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 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 $\Gamma$ 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 neuron 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 $\Gamma$ 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 the 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 standard SOM and the batch training algorithm, as
well as the 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 $\Gamma$ 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 SOM 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 the input patterns. Only one neuron wins at each iteration and becomes the winner of 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 $x_j$ the $j$th $D$-dimensional input pattern and by $w_i$ the $i$th $D$-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 Jacobi 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 $w_i$ 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 $w_j$. In the winner selection step, each input pattern $x_j$ is assigned to a Voronoi cell based on the nearest neighbor condition. That is, the BMU index, $c(j)$, of the input pattern $x_j$ is defined by

$$ c(j) = \arg \min_i \{ \| x_j - w_i \| \} $$  \hspace{1cm} (1)

where $\| \cdot \|$ denotes the Euclidean distance. Accordingly, the SOM can be treated as a vector quantization method [20]. Due to the fact that the input patterns $x_j$ are random vectors, the quantization error $\| x_j - w_i \|$ 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 $P(x)$. When the number of neurons is large, $P(x)$ 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. The Gaussian neighborhood around the winner neuron $c(j)$ is defined as [17]

$$ h_{ic}(t) = \exp \left( -\frac{d^2_{ic}(t)}{2\sigma^2} \right) $$  \hspace{1cm} (2)

where $d_{ic}(t) = \| r_{ic}(j) - r_i \|$ is the distance between map units $c(j)$ and $i$ on the map grid and $\sigma$, 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 $\{ x_j \}$ 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 $i$th weight vector $w_i$ is computed as [1,17]

$$ w_i(t + 1) = \frac{\sum_{j=1}^{M} h_{ic}(j)(t) \ x_j}{\sum_{j=1}^{M} h_{ic}(j)(t)} $$  \hspace{1cm} (3)

where $M$ denotes the number of patterns $x_j$ that have been assigned to the $i$th neuron up to the $t$th iteration.
The training is performed in two phases, namely the rough training and the fine tuning phases. In the rough training phase, large initial neighborhood radius \( \sigma_0 \) 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 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, SOM equals to an online learning that is performed through time, it is usual for a SOM to have “dead” units. The “dead” units are neurons which subsequently fail to be associated with any of the input patterns. The “dead” neurons 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 random 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 \([24]\). Another recent application of the MMSOM is in grouping and visualization of human endogenous retroviruses \([25]\).

In subsections 2.2.1 and 2.2.2, \( R_i(t) \) denotes the input patterns assigned to the \( i \)th neuron until the \( t \)th iteration and \( x(t) \) denotes the input pattern assigned to the neuron at the \( t \)th iteration.

2.2.1. Marginal median SOM

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 \( x_1, x_2, \ldots, x_M \), where \( x_j = (x_{1j}, x_{2j}, \ldots, x_{Dj})^T \), 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 \( i \)th neuron weighted by a factor that dictates the neighborhood. The marginal median is defined by \([26]\)

\[
\text{marginal median } \{x_1, x_2, \ldots, x_M\} = \begin{cases} \frac{\sum_{v=0}^{q-1} x_{(v)} + \sum_{v=q+1}^{M-1} x_{(v+1)}}{2}, & \text{if } M = 2v+1 \\ \frac{\sum_{v=q}^{M} x_{(v+1)} - \sum_{v=0}^{q-1} x_{(v)}}{2}, & \text{if } M = 2v \end{cases} \tag{5}
\]

where \( M \) denotes the number of patterns assigned to the neuron. The \( i \)th neuron is updated by

\[
w_i(t + 1) = h_i(x(t)) \text{ marginal median } \{R_i(t) \cup x(t)\}. \tag{6}
\]

2.2.2. Vector median SOM

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 \( i \)th neuron, which is the closest one to all the current input patterns. The vector median of \( M \) input patterns \( x_1, x_2, \ldots, x_M \) is defined by \([27]\)

\[
\text{vector median } \{x_1, x_2, \ldots, x_N\} = x_l \tag{7}
\]

where \( l = \arg \min_k \sum_{j=1}^{N} |x_j - x_k| \).

The \( i \)th neuron is updated by

\[
w_i(t + 1) = h_{i,c}(x(t)) \text{ vector median } \{R_i(t) \cup x(t)\}. \tag{8}
\]

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 \( I \) 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:

\[
\text{average over all neurons mean squared error } = \frac{1}{N} \sum_{i=1}^{N} (x_i - w_i)^2.
\]

\[
\text{Rand index } = \frac{\text{oc}(R_i) \text{ oc}(x(t))}{\text{oc}(R_i) + \text{oc}(x(t)) - \text{oc}(R_i \cup x(t))}.
\]

\[
\text{\( I \) statistic } = \frac{\text{oc}(R_i \cap x(t))}{\text{oc}(R_i) + \text{oc}(x(t)) - \text{oc}(R_i \cup x(t))}.
\]

\[
\text{overall entropy } = \sum_{i=1}^{N_f} \sum_{j=1}^{N_c} p_{ij} \log p_{ij}.
\]
Let $T$ be the total number of patterns that compose the test set and $g(x, y)$ 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 \cite{28}:

$$AC = \frac{1}{T} \sum_{j=1}^{T} \delta(g(x_j), \text{map}(\phi(x_j)))$$  \hspace{1cm} (13)$$

where $g(x_j)$ is the true label of pattern $x_j$, $\phi(x_j)$ is the label assigned to $x_j$ by the SOM or its variants, and $\text{map}(v_i)$ is the \textit{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 \cite{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 $G = (V \cup U, V \times U)$. The weight of the edge $(v_i, u_j)$ is denoted by $\xi(v_i, u_j)$. 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 $\Xi$, which represents the graph $G$, a permutation $\pi$ of $1, 2, \ldots, N_c$ must be found so that the following sum

$$\sum_{i=1}^{N_c} \xi(v_i, u_{\pi(i)})$$  \hspace{1cm} (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 $\Xi$.

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 $\xi(v_i, u_j)$ assigned to the edge $(v_i, u_j)$ corresponds to the profit made out, if the label assigned by the SOM is $v_i$ and the ground truth class label is $u_j$. 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 a $N_c \times N_c$ weight matrix $\Xi$. 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 (section 5), while for the VOWEL data the labels are integers between 1-15. The weight matrix $\Xi$ 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>
<thead>
<tr>
<th>Labels</th>
<th>$u_1$</th>
<th>$u_2$</th>
<th>$u_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>$v_2$</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>$v_3$</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

The output of the algorithm is the optimal matching, represented by a $N_c \times N_c$ 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>
<thead>
<tr>
<th>Labels</th>
<th>$u_1$</th>
<th>$u_2$</th>
<th>$u_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$v_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$v_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

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 $w_i$ is calculated as follows:

$$\text{MSE}_i = \frac{1}{M} \sum_{j=1}^{M} \|x_{ji} - w_i\|^2$$  \hspace{1cm} (15)$$

where $M$ is the total number of patterns assigned to the $i$th neuron and $x_{ji} \in R(t)$ 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 $\text{MSE}_i$ for all the neurons of the map

$$\text{AMSE} = \frac{1}{K} \sum_{i=1}^{K} \text{MSE}_i$$  \hspace{1cm} (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 [29]. In order to validate the clustering structure $C$ derived by the SOMs, a partition $P$ 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 $P$) but are not grouped into the same cluster (according to $C$), or that are not from the same class but are grouped into the same cluster. The Rand index is defined as follows [29]:

$$
\gamma = 1 + \frac{1}{M_e} \sum_{i=1}^{N_c} \sum_{j=1}^{N_p} n_{ij}^2 - \frac{1}{2} \sum_{i=1}^{N_c} n_i^2 - \frac{1}{2} \sum_{j=1}^{N_p} n_j^2
$$

(17)

The Rand index admits values in the range $[0,1]$. High value of the Rand index implies close agreement between $C$ and $P$. A perfect clustering (i.e. $\gamma = 1$) may not be achievable, when $C$ and $P$ have different number of clusters/classes [29].

3.4. $\Gamma$ statistic

The $\Gamma$ statistic is a special case of Hubert’s $\Gamma$ statistic [29]. It follows the idea of partitional structure validity. That is, the ground truth of the data must be available to be compared with the clustering $C$ derived by the SOM or its variants. Let us define the following auxiliary variables:

- $a = \frac{1}{2} \left( \sum_{i=1}^{N_c} \sum_{j=1}^{N_p} n_{ij}^2 - \left( \frac{N_c}{2} \right) \right)$;
- $b = \frac{1}{2} \left( \sum_{i=1}^{N_c} n_i^2 - \frac{1}{2} \sum_{i=1}^{N_c} \sum_{j=1}^{N_p} n_{ij}^2 \right)$;
- $c = \frac{1}{2} \left( \sum_{j=1}^{N_p} n_j^2 - \frac{1}{2} \sum_{i=1}^{N_c} \sum_{j=1}^{N_p} n_{ij}^2 \right)$;

$$
- m_1 = a + b;
- m_2 = a + c;
$$

The $\Gamma$ statistic calculates the correlation between the two partitions. It is defined as follows [29]:

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

(18)

Since the $\Gamma$ statistic is a correlation coefficient, its value ranges between -1 to 1. A high value of the $\Gamma$ statistic implies high correlation between $C$ and $P$ 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 $C$ can be evaluated according to the ground truth labels of patterns $P$. For each cluster, $c_i$, the cluster entropy $E_{c_i}$ is computed by [22]

$$
E_{c_i} = - \sum_{j=1}^{N_j} \frac{n_{ij}}{n_i} \log \frac{n_{ij}}{n_i}
$$

(19)

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

$$
E_c = \frac{1}{N} \sum_{i=1}^{N_c} n_i \cdot E_{c_i}
$$

(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 cluster 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]

$$
E_l = \frac{1}{N} \sum_{j=1}^{N_l} n_j \cdot E_{l_j}
$$

(21)

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

$$
E_{l_j} = - \sum_{i=1}^{N_c} \frac{n_{ij}}{n_j} \log \frac{n_{ij}}{n_j}
$$

(22)

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

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

(23)

with $\beta \in [0,1]$ functioning as a weight parameter that balances cluster and class entropies. In our experiments, $\beta$ 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 [30]. 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 [31, p.346]. Furthermore, the IRIS data set is widely used in clustering applications reported to bibliography [32].

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 fifteen 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 IRIS and VOWEL, 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 [33]. 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.

Cross validations. The best performance concerning all evaluation measures is indicated in boldface.

Three experiments were carried out. In the first experiment we worked on the IRIS data. The training set consists of 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, and the OE should decrease. A high quality clustering, as the number of the map neurons increases, the accuracy and the Rand index 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, and the OE of the SOM, the MMSOM, and the VMSOM for different map sizes on the IRIS data. The same ordering between the three 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.

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 for the VMSOM and the SOM are 0.0567 and 0.0765, respectively. The best ordering between the three 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.

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 batch 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 5
Accuracy, AMSE, Rand index, $\Gamma$ statistic, and OE of the VMSOM for different map sizes on the IRIS data.

<table>
<thead>
<tr>
<th>Neurons</th>
<th>VMSOM</th>
<th>AC(%)</th>
<th>AMSE</th>
<th>Rand</th>
<th>$\Gamma$</th>
<th>OE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (2 x 2)</td>
<td>89.88</td>
<td>0.53</td>
<td>0.9932</td>
<td>0.7455</td>
<td>0.1152</td>
<td></td>
</tr>
<tr>
<td>4 (2 x 2)</td>
<td>88.66</td>
<td>0.56</td>
<td>0.9921</td>
<td>0.7234</td>
<td>0.1140</td>
<td></td>
</tr>
<tr>
<td>5 (3 x 2)</td>
<td>95.55</td>
<td>0.41</td>
<td>0.9965</td>
<td>0.8749</td>
<td>0.0681</td>
<td></td>
</tr>
<tr>
<td>6 (3 x 2)</td>
<td>94.55</td>
<td>0.42</td>
<td>0.9957</td>
<td>0.8432</td>
<td>0.0647</td>
<td></td>
</tr>
<tr>
<td>7 (4 x 2)</td>
<td>93.88</td>
<td>0.35</td>
<td>0.9987</td>
<td>0.8645</td>
<td>0.0594</td>
<td></td>
</tr>
<tr>
<td>8 (4 x 2)</td>
<td>96.44</td>
<td>0.28</td>
<td>0.9972</td>
<td>0.8971</td>
<td>0.0584</td>
<td></td>
</tr>
<tr>
<td>9 (3 x 3)</td>
<td>96.33</td>
<td>0.31</td>
<td>0.9971</td>
<td>0.8931</td>
<td>0.0534</td>
<td></td>
</tr>
<tr>
<td>10 (3 x 3)</td>
<td>95.88</td>
<td>0.26</td>
<td>0.9966</td>
<td>0.8841</td>
<td>0.0585</td>
<td></td>
</tr>
<tr>
<td>11 (4 x 3)</td>
<td>96.22</td>
<td>0.25</td>
<td>0.9971</td>
<td>0.8959</td>
<td>0.0598</td>
<td></td>
</tr>
<tr>
<td>12 (4 x 3)</td>
<td>97.22</td>
<td>0.23</td>
<td>0.9977</td>
<td>0.9177</td>
<td>0.0567</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Neurons</th>
<th>SOM</th>
<th>AC(%)</th>
<th>AMSE</th>
<th>Rand</th>
<th>$\Gamma$</th>
<th>OE</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>44.04</td>
<td>1.49</td>
<td>0.9078</td>
<td>0.2228</td>
<td>0.5893</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>46.75</td>
<td>1.45</td>
<td>0.9048</td>
<td>0.2318</td>
<td>0.6067</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>49.20</td>
<td>1.422</td>
<td>0.9066</td>
<td>0.2536</td>
<td>0.5841</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>51.16</td>
<td>1.35</td>
<td>0.9102</td>
<td>0.2690</td>
<td>0.5666</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>53.75</td>
<td>1.31</td>
<td>0.9121</td>
<td>0.2916</td>
<td>0.5506</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>56.38</td>
<td>1.25</td>
<td>0.9102</td>
<td>0.2769</td>
<td>0.5573</td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>58.87</td>
<td>1.22</td>
<td>0.9144</td>
<td>0.3131</td>
<td>0.5247</td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>57.54</td>
<td>1.22</td>
<td>0.9175</td>
<td>0.3314</td>
<td>0.5133</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Neurons</th>
<th>MMSOM</th>
<th>AC(%)</th>
<th>AMSE</th>
<th>Rand</th>
<th>$\Gamma$</th>
<th>OE</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>53.54</td>
<td>1.02</td>
<td>0.9113</td>
<td>0.2920</td>
<td>0.5455</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>55.12</td>
<td>1.01</td>
<td>0.9122</td>
<td>0.3039</td>
<td>0.5274</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>56.33</td>
<td>0.99</td>
<td>0.9135</td>
<td>0.3183</td>
<td>0.5188</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>57.37</td>
<td>0.98</td>
<td>0.9161</td>
<td>0.3340</td>
<td>0.5046</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>59.62</td>
<td>0.95</td>
<td>0.9198</td>
<td>0.3596</td>
<td>0.4786</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>58.20</td>
<td>0.95</td>
<td>0.9168</td>
<td>0.3330</td>
<td>0.5044</td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>58.87</td>
<td>0.95</td>
<td>0.9172</td>
<td>0.3435</td>
<td>0.4918</td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>61.45</td>
<td>0.94</td>
<td>0.9215</td>
<td>0.3679</td>
<td>0.4761</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Neurons</th>
<th>VMSOM</th>
<th>AC(%)</th>
<th>AMSE</th>
<th>Rand</th>
<th>$\Gamma$</th>
<th>OE</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>58.62</td>
<td>1.12</td>
<td>0.9169</td>
<td>0.3424</td>
<td>0.5019</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>59.83</td>
<td>1.07</td>
<td>0.9182</td>
<td>0.3558</td>
<td>0.4887</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>61.95</td>
<td>1.03</td>
<td>0.9213</td>
<td>0.3765</td>
<td>0.4715</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>62.95</td>
<td>1.02</td>
<td>0.9227</td>
<td>0.3896</td>
<td>0.4556</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>64.79</td>
<td>0.97</td>
<td>0.9272</td>
<td>0.4185</td>
<td>0.4375</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>66.20</td>
<td>0.96</td>
<td>0.9290</td>
<td>0.4294</td>
<td>0.4286</td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>66.45</td>
<td>0.96</td>
<td>0.9290</td>
<td>0.4319</td>
<td>0.4239</td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>66.83</td>
<td>0.93</td>
<td>0.9298</td>
<td>0.4383</td>
<td>0.4150</td>
<td></td>
</tr>
</tbody>
</table>

As it can be noticed from 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 $\mu$, $\sigma$, 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 the SOM variants. For both the MMSOM and the VMSOM, the number of “bad” neurons is smaller for all map sizes, which explains their superior objective figures of merit than those of SOM.

Table 9
Number of “bad” neurons for different SOM sizes for the IRIS data.

<table>
<thead>
<tr>
<th>Neurons</th>
<th>SOM</th>
<th>MMSOM</th>
<th>VMSOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (2 x 2)</td>
<td>22</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4 (2 x 2)</td>
<td>22</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5 (3 x 2)</td>
<td>43</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6 (3 x 2)</td>
<td>43</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7 (4 x 2)</td>
<td>42</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8 (4 x 2)</td>
<td>72</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9 (3 x 3)</td>
<td>72</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10 (3 x 3)</td>
<td>72</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11 (4 x 3)</td>
<td>75</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>12 (4 x 3)</td>
<td>85</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16 (4 x 4)</td>
<td>84</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 10
Number of “bad” neurons for different SOM sizes for the VOWEL data.

<table>
<thead>
<tr>
<th>Neurons</th>
<th>SOM</th>
<th>MMSOM</th>
<th>VMSOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (2 x 2)</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4 (2 x 2)</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5 (3 x 2)</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>6 (3 x 2)</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>7 (4 x 2)</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8 (4 x 2)</td>
<td>7</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>9 (3 x 3)</td>
<td>7</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>10 (3 x 3)</td>
<td>7</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>11 (4 x 3)</td>
<td>7</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>12 (4 x 3)</td>
<td>8</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>16 (4 x 4)</td>
<td>8</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

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

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 $\Gamma$ statistic and the overall entropy as figures of merit. The assessment was first conducted on the well-known IRIS and the 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 research was co-funded by the European Union and the Hellenic Ministry of Education in the framework of the program Pythagoras II within the Operational Program for Education and Initial Vocational Training of the 3rd Community Support Program.

References