Vassiliki Moschou, Dimitrios Ververidis, Constantine Kotropoulos, "Assessment of self organizing map variants for clustering with application to redistribution of emotional speech patterns," Elsevier Neurocomputing, vol. 71, issues 1-3, pp. 147-156, 2007.

Assessment of self organizing map variants for clustering with application to redistribution of emotional speech patterns

Vassiliki Moschou, Dimitrios Ververidis, Constantine Kotropoulos *

Artificial Intelligence and Information Analysis Lab, Department of Informatics, Aristotle University of Thessaloniki, Box 451, Thessaloniki 54124, Greece

Abstract

Two well-known variants of the self-organizing map (SOM) that are based on multivariate order statistics are the marginal median SOM and the vector median SOM. In the past, their efficiency was demonstrated for color image quantization. We employ the well-known IRIS and VOWEL data sets and we assess the SOM variants' performance with respect to the accuracy, the average over all neurons mean squared error between the patterns that were assigned to a neuron and the neuron's weight vector, the Rand index, the Γ statistic, and the overall entropy. All figures of merit favor the marginal median SOM and the vector median SOM against the standard SOM. Based on the aforementioned findings, the marginal median SOM and the vector median SOM are used to redistribute emotional speech patterns from the Danish Emotional Speech database, that were originally classified as being neutral, to the emotional states of hot anger, happiness, sadness, and surprise.

Key words: Self-organizing map (SOM), Marginal Median SOM, Vector Median SOM, Emotional speech patterns, Danish Emotional Speech (DES) database.

1. Introduction

The neural networks constitute a powerful tool in pattern recognition. They have been an active research area for the past three decades due to their wide range of applications [1]. The self-organizing map (SOM) establishes a mapping from the input data space onto a low dimensional lattice of nodes so that a number of topologically ordered and well defined prototypes is produced. The nodes are organized on a map and they compete in order to win the input patterns [2]. The SOM is among the most popular neural networks. A number of 5384 related papers are reported [3,4]. Recent applications of SOM include exploratory analysis of high-dimensional data [5,6], human posture classification [7], and clustering [8,9] to mention a few.

We are interested in the class of SOM training algorithms that employ multivariate order statistics, such as the marginal median and the vector median [10]. These SOM variants as well as the standard SOM, that is trained with the batch algorithm (to be referred to as SOM hereafter), are applied to pattern clustering. The novel contribution of this work is in the assessment of SOM training algorithms in clustering with respect to the accuracy, the average over all neurons mean squared error, the Rand index, the Γ statistic, and the overall entropy. The superiority of the studied SOM variants against the SOM is demonstrated by experiments carried out using the well-known IRIS data and VOWEL data [11]. We also compare the SOM variants under study with the SOM in the redistribution of emotional speech patterns from the Danish Emotional Speech (DES) database [12], that were originally classified as being neutral, into the emotional states of hot anger, happiness, sadness, and surprise. The latter experiment is motivated by the following facts. First, on the one hand, there are emotional facial expression databases, such as the Action-Unit coded Cohn-Kanade database [13], where the neutral emotional class is not represented adequately. Accordingly, facial expression patterns are not assigned to the neutral emotional class [14]. On the other hand, for the emotional speech databases, there are utterances regularly classified as neutral. Accordingly, when the neutral class is not represented in one modality it is difficult to develop multimodal emotion recognition algorithms using feature fusion. Second, it is frequent that the ground truth information related to emotions, provided by human evaluators, is biased towards the neutral class. Therefore, the patterns classified as neutral might be needed to be redistributed among the non-neutral classes to enable further experimentation.

The outline of this paper is as follows. Section 2 describes briefly the SOM and the batch training algorithm as well as the

* Corresponding author.
Email addresses: vmoshou@aiia.csd.auth.gr (Vassiliki Moschou), jimver@aiia.csd.auth.gr (Dimitrios Ververidis), costas@aiia.csd.auth.gr (Constantine Kotropoulos).

SOM variants tested, namely the marginal median SOM (MMSOM) and the vector median SOM (VMSOM). In section 3, we define mathematically the evaluation measures employed, i.e. the accuracy, the average over all neurons mean squared error, the Rand index, the Γ statistic, and the overall entropy. This section also describes the Kuhn-Munkres algorithm [15] and how it is used to calculate the SOM accuracy. In section 4, the data we worked on are discussed. In section 5, the experimental results for clustering the IRIS and VOWEL data using the SOM, the MMSOM, and the VMSOM are demonstrated. Furthermore, figures of merit are presented and discussed for the redistribution of neutral speech patterns into four non-neutral emotional classes using the SOM, the MMSOM, and the VMSOM on the DES data. Finally, conclusions are drawn in section 6.

2. Self-organizing map and its variants

2.1. *Self-organizing map (SOM)*

The SOM forms a nonlinear mapping of an arbitrary D-dimensional input space onto a low (usually 2 or 3) dimensional lattice of nodes (the map). Each node is associated with a weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_D)^T$ in the input space. The SOM is trained iteratively and the weight vectors are updated properly, so that the nodes move to form clusters [16]. The training algorithm has two steps: *winner selection* and *weight adaptation*. In the winner selection step, the map nodes (*neurons*) compete each other in order to be activated by winning input patterns. Only *one* neuron wins at each iteration and becomes the winner or the *best matching unit* (BMU) [17]. In the weight adaptation step, the weight vector of every map neuron is updated by moving towards the input pattern. The amount of adaptation depends on how close each neuron is to the winner on the map. Hence, the map adapts to the input patterns in an ordered fashion.

Let us denote by $\mathbf{x}_j$ the jth D-dimensional input pattern and by $\mathbf{w}_i$ the ith D-dimensional weight vector. The *weight vector initialization* precedes both the winner selection and the weight adaptation steps and is crucial for the algorithm performance. Several initialization algorithms have been developed. For example, the linear initialization algorithm calculates the two eigenvectors that correspond to the two largest eigenvalues of the covariance matrix of the input patterns and defines the hyperplane on which the neuron grid lies onto. The eigenvectors can be calculated using the Jacobbi transformation algorithm or the Givens and Householder reduction algorithm [18]. The sample initialization algorithm initializes the weight vectors with random samples from the input data set, while the random initialization algorithm with small random values [17]. In our experiments, the linear initialization algorithm was used.

The weight vectors, $\mathbf{w}_i$, define the *Voronoi tessellation* of the input space [1,2,19]. Each Voronoi cell is represented by its centroid, that becomes the corresponding weight vector $\mathbf{w}_i$. In the winner selection step, each input pattern $\mathbf{x}_j$ is assigned to a Voronoi cell based on the nearest neighbor condition. That is, the BMU index, $c(j)$, of the input pattern $\mathbf{x}_j$ is defined by

$$c(j) = \arg\min_i \{\|\mathbf{x}_j - \mathbf{w}_i\|\} \tag{1}$$

where $\|.\|$ denotes the Euclidean distance. Accordingly, the SOM can be treated as a vector quantization method [20]. Due to the fact that the input patterns $\mathbf{x}_j$ are random vectors, the quantization error $\|\mathbf{x}_j - \mathbf{w}_i\|$ is also a random variable [21]. Furthermore, provided that the weight density is proportional to the input density, the map can be regarded as a non-parametric model of the input density $\mathcal{P}(\mathbf{x})$. When the number of neurons is large, $\mathcal{P}(\mathbf{x})$ is approximately constant in the Voronoi region of the neuron that has won the pattern [21].

The most important step of the SOM algorithm is the weight adaptation. The neurons are related by a *neighborhood function*, dictating the structure of the map topology. The neighborhood function determines how strongly the neurons are related to each other [1]. At each training step, the neuron updating depends on the neighborhood function, whose purpose is to correlate the directions of the weight updates of a large number of neurons around the BMU [21]. The larger the neighborhood, the more rigid the SOM is. A variety of neighborhood functions can be used. The neighborhood function can be defined to be decreasing or constant around the winner neuron. A Gaussian kernel can also be used. However, the latter kernel is computationally demanding, due to the exponential function that should be calculated. The Gaussian kernel can be approximated well by the bubble neighborhood function, which is simpler. Using the bubble neighborhood function, every neuron in the winner neighborhood is updated by the same proportion of the difference between the neuron and the incoming pattern. The Gaussian kernel was used for training in our experiments. Its neighborhood around the winner neuron $c(j)$ is defined as [17]

$$h_{ic(j)}(t) = \exp^{-d^2_{ic(j)}/2\sigma_t^2} \tag{2}$$

where $d_{ic(j)} = \|\mathbf{r}_{c(j)} - \mathbf{r}_i\|$ is the distance between map units $c(j)$ and i *on the map grid*, $\mathbf{r}_i$ denotes the coordinates of the ith neuron on the map, and σ_t is the neighborhood radius at time t.

To update the winner neurons and their neighbors, either a Least Mean Squared (LMS) type adaptation rule [1] or a batch algorithm can be employed. We are interested in the latter. In the batch training algorithm, for a fixed training set $\{\mathbf{x}_j\}$ of N patterns, we keep record of the weight updates, but their adjustment is applied only after all training samples have been considered. The algorithm does not depend on the order of presentation of input patterns. The learning stops when a predetermined number of iterations is reached [21]. At each training iteration, the BMU of each pattern is determined. Afterwards, all neurons that belong to the BMU neighborhood are updated. The updating rule of the ith weight vector $\mathbf{w}_i$ is computed as [1,17]

$$\mathbf{w}_i(t+1) = \frac{\sum_{j=1}^{M} h_{ic(j)}(t)\, \mathbf{x}_j}{\sum_{j=1}^{M} h_{ic(j)}(t)} \tag{3}$$

where M denotes the number of patterns $\mathbf{x}_j$ that have been assigned to the ith neuron up to the tth iteration.

The training is performed in two phases, namely the *rough training* and the *fine tuning* phases. In the rough training phase,

a large initial neighborhood radius σ_0 is used, in order to have a rigid SOM, that decreases through time. In the fine tuning phase, the neighborhood radius is initially small and shrinks as time passes [2,17]. This corresponds to first tuning the SOM to the input data and then to fine tuning the map. Neurons that belong to the BMU's neighborhood are closer to the BMU and are updated more heavily than others. As the distance from the BMU increases, the updating quantity decreases. Concerning the neighborhood, as its range is decreased, so does the number of neurons whose weight update direction is correlated. As a result, neighboring neurons will be specialized for similar input patterns [21]. The topological information of the map ensures that neighboring neurons on the grid possess similar attributes. When the SOM utilizes an one-dimensional map and the degree of neighborhood is zero, it is equivalent to a variant of the k-means algorithm [22].

It must be mentioned that, due to neighborhood shrinking that is performed through time, a SOM may have "dead" units. "Dead" units are neurons, which subsequently fail to be associated with any input pattern. They have a zero (or very low) probability to be activated [21].

2.2. *SOM variants based on order statistics*

The SOM has some disadvantages, such as lack of robustness against outliers and against erroneous choices for the winner vector due to the linear estimators, e.g. (3) [10]. In order to deal with these problems, variants of the standard SOM that employ multivariate order statistics (OS) can be used.

The SOM variants that are based on multivariate OS differentiate in the way they update the weight vectors. The MMSOM updates the weight vectors using the marginal median, while the VMSOM applies the vector median [10,19]. In contrast, the SOM calculates the weighted mean of the input patterns, as can be seen in (3). The MMSOM and the VMSOM treat efficiently the outliers, because they inherit the robustness properties of the OS [10,19]. The MMSOM has been successfully applied to color image quantization [10] and document organization and retrieval [23]. Another recent application of the MMSOM is in grouping and visualization of human endogenous retroviruses [24].

In subsections 2.2.1 and 2.2.2, $R_i(t)$ denotes the input patterns assigned to the ith neuron until the tth iteration and $\mathbf{x}(t)$ denotes the input pattern assigned to the neuron at the tth iteration.

2.2.1. *Marginal median SOM (MMSOM)*

The MMSOM updates the weight vectors of the neurons that belong to the neighborhood of the BMU. It calculates the marginal median of all patterns assigned to a neuron. The MMSOM relies on the concept of marginal ordering. The marginal ordering of M input patterns $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$, where $\mathbf{x}_j = (x_{1j}, x_{2j}, \ldots, x_{Dj})^T$, is performed by ordering the pattern components independently along each of the D dimensions [10,19]

$$x_{q(1)} \leq x_{q(2)} \leq \cdots \leq x_{q(M)}, \quad q = 1, 2, \ldots, D \tag{4}$$

with q denoting the pattern component index. The updated neuron weights emerge from the calculation of the marginal median of the patterns indexed by the ith neuron, weighted by a factor that dictates the neighborhood. The marginal median is defined by [25]

$$\text{marginal median } \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\} = \\ = \begin{cases} (\frac{x_{1(v)} + x_{1(v+1)}}{2}, \ldots, \frac{x_{D(v)} + x_{D(v+1)}}{2})^T, & \text{if } M = 2v \\ (x_{1(v+1)}, \ldots, x_{D(v+1)})^T, & \text{if } M = 2v+1 \end{cases} \tag{5}$$

where M denotes the number of patterns assigned to the neuron. The ith neuron is updated by

$$\mathbf{w}_i(t+1) = h_{ic(\mathbf{x}(t))}(t) \text{ marginal median } \{R_i(t) \cup \mathbf{x}(t)\}. \tag{6}$$

2.2.2. *Vector median SOM (VMSOM)*

The VMSOM updates the weight vectors of the neurons that belong to the neighborhood of the BMU. It calculates the vector median of the patterns assigned to a neuron. The vector median operator is the vector that belongs to the set of input vectors assigned to the ith neuron, which is the closest one to all the current input patterns. The vector median of M input patterns $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$ is defined by [26]

$$\text{vector median } \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} = \mathbf{x}_l$$
$$\text{where} \quad l = \arg\min_k \sum_{j=1}^{N} |\mathbf{x}_j - \mathbf{x}_k|. \tag{7}$$

The ith neuron is updated by

$$\mathbf{w}_i(t+1) = h_{ic(\mathbf{x}(t))}(t) \text{ vector median } \{R_i(t) \cup \mathbf{x}(t)\}. \tag{8}$$

3. Clustering evaluation measures

Five measures are employed in order to assess the performance of the SOMs under study, namely the accuracy, the average over all neurons mean squared error, the Rand index, the Γ statistic, and the overall entropy. Let

N_f be the total number of classes the patterns are initially grouped into according to the ground truth;
N_c be the total number of clusters created by the SOMs;
N be the total number of patterns;
n_{ij} be the total number of patterns in cluster i that belong to class j;
$n_{i.}$ be the total number of patterns in cluster i;
$n_{.j}$ be the total number of patterns that belong to class j;
M_c be the number of combinations of two patterns that can be taken out from the input data set;

Equations (9)-(12) indicate the relations between the aforementioned variables:

$$n_{i.} = \sum_{j=1}^{N_f} n_{ij} \tag{9}$$

$$n_{.j} = \sum_{i=1}^{N_c} n_{ij} \tag{10}$$

$$N = \sum_{i=1}^{N_c} \sum_{j=1}^{N_f} n_{ij} \tag{11}$$

$$M_c = \frac{N(N-1)}{2}. \tag{12}$$

3.1. *Accuracy*

Let T be the total number of patterns that compose the test set and $\delta(x, y)$ be the delta Kronecker, which equals 1 if $x = y$, and 0 otherwise. The accuracy of the assignment performed by the SOM is defined as [27]

$$AC = \frac{1}{T} \sum_{j=1}^{T} \delta(g(\mathbf{x}_j), map(\phi(\mathbf{x}_j))) \tag{13}$$

where $g(\mathbf{x}_j)$ is the true label of pattern $\mathbf{x}_j$, $\phi(\mathbf{x}_j)$ is the label assigned to $\mathbf{x}_j$ by the SOM or its variants, and $map(v_i)$ is the *optimal matching*, which maps the label assigned to the pattern onto the ground truth labels. The optimal matching can be derived by the Kuhn-Munkres algorithm [15].

The problem solved by the Kuhn-Munkres algorithm is stated as follows. Let us denote $V = \{v_i\}$ and $U = \{u_i\}$, where $i = 1, 2, \ldots, N_c$ with N_c being the number of nodes. Consider a complete weighted bipartite graph $G = (V \bigcup U, V \times U)$. The weight of the edge (v_i, u_i) is denoted by $\xi(v_i, u_i)$. The goal is to find the optimal matching from V to U. That is, the matching with the maximum sum of the edge weights that belong to it.

Mathematically, given an $N_c \times N_c$ weight matrix Ξ, which represents the graph G, a permutation π of $1, 2, \ldots, N_c$ must be found so that the following sum

$$\sum_{i=1}^{N_c} \xi(v_i, u_{\pi(i)}) \tag{14}$$

is maximized. The resulted set of edges is the optimal matching. A graph that is not complete, it must be forced to become a complete one, by adding zeros for the non-existing edges in the weight matrix Ξ.

Let us explain the use of the Kuhn-Munkres algorithm in the calculation of the SOM clustering accuracy. The accuracy of the assignment performed by the SOM is defined by (13). It is assumed that the patterns must be clustered into N_c clusters. That is, the number of nodes of the graph G is N_c. Ideally, N_c should equal N_f. The weight $\xi(v_i, u_i)$, assigned to the edge (v_i, u_i), corresponds to the profit made out, if the label assigned by the SOM is v_i and the ground truth class label is u_i. The purpose is to maximize the profit. Obviously, if the two labels are the same, the profit is maximized.

The input to the Kuhn-Munkres algorithm is an $N_c \times N_c$ weight matrix Ξ. Let $\xi(i, i) = 1$, when $i = 1, 2, \ldots, N_c$, and $\xi(i, j) = -1$, when $j = 1, 2, \ldots, N_c$ and $i \neq j$. Negative profit is made out when the labels assigned by the SOM differ from the actual ground truth labels (alternatively, a low profit value could be used instead of -1). For the IRIS data, the ground truth labels and the labels assigned by the SOM are 0, 1, and 2, while for the VOWEL data the labels are integers between 1-15 (section 5). The weight matrix Ξ provided as input to the Kuhn-Munkres algorithm for the IRIS data is shown in Table 1. Rows represent the labels assigned by the SOM, while columns represent the ground truth labels.

Table 1
Weight matrix Ξ provided as input to the Kuhn-Munkres algorithm for the IRIS data set.

Labels	u_1	u_2	u_3
v_1	1	-1	-1
v_2	-1	1	-1
v_3	-1	-1	1

The output of the algorithm is the optimal matching, represented by an $N_c \times N_c$ matrix OM. $OM(i, j)$ equals 1 if the edge (v_i, u_j) belongs to the optimal matching, otherwise it equals 0. Table 2 shows the optimal matching OM derived by the Kuhn-Munkres algorithm on the IRIS data. As it was expected, if the actual ground truth label coincides with the label assigned by the SOM, the corresponding edge belongs to the optimal matching.

Table 2
The optimal matching OM derived by the Kuhn-Munkres algorithm on the IRIS data.

Labels	u_1	u_2	u_3
v_1	1	0	0
v_2	0	1	0
v_3	0	0	1

3.2. *Average over all neurons Mean Squared Error (AMSE)*

In order to set the definition of the AMSE, we must first define the Mean Squared Error (MSE). The MSE of one neuron is the mean value of the Euclidean distances between its weight vector and all the patterns assigned to it. Mathematically, the MSE of the neuron $\mathbf{w}_i$ is calculated as follows:

$$MSE_i = \frac{1}{M} \sum_{j=1}^{M} \|\mathbf{x}_{j[i]} - \mathbf{w}_i\|^2 \tag{15}$$

where M is the total number of patterns assigned to the ith neuron and $\mathbf{x}_{j[i]} \in R_i(t)$ is the j-th pattern assigned to this neuron. The average over all neurons MSE, which from now on will be referred to as AMSE, is the average value of MSE_i for all the neurons of the map

$$AMSE = \frac{1}{K} \sum_{i=1}^{K} MSE_i \tag{16}$$

where K is the total number of the map neurons.

3.3. *Rand index*

The Rand index is a widely used cluster validity measure in partitional structures [28]. In order to validate the clustering

structure $\mathcal{C}$ derived by the SOMs, a partition $\mathcal{P}$ of the data (i.e. the ground truth) must be available. The Rand index indicates the number of input patterns that are either from the same class (according to $\mathcal{P}$) but are not grouped into the same cluster (according to $\mathcal{C}$), or that are not from the same class but are grouped into the same cluster. The Rand index is defined as follows [28, p. 173-174]:

$$\gamma = 1 + \frac{1}{M_c}\left[\sum_{i=1}^{N_c}\sum_{j=1}^{N_f} n_{ij}^2 - \frac{1}{2}\sum_{i=1}^{N_c} n_{i.}^2 - \frac{1}{2}\sum_{j=1}^{N_f} n_{.j}^2\right]. \quad (17)$$

The Rand index admits values in the range [0,1]. High value of the Rand index implies close agreement between $\mathcal{C}$ and $\mathcal{P}$. A perfect clustering (i.e. $\gamma = 1$) may not be achievable, when $\mathcal{C}$ and $\mathcal{P}$ have different number of clusters/classes [28].

3.4. Γ *statistic*

The Γ statistic is a special case of Hubert's Γ statistic [28]. It follows the idea of partitional structure validity. That is, the ground truth of the data $\mathcal{P}$ must be available to be compared with the clustering $\mathcal{C}$ derived by the SOM or its variants. Let us define the following auxiliary variables:

$a = \frac{1}{2}\sum_{i=1}^{N_c}\sum_{j=1}^{N_f} n_{ij}^2 - (N/2)$;
$b = \frac{1}{2}\sum_{j=1}^{N_f} n_{.j}^2 - \frac{1}{2}\sum_{i=1}^{N_c}\sum_{j=1}^{N_f} n_{ij}^2$;
$c = \frac{1}{2}\sum_{i=1}^{N_c} n_{i.}^2 - \frac{1}{2}\sum_{i=1}^{N_c}\sum_{j=1}^{N_f} n_{ij}^2$;
$m_1 = a + b$;
$m_2 = a + c$;

The Γ statistic calculates the correlation between the two partitions. It is defined as follows [28]:

$$\Gamma = \frac{(M_c\, a - m_1 m_2)}{[m_1 m_2\,(M_c - m_1)\,(M_c - m_2)]^{1/2}}\ . \quad (18)$$

Since the Γ statistic is a correlation coefficient, its value ranges between -1 to 1. A high value of the Γ statistic implies high correlation between $\mathcal{C}$ and $\mathcal{P}$ and thus, a good clustering result.

3.5. *Overall entropy (OE)*

In order to define the OE, we must first define the *cluster entropy* and the *class entropy*. The quality of a clustering structure $\mathcal{C}$ can be evaluated according to the ground truth labels of patterns, $\mathcal{P}$. For each cluster, c_i, the cluster entropy E_{c_i} is computed by [22]

$$E_{c_i} = -\sum_{j=1}^{N_f} \frac{n_{ij}}{n_{i.}} \log \frac{n_{ij}}{n_{i.}}. \quad (19)$$

The *overall cluster entropy*, E_c, is given by the weighted sum of the individual cluster entropies [22]

$$E_c = \frac{1}{N}\sum_{i=1}^{N_c} n_{i.}\ E_{c_i}. \quad (20)$$

The cluster entropy reflects the quality of individual clusters in terms of the homogeneity of the patterns in a cluster. It admits values in the range [0,1]. Low values of the cluster entropy indicate high homogeneity. However, the cluster entropy does not measure the cluster compactness in terms of the number of clusters generated. Some clustering algorithms might produce many clusters, which leads to low cluster entropy values, but is not usually desirable. For this reason, the *overall class entropy* is used to measure how the patterns of the same class are represented by the clusters created. Similarly to the cluster entropy, the overall class entropy E_l is defined as [22]

$$E_l = \frac{1}{N}\sum_{j=1}^{N_f} n_{.j}\ E_{l_j} \quad (21)$$

where E_{l_j} is the class entropy for class j given by [22]

$$E_{l_j} = -\sum_{i=1}^{N_c} \frac{n_{ij}}{n_{.j}} \log \frac{n_{ij}}{n_{.j}}. \quad (22)$$

The class entropy also admits values in the range [0,1]. The overall entropy is defined as

$$OE = \beta E_c + (1 - \beta) E_l \quad (23)$$

with $\beta \in [0, 1]$ functioning as a weight parameter that balances cluster and class entropies. In our experiments, β was chosen to be 0.5. Low OE values indicate better clustering performance than high OE values.

4. Data

The well-known IRIS data was used, in order to evaluate the performance of the algorithms for clustering. The IRIS data records information about 150 flower patterns [29]. Each pattern is characterized by 4 features, namely the sepal length, the sepal width, the petal length, and the petal width. The patterns are classified into 3 classes called Setosa, Versicolor, and Virginica. The most important feature of the IRIS data is the ground truth of the patterns, i.e. the actual class each pattern is classified to.

It must be noted that the IRIS data set *does* contain outliers for unsupervised learning, as can be seen in Figure 1. Accordingly, this data set is appropriate for studying the role of the outliers in clustering. This is not the case for supervised learning [30, p.346]. Furthermore, the IRIS data set is widely used in clustering applications reported to bibliography [31].

In addition, the VOWEL data was also used for experiments. The VOWEL data records information about the 11 steady state vowels of British English, namely i, I, E, A, Y, a, O, o, U, u, and e [11]. There are 15 individual speakers, each saying each vowel six times. For each utterance, 10 linear prediction coefficients-derived log area ratios have been extracted. The ground truth information, that is the vowel that corresponds to each feature vector, is available.

Motivated by the observations made on the IRIS and VOWEL data, we shall compare the SOM variants against the SOM for the redistribution of neutral emotional speech patterns from the DES database [12] into non-neutral emotional states. We decided to work on the DES database, because it is easily accessible and well annotated. A number of 1160 emotional speech

patterns are extracted. Each pattern consists of a 90-dimensional feature vector [32]. Each emotional pattern is classified into one of the five primitive emotional states, such as hot anger, happiness, neutral, sadness, and surprise. The ground truth for all patterns is also available.

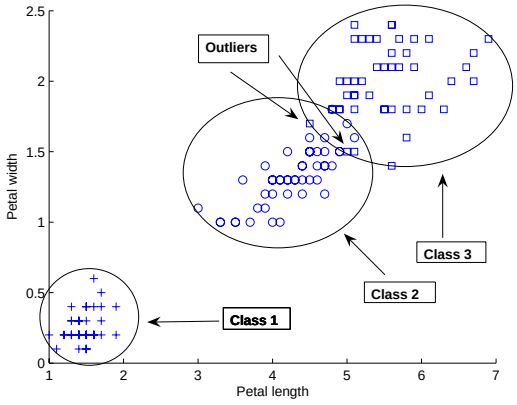


Fig. 1. Petal width vs. petal length for IRIS data.

5. Experimental Results

The performance of the SOM, the MMSOM, and the VMSOM on clustering was demonstrated through all the evaluation measures. Two experiments were carried out. In the first experiment, we worked on the IRIS data. The training set consists of 120 randomly selected patterns, while the test set is composed by the 30 remaining patterns. For the second experiment, the VOWEL data was used. To create the training set, we use 44 feature vectors for each speaker, while the remaining 22 feature vectors of each speaker are used for testing. The accuracy, the AMSE, the Rand index, the Γ statistic, and the OE were measured using 30-fold cross validation for different map sizes. For a high quality clustering, as the number of the map neurons increases, the accuracy and the Γ statistic should increase, while the AMSE, the Rand index, and the OE should decrease.

Tables 3, 4, and 5 summarize the accuracy, the AMSE, the Rand index, the Γ statistic, and the OE of the SOM, the MMSOM, and the VMSOM, respectively, using different map sizes on the IRIS data. The results presented are averaged over the 30 cross validations. The best performance concerning all evaluation measures is indicated in boldface.

As it can be noticed from Tables 3, 4, and 5, the MMSOM yields the best behavior concerning all the evaluation measures compared to the SOM and the VMSOM. In detail, the MMSOM yields the best accuracy (97.77%), the VMSOM follows (97.22%), while the SOM has the worst behavior with respect to the accuracy (94.44%). The best value for the Γ statistic (0.9353) is measured for the MMSOM. The Γ statistic values measured for the VMSOM and the SOM are 0.9177 and 0.8498, respectively. The same ordering between the three SOMs stands also with respect to the Rand index. Furthermore, the smallest AMSE is measured for the MMSOM (0.20). The VMSOM yields a larger AMSE than the MMSOM (0.23), and the SOM exhibits the worst performance with respect to the AMSE (0.45). Finally, the smallest OE value is demonstrated by the MMSOM (0.0458). The smallest OE values measured for the VMSOM and the SOM are 0.0567 and 0.0765, respectively. The best values for all the evaluation measures emerge for a 4×4 map.

Table 3
Accuracy, AMSE, Rand index, Γ statistic, and OE of the SOM for different map sizes on the IRIS data.

Neurons	SOM				
	AC(%)	AMSE	Rand	Γ	OE
3 (2×2)	81.88	1.65	0.9884	0.6488	0.1302
4 (2×2)	82.22	1.66	0.9890	0.6427	0.1298
5 (3×2)	89.88	1.29	0.9928	0.7601	0.1228
6 (3×2)	91.66	1.25	0.9939	0.7807	0.1146
7 (4×2)	93.88	0.71	0.9955	0.8388	0.0876
8 (4×2)	83.66	1.17	0.9898	0.6281	0.1360
9 (3×3)	84.44	1.21	0.9904	0.6496	0.1396
10 (3×3)	84.88	1.13	0.9902	0.6553	0.1363
11 (4×3)	90.88	0.57	0.9936	0.7650	0.0952
12 (4×3)	91.11	0.53	0.9938	0.7729	0.0898
16 (4×4)	**94.44**	**0.45**	**0.9958**	**0.8498**	**0.0765**

Table 4
Accuracy, AMSE, Rand index, Γ statistic, and OE of the MMSOM for different map sizes on the IRIS data.

Neurons	MMSOM				
	AC(%)	AMSE	Rand	Γ	OE
3 (2×2)	88.88	0.53	0.9925	0.7287	0.1077
4 (2×2)	89.66	0.53	0.9925	0.7287	0.1077
5 (3×2)	96.44	0.33	0.9972	0.9040	0.0515
6 (3×2)	96.55	0.34	0.9972	0.8988	0.0484
7 (4×2)	96.66	0.34	0.9974	0.9106	0.0489
8 (4×2)	96.44	0.23	0.9973	0.9024	0.0461
9 (3×3)	97.00	0.26	0.9977	0.9150	0.0507
10 (3×3)	96.11	0.25	0.9969	0.8888	0.0491
11 (4×3)	96.66	0.23	0.9975	0.9112	0.0541
12 (4×3)	96.44	0.22	0.9973	0.9005	0.0510
16 (4×4)	**97.77**	**0.20**	**0.9982**	**0.9353**	**0.0458**

Table 5
Accuracy, AMSE, Rand index, Γ statistic, and OE of the VMSOM for different map sizes on the IRIS data.

Neurons	VMSOM				
	AC(%)	AMSE	Rand	Γ	OE
3 (2×2)	89.88	0.53	0.9932	0.7455	0.1152
4 (2×2)	88.66	0.56	0.9923	0.7234	0.1140
5 (3×2)	95.55	0.41	0.9965	0.8749	0.0681
6 (3×2)	94.55	0.42	0.9957	0.8432	0.0647
7 (4×2)	93.88	0.35	0.9957	0.8465	0.0694
8 (4×2)	96.55	0.29	0.9973	0.9031	0.0588
9 (3×3)	96.44	0.28	0.9972	0.8971	0.0584
10 (3×3)	96.33	0.31	0.9971	0.8931	0.0534
11 (4×3)	95.88	0.26	0.9969	0.8841	0.0585
12 (4×3)	96.22	0.25	0.9971	0.8959	0.0598
16 (4×4)	**97.22**	**0.23**	**0.9977**	**0.9177**	**0.0567**

The Student t-test for unequal variances [18] has been used to check whether the difference between the mean accuracies

achieved by the following algorithm pairs (SOM, MMSOM), (SOM, VMSOM), and (MMSOM, VMSOM) is statistically significant at the 95% level of significance in a 30-fold cross validation experiment with a 4×4 map. The same assessment has also been performed for the AMSE, the Rand index, the Γ statistic, and the OE. It was proven that the differences *are* statistically significant.

The superiority of the MMSOM and the VMSOM compared to the SOM is demonstrated on the VOWEL data set for different map sizes in Tables 6 - 8. The experimental findings reveal that VMSOM yields the best clustering.

Table 6
Accuracy, AMSE, Rand index, Γ statistic, and OE of the SOM for different map sizes on the VOWEL data.

Neurons	SOM				
	AC(%)	AMSE	Rand	Γ	OE
110	44.04	1.49	0.9078	0.2228	0.5893
120	46.75	1.45	0.9048	0.2318	0.6067
130	49.20	1.40	0.9066	0.2536	0.5841
140	51.16	1.35	0.9102	0.2690	0.5666
150	53.75	1.31	0.9121	0.2916	0.5506
160	52.58	1.25	0.9096	0.2769	0.5573
170	55.87	1.22	0.9144	0.3131	0.5247
180	**57.54**	**1.22**	**0.9175**	**0.3314**	**0.5133**

Table 7
Accuracy, AMSE, Rand index, Γ statistic, and OE of the MMSOM for different map sizes on the VOWEL data.

Neurons	MMSOM				
	AC(%)	AMSE	Rand	Γ	OE
110	53.54	1.02	0.9113	0.2920	0.5455
120	55.12	1.01	0.9122	0.3039	0.5274
130	56.33	0.99	0.9135	0.3183	0.5188
140	57.37	0.98	0.9161	0.3340	0.5046
150	59.62	0.95	0.9198	0.3596	0.4786
160	58.20	0.95	0.9168	0.3330	0.5044
170	58.87	0.95	0.9172	0.3435	0.4918
180	**61.45**	**0.94**	**0.9215**	**0.3679**	**0.4761**

Table 8
Accuracy, AMSE, Rand index, Γ statistic, and OE of the VMSOM for different map sizes on the VOWEL data.

Neurons	VMSOM				
	AC(%)	AMSE	Rand	Γ	OE
110	58.62	1.12	0.9169	0.3424	0.5019
120	59.83	1.07	0.9182	0.3558	0.4887
130	61.95	1.03	0.9213	0.3765	0.4715
140	62.95	1.02	0.9227	0.3896	0.4556
150	64.79	0.97	0.9272	0.4185	0.4375
160	66.20	0.96	0.9290	0.4294	0.4286
170	66.45	0.96	0.9290	0.4319	0.4239
180	**66.83**	**0.93**	**0.9298**	**0.4383**	**0.4150**

As it can be noticed form Tables 4, 5, 7, and 8 both the MMSOM and the VMSOM have similar values that do not change significantly with the map size, concerning all the evaluation measures. In contrast, the SOM values (Tables 3 and 6) change significantly with the map size compared to the MMSOM and the VMSOM. This fact can be explained by the number of "bad" neurons of each SOM. Let us denote by μ, σ, and M the mean number of patterns, the standard deviation, and the exact number of patterns that a neuron wins during training. The "bad" neurons are those for which the following inequality holds: $M < \mu - \sigma$. Tables 9 and 10 present the number of "bad" neurons of each SOM for different map sizes on the IRIS and VOWEL data, respectively. It is obvious that the number of "bad" neurons for the SOM gets very large with increasing map size, causing the significant difference of its performance compared to that of the SOM variants. Both the MMSOM and the VMSOM, have a smaller number of "bad" neurons than the SOM for all map sizes, which explains their superior objective figures of merit with respect to those of SOM.

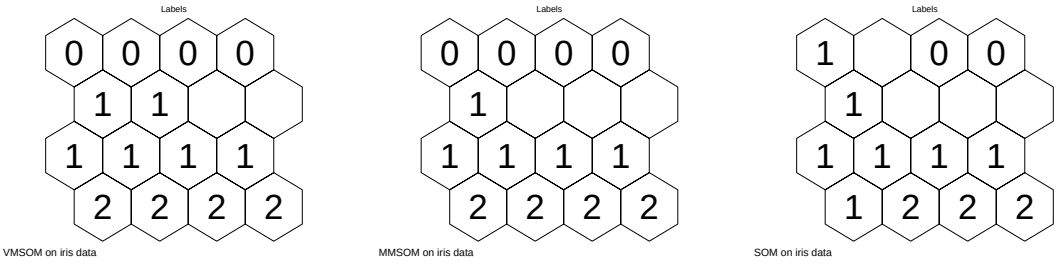
Table 9
Number of "bad" neurons for different SOM sizes on the IRIS data.

Neurons	SOM	MMSOM	VMSOM
3 (2×2)	2	2	0
4 (2×2)	2	2	0
5 (3×2)	4	3	0
6 (3×2)	4	3	0
7 (4×2)	4	2	1
8 (4×2)	7	2	0
9 (3×3)	7	2	0
10 (3×3)	7	2	0
11 (4×3)	7	5	2
12 (4×3)	8	5	1
16 (4×4)	8	4	5

Table 10
Number of "bad" neurons for different SOM sizes on the VOWEL data.

Neurons	SOM	MMSOM	VMSOM
110	33	20	5
120	34	22	13
130	46	30	18
140	54	35	21
150	58	40	23
160	68	55	33
170	71	53	35
180	74	63	36

Figure 2 depicts the 4×4 maps created for the IRIS data by the VMSOM, the MMSOM, and the SOM, respectively. The numbers indicate the class in which each neuron has been assigned to. The numbers correspond to the ground truth classes as follows: 0 is assigned to class Setosa, 1 to class Versicolor, and 2 to class Virginica. The empty neurons, that have not been assigned any label, are "dead" neurons. As it can be noticed, the SOM contains more "dead" neurons than the MMSOM and the VMSOM and cannot represent data well. The maps created by the SOM variants are more representative, due to their properties. The latter have less "dead" neurons than the



(a) (b) (c)

Fig. 2. Labeling of the 4×4 maps created for IRIS data by (a) the VMSOM, (b) the MMSOM, and (c) the SOM.

SOM and the clusters defined on the map are well separated. However, "dead" units are inevitable for a SOM [21].

The SOMs were also applied to the redistribution of emotional speech patterns extracted from the DES database. The primitive emotional states are anger, happiness, neutral, sadness, and surprise. Our purpose is to redistribute the emotional speech patterns, that were originally classified as neutral, into the other four emotional states. That is, to find out which class is closer to the neutral one and how each training algorithm acts on the data. The training set consists of all the non-neutral patterns and the test set consists of all the neutral patterns. The average assignment ratio was estimated using 15-fold cross validation.

Table 11 demonstrates the average assignment ratio of the neutral patterns that are labeled by each SOM as angry, happy, sad, and surprised. As it can be seen, all the algorithms classify the neutral patterns as *sad* with a very high ratio. This means that sadness resembles the neutral state more than the other emotional states. The largest reassignment ratio is measured for the MMSOM (61.86%), the next larger ratio is provided by the VMSOM (61.51%) and, finally, the SOM yields the lowest one (58.27%). Anger is the second closer to the neutral emotion, happiness follows, and, finally, surprise is the least similar to neutrality, according to the SOM and the VMSOM. The MMSOM yields a slightly different ordering. According to the MMSOM, happiness is the second closer to neutrality, anger follows, and surprise resembles neutrality the least.

Table 11
Average ratio of neutral emotional speech patterns assigned to non-neutral emotional classes using the SOM variants.

Emotion	Average assignment ratio (%)		
	SOM	MMSOM	VMSOM
Sadness	**58.27**	**61.86**	**61.51**
Anger	13.87	14.02	15.00
Happiness	13.56	14.81	13.62
Surprise	13.16	9.59	9.82

The Student t-test for unequal variances has also found that the differences in the average assignment ratio per emotion *are* statistically significant at the 95% level of significance in a 15-fold cross validation experiment with a 17×8 map.

Figure 3 depicts a partition of the 2D feature domain that has been resulted after selecting the five best emotional features by the Sequential Forward Selection algorithm and applying Principal Component Analysis (PCA), in order to reduce the dimensionality from five dimensions (5D) to two dimensions (2D) [32]. Only the samples which belong to the interquartile range of the probability density function for each class are shown. It can be seen that the neutral emotional class does not possess any overlap with the surprise, while such overlap is observed for sadness, anger, and happiness. Therefore, the results shown in Table 11 comply with the sample space depicted in Figure 3.

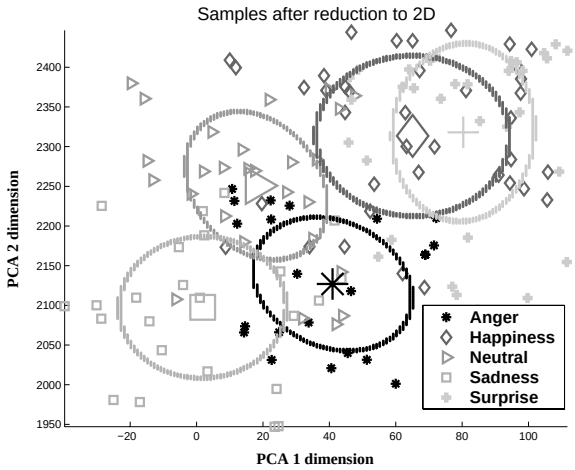


Fig. 3. Partition of the 2D domain into five emotional states derived by PCA. The samples which belong to the interquartile range of each pdf are shown. The big symbols denote the mean of each class. The ellipses denote the 60% likelihood contours for a 2-D Gaussian model.

6. Conclusions

Two variants of the self organizing map, the MMSOM and the VMSOM, that are based on order statistics, have been studied. These variants have been successfully used in color quantization, document organization and retrieval, and in grouping and visualizing human retroviruses. We presented experimental evidence for their clustering quality by using the accuracy, the average over all neurons mean squared error, the Rand index, the Γ statistic, and the overall entropy as figures of merit. The assessment was first conducted on the well-known IRIS and

VOWEL data sets. Motivated by the superiority of the SOM variants that are based on order statistics, we investigated their application in the redistribution of emotional neutral patterns to non-neutral emotional states. We demonstrated that the redistribution is consistent with the sample feature space.

Acknowledgments

This work has been supported by the "PYTHAGORAS II" Programme, funded in part by the European Union (75%) and in part by the Hellenic Ministry of Education and Religious Affairs (25%).

References

[1] S. Haykin, Neural Networks: A Comprehensive Foundation (2/e., Prentice-Hall, Upper Saddle River, N. Y., 1999).

[2] T. Kohonen, Self-Organizing Maps (Springer-Verlag, Berlin, 2000).

[3] S. Kaski, J. Kangas, and T. Kohonen, Bibliography of Self-Organizing Map (SOM) Papers: 1981-1997, Neural Computing Surveys, Vol. 1 (1998) 102-350.

[4] M. Oja, S. Kaski, and T. Kohonen, Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum, Neural Computing Surveys, Vol. 3 (2003) 1-156.

[5] A. Rauber, D. Merkl, and M. Dittenbach, The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data, Neural Networks, Vol. 13 (6) (November 2002) 1331-1341.

[6] J. Peltonen, A. Klami, and S. Kaski, Improved learning of Riemannian metrics for exploratory analysis, Neural Networks, Vol. 17 (8-9) (2004) 1087-1100.

[7] K. Takahashi and S. Sugakawa, Remarks on human posture classification using self-organizing map, in: Proc. IEEE Int. Conf. Systems, Man, and Cybernetics, Vol. 3 (October 2004) 2623-2628.

[8] I. Lapidot, H. Guterman, and A. Cohen, Unsurpervised speaker recognition based on competition between self-organizing maps, IEEE Trans. Neural Networks, Vol. 13 (4) (July 2002) 877-887.

[9] J. Sinkkonen and S. Kaski, Clustering based on conditional distributions in an auxiliary space, Neural Computation, Vol 14 (1) (January 2002) 217-239.

[10] I. Pitas, C. Kotropoulos, N. Nikolaidis, R.Yang, and M. Gabbouj, Order statistics learning vector quantizer, IEEE Trans. Image Processing, Vol. 5 (June 1996) 1048-1053.

[11] `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/undocumented/connectionist-bench/vowel/`

[12] I. S. Engberg and A. V. Hansen, Documentation of the Danish Emotional Speech Database DES, Internal Report, Center for Person Kommunikation, Aalborg University (1996).

[13] J. C. T. Kanade and Y. Tian, Comprehensive database for facial expression analysis, in: Proc. IEEE Int. Conf. Face and Gesture Recognition (March 2000) 46-53.

[14] I. Kotsia and I. Pitas, Real-time facial expression recognition from image sequences using support vector machines, in: Proc. Conf. Visual Communications Image Processing (Beijing, China, July 2005) 966-969.

[15] J. A. McHugh, Algorithmic Graph Theory (Prentice Hall, Upper Saddle River, N. Y., 1990).

[16] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, On clustering validation techniques, J. Intelligent Information Systems, Vol. 17 (2) (December 2001) 107-145.

[17] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, SOM Toolbox for Matlab 5 (2000) `www.cis.hut.fi`.

[18] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, Numerical Recipes in C: The Art of Scientific Computing (Cambridge University Press, Cambridge, U. K., 1992).

[19] C. Kotropoulos and I. Pitas, Self-organizing maps and their applications in image processing, information organization, and retrieval, in: K. E. Barner and G. R. Arce (eds.), Nonlinear Signal and Image Processing: Theory, Methods, and Applications (CRC Press, Boca Raton, FL., 2004) 387-444.

[20] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression (MA: Kluwer Academic Publishers, Boston, 1992).

[21] M. M. van Hulle, Faithful Representations and Topographic Maps. From Distortion- to Information-Based Self-Organization (John Wiley & Sons, N. Y., 2000).

[22] J. He, A. H. Tan, C. L. Tan, and S. Y. Sung, On quantitative evaluation of clustering systems in: W. Wu, H. Xiong, and S. Shekhar (eds.), Clustering and Information Retrieval (Kluwer Academic Publishers, Norwell, MA., 2003) 105-133.

[23] A. Georgakis, C. Kotropoulos, A. Xafopoulos, and I.Pitas, Marginal median SOM for document organization and retrieval, Neural Networks, Vol. 17 (3) (April 2004) 365-377.

[24] M. Oja, G. Sperber, J. Blomberg, and S. Kaski, Grouping and visualizing human endogenous retroviruses by bootstrapping median self-organizing maps, in: Proc. 2004 IEEE Symp. Computational Intelligence in Bioinformatics and Computational Biology (2004) 95-101.

[25] I. Pitas and P. Tsakalides, Multivariate ordering in color image restoration, IEEE Trans. Circuits and Systems for Video Technology, Vol. 1 (3) (September 1991) 247-259.

[26] J. Astola, P. Haavisto, and Y. Neuvo, Vector median filters, Proc. IEEE, Vol. 78 (4) (April 1990) 678-689.

[27] W. Xu, X. Liu, and Y. Gong, Document clustering based on non-negative matrix factorization, in: Proc. ACM SIGIR (Toronto, Canada, 2003) 267-273.

[28] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data (Prentice-Hall, Englewood Cliffs, N. J., 1988).

[29] R. A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen., Vol. 7 (1936) 179-188.

[30] K. V. Mardia, J. T. Kent, and J. M. Bibby, Multivariate Analysis (Academic Press, Harcourt Brace & Co., N. Y. , 1979).

[31] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification and Scene Analysis (John Wiley & Sons, N. Y. , 1973).

[32] D. Ververidis, C. Kotropoulos, and I. Pitas, Automatic emotional speech classification, in: Proc. 2004 IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Vol. 1 (May 2004) 593-596.