# MULTILEVEL KOHONEN NETWORK LEARNING FOR CLUSTERING PROBLEMS

Siti Mariyam Shamsuddin, Anazida Zainal and Norfadzila Mohd Yusof

*Soft Computing Research Group*
*Universiti Teknologi Malaysia*

*mariyam@utm.my*

## ABSTRACT

Clustering is the procedure of recognising classes of patterns that occur in the environment and assigning each pattern to its relevant class. Unlike classical statistical methods, self-organising map (SOM) does not require any prior knowledge about the statistical distribution of the patterns in the environment. In this study, an alternative classification of self-organising neural networks, known as multilevel learning, was proposed to solve the task of pattern separation. The performance of standard SOM and multilevel SOM were evaluated with different distance or dissimilarity measures in retrieving similarity between patterns. The purpose of this analysis was to evaluate the quality of map produced by SOM learning using different distance measures in representing a given dataset. Based on the results obtained from both SOM methods, predictions can be made for the unknown samples. The results showed that multilevel SOM learning gives better classification rate for small and medium scale datasets, but not for large scale dataset.

## INTRODUCTION

Classification/Clustering is one of the most active research and application areas of neural networks. The Kohonen self-organising map (SOM) is a feed forward neural network that has been successfully applied as a clustering tool

in various problem domains. The SOM network is a special type of neural network that can learn from complex, multi-dimensional data, and transform them into visually decipherable clusters. The theory of the SOM network was motivated by the observation of the brain operation. Various human sensory impressions are neurologically mapped into the brain in such a way that spatial or other relations among stimuli correspond to spatial relations among the neurons, which are organised into a two-dimensional map.

SOM is a form of competitive neural network (Kohonen, 1998), which transforms highly dimensional data onto a two dimensional grid, while keeping the data topology by mapping similar data items to the same cell on the grid (or to neighbouring cells). Moreover, the unsupervised training of Kohonen network does not require target output for training. The network is able to learn the pattern of data itself without knowing all the outputs. The nodes in the network converge to form clusters representing groups of entities with similar properties. The number and composition of clusters can be visually determined based on the output distribution generated by the training process.

In other neural network models, all neurons adjust their weights in response to a training presentation; while in competitive learning only one or few neurons are allowed to adjust their weights. Therefore, this property has made Kohonen networks more resource efficient compared to other networks.

Artificial neural networks based on competition often use some means of calculating distance between input vectors and weights (Gopalan & Titus, 2003). Clearly, an important part of this process is the comparison between input vector elements and weight vector elements. Mathematically, this comparison is achieved through the computation of distance between vectors; vectors with smallest distance are most similar.

The success of unsupervised algorithms such as SOM and other clustering algorithms, depends crucially on the metric the measure of distance between object of interest (Kaski, Sinkkonen & Peltonen, 2001). Thus, when comparing patterns, it is very useful if they are presented in a metric space. The choice of metric for a neural network that implements competetive learning rule such as SOM is directly connected to the representation of data and it crucially influences the efficiency, accuracy, and generalisation ability of the outcomes.

Different distance measures define a different kind of metric space. Some have very close behaviours in similarity queries, others may behave quite differently (Qian, Surat & Pramanik, 2002). Understanding the relationship

among distance measures is beneficial in deciding which distance measure to choose for a particular application. Changing the distance measure can affect the performance of classification system. In this paper, the performances of SOM using different distance measures in several real world classification problems were measured. Although the usage of distance measures has an effect on the performance of SOM in classification tasks, this study showed that the distance measure has little to no impact toward classification rate.

Thus, an enhancement on the learning methodology for SOM algorithm is proposed. Multilevel SOM learning was explored to improve SOM performance in terms of classification and convergence rates. The results showed the proposed method is able to give better performance compared to the standard SOM in terms of accuracy but not in terms of its convergence rates.

## KOHONEN SELF-ORGANISING MAP (SOM)

The SOM network consists of an input layer and the output (Kohonen) layer. The Kohonen layer is usually designed as a two-dimensional arrangement of neurons that maps N-dimensional input to two dimensions, preserving topological order (Figure 1).



**Fig. 1: Kohonen SOM Architecture**
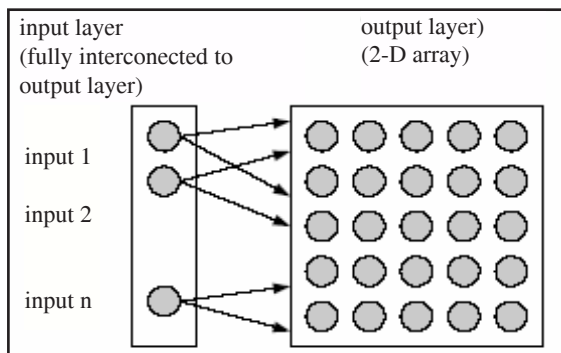
Each input dimension is called a feature. Each node in the grid is assigned the N-dimensional vector; the components of this vector are usually called weights. Initially weight components are small random values that usually fall in the range of 0 and 1. The weights are adjusted through the learning process by unsupervised competitive learning algorithm, a process of self organisation.

The input layer of neurons is fully connected to the Kohonen layer. The Kohonen layer computes the distance between the weight vector for each of the Kohonen neurons and the input pattern. The neuron with minimum distance is the winner, with an activation value of one, and others are zero. This neuron is called the winning node or best matching unit (BMU). This neuron best resembles the data input vector. Then, weight of the winning node is adjusted according to the input vector in the N-dimensional space.

The process goes through repeatedly until the solution is converged, i.e. the weight adjustments approaching zero. Each input vector is mapped to a closest grid in the N-dimensional space. This process corresponds to a projection of the input space onto the two-dimensional grid.

The main properties of such a feature map are:
1. The feature map preserves the distance relationships between the input data. While some distortion is inevitable, the mapping preserves neighbourhood relationships between the input data, and makes such relationships geographically explicit.
2. The feature map allocates different number of nodes to input, based on their occurrences. The more frequent input patterns are mapped to larger domains at the expense of the less frequent ones.

## SOM LEARNING ALGORITHM

The construction of feature map consists of two basic procedures, selecting winning node and updating both its weights and the neighbouring nodes. The flow of SOM learning is shown below:
1. One sample vector $x$ is randomly drawn from the input data set and its similarity (distance) to the codebook vectors is computed by a distance measure, for example Euclidean distance:

$$\left\| x - m_c \right\| = \min_i \left\{ \left\| x - m_i \right\| \right\}, \quad (1)$$

where,
$x$          input data vector,

$m_c$         best matching reference vector,

$m_i$         reference vector of unit $i$.

2.  Once the BMU is found, the codebook vectors get updated. The BMU with its neighbours are moved closer to the input vector in the input space. The magnitude of the attraction is governed by the learning rate. As the learning proceeds and new input vectors are fed to the map, the learning rate gradually decreases to zero together with the reduction of radius neighbourhood. The update rule for the reference vector of unit *i* is :

$$m_i(t+1) = m_i(t) + \alpha(t)[x(t) - m_i(t)] \quad (2)$$

where,

  *$m_i(t)$*            reference weight vector,

  *$m_i(t+1)$*       updated reference weight vector,

  *$x(t)$*             input vector,

  *$\alpha$*               learning rate, and

  *$t$*                learning time.

3.  Steps 1 and 2 constitute a single training step and they are repeated until the training ends. The number of training steps must be set prior to SOM training according to convergence and learning rates.

### DISTANCE MEASURE

The original SOM application was designed for real valued patterns. The Euclidean distance is usually used to find the network weights. Besides Euclidean distance, there are other types of distance measures that are based on different metric spaces. In this study, we investigated the following distance measures:

1.  Euclidean distance

$$d_{ij} = \sum_{k=1}^{n} \sqrt{\left(x_{ik} - x_{jk}\right)^2} \quad (3)$$

2.  Manhattan distance

$$d_{ij} = \sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right) \quad (4)$$

5

3. Bray Curtis distance

$$d_{ij} = \frac{\sum_{k=1}^{n} \left| x_{ik} - x_{jk} \right|}{\sum_{k=1}^{n} \left| x_{ik} \right| + \left| x_{jk} \right|} \qquad (5)$$

4. Canberra distance

$$d_{ij} = \frac{\sum_{k=1}^{n} \left| x_{ik} - x_{jk} \right|}{\sum_{k=1}^{n} \left| x_{ik} \right| + \left| x_{jk} \right|} \qquad (6)$$

5. Chebyshev distance

$$d_{ij} = max \left| x_{ik} - x_{jk} \right| \qquad (7)$$

where

$x_i$      input node *i* at time *k*,

$x_j$      output node *j* at time *k* with

        weight from input node *i,* and

$k$      input dimension.

## A COMPARISON OF STANDARD SOM AND MULTILEVEL SOM LEARNING

Multilevel learning is an enhancement of original SOM learning algorithm, and it is divided into two phases. The predetermined number of learning iterations is divided equally for each learning level, thus, each learning level will consume the same number of learning times. These two learning methods are different in terms of their computational procedure in finding the winning node. In standard SOM application, one type of distance measure is computed to determine the winning node during the learning process, while for the multilevel learning, two types of distance measure will be used. For each learning level, different types of distance measures are used for winning node computation. The standard SOM and multilevel SOM learning algorithm are shown in Fig. 2 and Fig. 3.
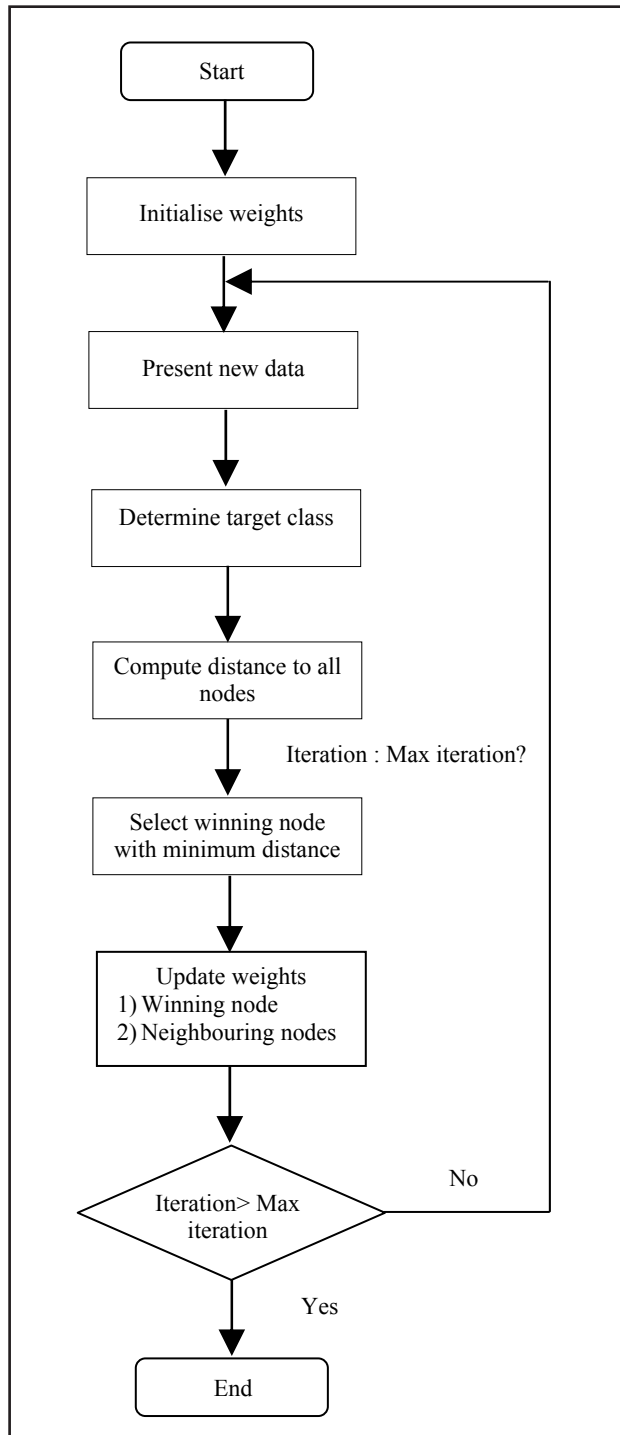
```
                    ┌──────────────┐
                    │    Start     │
                    └──────────────┘
                           │
                           ▼
                 ┌────────────────────┐
                 │ Initialise weights │
                 └────────────────────┘
                           │
                           ▼
                 ┌────────────────────┐
                 │  Present new data  │◄──────────┐
                 └────────────────────┘           │
                           │                       │
                           ▼                       │
                 ┌──────────────────────┐         │
                 │ Determine target class│         │
                 └──────────────────────┘         │
                           │                       │
                           ▼                       │
                 ┌──────────────────────┐         │
                 │ Compute distance to all│        │
                 │        nodes           │        │
                 └──────────────────────┘         │
                           │    Iteration : Max iteration?
                           ▼                       │
                 ┌──────────────────────┐         │
                 │   Select winning node │        │
                 │  with minimum distance│        │
                 └──────────────────────┘         │
                           │                       │
                           ▼                       │
                 ┌──────────────────────┐         │
                 │    Update weights     │        │
                 │  1) Winning node      │        │
                 │  2) Neighbouring nodes│        │
                 └──────────────────────┘         │
                           │                       │
                           ▼                       │
                        ◇─────◇      No            │
                     ◇  Iteration> ◇───────────────┘
                     ◇   Max       ◇
                        ◇ iteration ◇
                           │
                           │ Yes
                           ▼
                    ┌──────────────┐
                    │     End      │
                    └──────────────┘
```
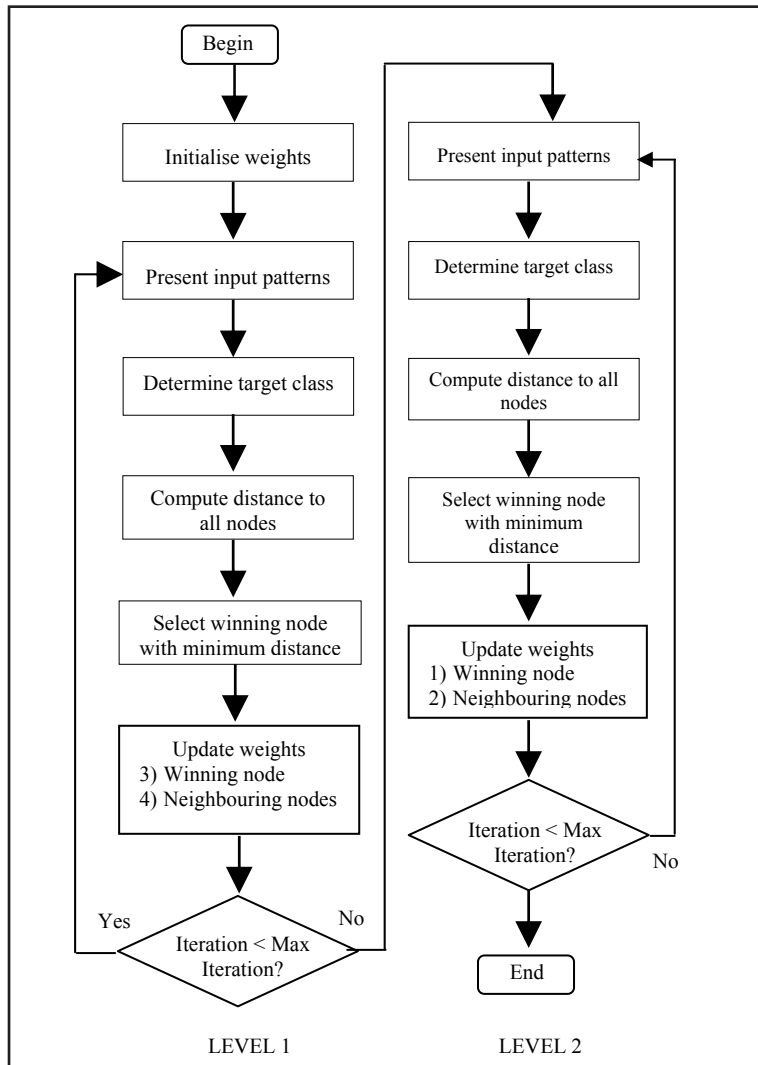
**Fig. 2: Standard SOM Algorithm**

**Fig. 3: Multilevel SOM Algorithm**

## PERFORMANCE MEASUREMENT

After training, the results need to be visualised, and these can be obtained through the mapping topology. The quality of obtained map after the learning process is calculated based on the mapping precision and topology preservation (James & Jacek, 2004). A common measure that calculates the precision of mapping is Average Quantisation Error (AQE) over the entire dataset.

Quantisation error of an input vector is defined by the differences between the input vector and the closest output vectors. The measures are dependent

because it is measured based on given data. By computing this measure, we can see how well the data is classified on the SOM map. Equation (8) is used for calculating the average quantisation errors:

$$E_q = \frac{1}{N} \sum_{k=1}^{N} \|x_i - m_c\|, \qquad (8)$$

where,

$N$          total number of input samples,

$x_i$         input data vector, and

$m_c$        best matching reference vector.

The analysis of testing phase consists of presenting and evaluating the distribution of test data. A number of classification assessment can be used to evaluate the peformance of the classifier, which include precision, recall, accuracy, (Mangiameli, Chen & West, 1996)) and processing time.

Retrieval performance in terms of precision and recall was implemented in this study to evaluate the performance of the trained SOM using different distance measures. The recall is the ratio of the number of data that are classified into a class correctly over the total number of data. Meanwhile, the precision is the ratio of the number of correct data that is classified into a class over the total number of data within the class. The accuracy is the ratio of the number of data that is correctly classified over the total number of data. The classification accuracy of new samples measures the generalisation of the results to deploy significant indications of the mapping quality.

The performance measures for recall, precision, and accuracy are defined and computed using the matrix classification shown in Table 1. The correctness of the assigned class to the data is counted in relation to the target class.

**Table 1: Matrix of Classification**

| | | Target class | |
|---|---|---|---|
| | | Yes correct | No correct |
| Predicted class | Assigned Yes | *a* | *b* |
| | Assigned No | *c* | *d* |

where,
- cell *a* counts the classes correctly assigned to the test set,
- cell *b* counts the class incorrectly assigned to the test set
- cell *c* counts the classes incorrectly rejected from the test set,
- cell *d* counts the classes correctly rejected from the test set.

From the scores, we can calculate the recall, precision, and accuracy as shown in Equation (9) , (10) and  (11).

$$Accuracy = \frac{a + d}{a + d + b + c} \tag{9}$$

$$Precision = \frac{a}{a + b} \tag{10}$$

$$Recall = \frac{a}{a + c} \tag{11}$$

## SOM NETWORK PARAMETER SETTINGS

Traditionally, SOM learning is controlled by three main parameters; the number of neurons in the output layer, the number of input samples (which will determine the number of iterations), the size of initial radius, and the maximum and minimum learning rates. A process of trial and error is necessary to determine a set of values that are suitable for the dataset used in an experiment. Here, the selection of parameters was based on some theoretical guidelines and suggestions from the literature.

Considerably large and extremely small learning rates can lead to poor network performance (Hagenbuchner, Sperduti & Tgoi, 2003). Ideally, the learning rate is set at $\alpha(t) \leq 0.3$. In our experiments, a linear learning rate with initial value of 0.3 was used. The number of neurons in the rectangular array should be large enough to allow adequate number of cluster formations. According to (Vesanto, Himberg, Alhoniemi & Parhankanges, 2000), the number of map units is usually in the range of 100 to 600. (Deboeck & Kohonen, 1998) and (Kohonen 1998) recommended using 10 times the dimension of the input patterns. Here, we adopted the later suggestion.

Neighbourhood function and the number of neurons determine the granularity of the resulting mapping. A larger neighbourhood is used in intial training and it is gradually decreased to a suitable final radius. In these experiments, the initial radius was set to half of the lattice size.

The accuracy of the map is highly dependent on the number of iterations of the SOM algorithm. For good statistical accuracy, the number of iterations should be at least 500 times the number of neurons. According to (Xu & Wang, 2004), the total learning time should be in the range of 100 to 10000. An excessive amount of iterations may cause an inaccurate clustering result. A more serious problem is that the topology preserving mapping is not guaranteed even with huge amounts of iterations. Thus, in this study, the iterations or learning steps were set to 5000.

## EXPERIMENTAL SETUP

The performance of standard SOM and multilevel SOM were evaluated with different distance or dissimilarity measures in retrieving similarity between patterns. For standard SOM, there were five different sets; each set is assigned with different distance measures as shown in Table 2(a), and eight different sets for Multilevel SOM learning. The selected distance measures were applied to each of the learning phases as shown in Table 2(b). The results obtained from the experiments were compared and analysed.

**Table 2(a): Standard SOM Classifiers**

| Classifier | Distance Measure |
|---|---|
| SOM 1 | Euclidean |
| SOM 2 | Manhattan |
| SOM 3 | Bray Curtis |
| SOM 4 | Canberra |
| SOM 5 | Chebyshev |

**Table 2(b): Multilevel SOM Classifiers**

| Classifier | Distance Measure | |
|---|---|---|
| | Level 1 | Level 2 |
| MSOM 1 | Manhattan | Euclidean |
| MSOM 2 | Euclidean | Manhattan |
| MSOM 3 | Bray Curtis | Euclidean |
| MSOM 4 | Euclidean | Bray Curtis |
| MSOM 5 | Canberra | Euclidean |
| MSOM 6 | Euclidean | Canberra |
| MSOM 7 | Chebyshev | Euclidean |
| MSOM 8 | Euclidean | Chebyshev |

Five real world datasets from the UCI Machine Learning database for classification problems were tested in this study: Iris, Wine, Glass, Diabetes, and Pendigits. These datasets represent small, medium, and large categories respectively. Before these data can be fed to SOM, preprocessing had to be performed to remove noise and inconsistencies.

All the inputs to be fed to the SOM network were subjected to the restriction; they have to lie within the symmetric bounds. Preprocessing was followed by normalisation. Normalisation is a process which maps the entire set of values of a given attribute to a new set of replacement values such that the previous value can be identified with one of the new values.

Normalisation of the inputs is necessary to ensure that the Kohonen layer finds the correct class for the problem. Without normalisation, larger input vectors may generate bias in the Kohonen processing elements, causing weaker value input sets improperly classified. Due to the competitiveness of the Kohonen layer, the larger input value vectors overpower the smaller vectors. The datasets were normalised to have values scaled between 0 and 1 using the linear transformation method. The equation for linear transformation is as follows:

$$X_n = \frac{X_0 - X_{min}}{X_{max} - X_{min}} \qquad (12)$$

where,

$X_n$        new $x$ value (after normalisation),
$X_0$        current $x$ value (before normalisation),
$X_{min}$    minimum value of $x$ in sample data, and
$X_{max}$    maximum value of $x$ in sample data.

In these experiments, the values for each parameter in SOM learning were fixed, except for the number of neurons that were determined based on the number of input features or dimensions. Different datasets contain different number of attributes and sizes. The number of neurons was determined based on the number of input features. Different map sizes were generated to match the different training sets. Other parameters such as the learning rate and number of learning iterations were kept constant for other experiments with various types of dataset.

12

## SIMULATION RESULTS

Comparisons were performed in two phases. Both of the SOM structures (standard and multilevel) were evaluated. Firstly, the analysis was carried out to evaluate the performance of both learning methods using various distance measures. Both standard SOM and multilevel SOM were run accordingly based on each dataset and only the best solutions in terms of classification accuracy performance were used for further analysis. The comparison between each classifier in standard SOM and multilevel SOM is discussed in detailed in the following sections.

In the second phase of comparison, the best classifier from standard SOM and multilevel SOM were compared, to evaluate the efficiency of the multilevel SOM in providing more accurate map and the ability to provide better classification performance on the testing data compared to the standard SOM. The comparative analysis of the SOM classifiers is described in the following sections.

### Comparison of Standard SOM with Different Distance Measures

In this experiment, performance of standard SOM trained using various types of distance measures were evaluated using different datasets. This experiment was conducted to examine the efficiency and validity of various distance measures in pattern classification tasks. The comparative results are reported in Table 3(a) to Table 7(b).

### ( i ) Performance of Iris Dataset

**Table 3(a): Performance of Standard SOM in Trainining Process on Iris Dataset**

| Classifier | Computation Time (ms) | AQE |
|---|---|---|
| SOM 1 | 74 | 0.00054 |
| SOM 2 | 77 | 0.00057 |
| SOM 3 | 82 | 0.00056 |
| SOM 4 | 80 | 0.00110 |
| SOM 5 | 78 | 0.00064 |

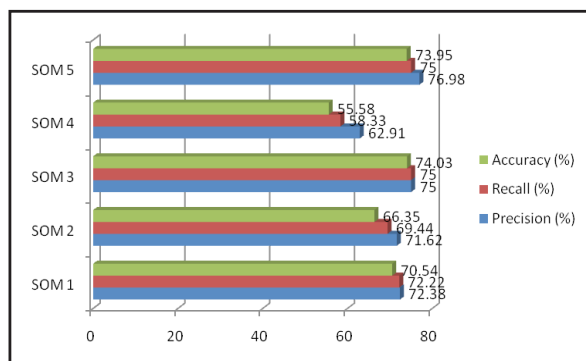**Table 3(b): Standard SOM Classifier Retrieval Performance on Test Data**
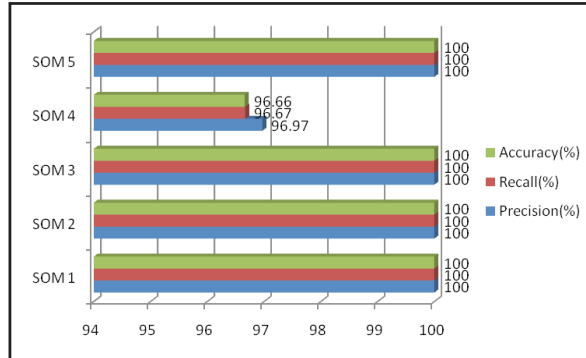


*( ii ) Performance of Wine Dataset*

**Table 4(a): Performance of Standard SOM in Training**

| Classifier | Computation Time (ms) | AQE |
|---|---|---|
| SOM 1 | 212 | 0.00002 |
| SOM 2 | 297 | 0.00002 |
| SOM 3 | 310 | 0.00002 |
| SOM 4 | 319 | 0.00017 |
| SOM 5 | 315 | 0.00002 |

**Table 4(b): Standard SOM Classifier Retrieval Performance on Test Data**



**(iii)** *Performance of Glass Dataset*

14

**Table 5(a): Performance of Standard SOM in Training**

| Classifier | Computation Time (ms) | AQE |
|---|---|---|
| SOM 1 | 147 | 0.00008 |
| SOM 2 | 186 | 0.00006 |
| SOM 3 | 193 | 0.00006 |
| SOM 4 | 188 | 0.00026 |
| SOM 5 | 200 | 0.00008 |

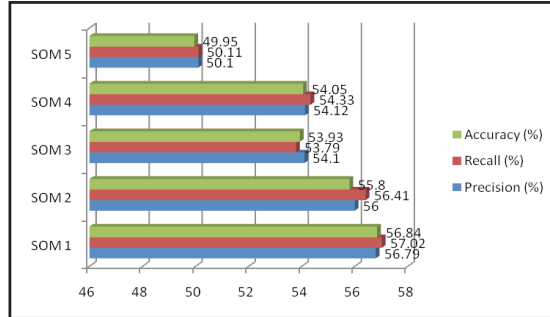**Table 5(b): Standard SOM Classifier Retrieval Performance on Test Data**
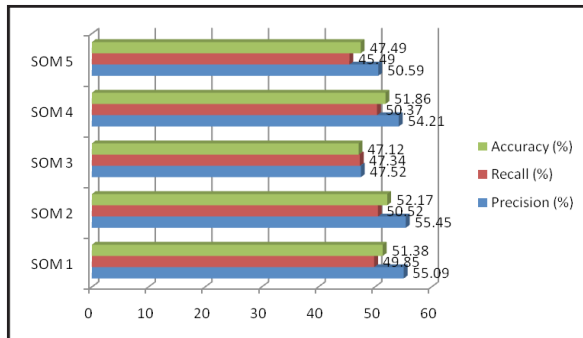


*(iv) Performance of Diabetes Dataset*

**Table 6(a): Performance of Standard SOM in Training**

| Classifier | Computation Time (ms) | AQE |
|---|---|---|
| SOM 1 | 332 | 0.00755 |
| SOM 2 | 412 | 0.00862 |
| SOM 3 | 468 | 0.00878 |
| SOM 4 | 469 | 0.05759 |
| SOM 5 | 440 | 0.00908 |

15

**Table 6(b): Standard SOM Classifier Retrieval Performance on Test Data**



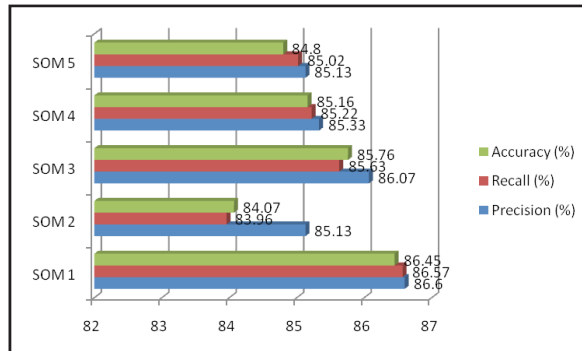*( v ) Performance of Pendigits Dataset*

**Table 7(a): Performance of Standard SOM in Training**

| Classifier | Computation Times (ms) | AQE |
|---|---|---|
| SOM 1 | 823 | 0.02719 |
| SOM 2 | 1509 | 0.02937 |
| SOM 3 | 1712 | 0.02950 |
| SOM 4 | 1797 | 0.03932 |
| SOM 5 | 1602 | 0.03327 |

**Table 7(b): Standard SOM Classifier Retrieval Performance on Test Data**



Based on the first analysis, we can see that the trained SOM with Euclidean distance, Manhattan distance, Bray Curtis distance, and Manhattan distance had very similar quantisation errors calculated on a trained map, except for Canberra distance.

16

Quantisation error computed on the SOM map produced by Canberra distance is vastly different from other distance measures and produced undesirable results. This was due to the Canberra distance that had to perform its own standardisation where absolute values of differences were divided by the absolute value of the sum of corresponding variables in the two observations. Special preprocessing treatment must be performed when one considers applying the Canberra distance as a distance measure. Depending on the values and distributions of the variables in clusterd data sets, these different distance measures may point out different aspects of the structure of the data set. The results obtained from the experiments also showed that Euclidean distance generate faster convergence time compared to other models.

For Iris datasets, classification accuracy is achieved almost by all classifiers except for SOM classifier using Canberra distance with slightly lower results. For the Wine dataset, trained SOM using Manhattan distance showed its ability to perform better than the other models, while for Glass datasets, good classifications were obtained by using Manhattan distance. The results showed that Euclidean distance has successfully performed better classification accuracy on Diabetes and Pendigits dataset. This conformed with the claims being made by previous researchers that Euclidean distance is the most appropriate technique for the classification of large scale data (for example, Diabetes and Pendigits).

**Comparison of Multilevel SOM with Different Distance Measures**

In this experiment, performance of multilevel SOM using various distance measures was evaluated. This experiment was conducted to investigate the efficiency and the validity of the proposed classifier in classification tasks. The results are tabulated in Table 8(a) to Table 12(b).

*(i ) Performance of  Iris Dataset*

**Table 8(a): Performance of Multilevel SOM**

| Classifier | Computation Time (ms) | AQE |
|---|---|---|
| MSOM 1 | 73 | 0.00057 |
| MSOM 2 | 74 | 0.00054 |
| MSOM 3 | 78 | 0.00056 |
| MSOM 4 | 79 | 0.00054 |
| MSOM 5 | 77 | 0.00110 |
| MSOM 6 | 80 | 0.00054 |
| MSOM 7 | 76 | 0.00064 |
| MSOM 8 | 82 | 0.00054 |

**Table 8(b): Multilevel SOM Classifier Retrieval Performance on Test Data**

| Classifier | Precision (%) | Recall (%) | Accuracy (%) |
|---|---|---|---|
| MSOM 1 | 100.00 | 100.00 | 100.00 |
| MSOM 2 | 100.00 | 100.00 | 100.00 |
| MSOM 3 | 100.00 | 100.00 | 100.00 |
| MSOM 4 | 100.00 | 100.00 | 100.00 |
| MSOM 5 | 96.97 | 96.67 | 96.66 |
| MSOM 6 | 100.00 | 100.00 | 100.00 |
| MSOM 7 | 100.00 | 100.00 | 100.00 |
| MSOM 8 | 100.00 | 100.00 | 100.00 |

*( ii ) Performance of Wine Dataset*

**Table 9(a): Performance of Multilevel SOM in Training**

| Classifier | Computation Times (ms) | AQE |
|---|---|---|
| MSOM 1 | 259 | 0.00002 |
| MSOM 2 | 256 | 0.00002 |
| MSOM 3 | 257 | 0.00002 |
| MSOM 4 | 263 | 0.00002 |
| MSOM 5 | 271 | 0.00021 |
| MSOM 6 | 270 | 0.00002 |
| MSOM 7 | 265 | 0.00002 |
| MSOM 8 | 267 | 0.00002 |

**Table 9(b): Multilevel SOM Classifier Retrieval Performance on Test Data**

| Classifier | Precision (%) | Recall (%) | Accuracy (%) |
|---|---|---|---|
| MSOM 1 | 71.95 | 69.44 | 68.91 |
| MSOM 2 | 69.44 | 69.44 | 68.02 |
| MSOM 3 | 64.09 | 61.11 | 59.99 |
| MSOM 4 | 80.95 | 77.78 | 77.95 |
| MSOM 5 | 78.47 | 72.22 | 70.49 |
| MSOM 6 | 75.38 | 75.00 | 74.53 |
| MSOM 7 | 63.17 | 61.11 | 60.47 |
| MSOM 8 | 66.27 | 66.67 | 63.90 |

*(iii) Performance of Glass Dataset*

**Table 10(a): Performance of Multilevel SOM for Training**

| Classifier | Computation Time (ms) | AQE |
|---|---|---|
| MSOM 1 | 166 | 0.00005 |
| MSOM 2 | 167 | 0.00005 |
| MSOM 3 | 171 | 0.00005 |
| MSOM 4 | 173 | 0.00006 |
| MSOM 5 | 176 | 0.00030 |
| MSOM 6 | 181 | 0.00007 |

**Table 11(b): Multilevel SOM Classifier Retrieval Performance on Test Data**

| Classifier | Precision (%) | Recall (%) | Accuracy (%) |
|---|---|---|---|
| MSOM 1 | 56.00 | 56.41 | 55.80 |
| MSOM 2 | 52.10 | 52.18 | 52.05 |
| MSOM 3 | 57.67 | 57.39 | 57.48 |
| MSOM 4 | 54.92 | 55.18 | 54.87 |
| MSOM 5 | 54.36 | 54.25 | 54.28 |
| MSOM 6 | 49.96 | 49.96 | 49.21 |
| MSOM 7 | 55.50 | 55.87 | 55.33 |
| MSOM 8 | 54.03 | 54.33 | 53.65 |

*( iv ) Performance of Pendigits Dataset*

**Table 12(a): Performance of Multilevel SOM in Training Process**

| Classifier | Computation Time (ms) | AQE |
|---|---|---|
| MSOM 1 | 1137 | 0.02852 |
| MSOM 2 | 1199 | 0.02735 |
| MSOM 3 | 1196 | 0.02848 |
| MSOM 4 | 1226 | 0.02710 |
| MSOM 5 | 1282 | 0.04091 |
| MSOM 6 | 1315 | 0.02797 |
| MSOM 7 | 1215 | 0.03355 |
| MSOM 8 | 1203 | 0.02739 |

**Table 12(b): Multilevel SOM Classifier Retrieval Performance on Test Data**

| Classifier | Precision (%) | Recall (%) | Accuracy (%) |
|---|---|---|---|
| MSOM 1 | 83.93 | 83.91 | 83.53 |
| MSOM 2 | 87.03 | 86.43 | 86.51 |
| MSOM 3 | 86.32 | 85.20 | 85.39 |
| MSOM 4 | 86.46 | 86.31 | 86.23 |
| MSOM 5 | 84.92 | 85.06 | 84.75 |
| MSOM 6 | 85.08 | 85.00 | 84.90 |
| MSOM 7 | 84.78 | 84.17 | 84.04 |
| MSOM 8 | 86.35 | 86.54 | 86.35 |

The second analysis was conducted to evaluate the performance of multilevel SOM using the same experimental setup as in standard SOM. Here, the multilevel learning approach was used in the SOM algorithm. In this approach, the standard SOM learning is divided into two levels, each with equal fixed training times (iterations).

In multilevel structure, a pair of different distance measures was deployed to discover how the learning would be affected. In either one of the layers, Euclidean distance was consistently used in all the experiments. Meanwhile, another layer was kept loose and we experimented it with different kinds of measures (Manhattan distance, Bray Curtis distance, Canberra distance, and Chebyshev distance). Refer to Table 2(b) for the distance measure pairings.

The results obtained showed that the multilevel SOM learning using Euclidean distance and Manhattan distance are faster in terms of computational time. From these experiments, it was show that multilevel learning with Euclidean distance and Bray Curtis gives better accuracy compared to other pairings in multilevel SOM classifiers. The efficiency and applicability of this method may confirm that the integration between these two measures is particularly suitable for pattern classification tasks on small and medium scale data. While for a large scale dataset, multilevel SOM using Euclidean and Manhattan distances is capable to achieve a slightly better level of classification accuracy on the test set compared to other models. The best classifiers from these experiments were selected for further analysis.

**Comparison between Standard SOM and Multilevel SOM**

This section investigates the capabilities of the proposed method through a comparison with standard SOM. In order to evaluate the classification

performance between both classifiers, the best classifier from standard SOM and multilevel SOM obtained from each experiment was then compared.

**Table 13: Comparison of Standard SOM and Multilevel SOM for Iris Dataset**

| Analysis Criteria | SOM Classifier | |
| --- | --- | --- |
| | Standard SOM | Multilevel SOM |
| Computation time (ms) | 74 | 74 |
| AQE | 0.00054 | 0.00054 |
| Precision (%) | 100 | 100 |
| Recall (%) | 100 | 100 |
| Accuracy (%) | 100 | 100 |

**Table 14: Comparison of  Standard SOM and Multilevel SOM for Wine Dataset**

| Analysis Criteria | SOM Classifier | |
| --- | --- | --- |
| | Standard SOM | Multilevel SOM |
| Computation time (ms) | 310 | 256 |
| AQE | 0.00002 | 0.00002 |
| Precision (%) | 75.00 | 80.95 |
| Recall (%) | 75.00 | 77.78 |
| Accuracy (%) | 74.03 | 77.95 |

**Table 15: Comparison of  Standard SOM and Multilevel SOM for Glass Dataset**

| Analysis Criteria | SOM Classifier | |
| --- | --- | --- |
| | Standard SOM | Multilevel SOM |
| Computation time (ms) | 186 | 171 |
| AQE | 0.00006 | 0.00005 |
| Precision (%) | 50.52 | 52.93 |
| Recall (%) | 55.45 | 57.77 |
| Accuracy (%) | 52.17 | 55.01 |

**Table 16: Comparison of Standard SOM and Multilevel SOM for Diabetes Dataset**

| Analysis Criteria | SOM Classifier | |
| --- | --- | --- |
| | Standard SOM | Multilevel SOM |
| Computation time (ms) | 332 | 395 |
| AQE | 0.00755 | 0.00854 |
| Precision (%) | 56.79 | 57.67 |
| Recall (%) | 57.02 | 57.39 |
| Accuracy (%) | 56.84 | 57.48 |

**Table 17: Comparison of Standard SOM and Multilevel SOM for Pendigit Dataset**

| Analysis Criteria | SOM Classifier | |
| --- | --- | --- |
| | Standard SOM | Multilevel SOM |
| Computation time (ms) | 823 | 1199 |
| AQE | 0.02719 | 0.02735 |
| Precision (%) | 86.60 | 87.03 |
| Recall (%) | 86.57 | 86.43 |
| Accuracy (%) | 86.45 | 86.51 |

The results from the experiments performed on each of the five datasets, showed that the multilevel learning approach is slightly better especially for small and medium scale dataset in terms of classification accuracy. Mean while for large scale data, better classification accuracy was achieved using standard SOM with Euclidean distance.

The computation time for standard SOM was a bit faster than multilevel SOM, especially for medium and large scale data. From the experiments performed, standard SOM with Euclidean distance is the most suitable approach to cluster data, especially in large scale datasets, due to its ability to produce better classification results with less computational time compared to other types of distance measures. When the feature vector is large, some distance measures may consume more computing resources than the others. This is because each distance measures is different in terms of its computational procedure and this will definitely affect the computational time of classification system when dealing with large and complex data.

However, multilevel SOM seems to be a bit slower in classifying large scale data. Nevertheless, it gives an acceptable level of accuracy. Multilevel SOM which deployed both Euclidean and Bray Curtis distance outperformed others. It produced slightly better classification results in all datasets, except Pendigits data.

## DISCUSSION

From the analysis, it was found that the choice of distance measures rely on its computational advantages, because some of the distance measures required less computation, such as Euclidean distance and Manhattan distance. Their simplicity has an advantage when dealing with large datasets. For larger datasets, there is also a greater degree of variation in run time. This shows that the computation time exponentially increases when training more nodes. Furhermore, the region described by a fixed distance differs greatly and it depends on the metric used. For example, an Euclidean distance describes a circle in 2D, a sphere in 3D, and a hyper sphere in more than 3D; the Manhattan distance describes a square, a cube, and a hypercube respectively. Because of this reason, we can absolutely see that by changing the distance measure, it would have little effect on the overall performance of a classification system.

Another perspective that must be taken into account when choosing an appropriate dissimilarity measure depends largely on the nature of data.

## CONCLUSION

Based on the experiments performed in this study, it can be concluded that the choice of an appropriate distance or dissimilarity measure depends largely on the nature of the data; some have very similar behaviours in similarity queries, others may behave quite differently. When the feature vector is large, some distance measures may consume more computing resources than the others.

Changing the distance measures can have a little effect on the overall performance of a classification system in terms of its classification accuracy and computational time. The choice of an appropriate metric depends on the given learning task and it is often difficult and time consuming to find the right distance measure for a concrete problem.

The multilevel SOM method can give very similar and satisfactory classification results as standard SOM, even though their processes are different. Our experiments can be extended by analysing the performance of the multilevel learning model with other families of distance that is based on correlation, such as the Pearson correlation.

**Suggestion for Future Work**

We plan to continue our work by extending it to cover other aspects of measurement.

(i) **Test on a different category of distance measure such as correlation.**
The distance measures used in this study are from the Euclidean category. They consider point-to-point differences and accounts for absolute differences. Results from these experiments showed that there are no significant differences on the performance of standard SOM and multilevel SOM models using different distance measures.

(ii) **Test on larger and complex datasets.**
From these experiments, it showed that there are no noticeable difference between the performance of multilevel SOM and the standard SOM in terms of its accuracy. Perhaps, an experiment which involves large and complex datasets can yield a result that can clearly differentiate the performance of these two structures (standard and multilevel SOM) in the classification and forecasting.

(iii) **Conduct more trial and error to determine the best set of parameters for the SOM algorithm that converges to an adequate and useful state.**
The most significant drawback of the SOM algorithm is that it requires the specification of many parameters such as:
a. the dimension in X and Y of the map,
b. number of clusters,
c. the number of iterations,
d. the initial learning rate $\alpha$ ,
e. the neighbourhood radius,
f. the type of neighbourhood function,
g. the type of vector initialization prior to training, and
h. the topology of map.

It is sometimes difficult or impractical to test all possibilities, however, by changing parameters such as learning rate and neighbourhood radius does not change the results significantly. The important elements are the network dimensions (number of clusters) and the number of iterations. By varying the network dimension and increasing number of iterations, the SOM learning performance can be improved and better classification results can be obtained.

# REFERENCES

Kohonen, T. (1998). *The self-organising maps.* Springler, Berlin: Heidelberg.

Gopalan, A., & Titus, A. H. (2003). A new wide range Euclidean distance circuit for neural network hardware implementations. *IEEE Transactions on Neural Networks,14*(5).

Kaski, S., Sinkkonen, J., & Peltonen, J. (2001). Bankruptcy analysis with self-organising maps in learning metrics. *IEEE Transactions on Neural Networks, 12,* 936-947.

Qian, G., Surat, S., & Pramanik, S. (2002). Similarity between Euclidean and consine angle distance for nearest neighbour queries. *IEEE ICIP.*

James, S. K., & Jacek M. Z. (2004). Topography-enhanced BMU search in self-organising maps. *ISNN 2004, LNCS 3174,* 695–700 .

Mangiameli, C., Chen, S. K., & West, D. (1996). A comparison of SOM neural network and hierarchichal clustering methods. *European Journal of Operation Research,  93*(2).

Hagenbuchner, M., Sperduti, A., & Tsoi, A. C. (2003). A self-organising map for adaptive processing of structured data. *IEEE Transactions on Neural Networks, 14*(3).

Vesanto, J., Himberg, J., Alhoniemi, E., & Parhan Kangas. J. (2000). *SOM toolbox for matlab.*

Deboeck, G., & Kohonen, T. (1998). *Visual explorations in finance with self-organising maps*. London: Spinger-Verlag.

Xu, J.S., & Wang, Z. O. (2004) Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*(3).