

On the Variants of the Self-Organizing Map That Are Based on Order Statistics

Vassiliki Moschou, Dimitrios Ververidis, and Constantine Kotropoulos*

Department of Informatics, Aristotle University of Thessaloniki
Box 451, Thessaloniki 54124, Greece
{vmoshou, jimver, costas}@aiaa.csd.auth.gr

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. In this paper, we employ the well-known IRIS data set and we assess their 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, and the Rand index. 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 re-distribute emotional speech patterns from the Danish Emotional Speech database that were originally classified as being neutral to four emotional states such as hot anger, happiness, sadness, and surprise.

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 an input data space onto a two or three 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 in [4, 5].

We are interested in the class of SOM training algorithms that employ multivariate order statistics, such as the marginal median and the vector median [8]. 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 the paper is in the assessment of SOM training algorithms in clustering with respect to the accuracy, the average over all neurons mean squared error, and the Rand index. The superiority of the studied SOM variants against the SOM is demonstrated by experiments carried out using the well-known IRIS data. We also compare the studied SOM variants with the SOM in the re-distribution of emotional speech patterns from the Danish Emotional Speech (DES) database [15], that were originally classified as being

* This work has been supported by the project PYTHAGORAS II "Efficient techniques for information organization, browsing, and retrieval in multimedia", funded by the Greek Ministry of Education and the European Union.

neutral, into four emotional states such as hot anger, happiness, sadness, and surprise. The latter experiment is motivated by the following fact. There are emotional facial expression databases such as the Action-Unit coded Cohn-Kanade database [17] where the neutral emotional class is not represented adequately. Accordingly, facial expression feature vectors are not clustered to the neutral emotional class [18]. For the emotional speech databases, the utterances are regularly classified as neutral. Accordingly, when the neutral class is not represented in one modality it is difficult to develop multimodal emotion recognition algorithms (e.g. feature fusion algorithms). Frequently, the ground truth information related to emotions that is provided by the human evaluators is biased towards the neutral class. Therefore, the patterns classified as neutral might be needed to be re-distributed 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, and the Rand index. This section also describes the Kuhn-Munkres algorithm [13] 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 data using the SOM, the MMSOM, and the VMSOM are demonstrated. Furthermore, figures of merit are presented and discussed for the re-distribution of neutral 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 two or three dimensional lattice of nodes (the map). Each node is associated with a weight vector $\mathbf{w} = (w_1, w_2, \dots, w_D)^T$ in the input space. The SOM is trained iteratively and learns the input patterns. The task of the self-organizing (unsupervised) learning lies to revealing the statistical properties of the input patterns, creating suitable representations for the features (i.e. weight vectors), and automatically creating new clusters. The map 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 or the *best matching unit* (BMU) [7].

Let us denote by \mathbf{x}_j the j th D -dimensional input feature vector and by \mathbf{w}_i the i th D -dimensional weight vector. The first step of the algorithm is the weight vector initialization performed using the linear initialization algorithm. The weight vectors \mathbf{w}_i define the *Voronoi tessellation* of the input space [1, 2]. Each Voronoi cell is represented by its centroid that corresponds to the weight vector \mathbf{w}_i . 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

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

where $\|\cdot\|$ denotes the Euclidean distance. Accordingly, the SOM can be treated as a vector quantization method [6]. The most important step of the SOM algorithm is the adaptation of the neuron weight vectors. The neurons are connected to adjacent neurons by a neighborhood function dictating the structure of the map (topology). It determines how strongly the neurons are connected to each other [7]. In each training step, the neurons update 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 [20]. The larger the neighborhood, the more rigid the SOM. In our experiments, the neighborhood function used is the Gaussian. 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. In this paper, we are interested in the latter. In the batch training algorithm, for a fixed training set $\{\mathbf{x}_j\}$, we keep record of the weight updates, but their adjustment is applied only after all samples of the set have been considered. The learning stops when a pre-determined number of iterations is reached [20]. At each training iteration, the BMU is determined. Afterwards, all the neurons that belong to the BMU's neighborhood are updated. The updating rule of the i th weight vector, \mathbf{w}_i , is computed as [7]

$$\mathbf{w}_i(t+1) = \frac{\sum_{j=1}^N \alpha(t) h_{ic(j)}(t) \mathbf{x}_j}{\sum_{j=1}^N h_{ic(j)}(t)} \quad (2)$$

where N defines the number of patterns \mathbf{x}_j that have been assigned to the i th neuron up to the t th iteration and $h_{ic(j)}(t)$ denotes the neighborhood function around the BMU $c(j)$. The learning rate $a(t)$ is a decreasing function of time.

During training, the neighborhood function shrinks through time [2, 7]. At the first training steps, large initial learning rates and neighborhood radii are used in order to have a rigid SOM, whereas small initial learning rates and radii are used during the following training steps. Concerning the neighborhood, as its range is decreased, so does the number of neurons whose weight update direction is correlated. As a result of this correlation, neighboring neurons will be specialized for similar input patterns [20]. The topological information of the map ensures that neighboring neurons on the map possess similar attributes. It must be mentioned that, due to the neighborhood shrinking and the decreasing learning rate 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 vectors, and, thus, are never organized by the input data. The “dead” neurons have zero (or very low) probability to be active [20].

2.2 SOM Variants Based on Order Statistics

The standard SOM has some disadvantages, such as lack of robustness against outliers and against erroneous choices for the winner vector due to the linear estimators [8]. In order to face these problems, the variants of the standard SOM that employ multivariate order statistics can be used. The MMSOM and the VMSOM treat efficiently the outliers, because they inherit the robustness properties of the order statistics [9].

The SOM variants under discussion 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 [8, 9]. In contrast, the SOM calculates the weighted

mean value of the input patterns, as can be seen in (2). Although that the MMSOM and the VMSOM, used in this paper, update only the BMU, while the SOM updates also the BMU's neighboring neurons, in principal, such update is not prohibited for MMSOM and VMSOM as well. The MMSOM and VMSOM updating rules, discussed here, can be seen as special cases of vector quantizers that employ a generalized centroid condition [6].

In subsections 2.1 and 2.2, $R_c(t)$ denotes the input patterns assigned to the BMU until the t th iteration and $\mathbf{x}(t)$ denotes the input pattern assigned to the BMU in the t th iteration.

Marginal Median SOM. The MMSOM calculates the marginal median of all patterns assigned to the winner neuron and updates only the BMU's weight vector. The MMSOM relies on the concept of marginal ordering. The marginal ordering of N input vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$, where $\mathbf{x}_j = (x_{1j}, x_{2j}, \dots, x_{Dj})^T$, is performed by ordering the winner neuron's vector components independently along each of the D dimensions [8, 9]:

$$x_{q(1)} \leq x_{q(2)} \leq \dots \leq x_{q(N)}, \quad q = 1, 2, \dots, D \quad (3)$$

where q denotes the index of the vector component into consideration. The new weight vector of the BMU emerges from the calculation of the marginal median of all patterns indexed by the BMU. The calculation of the marginal median is defined by [11]

$$\begin{aligned} \text{marginal median } \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} &= \\ &= \begin{cases} (x_{1(v+1)}, \dots, x_{D(v+1)})^T, & N = 2v + 1 \\ \left(\frac{x_{1(v)} + x_{1(v+1)}}{2}, \dots, \frac{x_{D(v)} + x_{D(v+1)}}{2} \right)^T, & N = 2v \end{cases} \end{aligned} \quad (4)$$

where N denotes the number of patterns assigned to the BMU, \mathbf{w}_c . The winner neuron is updated by

$$\mathbf{w}_c(t+1) = \text{marginal median } \{R_c(t) \cup \mathbf{x}(t)\}. \quad (5)$$

Vector Median SOM. The VMSOM calculates the vector median of all patterns assigned to the winner neuron and updates only the BMU's weight vector. The vector median operator is the vector that belongs to the set of input vectors indexed by the BMU, which is the closest one to all the current input vectors. The vector median of N input vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ is defined by [10]

$$\begin{aligned} \text{vector median } \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} &= \\ &= \mathbf{x}_l, \text{ where } l = \arg \min_k \sum_{j=1}^N |\mathbf{x}_j - \mathbf{x}_k| \end{aligned} \quad (6)$$

The winner neuron is updated by

$$\mathbf{w}_c(t+1) = \text{vector median } \{R_c(t) \cup \mathbf{x}(t)\}. \quad (7)$$

3 Clustering Evaluation Measures

Three 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 (AMSE), and the Rand index.

3.1 Accuracy

Let M be the total number of patterns that compose the test set, \mathbf{x}_j be the j th pattern, 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 [12]

$$AC = \frac{1}{M} \sum_{j=1}^M \delta(g(\mathbf{x}_j), \text{map}(\phi(\mathbf{x}_j))) \tag{8}$$

where $g(\mathbf{x}_j)$ is the true label of the pattern, $\phi(\mathbf{x}_j)$ is the label assigned to the pattern by the SOM, and $\text{map}(v_i)$ is the *optimal matching*, which maps the label assigned to the pattern by the SOM or its variants onto the ground truth labels. The optimal matching is needed because SOM is an unsupervised training algorithm. It can be derived by the Kuhn-Munkres algorithm [13].

The problem solved by the Kuhn-Munkres algorithm is stated as follows. Consider a complete weighted bipartite graph $\Gamma = (V \cup U, V \times U)$. Let us denote $V = \{v_i\}$ and $U = \{u_i\}$, where $i = 1, 2, \dots, K$ and K being the number of nodes. 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 a $K \times K$ weight matrix Ξ , which represents the graph Γ , a permutation π of $1, 2, \dots, K$ must be found so that the following sum

$$\sum_{i=1}^K \xi(v_i, u_{\pi(i)}) \tag{9}$$

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

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 (8). Let us consider that the patterns must be clustered into K clusters. That is the number of nodes of the graph Γ equals K . 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 label is u_i . The purpose is to maximize the profit. Obviously, if the two labels are the same, the profit is maximized, since the SOM has assigned the patterns to the correct ground truth class.

The input of the algorithm is a $K \times K$ weight matrix. The weights of the elements (i, i) with $i = 1, 2, \dots, K$ are set to be $\xi(i, i) = 1$, while the weights of the elements (i, j) with $j = 1, 2, \dots, K$ and $i \neq j$ are set to be 0 (or a very low value). The output of the algorithm is a $K \times K$ matrix. The (i, j) matrix element equals 1 if the edge (v_i, u_j) belongs to the optimal matching, otherwise it equals 0.

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

$$MSE_i = \frac{1}{N} \sum_{j=1}^N \|\mathbf{x}_{j[i]} - \mathbf{w}_i\|^2 \quad (10)$$

where N is the total number of patterns assigned to the i th neuron and $\mathbf{x}_{j[i]}$ 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 mean value of MSE_i for all the neurons of the map:

$$AMSE = \frac{1}{K} \sum_{i=1}^K MSE_i \quad (11)$$

where K is the total number of the map neurons.

3.3 Rand Index

The Rand index is a widely used evaluation measure in clustering applications. The Rand index indicates the number of input patterns that are either from the same class but are not grouped into the same cluster, or that are not from the same class but are grouped into the same cluster. The Rand index is defined as follows [3, p. 173-174]:

$$\gamma = \frac{1}{2} \sum_{i=1}^{N_c} n_{i.}^2 + \frac{1}{2} \sum_{j=1}^{N_f} n_{.j}^2 - \sum_{i=1}^{N_c} \sum_{j=1}^{N_f} n_{ij}^2 \quad (12)$$

where N_c denotes the total number of clusters that are created after training the SOMs, N_f the total number of classes that the patterns are initially grouped into according to the ground truth, $n_{i.}$ and $n_{.j}$ the total number of patterns assigned to clusters i and j , respectively, and n_{ij} the total number of patterns assigned to cluster i that belong to class j . Rand index values lie in the range $0 \leq \gamma \leq \binom{N}{2}$, with $\binom{N}{2}$ denoting the number of combinations of two patterns that can be taken out from the total set. The lower the Rand index, the better the clustering is. A perfect clustering should produce a zero Rand index [3].

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 [14]. 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. Accordingly, this data set is appropriate for studying the role of the outliers in clustering. This is not the case for supervised learning [19, p.346].

Motivated by the observations made on IRIS, we have compared the SOM variants against the SOM for the redistribution of neutral emotional speech feature vectors from the DES database [15] into non-neutral emotional speech patterns. 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 [16]. 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.

5 Experimental Results

The performance of the SOM, the MMSOM, and the VMSOM on clustering are demonstrated through the accuracy, the AMSE, and the Rand index on the IRIS data. The training set consists of 120 randomly selected patterns, while the test set is composed by the 30 remaining patterns. The accuracy, the AMSE, and the Rand index were measured using 30-fold cross validation. The accuracy should increase, while the AMSE and the Rand index should decrease for a high quality clustering.

Table 1 summarizes the accuracy, the AMSE, and the Rand index of the three SOMs using different number of neurons to build the map, respectively, averaged over the 30 cross validations. The best performance concerning the accuracy, the AMSE, and the Rand index is indicated in boldface. As it can be noticed from Table 1, the MMSOM yields the best accuracy (97.33%), the VMSOM follows (97.00%), while the SOM has the worst behavior with respect to the accuracy (91.22%). Table 1 indicates that the same ordering between the three SOMs stands also with respect to the AMSE. The smallest AMSE is measured for the MMSOM (0.221). The VMSOM yields a larger AMSE than the MMSOM (0.238) and, finally, the SOM exhibits the worst performance with respect to the AMSE (0.441). In addition, the SOM yields the worst Rand index for every map size compared to both the MMSOM and the VMSOM. The best Rand index values are 33.866 for a 4 × 4 map, 13.233 for a 3 × 3 map, and 15.266 for a 3 × 3 map, for the SOM, the MMSOM, and the VMSOM, respectively.

As it can be noticed from Table 1 both the MMSOM and the VMSOM have similar values that do not change significantly with the number of neurons, concerning all the evaluation measures. In contrast, the SOM values change significantly with the map size

Table 1. Accuracy, AMSE, and Rand index of SOM, MMSOM, and VMSOM averaged over 30 cross validations for different map sizes on the IRIS data

Neurons	Average accuracy			Average AMSE			Average Rand index		
	SOM	MMSOM	VMSOM	SOM	MMSOM	VMSOM	SOM	MMSOM	VMSOM
3 (2 × 2)	60.66	89.00	89.67	1.599	0.501	0.557	76.633	51.533	52.533
4 (2 × 2)	82.45	90.67	88.89	1.788	0.516	0.547	98.200	49.800	58.166
5 (3 × 2)	90.56	97.33	95.22	1.592	0.337	0.367	56.133	22.166	25.966
6 (3 × 2)	91.22	96.56	95.11	1.229	0.338	0.371	45.600	23.266	23.366
7 (4 × 2)	90.78	95.56	94.67	0.658	0.321	0.339	48.766	21.266	23.866
8 (4 × 2)	82.89	96.67	94.11	1.189	0.250	0.184	78.966	17.566	16.366
9 (3 × 3)	84.56	97.00	96.67	1.187	0.249	0.298	71.633	17.966	15.266
10 (3 × 3)	84.56	97.11	96.33	1.175	0.266	0.337	68.533	13.233	23.833
11 (4 × 3)	88.78	96.11	94.89	0.496	0.227	0.280	46.266	19.300	18.033
12 (4 × 3)	90.67	97.00	94.33	0.517	0.233	0.272	41.200	18.466	22.233
16 (4 × 4)	91.22	97.33	97.00	0.441	0.221	0.238	33.866	14.933	20.100

compared to the MMSOM and the VMSOM. This fact can be explained by the number of “dead” neurons of each SOM. Let us denote by μ the mean number of patterns that a neuron wins, by σ the standard deviation, and by N the exact number of patterns a neuron wins during training. The “dead” neurons are those for which the following inequality holds: $N < \mu - \sigma$. Table 2 presents the number of “dead” neurons of each SOM for different map sizes. It is obvious that for the SOM, the number of “dead” neurons gets very large with increasing number of neurons, causing the significant difference of its performance compared to the SOM variants for different map sizes. For both the MMSOM and the VMSOM, the number of “dead” neurons is small for all map sizes, which explains their similar behavior and the small range of values they get.

Table 2. Number of “dead” neurons for different map sizes for the SOMs

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

The Student t -test for unequal variances 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 and the Rand index. The tests have shown that the performance differences are statistically significant.

The superiority of the MMSOM was expected, because the marginal median is not affected by the outliers in contrast to the mean. Moreover, the weight vector is not constrained to be among the input vectors assigned to a neuron as the vector median does. Furthermore, the SOM contains many “dead” units and cannot represent data well. Due to the order statistic properties of the MMSOM and VMSOM, it is expected, though, that the maps created by the SOM variants are more representative, as demonstrated by Table 2. The maps created by the MMSOM and the VMSOM have less “dead” units and the classes defined on the map are well separated. However, “dead” units are inevitable for a SOM to train a non-stationary data set [20].

The SOMs were also applied to the re-distribution of emotional speech feature vectors extracted from the DES database. The primitive emotional states are anger, happiness, neutral, sadness, and surprise. Our purpose is to re-distribute 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 the 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 3 demonstrates the average assignment ratio of the neutral patterns that are labeled as angry, happy, sad, and surprised by each SOM. As can be seen, all the algorithms classify the neutral patterns as sad with a very high percentage. This means that sadness resembles the neutral state more than the other emotional states. The largest percentage is measured for the MMSOM (61.86%), the next larger percentage is provided by

the VMSOM (61.51%) and, finally, the SOM yields the lowest one (58.27%). It was expected that the MMSOM would re-distribute the neutral patterns in a better manner than VMSOM and SOM. Anger is the second closer to neutrality emotion, happiness follows and, finally, surprise is the least similar to neutrality emotion. All the algorithms conform to this order.

Table 3. 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 for a 15-fold cross validation experiment.

Figure 1 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 in order to reduce the dimensionality from five dimensions (5D) to two dimensions (2D) [16]. 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 an overlap is observed for sadness, anger, and happiness. Therefore, the results shown in Table 3 comply with the sample space depicted in Figure 1.

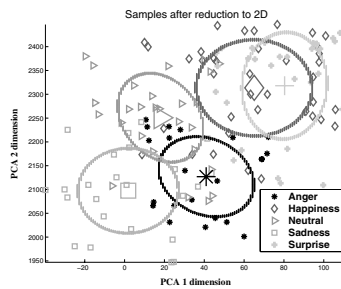


Fig. 1. Partition of the 2D domain into five emotional states derived by PCA. The samples which belong to the interquartile range of each pdf are shown. The big symbols denote the mean of each class. The ellipses denote the 60% likelihood contours for a 2-D Gauss model.

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 and document organization and retrieval. In this paper, we presented experimental evidence for their clustering quality by using the accuracy, the average over all neurons mean squared error, and the Rand index as figures of merit. The assessment was first conducted on the well-known IRIS data set. Motivated by the superiority of the SOM variants that are based on order statistics, we investigated their application in the re-distribution of emotional neutral feature vectors to non-neutral emotional states. We demonstrated that the re-distribution is consistent with the sample feature space.

References

1. S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, N.Y.: Prentice-Hall, 1999.
2. T. Kohonen, *Self-Organizing Maps*, 3/e. Berlin, Germany: Springer-Verlag, 2000.
3. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, N.J.: Prentice-Hall, 1988.
4. S. Kaski, J. Kangas, and T. Kohonen, "Bibliography of Self-Organizing Map (SOM) Papers: 1981-1997," *Neural Computing Surveys*, vol. 1, pp. 102-350, 1998.
5. M. Oja, S. Kaski, and T. Kohonen, "Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum," *Neural Computing Surveys*, vol. 3, pp. 1-156, 2003.
6. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer Academic Publishers, 1992.
7. J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, *SOM Toolbox for Matlab 5*, Finland, 2000, www.cis.hut.fi.
8. I. Pitas, C. Kotropoulos, N. Nikolaidis, R. Yang, and M. Gabbouj, "Order statistics learning vector quantizer", *IEEE Trans. Image Processing*, vol. 5, pp. 1048-1053, 1996.
9. C. Kotropoulos and I. Pitas, "Self-organizing maps and their applications in image processing, information organization, and retrieval," in *Nonlinear Signal and Image Processing: Theory, Methods, and Applications* (K. E. Barner and G. R. Arce, Eds.), Boca Raton, FL: CRC Press, 2004.
10. J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proceedings of the IEEE*, vol. 78, no. 4, pp. 678-689, April 1990.
11. I. Pitas and P. Tsakalides, "Multivariate ordering in color image restoration," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 1, no. 3, pp. 247-259, September 1991.
12. W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. ACM SIGIR 03*, pp. 267-273, Toronto, Canada, 2003.
13. J. A. McHugh, *Algorithmic Graph Theory*. Prentice Hall, 1990.
14. R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugen.*, vol. 7, pp. 179-188, 1936.
15. I. S. Engberg and A. V. Hansen, Documentation of the Danish Emotional Speech Database DES, Internal Report, Center for Person Kommunikation, Aalborg University, 1996.
16. D. Ververidis, C. Kotropoulos, and I. Pitas, "Automatic emotional speech classification," in *Proc. 2004 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. I, pp. 593-596, Montreal, Canada, May 2004.
17. J. C. T. Kanade and Y. Tian, "Comprehensive database for facial expression analysis," in *Proc. IEEE Int. Conf. Face and Gesture Recognition*, pp. 46-53, March 2000.
18. 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 12-15, 2005.
19. K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*. Academic Press, Harcourt Brace & Co., New York, 1979.
20. M. M. Van Hulle, *Faithful Representations and Topographic Maps. From Distortion- to Information-Base Self-Organization*. N.Y.: J. Wiley, 2000.