

Fuzzy Classification Using Self-Organizing Map and Learning Vector Quantization

Ning Chen

Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal
Institute of Mechanics, Chinese Academy of Sciences, P. R. China
ningchen74@yahoo.com

Abstract. Fuzzy classification proposes an approach to solve uncertainty problem in classification tasks. It assigns an instance to more than one class with different degrees instead of a definite class by crisp classification. This paper studies the usage of fuzzy strategy in classification. Two fuzzy algorithms for sequential self-organizing map and learning vector quantization are proposed based on fuzzy projection and learning rules. The derived classifiers are able to provide fuzzy classes when classifying new data. Experiments show the effectiveness of proposed algorithms in terms of classification accuracy.

Keywords: fuzzy classification, self-organizing map (SOM), learning vector quantization (LVQ).

1 Introduction

Classification is a supervised machine learning method to derive models between features (independent variables) and class (target variable). In the past decades, classification has been widely applied to solve a great variety of classifying tasks, e.g., product marketing, medical diagnosis, credit approval, image segmentation, qualitative prediction, customer attrition causes analysis. The process of classification is to first produce models from training data in which each sample is assumed to have a predefined class and then use the models to classify new data in which the class label is unknown. Classification can be divided into crisp classification and fuzzy classification. Crisp classification produces a number of classical sets of data, in which an element is classified to only one class. However, there also exist uncertainty cases in which samples are not clear members of any class [4]. Fuzzy classification is advantageous over crisp classification on solving uncertainty problems by assigning a sample to multiple classes with different membership degrees. The membership degrees given by fuzzy classification provide some valuable information, e.g., significance of a sample belonging to a class.

Self-organizing map (SOM) and learning vector quantization (LVQ) are artificial neuron network algorithms. SOM is trained in an unsupervised way. It projects the data into neurons through a topology preserving transformation so that the neurons close to each other have similar features in the input space. Although SOM is an unsupervised learning method by nature, it can be also used for supervised tasks after the map is labeled. LVQ is performed in a supervised way which defines class regions in the data space rather than preserves topological property of data.

Fuzzy SOM and LVQ have been studied in literature. FLVQ [2] is a batch SOM algorithm combining online weight adaptation rule with fuzzy membership assignment. The relative membership is calculated directly from the distances between the input instance and map neurons rather than the topological neighbors. Replacing crisp class with fuzzy class membership for both input samples and map neurons, a fuzzy SOM classifier is presented in [6]. The crisp labels of input samples are fuzzified by a k-nearest neighbor rule. After training, each map neuron is assigned to a class with a membership degree based on the typicalness of patterns projected on it. In this method, the fuzzy paradigm is only used in the labeling phase and has no impact on map organization. Some competitive algorithms: FALVQ 1, FALVQ 2, and FALVQ 3 support fuzzy classification using different membership functions [3]. These algorithms optimize some fuzzy cost functions, formed as the weighted sum of squared Euclidean distances between input vectors and reference vectors of neurons. However, it is noted that the optimization procedures are plagued with local minima.

In this paper, we propose two sequential algorithms for fuzzy classification using SOM and LVQ. Sequential algorithms are 'on-line' in the sense that the neurons are updated after the presentation of each input. Discarding each sample once it has been used, sequential algorithms avoid the storage of complete data set. The fuzzy SOM and LVQ algorithms are based on fuzzy projection, in which the membership values are calculated from the distance between input samples and map neurons. As opposed to [6], the fuzzy paradigm is used in both model training and model classification. In training phase, the neurons are updated according to the membership values. In classifying phase, each instance is assigned to multiple classes with different degrees. Finally, a hybrid classifier is presented combining fuzzy SOM and LVQ to hold both topology preserving property and pattern recognition capability. The performance of proposed fuzzy algorithms is investigated in terms of classification accuracy. In the remaining of the paper, section 2 describes the methodology of fuzzy classification algorithms. Experiments and results are given in section 3. Lastly, section 4 concludes the paper.

2 Fuzzy Sequential SOM and LVQ

2.1 Fuzzy Projection

SOM is an artificial neural network (ANN) which attempts to represent the input data in low dimensional grid space through a topology preserving mapping [5]. The neurons are organized on a regular grid with usually one or two dimensions. Each neuron is associated with input samples by a reference vector and connected to adjacent neurons by a neighborhood function. Suppose x_i is the i^{th} input vector and m_j is the reference vector of the j^{th} neuron. Crisp SOM projects x_i to map unit c which best matches to it, i.e., $c = \arg \min_j d(m_j, x_i)$, where d is the Euclidean distance. Fuzzy SOM projects x_i to m_j with a membership degree ω_{ij} , satisfying $\omega_{ij} \in [0, 1]$ and $\sum_{j=1}^m \omega_{ij} = 1$. Given a fuzzy parameter $\alpha \geq 1$ which controls the degree of fuzziness, the membership matrix can be calculated from the distance between input vector and reference vectors [1].

$$\begin{aligned}
\text{If } \alpha = 1: \quad \omega_{ij} &= \begin{cases} 1 & d(x_i, m_j) \leq d(x_i, m_l), l \neq j \\ 0 & \text{otherwise} \end{cases} \\
\text{If } \alpha > 1: \quad \omega_{ij} &= \begin{cases} 1 & \text{if } x_i = m_j \\ 0 & \text{if } x_i = m_l, l \neq j \\ \frac{1}{\sum_{l=1}^m \left(\frac{d(x_i, m_j)}{d(x_i, m_l)} \right)^{\frac{1}{\alpha-1}}} & \text{otherwise} \end{cases} \quad (1)
\end{aligned}$$

2.2 Fuzzy Sequential SOM (FSSOM)

Fuzzy sequential SOM uses fuzzy match instead of crisp match and updates reference vectors according to membership degrees. The map is trained iteratively by updating reference vectors according to input samples. An input vector x_i is assigned to each neuron with a membership degree obtained by Equation 1. Then the units are updated towards the input case with a proportion of the distance between them. The incremental vector is the sum of the neighborhood function values weighted by the exponent on the membership degrees. After training, the neurons become topologically ordered on the map. Finally, the map neurons are labeled by specific classes according to classified samples. For the purpose of constructing a classifier, the fuzzy sequential SOM algorithm for model derivation is described as follows:

- Step 1: Initialize the map with a lattice of neurons and reference vectors.
- Step 2: Choose a sample x_i from the training data set at time t .
- Step 3: Calculate the distances between x_i and reference vectors.
- Step 4: Compute the membership degree of neurons with respect to x_i .
- Step 5: Update the reference vectors of all neurons using fuzzy update rule:

$$\begin{aligned}
m_p(t+1) &= m_p(t) + \Delta m_p(t) \\
&= m_p(t) + \gamma(t) \sum_{j=1}^m h_{jp} \omega_{ij}^{\alpha} (x_i - m_p(t)) \quad (2)
\end{aligned}$$

where $\gamma(t)$ is the learning rate at time t , and h is the neighborhood function of radius $\delta(t)$. Both $\delta(t)$ and $\gamma(t)$ are non-increasing functions of time.

- Step 6: Repeat from Step 2 to Step 5 enough iterations until the status of map is stable.
- Step 7: Input the samples of classified data and project them to best-matching units.
- Step 8: Label a neuron with the class of maximal frequency occurring in the projected samples.

Usually the neighborhood radius and learning rate are bigger values at first and decrease to zero with training steps [5]. Apparently, when $\alpha = 1$, it produces a hard projection that $\omega_{ic} = 1$ (c is the best-matching unit which has the minimal distance to the input) and $\omega_{ij} = 0$ ($j \neq c$). In such case, FSSOM is equivalent to classic sequential SOM.

2.3 Fuzzy Sequential LVQ (FSLVQ)

Learning vector quantizer (LVQ), a variant of SOM, uses a supervised approach during learning. The map units are assigned by class labels in the initialization and then updated

at each training step. The update way depends on the match of class labels between best-matching unit and input. If the unit has the same class to the input, the reference vector is moved close to the input, otherwise, it is moved away from the input. In contrast to crisp LVQ which updates only the best-matching unit, FSLVQ updates all units according to the memberships. As an extension of LVQ1 [5], one of basic LVQ algorithms, the fuzzy sequential LVQ algorithm is described as follows:

- Step 1: Initialize the reference vector and class label for each neuron.
- Step 2: Choose a sample x_i from the training data set at time t .
- Step 3: Calculate the distances between x_i and reference vectors.
- Step 4: Compute the membership degree of neurons with respect to x_i .
- Step 5: Update the reference vector of all neurons:

$$m_p(t+1) = \begin{cases} m_p(t) + \gamma(t)\omega_{ip}^\alpha(x_i - m_p(t)) & \text{if } \text{class}(m_p) = \text{class}(x_i) \\ m_p(t) - \gamma(t)\omega_{ip}^\alpha(x_i - m_p(t)) & \text{otherwise} \end{cases} \quad (3)$$

Step 6: Repeat from Step 2 to Step 5 enough iterations.

When $\alpha = 1$, only the best-matching unit is updated, so that FSLVQ is essentially equivalent to crisp LVQ1.

2.4 Fuzzy Classifying

Once a map is trained and labeled, it can be used as a classifier for unclassified data. Fuzzy classification offers more insight of class assignment to decision makers. After calculating the membership degrees of a sample with respect to all units, the degree of the sample to one class is calculated as the sum of membership degrees with respect to the units having the same class. When a crisp assignment is needed, the classification can be done according to the class with maximal degree. If there are more than one class having the maximal degree, the first one is chosen. When $\alpha = 1$, it yields to a crisp classification, that simply assigns the input to the class of best-matching unit. Suppose $\{c_1, c_2, \dots, c_k\}$ is the set of class labels, then the class of sample x_i is determined as follows:

$$P(x_i, c_r) = \sum_{j=1}^m (\omega_{ij} | \text{class}(m_j) = c_r)$$

$$\text{class}(x_i) = \arg \max_r P(x_i, c_r) \quad (4)$$

3 Experiments and Results

3.1 Hybrid Fuzzy Classifier

Although both SOM and LVQ can be used for classification, they are different in some aspects. Firstly, SOM attempts to approximately preserve the neighborhood relationship of data in a topological order fashion. LVQ tries to recognize the patterns of class with respect to other features; Secondly, SOM is trained from an initial map without class labels. LVQ needs the assignment of labels for neurons in the initialization; Next, SOM

is trained in an unsupervised way without the direction of class labels. The training process is extreme data driven based on intrinsic similarity of data. LVQ is trained in a supervised way under the direction of class information. In order to possess both topology preserving property and pattern recognition capability, the unsupervised and supervised scheme can be combined in either simultaneous manner, e.g. LVQ-SOM [5], HLVQ [7], or successive manner [9].

In the hybrid fuzzy classifier, the models are derived using FSSOM followed by FSLVQ. The combination of FSSOM and FSLVQ is inspired by three reasons. First, the local neighborhood properties of trained SOM contribute to easier pattern recognition tasks, hence no pre-classified samples are required in the initial training, and only a limited number of known samples is needed in the labeling phases. This feature makes SOM particularly suitable for classification cases where there are few classified samples and allow users to avoid the expensive and tedious process of known sample collection [9]. Next, the objective of SOM is to preserve topology property of data without any consideration of class assignment. FSLVQ can be used to adjust the map neurons for better performance on pattern recognition. With labels and reference vectors induced from data clustering, FSSOM offers a better starting condition for FSLVQ training than random initialization. Next, FSSOM is very close to FSLVQ in data structure and learning scheme. In fact, to stabilize the status of FSSOM, neighborhood region usually shrinks to zero in fine-tuning step so that it is easy to change to FSLVQ in a straightforward way.

3.2 Effectiveness Study

The proposed classification algorithms are implemented based on SOM & LVQ software [8]. The following experiments are performed on Iris data set in a machine with 256M memory and intel celeron 1.03 GHz processor running windows XP professional operating system. Iris data set has 150 Iris flowers, described by four numeric features: sepal length, sepal width, petal length and petal width. The samples belong to three classes respectively: 'setosa', 'versicolor', and 'virginica'. The experiments are performed in four steps.

- Step 1: The performance of proposed algorithms is evaluated using 10-fold cross validation. In each trial, nine folds are used for model exploration and the remaining is for model validation.
- Step 2: For each training data, a map is initialized linearly in the two-dimensional subspace corresponding to the largest eigenvalues of autocorrelation matrix of the training data. Afterwards, the map is trained by FSSOM using a variant of fuzzy parameter from 1 to 10 in an unsupervised manner, and then labeled according to the known samples in a supervised manner. After that, FSLVQ is performed on the resultant map with the same parameters as previous training.
- Step 3: In the validation, each sample of the test data set is compared to map units and assigned by the label of best-matching unit. Then the accuracy is calculated as the percent of the correctly classified samples.
- Step 4: The final accuracy is obtained by calculating the average results on distinct trials.

Table 1 lists the arguments used in the experiment. The intermediate values of learning rate and neighborhood radius are linearly interpolated from the initial values to the

Table 1. Arguments of FSSOM and FSLVQ

parameters	FSSOM	FSLVQ
lattice	hexagonal	-
shape	sheet	-
map size	15×4	-
neighborhood function	gaussian	-
initial radius	2	-
final radius	1	-
training epoches	5	20
initial learning rate	0.5	0.05
learning type	inverse	const

end values. After the labeling phase, some units are not labeled because no sample is projected on them. Although these neurons maybe useful on recognizing uncertain cases in future decision making, the existence of non-labeled neurons will influence the classification accuracy. This problem is exacerbated by big maps which probably result in more unlabeled neurons. Hence, these neurons are discarded before classifying. In Table 2, the average accuracy ratios for three classes and whole test data at a varied fuzzy parameter are given. It was observed that the accuracy of fuzzy configuration has an obvious increase compared to crisp configuration. The overall accuracy increases from 94.67% to 97.33%. Fuzzy parameter over 3 do not end up with any improvement on accuracy. Starting from the resulting map of FSSOM, FSLVQ does not result in significant improvement (less than 1%) on the accuracy. This is due to the fact that Iris data has an almost unmixed cluster formulation of class regions so that FSSOM classifier performs as well as hybrid classifier.

In Figure 1, the test data is projected to a 2-dimensional subspace spanned by its two eigenvectors with greatest eigenvalues using principal component analysis (PCA). Figure 2 is the projection of classified data using crisp SOM. Figure 3 is the projection of classified data using FSSOM at fuzzy parameter of 2. In each visualization, three classes are plotted in different markers: \bullet for 'setosa', \times for 'versicolor' and $*$ for 'virginica'. For the sake of easy detection, the misclassified samples are marked by \blacktriangleright in the last two figures. Compared to the crisp classifier which misclassifies two samples, fuzzy classifier results in only one error. It was also found that the misclassified samples occur on the boundary of class regions, where uncertain cases usually locate.

3.3 Fuzzy Classifying

In the following experiment, we use a more complex approach to classification phase which takes fuzzy parameter into account. The membership degrees of a sample to the units are calculated by Equation 1 and then the class with maximum degree is obtained by Equation 4. In each trial, a model is trained by FSSOM and FSLVQ using a random fuzzy parameter between 1 and 10 and a map size of $[4 \times 3]$. Each obtained model is validated by the same test data using different fuzzy parameters. Increasing the fuzzy parameter from 1 to 3 with a step of 0.2, the results achieved on test data are listed in Table 3. It was observed the result is quite good when fuzzy parameter is below 3, showing

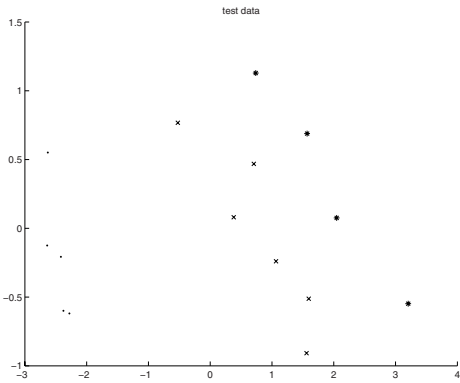


Fig. 1. PCA projection of test data

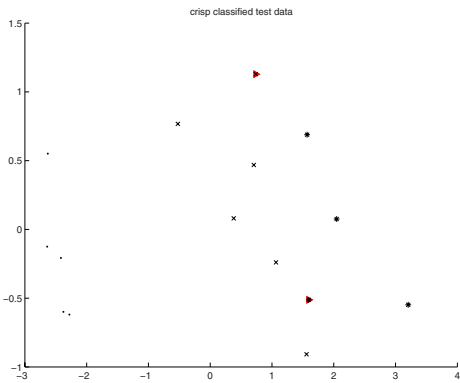


Fig. 2. PCA projection of crisp classified test data (2 errors)

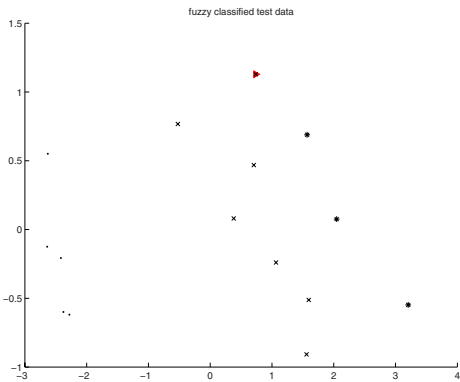


Fig. 3. PCA projection of fuzzy classified test data (1 error)

Table 2. Classification accuracy using FSSOM and FSLVQ

fuzzy	FSSOM (%)				FSSOM + FSLVQ (%)			
	setosa	versicolor	virginica	overall	setosa	versicolor	virginica	overall
1	100	90.06	95.65	94.67	100	93.39	96.90	96.00
2	100	93.81	98.33	97.33	100	93.81	98.33	97.33
3	100	92.14	97.08	96.00	100	92.14	97.08	96.00
4	100	92.14	97.08	96.00	100	92.14	97.08	96.00
5	100	92.14	97.08	96.00	100	92.14	97.08	96.00
6	100	92.14	97.08	96.00	100	92.14	97.08	96.00
7	100	92.14	97.08	96.00	100	92.14	97.08	96.00
8	100	92.14	97.08	96.00	100	92.14	97.08	96.00
9	100	92.14	97.08	96.00	100	92.14	97.08	96.00
10	100	92.14	97.08	96.00	100	92.14	97.08	96.00

that the fuzziness of classification does not degrade the accuracy while providing more information of class assignment.

Figure 4 shows the test data and map neurons in a 2-dimensional subspace. The neurons are displayed in different makers according to their labels and the number of samples is shown. For each sample, the class assignment of crisp classification and class memberships of fuzzy classification ($\alpha=2$) are given in Table 4. From the membership, the significance of an instance belonging to a class is known. Some misclassified samples are classified correctly under fuzzy strategy, for example, sample 8 is misclassified to 'virginica' in crisp case, while it is assigned to 'versicolor' in fuzzy case. Also, sample '4' and '10' are two members of 'setosa', while the latter has bigger membership (0.98) than the former (0.84). In fact, the latter is much closer to the representative neurons of 'setosa' than the former in Figure 4. It can be stated that replacing exact project with fuzzy project at a certain level in classification does not compromise the benefit of models.

Table 3. Fuzzy classification accuracy

fuzzy	FSSOM (%)				FSSOM + FSLVQ (%)			
	setosa	versicolor	virginica	overall	setosa	versicolor	virginica	overall
1.0	100	92.17	85.17	92.67	100	93.83	85.17	93.33
1.2	100	92.17	85.17	92.67	100	93.83	85.17	93.33
1.4	100	92.17	85.17	92.67	100	93.83	85.17	93.33
1.6	100	92.17	85.17	92.67	100	93.83	85.17	93.33
1.8	100	92.17	85.17	92.67	100	93.83	85.17	93.33
2.0	100	92.17	85.17	92.67	100	93.83	85.17	93.33
2.2	100	94.17	85.17	93.33	100	93.83	82.67	92.67
2.4	100	94.17	83.17	92.67	100	95.83	82.67	93.33
2.6	100	94.17	80.67	92.00	100	97.50	82.67	94.00
2.8	100	95.83	78.67	92.00	100	97.50	80.67	93.33
3.0	100	95.83	78.67	92.00	100	97.50	77.42	92.00

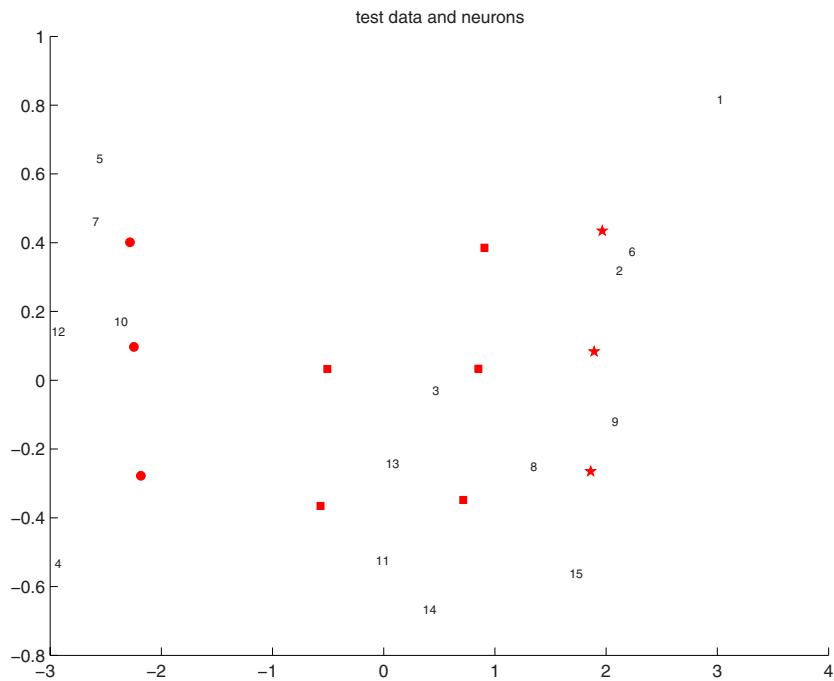


Fig. 4. PCA projection of test data and map neurons in a 2-dimensional subspace. Three classes of neurons are shown in different markers: ● for 'setosa', ■ for 'versicolor' and ★ for 'virginica'. The samples of test data are marked by their numbers

Table 4. Crisp and fuzzy classification

sample	crisp class	fuzzy classification			sample	crisp class	fuzzy classification		
		setosa	versicolor	virginica			setosa	versicolor	virginica
1	virginica	0.04	0.29	0.67	9	virginica	0.01	0.08	0.92
2	virginica	0.01	0.08	0.92	10	setosa	0.98	0.02	0.00
3	versicolor	0.02	0.90	0.08	11	versicolor	0.07	0.85	0.09
4	setosa	0.84	0.13	0.03	12	setosa	0.88	0.10	0.02
5	setosa	0.92	0.07	0.01	13	versicolor	0.05	0.86	0.08
6	virginica	0.01	0.11	0.88	14	versicolor	0.04	0.85	0.11
7	setosa	0.94	0.05	0.01	15	virginica	0.02	0.30	0.68
8	virginica	0.02	0.52	0.46					

4 Conclusion

Fuzzy classification is an extension of crisp classification using fuzzy set theory. In this paper, two fuzzy classification algorithms are proposed using sequential SOM and LVQ based on fuzzy projection. The resulting map of SOM can be used as the initialization

of LVQ in a hybrid classifier, which can improve the pattern recognition ability while preserves the topology property approximately. Experimental results show that the proposed algorithms at a certain fuzzy level improve the accuracy of classification compared to crisp algorithms. It could be stated that fuzzy classification solves the uncertainty problem of samples belonging to several classes, and improves classification accuracy in future decision. Future work will mainly focus on the qualitative description of classification models.

Acknowledgements

Parts of this reported work were supported by NSFC-RGC #70201003 from National Science Foundation of China and Head Fund #0347SZ from Institute of Policy and Management, CAS.

References

1. Bezdek, James C.: Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York (1981)
2. Bezdek, James C., Pal, Nikhil R.: Two soft relative of learning vector quantization. *Neural Networks* **8**(5) (1995) 729-743
3. Karayiannis, Nicolaos B., Pai, Pin-I: Fuzzy algorithms for learning vector quantization: generalizations and extensions. In: Steven K. Rogers (ed.): *Applications and Science of Artificial Neural Networks*. Proceedings of SPIE, Air Force Institute of Technology, Wright-Patterson AFB, OH, USA **2492** (1995) 264-275
4. Keller, James M., Gary, Michael R., Givens, James A.: A fuzzy k-nearest neighbor algorithm. *IEEE Trans. on Systems, Man, and Cybernetics* **15**(4) (1985) 580-585
5. Kohonen, T.: *Self-organizing maps*. Springer Verlag, Berlin, Second edition (1997)
6. Sohn, S., Dagli, Cihan H.: Self-organizing map with fuzzy class memberships. In *Proceedings of SPIE International Symposium on AeroSense* **4390** (2001) 150-157
7. Solaiman, B., Mouchot, Marie C., Maillard, Eric P.: A hybrid algorithm (HLVQ) combining unsupervised and supervised learning approaches. In *Proceedings of IEEE International Conference on Neural Networks(ICNN)*, Orlando, USA (1994) 1772-1778
8. Laboratory of computer and information sciences & Neural networks research center, Helsinki University of Technology: SOM Toolbox 2.0. <http://www.cis.hut.fi/projects/somtoolbox/>
9. Visa, A., Valkealahti, K., Iivarinen, J., Simula, O.: Experiences from operational cloud classifier based on self-organising map. In *Proceedings of SPIE, Orlando, Florida, Applications of Artificial Neural Networks V* **2243** (1994) 484-495