added

dimension as K increases, but never more than 0.5 (or tolerance of 50%) even for very high dimensional data space. Table A-1 shows the values of θ computed by the function (A.4) for some selected values of K along with the changes in their values as K increases.

Table A-1 Values of θ as a function of the dimensionality of input data space

Input Dimension K	Value of maximum tolerated MNDE θ	Change in θ $\theta(K) - \theta(K\text{-}1)$
1	0.009900663	0.009900663
2	0.019605280	0.009704617
3	0.029117733	0.009512453
4	0.038441827	0.009324094
5	0.047581291	0.009139464
6	0.056539782	0.008958491
7	0.065320882	0.008781101
8	0.073928106	0.008607223
9	0.082364894	0.008436789
10	0.090634623	0.008269729
13	0.114474207	0.007788138
17	0.144114839	0.007189357
20	0.164839977	0.006770682
25	0.196734670	0.006126366
30	0.225594182	0.005543365
35	0.251707348	0.005015844
40	0.275335518	0.004538524
45	0.296715170	0.004106626
50	0.316060279	0.003715829

As can be seen from (A.4) and Table A-1, the function used in the proposed model to project an appropriate value of θ is a logistic function of the dimensionality of the input data space approaching the growth limit of 0.5 starting from an initial θ value of about 0.01 when $K = 1$ and growing the value of θ with diminishing increases as K increase resulting in an exponential decay in the changes of the values of θ as the dimensionality of the input data space grows.

A set of experiments have been conducted to test the premises assumed in the design of the function (A.4) and to verify whether the values of θ projected by the function leads to a suitable stopping point culminating in the production of feature maps of appropriate sizes. In the experiments, we test the automatic stopping condition used in the proposed model against another comparative stopping condition adapting different values of θ by applying the proposed model to a set of training data samples using the two different automatic stopping conditions while keeping all other aspects of growing and learning mechanisms of the proposed model same, and by evaluating and comparing the quality of the two feature maps generated using the corresponding test data samples.

The suitability of the function (A.4) as the basis of the automatic stopping condition for the proposed model is assessed by testing it against another function that generates relatively lower values of θ and by comparing the MNQE and the RCCI of the two feature maps generated by the proposed model adapting the two different stopping conditions along with the overall training times taken to generate the respective feature maps. Smaller θ values will generally lead to more growing and learning cycles resulting in larger feature maps. If the larger feature maps do not show any significant improvements in the MNQE and RCCI, it can be presumed that the stopping condition based on the function (A.4) used in the proposed model avoid the problems of undergrowth and overgrowth and manifest feature maps of suitable sizes for given data. We use the following function to generate θ values that are about 60% of the values generated by the function (A.4) as the basis of the comparison in the experiments:

$$\theta = 0.3 \times (1 - \exp(-0.02 \times K)) \quad \text{(A.5)}$$

The values of θ computed by the functions (A.4) and (A.5) are shown for some selected values of K in Table A-2.

Table A-2 Values of θ generated by the functions (A.4) and (A.5)

Input Dimension K	Value of $\theta = 0.5 \times (1 - \exp(-0.02 \times K))$	Value of $\theta = 0.3 \times (1 - \exp(-0.02 \times K))$
1	0.009900663	0.005940398
2	0.019605280	0.011763168
3	0.029117733	0.01747064
4	0.038441827	0.023065096
5	0.047581291	0.028548775
6	0.056539782	0.033923869
7	0.065320882	0.039192529
8	0.073928106	0.044356863
9	0.082364894	0.049418937
10	0.090634623	0.054380774
13	0.114474207	0.068684524
17	0.144114839	0.086468903
20	0.164839977	0.098903986
25	0.196734670	0.118040802
30	0.225594182	0.135356509
35	0.251707348	0.151024409
40	0.275335518	0.165201311
45	0.296715170	0.178029102
50	0.316060279	0.189636168

The experiments are carried out using the three pairs of training data set and test data set constructed by random selection process from the iris data set and the Italian olive oil data set, which were used for the experiments reported in Chapter 4. The results of the three experiments using the three different pairs of training data set and test data set on the iris data set and the Italian olive oil data set are presented in Table A-3 and Table A-4 respectively, summarizing the configurations of the final feature maps generated by the proposed model adapting the two different stopping conditions, training times taken to generate the corresponding final feature maps, and their quality measures in terms of MNQE and RCCI.

Table A-3 Results of the experiments conducted for evaluation of stopping conditions using iris data set

Training – Test Data Sets	$\theta = 0.5 \times (1 - \exp(-0.02 \times K))$				$\theta = 0.3 \times (1 - \exp(-0.02 \times K))$			
	Feature Map Size	Training Time	MNQE	RCCI	Feature Map Size	Training Time	MNQE	RCCI
Set 1	26 x 28	99.578	0.0650	1.000	30 x 35	279.61	0.0657	1.000
Set 2	21 x 30	76.515	0.0937	0.952	35 x 29	264.31	0.0964	0.952
Set 3	25 x 30	111.25	0.0760	1.000	29 x 35	261.58	0.0732	1.000

Table A-4 Results of the experiments conducted for evaluation of stopping conditions using Italian olive oil data set

Training – Test Data Sets	$\theta = 0.5 \times (1 - \exp(-0.02 \times K))$				$\theta = 0.3 \times (1 - \exp(-0.02 \times K))$			
	Feature Map Size	Training Time	MNQE	RCCI	Feature Map Size	Training Time	MNQE	RCCI
Set 1	40 x 42	2049.39	0.1249	1.000	51 x 59	10317.5	0.1128	0.988
Set 2	45 x 35	1883.77	0.1345	0.988	57 x 54	10834.2	0.1239	0.988
Set 3	43 x 36	1713.55	0.1252	1.000	55 x 54	9829.34	0.1127	0.988

As expected, the proposed model generates relatively larger feature maps when smaller values of θ are used as the basis of the automatic stopping condition since more growing and learning cycles are needed to reduce the MNDE to the required value. However, there is very little difference in the quality of the feature maps between the larger feature maps and the corresponding smaller feature maps produced by the proposed model adapting smaller values of θ. The MNQE and the RCCI of the corresponding feature maps for the iris data set are very much alike. For the Italian olive data set, the larger feature maps show slightly lower MNQE values (average reduction of about 0.012 in their values) and equal or slightly worse RCCI values, but the differences in the MNQE and the RCCI are not significant enough to indicate any variance in the quality. However, training times taken to produce the larger feature maps are significantly higher in comparison to the training times taken

to generate the corresponding smaller feature maps, exhibiting almost 3 fold increase on average for the iris data set and the increase of more than 5 times on average for the Italian olive oil data set.

The results of the experiments clearly indicate that the lower values of θ used in the alternative automatic stopping condition do not lead to the formation of higher quality feature maps while causing significant increase in the training time, which can be described as unnecessary overgrowth. At the same time, the fact that quality of the feature maps produced by the proposed model adapting the suggested automatic stopping condition exhibit very high RCCI values and the MNQE values that are in line with the MNQE value of the larger feature maps (including the larger feature maps produced by the standard SOM algorithm as shown in Chapter 4) indicate that the proposed automatic stopping condition does not cause the problem of undergrowth that may lead to production of inferior quality feature maps.

Therefore, our postulation assumed in the design of the function (A.4) used to project an appropriate value of θ based the dimensionality of the input data space has been shown to hold at least for the data sets used for testing and the results of the experiments presented in this appendix and the results of the experiments described in Chapter 4 suggest that the proposed model of self-growing and self-organizing batch map with the suggested automatic stopping condition is effective in automatically discovering an appropriate topology of the network suitable for the given data manifold while avoiding undergrowth and overgrowth, and that the proposed model is capable of growing the network dynamically during the training process to manifest a feature map of suitable size and shape for a given data set in significantly less time than the time taken by the standard SOM algorithm to produce a feature map of similar quality on a predetermined network structure.