Classification with Reject Option Using the Self-Organizing Map

Ricardo Sousa[1,⋆], Ajalmar R. da Rocha Neto[3,⋆],
Jaime S. Cardoso[4], and Guilherme A. Barreto[2]

[1] INEB – Instituto de Engenharia Biomédica, Universidade Porto
[2] Departamento Engenharia de Teleinformática, Universidade Federal do Ceará (UFC)
[3] Departamento de Telemática, Instituto Federal do Ceará (IFCE)
[4] INESC Porto, FEUP, Universidade Porto

Abstract. Reject option is a technique used to improve classifier's reliability in decision support systems. It consists on withholding the automatic classification of an item, if the decision is considered not sufficiently reliable. The rejected item is then handled by a different classifier or by a human expert. The vast majority of the works on this issue have been concerned with implementing a reject option by endowing a supervised learning scheme (e.g., Multilayer Perceptron, Learning Vector Quantization or Support Vector Machines) with a reject mechanism. In this paper we introduce variants of the Self-Organizing Map (SOM), originally an unsupervised learning scheme, to act as supervised classifiers with reject option, and compare their performances with that of the MLP classifier.

Keywords: Self-Organizing Maps, Reject Option, Robust Classification, Prototype-based Classifiers, Neuron Labeling

1 Introduction

The field of machine learning has been evolving at a very fast pace, being mostly motivated and pushed forward by increasingly challenging real world applications. For instance, in credit scoring modeling, models are developed to determine how likely applicants are to default with their repayments. Previous repayment history is used to determine whether a customer should be classified into a 'good' or a 'bad' category [1].

Notwithstanding, real world problems still pose challenges which may not be solvable satisfactorily by the existing learning methodologies used by automatic decision support systems [2], leading to many incorrect predictions. However, there are situations in which the decision should be postponed, giving the support system the opportunity to identify critical items for posterior revision, instead of trying to automatically classify every and each item. In such cases, the system automates only those decisions which can be reliably predicted, letting the critical ones for a human expert to analyze. Therefore, the development of binary classifiers with a third output class, usually called the *reject class*, is attractive. This approach is known as classification with reject option [3, 4] or soft decision making [5]. Roughly speaking, reject option comprises a set

[⋆] This work was partially supported through Program CNPq/Universidade do Porto/590008/2009-9 and conducted when Ricardo Sousa was in internship at Universidade Federal do Ceará, Brazil. This work was also partially funded by Fundação para a Ciência e a Tecnologia (FCT) - Portugal through project PTDC/SAU-ENB/114951/2009 and by FEDER funds through the Programa Operacional Factores de Competitividade - COMPETE in the framework of the project PEst-C/SAU/LA0002/2013. First and second authors contributed equally to this article.

of techniques aiming at improving the classification reliability in decision support systems, being originally formalized in the context of statistical pattern recognition in [4], under the minimum risk theory. Basically, it consists in withholding the automatic classification of an item, if the decision is considered not sufficiently reliable. Rejected patterns can then be handled by a different classifier, or manually by a human. Recently, this paradigm was tailored to obtain efficient learning models [6].

In this paper we develop two novel variants of the SOM network to act as supervised classifiers with reject option, and compare their performances with that of the MLP classifier. To the best of our knowledge, this is the first time such approach is developed for the self-organizing map or similar neural networks. Computational simulations conducted in this study shows the robustness for our proposal.

2 Basics of Classification with Reject Option

As mentioned before, in possession of a "complex" dataset (e.g. from a medical diagnosis problem), every classifier is bound to misclassify some data samples. For that, we assume that the problem (and hence, the data) involves only two classes, say $\{\mathcal{C}_{-1}, \mathcal{C}_{+1}\}$, but the classifier must be able to output a third one, the reject class $\{\mathcal{C}_{-1}, \mathcal{C}_{\text{Reject}}, \mathcal{C}_{+1}\}$. Assuming that the input information is represented by an n-dimensional real vector $\mathbf{x} = [x_1 \;\; x_2 \;\; \cdots \;\; x_n]^T \in \mathbb{R}^n$, the design of classifiers with reject option can be systematized in three different approaches for the binary problem[5]:

Method 1: It involves the design of a single, standard binary classifier. If the classifier provides some approximation to the a posteriori class probabilities, $\mathbb{P}(\mathcal{C}_k|\mathbf{x})$, $k = 1, 2, ..., K$, then a pattern is rejected if the largest value among the K posterior probabilities is lower than a given threshold, say β ($0 \leq \beta \leq 1$);

Method 2: The design of two, *independent*, classifiers. A first classifier is trained to output $\mathcal{C}_{-1}$ only when the probability of $\mathcal{C}_{-1}$ is high and a second classifier trained to output $\mathcal{C}_{+1}$ only when the probability of $\mathcal{C}_{+1}$ is high. When both classifiers agree on the decision, the corresponding class is outputted. Otherwise, in case of disagreement, the reject class is the chosen one;

Method 3: The design of a single classifier with embedded reject option; that is, the classifier is trained following optimality criteria that automatically take into account the costs of misclassification and rejection in their loss functions, leading to the design of algorithms specifically built for this kind of problem.

In this paper we will introduce two SOM-based strategies for the classification with reject option paradigm under Methods 1 and 2.

3 The Self-Organizing Map

The Self-Organizing Map (SOM) [8] is one of the most popular neural network architectures. It belongs to the category of unsupervised competitive learning algorithms and it is usually designed to build an ordered representation of spatial proximity among vectors of an unlabeled data set. The neurons in the SOM are put together in an output layer, $\mathcal{A}$, in one-, two- or even three-dimensional arrays. Each neuron $j \in \mathcal{A}$, $j = 1, 2, \ldots, q$, has a weight vector $\mathbf{w}_j \in \mathbb{R}^d$ with the same dimension of the input vector $\mathbf{x} \in \mathbb{R}^d$. The network weights are trained according to a competitive-cooperative learning scheme

[5] Please consider reading [7] for further information

in which the weight vectors of a winning neuron (also called, the best-matching unit – BMU) and its neighbors in the output array are updated after the presentation of an input vector (see [9, 10]).

3.1 SOM for Supervised Classification

In order to use the SOM for supervised classification, modifications are necessary in its original learning algorithm. There are many ways to do that (see [11] and references therein), but in the present paper we will resort to two well-known strategies.
Strategy 1: The first strategy involves a post-training neuron labeling. SOM is trained in the usual unsupervised way and once done the whole training data is presented to the SOM in order to find the winning neuron for each pattern vector. The labelling of the winning neuron is conducted according to the majority voting basis, for instance. Two undesirable situations may occur: (*i*) ambiguity or (*ii*) dead neurons. In these cases, the neuron could be pruned (i.e. disregarded) from the map, or even be tagged with a "rejection class" label. In this paper, we extend Strategy 1 in order to allow the SOM network to handle pattern classification problems with reject option. For this purpose, we follow a more systematic and principled approach based on Chow's concept of rejection cost [4], instead of simply tagging ambiguous or dead neurons with "rejection class" labels.
Strategy 2: The second strategy, usually called the *self-supervised SOM* training scheme, is the one used by Kohonen for the neural phonetic typewriter [12]. According to this strategy, the SOM is made supervised by adding class information to each input pattern vector. Specifically, the input vectors $\mathbf{x}(n)$ are now formed of two parts, $\mathbf{x}_p(n)$ and $\mathbf{x}_l(n)$, where $\mathbf{x}_p(n)$ is the pattern vector itself, while $\mathbf{x}_l(n)$ is the corresponding class label of $\mathbf{x}_p(n)$. During training, these vectors are concatenated to build augmented vectors $\mathbf{x}(n) = [\mathbf{x}_p(n)\ \mathbf{x}_l(n)]^T$ which are used as inputs to the SOM. The corresponding augmented weight vectors, $\mathbf{w}_j(n) = [\mathbf{w}_j^p(n)\ \ \mathbf{w}_j^l(n)]^T$, are adjusted as in the usual SOM training procedure.

4 Incorporating Reject Option into the SOM: Two Proposals

Before proceeding with the description of the two proposals, it is worth exposing the main reasons that led to the choice of the SOM for supervised classification with rejection option instead of other prototype-based classifiers. Firstly, it has been verified that the use of a neighborhood function makes the SOM less sensitive to weight initialization [13] and accelerates its convergence [14] when compared with other prototype-based classifiers, such as the LVQ. Once trained, one can also take advantage of the SOM's density matching and topology-preserving properties to extract rules from a trained SOM network [15] in order to permit further analysis of the results towards better decision making. In particular, the density matching and topology-preserving properties will be used by both proposals to be described in order to estimate $\mathbb{P}(\mathbf{x}|\mathcal{C}_k)$ (or $\mathbb{P}(\mathcal{C}_k|\mathbf{x})$) using the distribution of SOM's weight vectors. An optimal threshold value has to be determined in order to re-tag some of the weight vectors with the rejection class label. In this paper we will also provide techniques to obtaining suitable estimates of the likelihood function $\mathbb{P}(\mathbf{x}|\mathcal{C}_k)$ or the posterior probability $\mathbb{P}(\mathbf{x}|\mathcal{C}_k)$. As a final remark, it is worth mentioning that the design methodologies of the classifiers proposed in this work, Reject Option SOM with 1 map for Classification (ROSOM-1C) and Reject Option SOM with 2 maps for Classification (ROSOM-2C), are general enough in the sense that they can be used to develop pattern classifiers with reject option using, in principle, any topology-preserving prototype-based neural networks, such as the Growing Neural Gas (GNG) [16] and the Parameterless SOM (PLSOM) [17] algorithms.

4.1 SOM with Reject Option Using One Classifier

Initially, the ROSOM-1C requires post-training neuron labeling via Strategy 1, as described in Section 3.1. Additional steps are included in order to change the labels of some neurons to *rejection class*. The main idea behind the proposal of the ROSOM-1C approach relies exactly on developing formal techniques to assign the rejection class label to a given neuron. In greater detail, the design of the ROSOM-1C requires the following steps.
$\triangleright$ **STEP 1:** For a given data set, a number of training realizations are carried out using a single SOM network in order to find the best number of neurons and suitable map dimensions. For this purpose, the conventional unsupervised SOM training is adopted.
$\triangleright$ **STEP 2:** Present the training data once again and label the prototypes $\mathbf{w}_j$, $j = 1,...,q$, according to the mode of the class labels of the patterns mapped to them. No weight adjustments are carried out at this step.
$\triangleright$ **STEP 3:** Based on the SOM's ability to approximate the input data density, we approximate $\mathbb{P}(\mathbf{x}|\mathcal{C}_k)$ with $\mathbb{P}(\mathbf{w}_j|\mathcal{C}_k,\mathbf{x})$, for $j = 1,...,q$ and $k = 1,...,K$. In the end of this Section, we describe two techniques to compute $\mathbb{P}(\mathbf{w}_j|\mathcal{C}_k,\mathbf{x})$ based on standard statistical techniques, namely, Parzen Windows and Gaussian Mixture Models.
$\triangleright$ **STEP 4:** Finding an optimum value for the rejection threshold β requires the minimization of the empirical risk as proposed in [4]:

$$\widehat{R} = w_r R + E \tag{1}$$

where R and E are, respectively, the ratio of rejected and misclassified patterns (computed using validation data), while w_r is the rejection cost (whose value must be specified in advance by the user).
$\triangleright$ **STEP 5:** Re-label the prototypes as belonging to the rejection class if $\max_k\{\mathbb{P}(\mathcal{C}_k)\mathbb{P}(\mathbf{w}_i|\mathcal{C}_k,\mathbf{x})\} < \beta$ verifies.

On the Estimation of $\mathbb{P}(\boldsymbol{w}_j|\mathcal{C}_k,\boldsymbol{x})$*:* The first approach to be used to compute SOM-based estimates of $\mathbb{P}(\mathbf{w}_j|\mathcal{C}_k)$ is through the Parzen windows nonparametric method. The estimation is usually performed by some kernel function, usually a Gaussian, averaged by the number of points belonging to a given class. Another approach that can also be used to estimate $\mathbb{P}(\mathbf{w}_j|\mathcal{C}_k,\mathbf{x})$ based on the distribution of weight vectors of the SOM is the Gaussian Mixture Models (GMM) [18–20]. In this paper we follow the approach developed by [19], which is implemented in the SOM toolbox[6].

Neuron Re-Labeling Based on Gini Index*:* For the application of the decision rule in **STEP 5**, one has to store all the values of the posterior probabilities estimates

[6] Available for download at `http://www.cis.hut.fi/somtoolbox/`

$\mathbb{P}(\mathcal{C}_k|\mathbf{w}_j,\mathbf{x}) \propto \mathbb{P}(\mathcal{C}_k)\mathbb{P}(\mathbf{w}_j|\mathcal{C}_k,\mathbf{x})$ for each neuron j. The quantity $\mathbb{P}(\mathcal{C}_k|\mathbf{w}_j,\mathbf{x})$ express the probability of an instance that has fallen within the Voronoi cell of neuron j to belong to class $\mathcal{C}_k$. By means of concepts borrowed from information theory, it is possible to merge all the probabilities $\mathbb{P}(\mathcal{C}_k|\mathbf{w}_j,\mathbf{x})$, $k = 1,...,K$, associated with a given neuron, into a single quantity to be called *cell impurity*.

4.2 SOM with Reject Option Using Two Classifiers

In comparison to the ROSOM-1C, the individual SOM networks that comprise the ROSOM-2C have an extra feature: the ability to control the preference for patterns of a given class by the inclusion of cost parameter w_r into the learning rules of the individual networks. In other words, one individual network is trained to become specialized, say, on class $\mathcal{C}_{-1}$, while the other is trained to become specialized on class $\mathcal{C}_{+1}$.

By allowing one of the networks to have preference for (i.e. to be biased toward) the patterns of class C_{+1}, while the other has preference for the patterns of class C_{-1}, makes the decision rule of ROSOM-2C more reliable. More reliable in the sense that a pattern is classified only when the outputs of both network coincides, otherwise the pattern is rejected. The design of the ROSOM-2C requires the following steps.
▷ **STEP 1:** Choose a rejection cost w_r.
▷ **STEP 2:** Train two SOM networks following the self-supervised SOM training scheme describe in Section 3.1.

2.1) Train the first SOM network, henceforth named SOM-1 classifier, to become specialized on the class $\mathcal{C}_{-1}$. For that, we replace the standard SOM learning rule with Equation (2).

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \begin{cases} \eta(n)h(i,j;n)[\mathbf{x}(n) - \mathbf{w}_j(n)]w_r, & \text{if class}(\mathbf{x}(n)) = \mathcal{C}_{+1} \\ \eta(n)h(i,j;n)[\mathbf{x}(n) - \mathbf{w}_j(n)](1-w_r), & \text{if class}(\mathbf{x}(n)) = \mathcal{C}_{-1}. \end{cases} \tag{2}$$

2.2) Train the second SOM network, henceforth named SOM-2 classifier, to become specialized on the class $\mathcal{C}_{+1}$. For that, we replace the standard SOM learning rule with Equation (3).

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \begin{cases} \eta(n)h(i,j;n)[\mathbf{x}(n) - \mathbf{w}_j(n)](1-w_r), & \text{if class}(\mathbf{x}(n)) = \mathcal{C}_{+1} \\ \eta(n)h(i,j;n)[\mathbf{x}(n) - \mathbf{w}_j(n)]w_r, & \text{if class}(\mathbf{x}(n)) = \mathcal{C}_{-1}. \end{cases} \tag{3}$$

▷ **STEP 3, 4 and 5:** The same as the ones described for the ROSOM-1C classifier. The Gini coefficient approach can also be used to re-label the prototypes of the ROSOM-2C classifier.

A final remark is necessary here. Extension of the ROSOM-2C approach to multiclass problems is straightforward. For this, one should adopt a One-Against-One strategy, which is commonly used to extend SVM binary classifiers to multiclass problem. In this case the algorithm would be the following: For K classes, construct $K(K-1)/2$ ROSOM-2C classifiers. Each classifier discriminates between two classes. A new incoming pattern is assigned using each classifier in turn and a majority vote taken. In case of ambiguity of the majority vote, with no clear decision for some patterns, the pattern is rejected.

5 Experimental Study and Discussion

The performance of the classification methods were assessed over two datasets: One synthetic dataset was generated as in [7, 21] (syntheticI) and a real-world dataset. The real-world data set represents the discrimination of normal subjects from those with a pathology on the vertebral column (VC). This database, also publicly available on the UCI machine learning repository, contains information about 310 patients obtained from sagittal panoramic radiographies of the vertebral column described by 6 different biomechanical features. See [22] for more detail on this data set.

In the experiments, we used the SOM toolbox for implementing the ROSOM-1C and ROSOM-2C classifiers and the Matlab$^{\text{TM}}$ Neural Networks toolbox for MLP-based classifiers. For fair performance comparison, we have instantiated the same rejection option strategies used for the SOM-based classifiers into the MLP-based classifiers, giving rise to the MLP-1C and MLP-2C classifiers. For the SOM-based classifiers we used a two-dimensional map with a hexagonal neighborhood structure and a Gaussian neighborhood function. A 5-fold cross validation was conducted to find the best number of neurons and the initial radius size for the neighborhood function. Our search considered a squared map spanning 5×5 to 25×25 neurons. The learning phase stopped after 200 epochs. A similar search was conducted for the MLP-based classifiers to find the best number of neurons that composed the network: 5 to 20 neurons with one hidden layer, a single output neuron, and logistic sigmoid as activation function for all neurons. We defined a maximum number of 15 epochs as the stopping criterion in order to avoid overfitting. The resilient back-propagation training algorithm was used.

It is important to point out that, in the absence of further insights about the problem at our disposal (other than the data itself), we cannot select only one value for w_r, since its selection is intrinsically application-dependent. Thus, we started by running the classifiers spanning three values for w_r in Equation (1): 0.04, 0.24 and 0.44[7]. As mentioned the w_r value is directly related to how many patterns an expert is willing to reject. To assess the stability of the proposed approaches the experiments were repeated 50 times by averaging the results. The performance of our methods are plotted on an Accuracy-Reject (A-R) curve where each point break in the curves corresponds to a given w_r value.

By analyzing the performance on an A-R curve one can easily read the performance achieved by a given method and how much it was rejected for a given w_r: the highest the curve, the better the performance is. For example, for the A-R curves shown in Fig. 1, the ROSOM-1C using the Parzen and Gini coefficient approaches achieved the best overall results. We can also see that the performances of all ROSOM-2C variants and the MLP-2C were equivalent. For the VC dataset, the A-R curves in Fig. 1 (second row) indicate that the ROSOM-1C/Gini achieved the best overall performance. Also in the same Figure, we see that all the ROSOM-2C variants performed better than the MLP-2C. It is worth mentioning that to verify that the performances of the SOM-based and MLP-based classifiers are equivalent is *not* a bad thing for the SOM-based classifiers. On the contrary, it is a good thing. Let us recall that the SOM is being adapted to work as a supervised classifier, since it is originally an unsupervised learning algorithm. But even so, the proposed SOM-based approaches achieved very competitive results in

[7] Values of w_r higher than 0.5 are equivalent to random guesses.

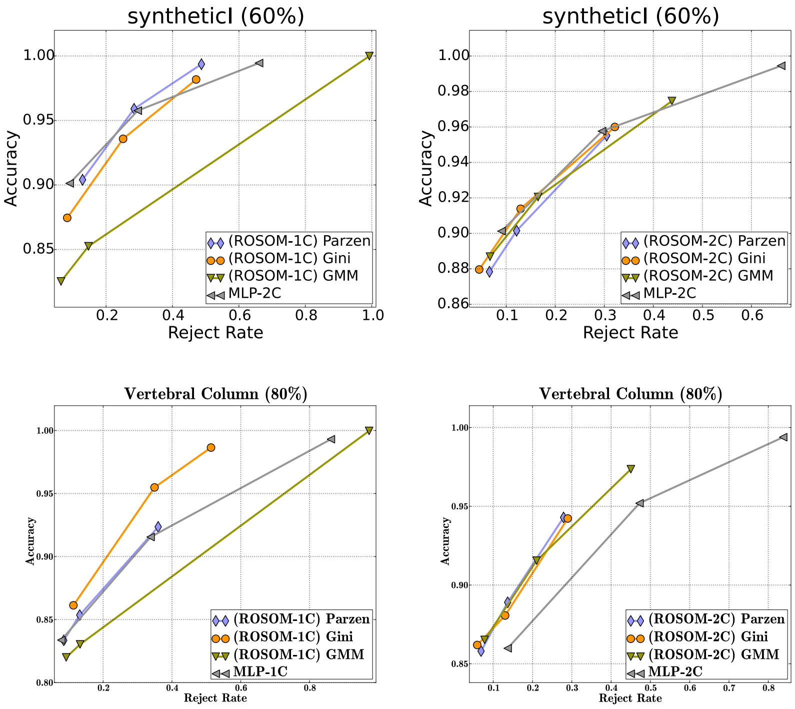

Fig. 1: The A-R curves for the `SyntheticI` dataset using 60% of training data (first row); and, `VC` dataset using 80% of training data (second row).

comparison with the MLP-based approaches. For all datasets the ROSOM-1C/GMM achieved in average the worst results. However, the ROSOM-2C/GMM achieved competitive results in comparison with the other approaches based on two classifiers. Such behavior can be partly explained by the fact that the proposed modified learning rules in (2) and (3) provide additional improvement over the raw estimates of the posterior probabilities in the performances of the ROSOM-2C classifier. As a general conclusion, although neither the Parzen windows nor the Gini coefficient approaches outperformed one another over all datasets, Parzen and Gini attained better performances than the MLP-based counterparts. Future experiments will be conducted in a larger set of datasets.

6 Conclusions

In this paper we presented two SOM-based pattern classifiers that incorporate the rejection class option: (a) ROSOM-1C, encompassing a single SOM network trained in the usual unsupervised way; and (b) ROSOM-2C, requiring two SOMs which are trained in the self-supervised learning scheme. For both proposals we analysed the advantages on using existing estimates for the likelihood function or the posterior probability tailored for the rejection problem. For ROSOM-2C a new learning rule was proposed. The simulations show that our classifiers are very robust in terms of confidence in decision making process, attaining higher performances than their siblings.

References

1. Thomas, L.C., Edelman, D.B., Crook, J.N.: Credit Scoring and Its Applications. 1st edn. SIAM (2002)
2. Han, J., Gao, J.: Research challenges for data mining in science and engineering. In Kargupta, H., Han, J., Yu, P.S., Motwani, R., Kumar, V., eds.: Next Generation of Data Mining. Chapman & Hall / CRC Press (2009) 1–18
3. El-Yaniv, R., Wiener, Y.: On the foundations of noise-free selective classification. Journal of Machine Learning Research **11** (2010) 1605–1641
4. Chow, C.: On optimum recognition error and reject tradeoff. IEEE Transactions on Information Theory **16**(1) (1970) 41–46
5. Ishibuchi, H., Nii, M.: Neural networks for soft decision making. Fuzzy Sets and Systems **34**(115) (2000) 121–140
6. Sousa, R., da Rocha Neto, A.R., Barreto, G.A., Cardoso, J.S., Coimbra, M.T.: Reject Option Paradigm for the Reduction of Support Vectors. In: ESANN. (2014) accepted.
7. Sousa, R., Cardoso, J.S.: The Data Replication Method for the Classification with Reject Option. AI Communications **26** (2013) 281–302
8. Kohonen, T.: The self-organizing map. Proceedings of the IEEE **78**(9) (1990) 1464–1480
9. van Hulle, M.: Self-organizing maps. In Rozenberg, G., Baeck, T., Kok, J., eds.: Handbook of Natural Computing: Theory, Experiments, and Applications. Springer-Verlag (2010) 1–45
10. Yin, H.: The self-organizing maps: Background, theories, extensions and applications. In Fulcher, J., Jain, L.C., eds.: Computational Intelligence: A Compendium. Volume 115 of Studies in Computational Intelligence. Springer-Verlag (2008) 715–762
11. Mattos, C.L.C., Barreto, G.A.: ARTIE and MUSCLE models: building ensemble classifiers from fuzzy ART and SOM networks. Neural Computing & Applications (2012)
12. Kohonen, T.: The 'neural' phonetic typewriter. Computer **21**(3) (1988) 11–22
13. Kohonen, T.: Self-Organizing Maps. 3rd edn. Springer (2001)
14. de Bodt, E., Cottrell, M., Letremy, P., Verleysen, M.: On the use of self-organizing maps to accelerate vector quantization. Neurocomputing **56** (2004) 187–203
15. Malone, J., McGarry, K., Wermter, S., Bowerman, C.: Data mining using rule extraction from Kohonen self-organising maps. Neural Computing and Applications **15** (2005) 9–17
16. Fritzke, B.: A growing neural gas network learns topologies. In: Advances in Neural Information Processing Systems 7. MIT Press, Cambridge, MA (1995) 625–632
17. Berglund, E., Sitte, J.: Parameterless self-organizing map algorithm. IEEE Transactions on Neural Networks **17**(2) (2006) 305–316
18. Yin, H., Allinson, N.M.: Self-organizing mixture networks for probability density estimation. IEEE Transactions on Neural Networks **12**(2) (2001) 405–411
19. Alhoniemi, E., Himberg, J., Vesanto, J.: Probabilistic measures for responses of self-organizing map units. In: International ICSC Congress on Computational Intelligence Methods and Applications (CIMA, ICSC Academic Press (1999) 286–290
20. Holmström, L., Hämäläinen, A.: The self-organizing reduced kernel density estimator. In: Proceedings of the 1993 IEEE International Conference on Neural Networks (ICNN'93). (1993) 417–421
21. Cardoso, J.S., da Costa, J.F.P.: Learning to classify ordinal data: the data replication method. Journal of Machine Learning Research **8** (2007) 1393–1429
22. Rocha-Neto, A.R., Sousa, R., Cardoso, J.S., Barreto, G.A.: Diagnostic of pathology on the vertebral column with embedded reject option. In: Proceedings of the 5th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2011). Volume LNCS-6669. (2011) 588–595