Hesitant Neural Gas for Supervised and Semi-supervised Classification

Piotr Płoński and Krzysztof Zaremba

Institute of Radioelectronics, Warsaw University of Technology,
Nowowiejska 15/19,00-665 Warsaw, Poland
{pplonski,zaremba}@ire.pw.edu.pl

Abstract. Neural Gas is a neural network algorithm for vector quantization. It has not arbitrary established network topology, instead its topology is changing dynamically during training process. Originally, the Neural Gas is an unsupervised algorithm. However, there are several extensions that enables Neural Gas to use the information about sample's class. This significantly improves the accuracy of obtained clusters. Therefore, the Neural Gas was successfully used in classification problems. In this paper we present a novel method to learn the Neural Gas with fully and partially labelled data sets. Proposed method simulates the neuron's hesitation between membership to the classes during the learning. Hesitation process is based on neuron's class membership probability and Metropolis-Hastings algorithm. The proposed method was compared with state-of-art extensions of Neural Gas on supervised and semi-supervised classification tasks on benchmark data sets. Experimental results yield better or the same classification accuracy on both types of supervision.

Keywords: Neural Gas, Supervised clustering, Semi-supervised clustering, Classification, Metropolis-Hastings algorithm.

1 Introduction

Neural Gas (NG) is an algorithm for cluster analysis [2], first presented by Martinez and Shulten [11]. In contrary to well known Self-Organising Maps [10] it has not arbitrary established network topology, instead its topology is changing dynamically during the training process. There are many extensions of NG mainly focused on finding optimal neurons number [3] or using more sophisticated similarity measures than Euclidean [14], [5]. Originally, NG optimises clusters in unsupervised way, although there are various examples that use NG in classification tasks [14], [17]. The methods that enables use of NG for classification can be divided into three groups.

The first group of methods uses standard NG in an unsupervised manner. After training for each neuron the class label is designated based on major vote of sample's class, which belongs to the neuron. This method is also so-called 'winner-takes-all' (WTA) strategy [14].

L. Rutkowski et al. (Eds.): ICAISC 2013, Part I, LNAI 7894, pp. 474–482, 2013.

The second approach combines information about class label in binary coded manner in attribute vector [13]. Each neuron has two types of weights, corresponding to attributes and class. The part of input vector with class information is presented only during training. In testing phase, the information of neuron class label is coded in class weights. This can be interpreted as a fuzzy class membership. There are several approaches to measure similarity between neuron's weights and input vector [18], [19].

Third group of methods arbitrary assigns neurons to the class label [14]. The neuron is learned only with samples from the corresponding class. During the testing, the output class label is designed upon the closest neuron's class. There are some more sophisticated methods of learning with arbitrary assigned neurons in NG[5], [7], [4].

Contemporary, more often in data mining are situations that class labels are not available for all samples in data set. This is because labelling data by human expert can be expensive. Learning with partially labelled data is so-called semi-supervised [8].

In this paper we present a novel method for controlling supervision in Neural Gas algorithm. It is based on neuron's class membership probability and Metropolis-Hastings (MH) algorithm [12], [6]. The MH is well known from Simulated Annealing (SA) method [9]. Proposed method can be used on both data type: fully and partially labelled. We so-called proposed method as 'Hesitant Neural Gas' (HNG). Recently, we proposed a similar method for controlling learning of neurons in Self-Organising Maps [15].

Firstly, we provide a description of Neural Gas algorithm and three methods to use it for classification (one from each group). Secondly, the Hesitant Neural Gas algorithm is described. Then, the comparison of the HNG with other methods is presented on fully and partially labelled sets. Additionally, on fully labelled sets HNG is compared to Learning Vector Quantization (LVQ) algorithm [10], which is a state-of-art method in prototype-based supervised classification.

2 Methods

Let's denote data set as $D = \{(\boldsymbol{x_i}, c_i)\}$, where $\boldsymbol{x_i}$ is an attribute vector, $\boldsymbol{x} \in \mathcal{R}^d$ and c_i is a discrete class number of i-th sample, $i = [1, 2, ..., M]$ and $c = [1, 2, ..., C]$. Sometimes the class number will be encoded as a binary vector and denoted as $\boldsymbol{y_i}$, where $\boldsymbol{y_{ij}} = 1$ for $j = c_i$ and $\boldsymbol{y_{ij}} = 0$ otherwise.

2.1 Neural Gas

In the Neural Gas algorithm each neuron is described by weights vector $\boldsymbol{w}_j$, where $j = \{1, 2, .., N\}$. For each input sample D_i are computed distances to neurons by following equation:

$$Dist(\boldsymbol{w}_j, D_i) = (\boldsymbol{x}_i - \boldsymbol{w}_j)^T(\boldsymbol{x}_i - \boldsymbol{w}_j). \tag{1}$$

Then distances are sorted and for each neuron a k_j rank is assigned, $k = \{0, 1, 2, .., N - 1\}$. The rank $k_j = 0$ is assigned to the closest neuron, whereas consecutive k are for neurons with greater distance. The $k_j = N-1$ is for the furthest neuron. Then, weight update step is executed. The weights of each neuron are updated with the following formula:

$$\boldsymbol{w}'_j = \boldsymbol{w}_j + \eta e^{-k_j/\lambda}(\boldsymbol{x}_i - \boldsymbol{w}_j), \tag{2}$$

where η is a learning rate and λ is a neighbourhood range. The η is decreasing during learning:

$$\eta = \eta_0 e^{-t/\sigma}, \tag{3}$$

where t is a current epoch number and σ controls speed of decreasing. Network is trained till chosen number of learning procedure iterations t_{stop} is exceeded. In original Neural Gas presented by Martinez and Schulten [11] there were also optimised edges, which connect similar neurons. This can be useful for visualization purposes. However, this is not in the scope of this paper.

2.2 WTA Neural Gas

In the WTA Neural Gas after unsupervised training process the class membership for each neuron is computed. The neuron's class label is designated base on major votes of sample's class for which neuron was selected as the closest ($k_j = 0$). The disadvantage of this method are so-called 'empty neurons', when neuron has no assigned label. This situation is observed, when neuron has never been selected as the closest during training but is selected for the testing sample. In case of partially labelled data set, only labelled samples participate in class voting.

2.3 Fuzzy Neural Gas

The other approach to use NG as classifier is so-called 'Fuzzy Neural Gas'. In the training process, it takes into consideration the class vector $\boldsymbol{y}_j$ additionally to input attributes. Each neuron contains part of weights corresponding to the attributes $\boldsymbol{w}^x_j$ and class $\boldsymbol{w}^y_j$. The similarity measure between input sample and neuron is computed during learning process by equation:

$$Dist_{train}(\boldsymbol{w}_j, D_i) = \gamma(\boldsymbol{w}^x_j - \boldsymbol{x}_i)^T(\boldsymbol{w}^x_j - \boldsymbol{x}_i) + (1-\gamma)(\boldsymbol{w}^y_j - \boldsymbol{y}_i)^T(\boldsymbol{w}^y_j - \boldsymbol{y}_i). \tag{4}$$

The γ coefficient controls the balance between distance from attributes and class. The update step is performed with equations:

$$\boldsymbol{w}^{x'}_j = \boldsymbol{w}^x_j + \eta\gamma e^{-k_j/\lambda}(\boldsymbol{x}_i - \boldsymbol{w}^x_j), \tag{5}$$

$$\boldsymbol{w}^{y'}_j = \boldsymbol{w}^y_j + \eta(1-\gamma) e^{-k_j/\lambda}(\boldsymbol{y}_i - \boldsymbol{w}^y_j). \tag{6}$$

In the testing phase, to the network is presented an input vector only with attributes. This step is also so-called 'exploitation phase'. The distance is computed by:

$$Dist_{test}(\boldsymbol{w}_j, D_i) = (\boldsymbol{w}_j^x - \boldsymbol{x}_i)^T(\boldsymbol{w}_j^x - \boldsymbol{x}_i). \tag{7}$$

The output class label is designated based on position of maximum value in the $\boldsymbol{w}_j^y$ weights of the closest neuron. For semi-supervised learning, the second part of equation (4) is considered only when sample's class label is available, otherwise is omitted.

2.4 Class Neural Gas

The last approach arbitrary assigns neurons to the classes. In the training process neurons take part in the learning only with samples from corresponding class. During testing, all neurons are considered for distance computation. The output class label is designated from the closest neuron. We so-called this method as 'Class Neural Gas' (CNG). In case of learning with samples without class label all neurons participate in the distance computation during training and testing.

2.5 Proposed Method - Hesitant Neural Gas

In the proposed method, neuron's class membership is described by a probability. We note $P_j(h)$ as a probability of j-th neuron's membership in class number h. In the training phase, for each sample is selected a group of neurons that will take part in the weights optimisation. Selection is described by a matrix T, where $T_j^i = 1$ means that j-th neuron will participate in the learning with i-th sample, $T_j^i = 0$ otherwise. Neurons are selected in two steps. First choose neurons having maximum probability for the class matching the class c_i of the input sample:

$$T_j^{i(1)} = \begin{cases} 1 & \text{if } \arg\max_h(P_j(h)) = c_i; \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

In the second step, remaining neurons are considered, with $T_j^{i(1)} = 0$. The decision on joining into the training with i-th sample is taken upon MH algorithm. The probability of joining is computed using following equation:

$$J_j^i = 1 - exp(-\rho P_j(c_i) t_{stop}/t), \tag{9}$$

where ρ is the parameter that controls the intensity of hesitation, $\rho \in [0, 1]$. The greater ρ value, the more neurons are selected additionally to learning in the MH step. In the eq.(9) the number of training iteration t is used, therefore neurons will be selected less frequently at the end of learning process than at its beginning. This can be interpreted as a hesitation of the neuron, which decreases during the training. Whether the MH decision will be positive ($T_j^{i(2)} = 1$), we draw random number a from an uniform distribution, $a \in [0, 1]$. The neuron will be added to the training group if a value is smaller than J_j^i:

$$T_j^{i(2)} = \begin{cases} 1 & \text{if } a < J_j^i; \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The final decision on neuron selection is a logical 'or' of the decisions from two steps: $T_j^i = T_j^{i(1)} \vee T_j^{i(2)}$. Neurons with $T_j^i = 0$ will not take part in distance computation step neither in weights update step. After all training samples presentation, neuron's class membership probability is updated. During the training for each i-th sample the neighbourhood value $e^{-k_j/\lambda}$ is added to the neuron's probability of membership in a given class:

$$P'_j(h) = \sum_i^N T_j^i e^{-k_j/\lambda}, \text{ for } h = c_i. \tag{11}$$

Note, that the neighbourhood is considered only if j-th neuron was selected for training with i-th sample. The neighbourhood value represents the belonging of the neuron to the input sample's class. After all iterations in a given epoch, the probability is normalized and updated with formula:

$$P_j(h) = \frac{P'_j(h)}{\sum_{l=1}^C P'_j(l)}. \tag{12}$$

In case of partially labelled data, we assume that all neurons take part in the training for samples without class label, thus $T_j^i = 1$ for all neurons. However, unlabelled samples do not take part in probability of class membership update (eq. 11). For labelled samples the procedure described above is used.

3 Results

To test performance of the Hesitant Neural Gas method on fully labelled data, we will compare it to the Learning Vector Quantization algorithm (LVQ) [10], WTA NG, Fuzzy NG, Class NG, Hesitant NG. The LVQ is not used in comparison on partially labelled data sets. The comparison is made on 6 real data sets. We used data sets 'Wine', 'Ionosphere', 'Iris', 'Sonar', 'Glass' from the 'UCI Machine Learing Repository' [1] [1], and set 'Faces' are from the 'The ORL Database of Faces'[2]. Data sets are described in Table 1. In all experiments we train algorithms with number of iterations $t_{stop} = 200$. We use learning rate $\eta_1 = 0.1$, exponentially decreasing to $\eta_{200} = 0.001$. The neighbourhood range was $\lambda = 1$. All algorithms were initialized with random samples. For all data sets, we arbitrarily chose the neurons number - selecting optimal network size is not in the scope of this paper. The selected values are presented in Table 1. The total number of neurons for each algorithm type is equal. Additionally, the ρ parameter for the HNG must be tuned. We checked several values of ρ, $\rho = \{0.05, 0.25, 0.5, 0.75, 1.0\}$. The optimal value was selected by cross-validation. Selected ρ values for each data set are presented in Table 1. To demonstrate the impact on number of positive MH decision depending on different ρ values, we count the number of positive MH decisions in each learning epoch for all neurons in the network for all considered

[1] http://archive.ics.uci.edu/ml/

[2] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Table 1. Description of data sets used to test performance, number of neurons used to each data set and optimal ρ in the Hesitant Neural Gas. (*In 'Faces' data set, the number of attributes was reduced with PCA.)

	Train examples	Test examples	Attributes	Classes	# neurons	MH ρ
Faces	320	80	50*	40	80	0.75
Sonar	166	42	60	2	36	1
Glass	171	43	9	6	24	0.05
Iris	120	30	4	3	12	0.25
Ionosphere	280	71	34	2	24	0.5
Wine	142	36	13	3	12	0.25

ρ values. The demonstration is made on 'Iris' set and presented in the Fig.1. It can be observed that, the greater ρ value is, the more positive MH decisions are made and the more frequently neuron takes part in the training with the sample from the class different than its major class. For each data set we made 10 repetitions to avoid effect of local minima. At each time training and testing subsets were redrawn. For comparison measure, we take a percentage of incorrect classifications. The obtained mean results on testing subsets are presented in the Table.2. The results were obtained using all labels from data sets in the training.

Table 2. Percent of incorrect classification on the testing subsets. Networks were learned with fully labelled samples. Results are mean and σ over 10 runs.

	LVQ	WTA Neural Gas	Fuzzy Neural Gas	Class Neural Gas	Hesitant Neural Gas
Faces	8.25±3.34	21.38±4.62	18.50±6.66	**4.00±2.55**	4.50±2.44
Sonar	14.52±7.48	23.1±5.39	19.76±6.92	**13.33±6.07**	13.57±5.50
Glass	31.16±6.95	34.42±5.98	37.67±9.29	35.35±5.46	**29.77±9.79**
Iris	4.00±2.11	4.33±4.46	**3.67±1.89**	4.00±2.11	4.00±2.11
Ionosphere	10.99±2.95	9.44±3.26	8.73±3.44	8.17±2.18	**7.89±2.75**
Wine	5.00±3.66	5.28±2.76	**3.06±2.43**	**3.06±2.43**	3.33±2.87
All sets error	73.92	97.95	91.39	67.91	**63.06**

The overall classification error on all data sets was the smallest for the proposed HNG method. However, the CNG was the best method on three sets. It gains the lowest error on 'Faces', 'Sonar' and 'Wine' sets. The HNG was the best method on two sets: 'Sonar' and 'Ionosphere'. The FNG method was the best on two data sets, namely: 'Iris' and 'Wine'. The HNG and CNG obtained smaller overall error than the LVQ algorithm. Although, the LVQ method was better than WTA-NG and FNG. The WTA-NG has the poorest accuracy on all sets, which can be expected as only this method does not use information about class labels directly in the learning.

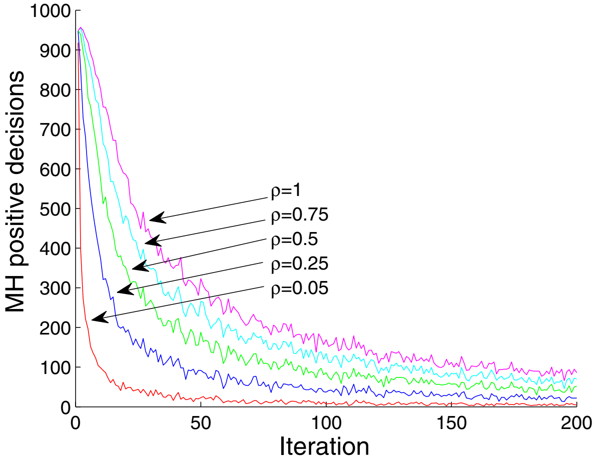

Fig. 1. Number of positive MH decisions in Hesitant Neural Gas algorithm taken in each training iteration for different ρ values on 'Iris' data set

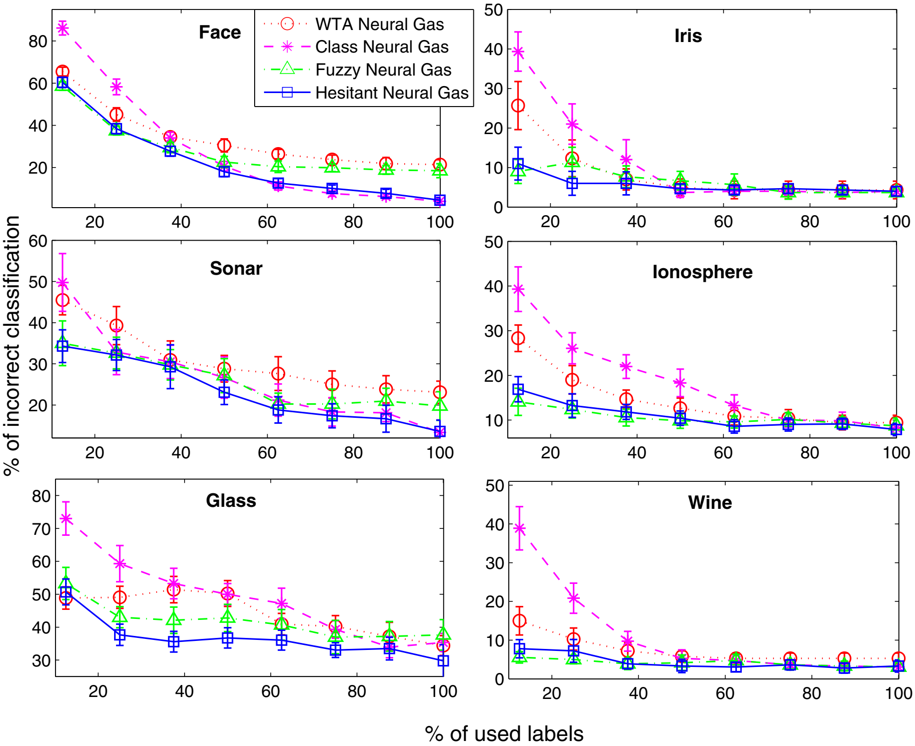

Fig. 2. Percent of incorrect classification on the testing subsets. Networks were learned with partially labelled samples. Results are mean and σ over 10 runs.

To test performance of the proposed HNG method on partially labelled data, we used only part of available labels from the training subsets in the learning process, in per cent $r = \{12.5, 25, 37.5, 50, 75, 87.5, 100\}$. The results are presented in the Fig.2. The HNG method achieved the smallest classification error for 'Faces', 'Sonar', 'Glass' and 'Iris' data sets when less than a half of available labels were used during the learning, $r < 50$. For 'Ionosphere' and 'Wine' data sets when r was smaller than 50, the FNG has the slightly better performance than the HNG. When small number of labels was used ($r < 50$), it can be observed that the CNG has the largest classification error on all data sets. Though, when the number of used labels grows the performance of the CNG significantly increases. This can be caused by arbitrary assigning class labels to the neuron. When the number of samples with class labels is smaller than number of samples without labels, the impact of labelled samples on neurons' weights is not enough to force unlabelled samples to belong to correct neurons. For 'Iris' and 'Wine' data sets, for $r > 50$ all methods seems to give similar results. These sets are rather simple, therefore all methods obtained similar local minima.

4 Conclusions

In this paper we present a novel method that extends Neural Gas algorithm for supervised and semi-supervised learning. It is so-called the 'Hesitant Neural Gas'. It controls the neuron's weights optimisation by selecting a group of neurons which will participate in the training of the presented sample. At first, neurons with the same as sample's class are selected. In the next step, the hesitation mechanism is introduced, which enables neurons with different class to take part in weights optimisation. The hesitation is based on neuron's class membership probability and Metropolis-Hastings algorithm. The hesitation intensity is controlled by ρ parameter and current training epoch number. The number of MH positive decisions decrease during learning, which can be interpreted as making neurons more confident. For unlabelled samples all neurons participate in the training. The proposed HNG method was compared to other state-of-art extensions of NG and LVQ algorithm on classification tasks. The results confirm that proposed method obtains better or similar accuracy than other methods on both types of supervision. Matlab implementation of the HNG algorithm is available at `http://home.elka.pw.edu.pl/~pplonski/hesitant_neural_gas`.

References

1. Asuncion, A., Newman, D.J.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2007)
2. Du, K.-L.: Clustering: A neural network approach. Neural Networks 23, 89–107 (2010)
3. Fritzke, B.: A Growing Neural Gas Network Learns Topologies. In: Advances in Neural Information Processing Systems (NIPS 1994), pp. 625–632 (1994)

4. Hammer, B., Hasenfuss, A., Schleif, F.-M., Villmann, T.: Supervised Batch Neural Gas. In: Schwenker, F., Marinai, S. (eds.) ANNPR 2006. LNCS (LNAI), vol. 4087, pp. 33–45. Springer, Heidelberg (2006)
5. Hammer, B., Strickert, M., Villmann, T.: Supervised Neural Gas with General Similarity Measure. Neural Processing Letters 21, 21–44 (2005)
6. Hastings, W.K.: Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika 57, 97–109 (1970)
7. Herrmann, M., Villmann, T.: Vector Quantization by Optimal Neural Gas. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 625–630. Springer, Heidelberg (1997)
8. Kästner, M., Villmann, T.: Fuzzy Supervised Self-Organizing Map for Semi-supervised Vector Quantization. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS, vol. 7267, pp. 256–265. Springer, Heidelberg (2012)
9. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220, 671–680 (1983)
10. Kohonen, T.: The Self-Organizing Map. Proceedings of the IEEE 78, 1464–1480 (1990)
11. Martinetz, T., Schulten, K.: A Neural-Gas Network Learns Topologies. Artificial Neural Networks 1, 397–402 (1991)
12. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equations of State Calculations by Fast Computing Machines. Journal of Chemical Physics 21, 1087–1092 (1953)
13. Midenet, S., Grumbach, A.: Learning Associations by Self-Organization: The LASSO model. Neurocomputing 6, 343–361 (1994)
14. Möller, R., Hoffmann, H.: An extension of neural gas to local PCA. Neurocomputing 62, 305–326 (2004)
15. Płoński, P., Zaremba, K.: Self-Organising Maps for Classification with Metropolis-Hastings Algorithm for Supervision. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part III. LNCS, vol. 7665, pp. 149–156. Springer, Heidelberg (2012)
16. Schenck, W., Welsch, R., Kaiser, A., Möller, R.: Adaptive learning rate control for neural gas principal component analysis. In: European Symposium on Artificial Neural Networks (ESANN 2010), pp. 213–218. d-side pub. (2010)
17. Schleif, F.-M., Villmann, T., Hammer, B.: Supervised Neural Gas for Classification of Functional Data and Its Application to the Analysis of Clinical Proteom Spectra. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 1036–1044. Springer, Heidelberg (2007)
18. Villmann, T., Geweniger, T., Kästner, M., Lange, M.: Fuzzy Neural Gas for Unsupervised Vector Quantization. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS, vol. 7267, pp. 350–358. Springer, Heidelberg (2012)
19. Villmann, T., Hammer, B., Schleif, F.-M., Geweniger, T., Hermann, W.: Fuzzy classification by fuzzy labeled neural gas. Neural Networks 19, 772–779 (2006)