

Overcoming the Linearity of Ordinal Logistic Regression Adding Non-linear Covariates from Evolutionary Hybrid Neural Network Models

Manuel Dorado-Moreno^(✉), Pedro Antonio Gutiérrez,
Javier Sánchez-Monedero, and César Hervás-Martínez

Department of Computer Science and Numerical Analysis, University of Cordoba,
Campus de Rabanales, Albert Einstein building, 14071 Cordoba, Spain
{i92domom,pagutierrez,jsanchezm,chervas}@uco.es

Abstract. This paper proposes a non-linear ordinal logistic regression method based on the combination of a linear regression model and an evolutionary neural network with hybrid basis functions, combining Sigmoidal Unit and Radial Basis Functions neural networks. The process for obtaining the coefficients is carried out in several steps. Firstly we use an evolutionary algorithm to determine the structure of the hybrid neural network model, in a second step we augment the initial feature space (covariate space) adding the non-linear transformations of the input variables given by the hybrid hidden layer of the best individual of the evolutionary algorithm. Finally, we apply an ordinal logistic regression in the new feature space. This methodology is tested using 10 benchmark problems from the UCI repository. The hybrid model outperforms both the RBF and the SU pure models obtaining a good compromise between them and better results in terms of accuracy and ordinal classification error.

Keywords: Artificial neural networks · Hybrid basis functions · Proportional odds model · Evolutionary algorithms · Ordinal classification · Ordinal regression

1 Introduction

Artificial Neural Networks (ANNs) are a very flexible modelling technique based on biological neural systems, whose computing power is developed using an adaptive learning process. Properties and characteristics of ANNs have made them a common tool when successfully solving high complexity problems from different areas, e.g. medical diagnosis, financial data modelling, predictive microbiology, remote sensing, analytical chemistry... For some of these problems, items have to be classified into naturally ordered classes. They are traditionally handled by conventional methods intended for classification of nominal classes, where the order relation is ignored. This kind of supervised learning problems are referred

to as ordinal regression, where an ordinal scale is used to label the examples. Therefore, in ordinal classification problems, the goal is to learn how to classify examples in the correct class. But one should take into account that the higher distance between predicted and real labels is (with respect to the ordinal scale), the more the misclassification error should be penalised.

In general, Logistic Regression (LR) is a simple and useful classification procedure, although it poses problems when applied to real-problems, where, frequently, we cannot make the assumption of additive and purely linear effects of the covariates. As suggested by [13], an obvious way to generalise the linear logistic regression is to replace the linear predictors with structured non-parametric models such as an additive model of basis function. In this paper, we extend the ideas introduced in [12], where a combination of LR and Neural Networks models was used to solve nominal classification problems. We present an adaptation of the corresponding algorithm to tackle ordinal classification combined with an ordinal ANN model with hybrid basis functions.

As exposed in [19], the motivation for studying hybrid basis functions for neural networks comes from many different branches. Biological neural systems are built from a large diversity of neuron types. We can also think about computational efficiency; neural networks with high diversity performs better [7]. That leads us to a third motivation, the reduction of complexity in the neural network; adding diversity to a neural network allows the number of nodes (and links) to be reduced [25].

One of the first models specifically designed for ordinal classification, and the one our model is built on, is the Proportional Odds Model (POM) [20], which is basically an ordinal logistic regression. This model is based on the assumption of stochastic ordering in the input space, and the use of thresholds to split the projected input space into different ordered classes. We propose a hybrid neural network ordinal model using a combination of projection functions (sigmoidal unit, SU) and kernel functions (radial basis function, RBF) in the hidden layer of a feed-forward neural network [11]. An evolutionary algorithm is adapted to train this model and applied for learning the model architecture, link weights and node topology. In order to obtain further conclusions, the hybrid basis neural network proposed is compared to its corresponding pure models: SU and RBF neural networks. A mixture of different kinds of basis functions [11] is an interesting alternative, which could be able to take advantage from the benefits of each one. Our proposal follows the idea of augmenting the vector of inputs with non-linear covariates obtained by the neural network hybrid hidden layer, and then, use the POM in this new space of derived input features.

The estimation of the coefficients is carried out in several steps. In a first step, an evolutionary algorithm [27] (EA) is applied to design the structure and train the weights of a hybrid SURBF neural network [5, 15] (SURBFNN). Evolutionary computation has been widely used in the late years to evolve NN architectures and weights. There have been many applications for parametric learning [22] and for both parametric and structural learning [1, 17, 26]. This evolutionary process determines the number of neurons in the model and the corresponding variables,

which will be the new covariates in the non-linear LR model. The best model in the last generation is used for that purpose. In a second step, we augment the input space adding the SURBF non-linear covariates to the linear ones. That led us to add a third step, where we perform a local optimization algorithm using the new input covariates, with a maximum likelihood method for ordinal LR, based on the structure of the POM.

The rest of the paper is organized as follows. Section 2 introduces the hybrid neural networks. In Sect. 3, the neural network model for ordinal regression is explained. Section 4 presents the algorithm developed in order to obtain the coefficients for the hybrid model. Section 5 includes the experiments: experimental design, information about the datasets and results of the experiments. Finally, in Sect. 6, we present the conclusions of the paper.

2 Hybrid Artificial Neural Networks

Different types of neural networks are being used today for classification purposes, including neural networks based on a sigmoidal basis (SU), radial basis function (RBF) [15] and a class of multiplicative basis functions, called the product unit (PU) [18, 23]. The combination of different basis functions in the hidden layer of a neural network has been proposed as an alternative to traditional neural networks [16]. We use RBF neurons and SU neurons according to Cohen and Intrator insights [7], based on the duality and complementary properties of projection-based functions (SU and PU) and kernel typology (RBF). These models have also been theoretically justified by Donoho [8], who demonstrated that any continuous function can be decomposed into two mutually exclusive functions, such as radial (RBF) and crest ones (SU and PU). In this way, RBF neurons contribute to a local recognition model [4], while SU neurons contribute to a global recognition one [18]. Their combination results in a high degree of diversity because the submodels differ from one another.

3 Proposed Neural Network Model

The POM model [20], as the majority of existing ordinal regression models, can be represented in the following general form:

$$C(\mathbf{x}) = \begin{cases} \mathcal{C}_1, & \text{if } f(\mathbf{x}, \boldsymbol{\theta}) \leq \beta_0^1 \\ \mathcal{C}_2, & \text{if } \beta_0^1 < f(\mathbf{x}, \boldsymbol{\theta}) \leq \beta_0^2 \\ \dots & \\ \mathcal{C}_J, & \text{if } f(\mathbf{x}, \boldsymbol{\theta}) > \beta_0^{J-1}, \end{cases} \quad (1)$$

where $\beta_0^1 < \beta_0^2 < \dots < \beta_0^{J-1}$ (this will be the most important constraint in order to adapt the nominal classification model to ordinal classification), J is the number of classes, \mathbf{x} is the input pattern to be classified, $f(\mathbf{x}, \boldsymbol{\theta})$ is a ranking function and $\boldsymbol{\theta}$ is the vector of parameters of the model. Indeed, the analysis of

Eq. (1) uncovers the general idea presented in [20]: patterns, \mathbf{x} , are projected to a real line by using the ranking function, $f(\mathbf{x}, \boldsymbol{\theta})$, and the biases or thresholds, β_0^j , are separating the ordered classes.

We are using an adaptation of the POM to artificial neural networks. This adaptation is based on two elements: the first one is a second hidden linear layer with only one node whose inputs are the non-linear transformations of the first hidden layer. The task of this node is to project the values into a line, to make them have an order. After this one node linear layer, an output layer is included with one bias for each class, whose objective is to set the optimum thresholds to classify the patterns in the class they belong to.

The structure of our model is presented in Fig. 1 which has two main parts. The lower one shows the SURBFNN model, where $\mathbf{x} = (x_1, \dots, x_k)$, is the vector of input variables and k is the number of variables in the database. $\boldsymbol{\omega} = (\omega_{1,10}, \dots, \omega_{1,m_1k}, \omega_{2,10}, \dots, \omega_{2,m_2k})$ is the matrix of weights of the connections from the input nodes to the hidden layer SU nodes ($\omega_{1,10}, \dots, \omega_{1,m_1k}$) and to the RBF ones ($\omega_{2,10}, \dots, \omega_{2,m_2k}$) and B are the nodes in the hybrid hidden layer, m_1 is the number of nodes of the first type and m_2 is the number of nodes of the second type, SU and RBF respectively in our case, and “1” is the bias of the layer, which takes part in the calculations.

The upper part of the figure shows a single node in the second hidden layer of the model, which is the one that performs the linear transformation of the POM model. Its result, $f(\mathbf{x}, \boldsymbol{\theta})$, is connected, together with a second bias, to the output layer, where J is the number of classes, and $\beta_0^0, \dots, \beta_0^{J-1}$ are the thresholds for the different classes. These $J-1$ thresholds are able to separate the J classes, but they have to fulfil the order constraint shown in the figure. Finally, the output layer obtains the outputs of the model, $f_j(\mathbf{x}, \boldsymbol{\theta}, \beta_0^j)$, for $j \in \{1, \dots, J-1\}$. These outputs are transformed using the function of the POM model, which transforms them into a probability ($g_j(\mathbf{x}, \boldsymbol{\theta}, \beta_0^j)$). This is the probability that each pattern has to belong to the different classes, and the class with the greatest probability is the one selected by the NN to be the class of the pattern.

4 Estimation of the Coefficients

The methodology proposed is based on the combination of an EA and an ordinal maximum likelihood optimization method. Figure 2 represents the different steps of the algorithm and the models obtained for the experiments. The different steps of the algorithm are now explained:

Step 1: We apply and EA to find the basis functions (SUs and RBFs):

$$\mathbf{B}(\mathbf{x}, \mathbf{W}) = \{B_{1,1}(\mathbf{x}, \mathbf{w}_{m_1}), \dots, B_{1,m_1}(\mathbf{x}, \mathbf{w}_2), \dots, B_{2,1}(\mathbf{x}, \mathbf{w}_2), \dots, B_{2,m_2}(\mathbf{x}, \mathbf{w}_{m_2})\}$$

corresponding to the non-linear part of the hybrid logistic regression model presented in this paper. The NN model for the EA is presented in Fig. 1. The EA begins with a random initial population, and each iteration the population is updated using a population-update algorithm [9]. The population is subject to

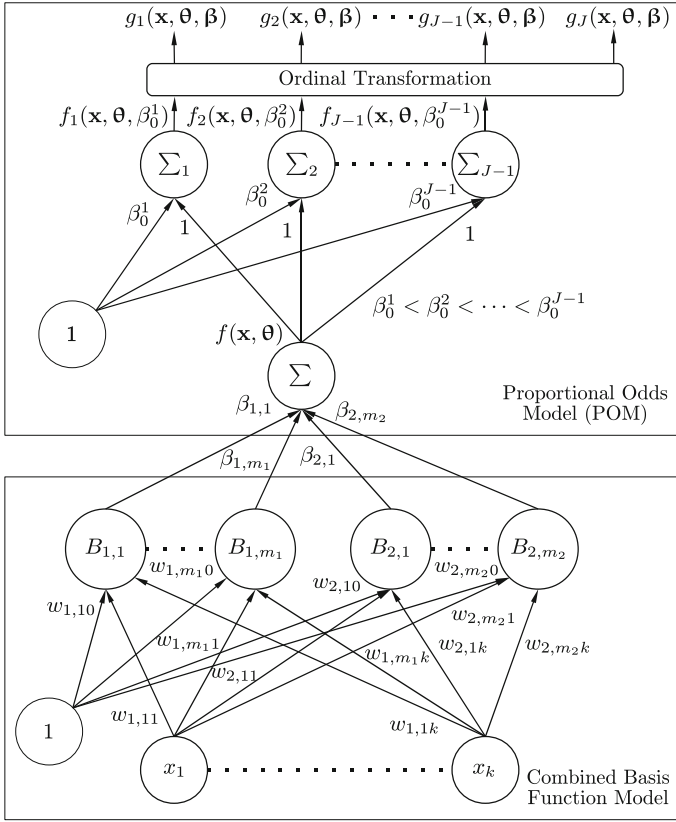


Fig. 1. Neural Network model for ordinal classification

operations of mutation and replication with ordinal constraints. Crossover is not used because of its disadvantages in evolving NNs [1].

Step 2: We perform a transformation of the input space, including the non-linear transformations of the inputs obtained by the EA in Step 1:

$$\begin{aligned} \mathbb{H} : \mathbb{R}^k &\rightarrow \mathbb{R}^{k+m} \\ (x_1, x_2, \dots, x_k) &\rightarrow (x_1, x_2, x_k, \dots, z_1, z_2, \dots, z_m), \end{aligned}$$

where $z_1 = B_1(\mathbf{x}, \mathbf{w}_1), z_2 = B_2(\mathbf{x}, \mathbf{w}_2), \dots, z_m = B_m(\mathbf{x}, \mathbf{w}_m)$.

Step 3: We apply an ordinal maximum likelihood optimization method in the new input space obtained in step 2. The optimization of the maximum likelihood is performed using a gradient descent algorithm called iRProp+ [14], which optimises the non-linear ordinal logistic regression for a defined number of epochs.

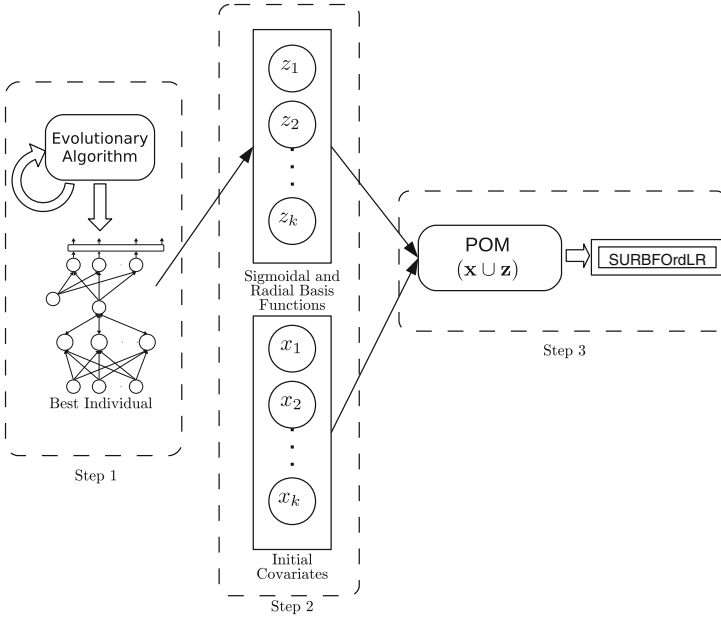


Fig. 2. Steps in the proposed methodology. The different models associated with this methodology are presented in a double squared box

5 Experiments

In order to analyse the performance of the proposed models, ten datasets have been tested, their characteristics being shown in Table 1. The collection of datasets is taken from the UCI [2] and the mldata.org [21] repositories. The experimental design was conducted using 10 random holdout procedures.

Three different neural networks have been compared: RBFOrdLR is the LR combined with the hidden nodes of an evolutionary RBFNN, SUOrdLR is the same model with SU functions on its hidden layer, finally SURBFOrdLR is the LR combined with both SU and RBF from the hybrid hidden layer of an evolutionary neural network. We also compare the results against the original POM model, SVOREX [6] and SVR [24], which is an ordinal regression transformed into an standard regression, changing the ordinal labels (C_1, C_2, \dots), for numbers $(0, 1/(Q - 1), 2/(Q - 1), \dots, Q)$.

All the parameters of the algorithm are common to these ten problems. The main parameters of the algorithm are: number of generations: 100; population size: 250; mutation percentage: 10 %; minimum number of hidden nodes: 4; maximum number of hidden nodes: 14.

In order to set up the minimum number of hidden neurons, a preliminary experiment was done with one partition of each dataset. A 5-fold cross-validation (using only the training split) was done and repeated with the following minimum number of hidden nodes, $\{1, 2, 4, \dots, 20\}$. We concluded that the optimum

minimum number of hidden nodes was 4 and we added 10 nodes for the maximum, in order to give the EA some freedom to optimise the NN.

The idea of having a small population size and a small number of generations is to give less importance to the evolutionary algorithm because its computational time is much higher than the optimisation of the non-linear ordinal logistic regression. After finishing the EA, we took the SURBF hidden layer from the best model in the last generation to augment the input space and applied a gradient descent algorithm with 1000 epochs to optimise the non-linear ordinal logistic regression model.

Table 1. Characteristics of the ten datasets used for the experiments: number of instances (Size), inputs (#In.), classes (#Out.) and patterns per-class (#PPC)

Dataset	Size	#In.	#Out.	#PPC
Bondrate	57	37	5	(6,33,12,5,1)
Balance	625	4	3	(288,49,288)
Contact-lenses	24	6	3	(15,5,4)
Car	1728	21	4	(1210,384,69,65)
ESL	488	4	9	(2,12,38,100,116,135,62,19,4)
LEV	1000	4	5	(93,280,403,197,27)
Newthyroid	215	5	3	(30,150,35)
Pasture	36	25	3	(12,12,12)
Squash-stored	52	51	3	(24,24,4)
SWD	1000	10	4	(32,352,399,217)

The following two measures have been used for comparing the models:

- *CCR*: The Correct Classification Rate (*CCR*) is the rate of correctly classified patterns: $CCR = \frac{1}{n} \sum_{i=1}^N J y_i^* = y_i K$, where y_i is the true label, y_i^* is the predicted label and $J \cdot K$ is a Boolean test which is 1 if the inner condition is true and 0 otherwise.
- *MAE*: The Mean Absolute Error (*MAE*) is the average deviation (number of categories) in absolute value of the predicted class from the true class [3]: $MAE = \frac{1}{n} \sum_{i=1}^N e(\mathbf{x}_i)$, where $e(\mathbf{x}_i) = |\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|$ is the distance between the true and the predicted ranks, $\mathcal{O}(\mathcal{C}_j) = j$. This is a way of evaluating the ordering performance of the classifier.

Table 2 shows the mean test value and standard deviation of the correct classified rate (*CCR*) and the mean absolute error (*MAE*) over the 10 models obtained.

To determine the statistical significance of the rank differences observed for each method in the different datasets, we have carried out a non-parametric Friedman test [10] with the ranking of *CCR* and *MAE* of the best models

Table 2. Generalization results obtained for benchmark datasets

Func	<i>CCR</i> (%)	<i>MAE</i>	<i>CCR</i> (%)	<i>MAE</i>
	Mean \pm SD	Mean \pm SD	Mean \pm SD	Mean \pm SD
	Bondrate		Balance	
SURBFOrdLR	56.66 \pm 5.94	0.5467 \pm 0.0883	<i>96.43 \pm 1.87</i>	<i>0.0388 \pm 0.0225</i>
RBFOrdLR	54.00 \pm 5.97	0.5667 \pm 0.0864	95.09 \pm 1.87	0.0535 \pm 0.0217
SUOrdLR	52.66 \pm 5.08	<i>0.5600 \pm 0.0922</i>	96.17 \pm 1.57	0.0414 \pm 0.0314
POM	34.44 \pm 1.94	0.9467 \pm 0.6868	90.55 \pm 1.85	0.1067 \pm 0.0208
SVOREX	<i>54.66 \pm 8.04</i>	0.6222 \pm 0.1181	99.78 \pm 8.19	0.0021 \pm 0.0081
SVR	54.22 \pm 9.20	0.5933 \pm 0.1424	83.52 \pm 1.47	0.1679 \pm 0.0188
	Contact-lenses		Car	
SURBFOrdLR	<i>65.00 \pm 5.54</i>	0.4800 \pm 0.0883	93.12 \pm 2.60	0.0721 \pm 0.0307
RBFOrdLR	65.00 \pm 5.97	0.4167 \pm 0.0864	88.37 \pm 2.72	0.1254 \pm 0.0335
SUOrdLR	63.33 \pm 5.08	<i>0.4000 \pm 0.0922</i>	90.92 \pm 1.72	0.0949 \pm 0.0200
POM	61.66 \pm 1.94	0.5333 \pm 0.6868	15.74 \pm 3.06	1.4505 \pm 0.5482
SVOREX	64.44 \pm 8.04	0.4833 \pm 0.1181	98.78 \pm 2.24	0.0125 \pm 0.0132
SVR	68.33 \pm 9.20	0.3778 \pm 0.1424	<i>97.30 \pm 1.12</i>	<i>0.0270 \pm 0.0110</i>
	ESL		LEV	
SURBFOrdLR	73.19 \pm 2.39	0.2278 \pm 0.0276	<i>62.49 \pm 2.16</i>	0.4032 \pm 0.0295
RBFOrdLR	<i>72.21 \pm 2.74</i>	<i>0.2893 \pm 0.0304</i>	64.40 \pm 2.21	0.4260 \pm 0.0309
SUOrdLR	71.55 \pm 2.32	0.2975 \pm 0.0344	60.92 \pm 3.73	0.4204 \pm 0.0428
POM	70.54 \pm 3.36	0.3103 \pm 0.0380	62.33 \pm 2.79	0.4093 \pm 0.0303
SVOREX	70.98 \pm 2.37	0.3019 \pm 0.0209	62.45 \pm 2.37	0.4105 \pm 0.0227
SVR	70.38 \pm 3.40	0.3117 \pm 0.0298	62.48 \pm 2.69	<i>0.4086 \pm 0.0288</i>
	Newthyroid		Pasture	
SURBFOrdLR	96.60 \pm 1.30	0.0314 \pm 0.0130	71.11 \pm 5.94	0.2889 \pm 0.0883
RBFOrdLR	95.37 \pm 1.57	0.0462 \pm 0.1573	66.66 \pm 5.97	0.3444 \pm 0.0864
SUOrdLR	96.29 \pm 2.76	0.0370 \pm 0.0276	<i>66.66 \pm 5.08</i>	<i>0.3333 \pm 0.0922</i>
POM	96.22 \pm 2.21	0.3277 \pm 0.0221	49.62 \pm 1.94	0.5818 \pm 0.6868
SVOREX	<i>96.48 \pm 3.02</i>	0.0351 \pm 0.0462	65.18 \pm 8.04	0.3333 \pm 0.1181
SVR	95.55 \pm 6.04	<i>0.0322 \pm 0.0604</i>	66.29 \pm 9.20	0.3222 \pm
	Squash-stored		SWD	
SURBFOrdLR	66.92 \pm 5.94	<i>0.3604 \pm 0.0883</i>	58.88 \pm 1.40	0.4302 \pm 0.0154
RBFOrdLR	<i>66.15 \pm 5.97</i>	0.3584 \pm 0.0864	<i>58.36 \pm 1.23</i>	<i>0.4348 \pm 0.0145</i>
SUOrdLR	65.38 \pm 5.08	0.3615 \pm 0.0922	57.20 \pm 1.80	0.4528 \pm 0.0167
POM	38.20 \pm 1.94	0.8121 \pm 0.6868	56.78 \pm 2.95	0.4501 \pm 0.0304
SVOREX	62.56 \pm 8.04	0.3667 \pm 0.1181	56.78 \pm 2.66	0.4464 \pm 0.0339
SVR	61.28 \pm 9.20	0.3692 \pm 0.1424	56.52 \pm 2.06	0.4513 \pm 0.0316

The best result is shown in bold and the second best in italics

as the test variables. For *CCR*, the test shows that the effect of the method used for classification is statistically significant at a significance level of 10 %, as

the confidence interval is $C_0(0, F_{0.10} = 1.97)$ and the F-distribution statistical value is $F^* = 6.78 \notin C_0$. For MAE , the test concludes the same, obtaining $C_0(0, F_{0.10} = 1.97)$ as confidence interval and $F^* = 4.73 \notin C_0$ as F-distribution variable. Consequently, we reject the null-hypothesis stating that all algorithms perform equally in mean ranking for both variables.

Based on this rejection, the Holm post-hoc test is used to compare all classifiers to each other using both CCR and MAE . The results of the Holm test for $\alpha = 0.10$ can be seen in Table 3, using the corresponding p and α'_{Holm} values. From the results of this test, it can be concluded that SURBFOrdLR obtains a significantly higher ranking of CCR and MAE when compared to the remaining methods, which justifies the proposed method in this paper. As MAE is a metric that needs to be minimised the best ranking is the higher one.

Table 3. Comparison of p -Value and α' for the Holm post-hoc non-parametric tests in CCR and MAE with a $\alpha = 0.1$ (SURBFOrdLR is the control method)

Algorithm	Mean CCR Rank	p -Value $_{CCR}$	α'_{CCR}	Mean MAE Rank	p -Value $_{MAE}$	α'_{MAE}
SURBFOrdLR	1.60	–	–	5.20	–	–
RBFOrdLR	3.25	0.04860	0.05000	3.80	0.01683	0.02500
SUOrdLR	3.45	0.02702	0.03333	3.30	0.02315	0.03333
POM	5.35	0.00001	0.02000	1.75	0.00004	0.02000
SVOREX	3.25	0.04860	0.10000	3.55	0.04860	0.10000
SVR	4.10	0.00281	0.02500	3.40	0.03144	0.05000

6 Conclusions

This work proposes to improve an ordinal linear logistic regression model and transform it into a non-linear one. To this end, the linear model is extended with non-linear covariates from the outputs of the hidden neurons of a hybrid SURBFNN. These neural network models are trained using an evolutionary algorithm that optimizes both its architecture and weights.

Moreover, the coefficients of the ordinal logistic regression model, consisting of the initial covariates and the SURBF non-linear outputs, are estimated by a gradient descent algorithm that tries to optimize the maximum likelihood.

Experiments show that this hybrid approach is promising and generally improves accuracy and order quality, performing better than the corresponding pure models. The model also obtains competitive results when compared to state-of-the-art ordinal classification algorithms.

Acknowledgements. This work has been partially subsidised by the TIN2014-54583-C2-1-R project of the Spanish MINECO, FEDER funds and P11-TIC-7508 project of the “Junta de Andalucía(Spain)”.

References

1. Angeline, P.J., Saunders, G.M., Pollack, J.B.: An evolutionary algorithm that constructs recurrent neural networks. *IEEE Trans. Neural Netw.* **5**(1), 54–65 (1994)
2. Asuncion, A., Newman, D.: UCI machine learning repository (2007). <http://www.ics.uci.edu/mllearn/MLRepository.html>
3. Baccianella, S., Esuli, A., Sebastiani, F.: Evaluation measures for ordinal regression. In: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA 09), Pisa, Italy, December 2009
4. Bishop, C.M.: Improving the generalization properties of radial basis function neural networks. *Neural Comput.* **8**, 579–581 (1991)
5. Buchtala, O., Klimek, M., Sick, B.: Evolutionary optimization of radial basis function classifiers for data mining applications. *IEEE Trans. Neural Netw. Part B* **35**(5), 928–947 (2005)
6. Chu, W., Keerthi, S.S.: Support vector ordinal regression. *Neural Comput.* **19**(3), 792–815 (2007)
7. Cohen, S., Intrator, N.: A hybrid projection-based and radial basis function architecture: initial values and global optimisation. *Pattern Anal. Appl.* **5**, 113–120 (2002)
8. Donoho, D.: Projection-based approximation and a duality with kernel methods. *Ann. Stat.* **5**, 58–106 (1989)
9. Dorado-Moreno, M., Gutiérrez, P.A., Hervás-Martínez, C.: Ordinal classification using hybrid artificial neural networks with projection and kernel basis functions. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part II. LNCS, vol. 7209, pp. 319–330. Springer, Heidelberg (2012)
10. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* **11**, 86–92 (1940)
11. Gutiérrez, P.A., Hervás-Martínez, C., Carbonero-Ruz, M., Fernandez, J.C.: Combined projection and kernel basis functions for classification in evolutionary neural networks. *Neurocomputing* **27**(13–15), 2731–2742 (2009)
12. Gutiérrez, P.A., Hervás-Martínez, C., Martínez-Estudillo, F.J.: Logistic regression by means of evolutionary radial basis function neural networks. *IEEE Trans. Neural Netw.* **22**(2), 246–263 (2011)
13. Hastie, T., Tibshirani, R.: *Generalized Additive Models*. Chapman and Hall, London (1990)
14. Igel, C., Hüsken, M.: Empirical evaluation of the improved rprop learning algorithms. *Neurocomputing* **50**(6), 105–123 (2003)
15. Lee, S.H., Hou, C.L.: An art-based construction of RBF networks. *IEEE Trans. Neural Netw.* **13**(6), 1308–1321 (2002)
16. Lippmann, R.P.: Pattern classification using neural networks. *IEEE Trans. Neural Netw.* **27**, 47–64 (1989)
17. Maniezzo, V.: Genetic evolution of the topology and weight distribution of neural networks. *IEEE Trans. Neural Netw.* **5**, 39–53 (1994)

18. Martínez-Estudillo, A.C., Martínez-Estudillo, F.J., Hervás-Martínez, C., García, N.: Evolutionary product unit based neural networks for regression. *Neural Netw.* **19**(4), 477–486 (2006)
19. Maul, T.: Early experiments with neural diversity machines. *Neurocomputing* **113**, 136–48 (2013)
20. McCullagh, P.: Regression models for ordinal data (with discussion). *J. Roy. Stat. Soc.* **42**(2), 109–142 (1980)
21. PASCAL: Pascal (pattern analysis, statistical modelling and computational learning) machine learning benchmarks repository (2011). <http://mldata.org/>
22. van Rooij, A.J.F., Jain, L.C., Johnson, R.P.: *Neural Networks Training Using Genetic Algorithms*. Series in Machine Perception and Artificial Intelligence, vol. 26. World Scientific, Singapore (1996)
23. Schmitt, M.: On the complexity of computing and learning with multiplicative neural networks. *Neural Comput.* **14**, 241–301 (2001)
24. Smola, A., Scholkopf, B.: A tutorial on support vector regression. *Stat. Comput.* **14**(3), 199–222 (2004)
25. Soltesz, I.: *Diversity in the Neuronal Machine: Order and Variability in Interneuronal Microcircuits*. Oxford University Press, New York (2002)
26. Yao, X., Liu, Y.: A new evolutionary system for evolving artificial neural networks. *IEEE Trans. Neural Netw.* **8**, 694–713 (1997)
27. Yao, X.: Evolving artificial neural networks. *Proc. IEEE* **87**(9), 1423–1447 (1999)