Behaviour of FeRaNGA method for Feature Ranking during learning process using Inductive Modelling

Aleš Pilný[1], Pavel Kordík[1], Miroslav Šnorek[1]

[1]*Dept. of Computer Science and Engineering, Karlovo nám. 13, 121 35 Praha 2, Czech Republic*

pilnya1@fel.cvut.cz, kordikp@fel.cvut.cz, snorek@fel.cvut.cz

Abstract. *Nowadays a Feature Ranking (FR) is commonly used method for obtaining information about a large data sets with various dimensionality. This knowledge can be used in a next step of data processing. Accuracy and a speed of experiments can be improved by this. Our approach is based on Artificial Neural Networks (ANN) instead of classical statistical methods. We obtain the knowledge as a by-product of Niching Genetic Algorithm (NGA) used for creation of a feedforward hybrid neural network called GAME. In this paper we present a behaviour of FeRaNGA (Feature Ranking method using Niching Genetic Algorithm(NGA)) during a learning process, especially in every layer of generated GAME network. We want to answer how important is NGA configuration and processing procedure for FR results because behaviour of GA is nondeterministic and thereby were results of FeRaNGA also indefinitive. This method ranks features depending on a percentage of processing elements that survived a selection process. Processing elements transforms parent input features to an output. The selection process is realized by means of NGA where units connected to the least significant features starve and fade from population. To obtain the best results and to find optimal configuration is behaviour of the FeRaNGA algortithm tested using various parameters of NGA and number of ensemble GAME models on well known artificial data sets.*

Keywords

Inductive modelling, Feature Ranking, Artificial Neural Networks, FeRaNGA, Nitching Genetic Algorithm, FAKE-GAME

1 Introduction

Feature Ranknig is one of the most important preprocessing methods which ranks all input features in correspondence to their relevance. Other methods are Feature Selection(FS) which search for subset of relevant features from an initial set of features [4] and Feature Extraction(FE) methods which creates new subset of features containing information extracted from original set of features. One part of FS methods are embedded methods which select relevant features within a learning process. FeRaNGA, as an embedded method, selects relevant features within learning process of internal parameters (e.g. weights betwen layers of neural networks). This knowledge can be used by choosing inputs in each successive created layer of an ANN. FeRaNGA is type of FS methods(embedded method) but is also able to be FR method and ranks all of features in dependency on its configuration.

During experiments we want to find how are ranks changing between layers - if some redundant or irrelevant features replace the relevant features. Next experiment is focused how can one process the result ranks from more layers e.g. combination of results between layers or models. Finaly we describe the changes in ranks betwen layers.

2 Theoretical Part

Niching methods extend genetic algorithms to domains which require the location of multiple solutions. They promote the formation and maintenance of stable subpopulations in genetic algorithms. We use deterministic crowding method where offspring replaces the most similar parent if it is less fit. The reason is to suppress domination of the most important inputs.

2.1 FeRaNGA

FeRaNGA is novel method for FR presented on ICANN 2008 conference[3]. It is a part of FAKE-GAME data mining tool[1]. In GAME(Group of Adaptive Models Evolution) are ANN constructed layer by layer during the stage and the GAME algorithm proceeds from the MIA algorithm(Multilayered Iterative Algorithm). Both algorithms are from the GMDH family algorithms for inductive modeling known as a Group Method of data Handling introduced by Ivachknenko. The more detailed description can be found in [2].

The GAME engine use the Niching Genetic Algorithm(NGA) for creating of ANN. It helps to promote the formation and maintenance of stable subpopulation in genetic algorithms(GAs). NGA extend GAs to domains that require the location of multiple solution. One of this method is deterministic crowding where the offspring replace the most similar parent with higher fitness [3].

FeRaNGA method ranks features depending on a percentage of processing elements that survived a selection process. Processing elements transforms parent input features to an output. The selection process is realized by means of NGA where units connected to the least significant features starve and fade from population. Results obtained by FeRaNGA from one GAME model of AAN are not so demonstrable because of behaviour of GA and thereby is for improvement stability of ranks used modificated FeRaNGA method - FeRaNGA-n where *n* is number of models from which FeRaNGA is computed. Computed in this case means to determinate median from all created models as a final rank of feature.

3 Data sets overview

During a learning process of an ANN type GAME we used two artificial data sets to compare accuracy of ranks in each layer.

Both artificial data sets were generated to measure the performance of the AMFIS feature selection algorithm [5]. First is Gaussian multivariate data set consists of two clusters of points generated from two different 10th-dimensional normal Gaussian distributions. Class 1 corresponds to points generated from N(0, 1) for each dimension and Class 2 to points generated from N(4, 1). This data set consists of 50 features and 500 samples per class. By construction, features 1-10 are equally relevant, features 11-20 are completely irrelevant and features 21-50 are highly redundant with the first ten features. Ideally, the order of selection should be: at first relevant features 1-10, then the redundant features 21-50, and finally the irrelevant features 11-20.

Second artificial data set is Uniform Hypercube data set which consists of two clusters of points generated from two different 10th-dimensional hypercube $[0, 1]^1 0$, with uniform distribution. The relevant feature vector (f1, f2, . . . , f10) was generated from this hypercube in decreasing order of relevance from feature 1 to 10. A parameter $\alpha = 0.5$ was defined for the relevance of the first feature and a factor $\alpha = 0.8$ for decreasing the relevance of each feature. A pattern belongs to Class 1 if ($f_i < \gamma^{i-1} * \alpha$ / i = 1, . . . , 10), and to Class 2 otherwise. This data set consists of 50 features and 500 samples per class. By construction, features 1-10 are relevant, features 11-20 are completely irrelevant, and features 21-50 are highly redundant with first 10 features. Ideally, the order of selection should be: at first relevant features 1-10 (starting with feature 1 until feature 10 in the last position), then the redundant features 21-50, and finally the irrelevant features 11-20.

4 Experiments

In experiments we used three configuration of FeRaNGA(resp. NGA). First configuration is defaul configuration of FAKE-GAME data mining tool and represents 30 individuals and 15 epochs as a parameters of NGA. Next two configuration are 75 and 150 where the number has the same meaning like by default configuration(75 epochs and individuals, 150 epochs and individuals).

4.1 How can configuration improve the results

In this experiment we tested all configuration on Gaussian data set. Results in next three tables are ranks in all layers from sevnth model which depends on six previous models because of median. These results are obtained from GAME ensemble models in FAKE-GAME tool which allows to make more than one model of ANN. Ranks are in every layer computed like a median from all layers with the same number in previous models.

Results in tables 1, 2 and 3 acknowledges hypothesis that icreasing number of epochs and individuals as a parameters of NGA causes better results of FeRaNGA. Redundant features on first ten positions in tables 1 and 2 are in the best configuration replaced by relevant features

In referenced tables one can see also last row with label *Overall*. It represents results of FeRaNGA method computed over all layers in actual model and is corresponding with confirmed hypothesis. In this results is changing order of ranks all right because first ten features (from one to ten) has the same importance and last thirty features(from 21 to 50) has the same importance as well.

Tab. 1. Results of FeRaNGA-7 on the Gaussian data set with default NGA configuration. At least nine features are in each layer ranked. Except layer nr. 2 are all ranks correct, in layer two are some redundant ranks on position of relevant ranks istead of them and some irrelevant features as well. Zeros in the table means that no more features were used in this layer.

Layer	Important part of ranks of features																			
0	9	1	2	3	4	5	6	7	10	0	0	0	0	0	0	0	0	0	0	0	0
1	9	10	5	6	7	4	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0
2	9	7	3	5	10	4	6	2	1	31	8	28	38	46	20	37	26	29	41	12	39
Overall	9	5	6	7	10	3	4	1	2	0	0	0	0	0	0	0	0	0	0	0	0

Tab. 2. Results of FeRaNGA-7 on the Gaussian data set. Configuration of NGA was 75 epochs and 75 individuals in population. In each layer is ten or more ranks of features. Layer zero has correct ranks of all features. Layer 1 has more than ten ranked features(in correct order) and layer 2 as a last layer contains errors in ranks again.

Layer	Important part of ranks of features																				
0	9	6	1	10	3	7	2	5	8	4	0	0	0	0	0	0	0	0	0	0	0
1	9	1	2	6	10	7	5	3	8	4	31	45	24	25	29	30	33	40	41	49	0
2	3	2	6	9	5	10	8	45	28	33	1	7	47	26	49	50	13	4	37	41	25
Overall	9	6	2	1	10	3	7	5	8	4	45	25	33	41	49	24	31	30	0	0	0

Tab. 3. Results of FeRaNGA-7 on the Gaussian data set. Configuration of NGA was 150 epochs and 150 individuals in population. For all layers are ranks absolutely correct. In layer one is more than ten features. Correct results are also caused by long time for creating of GAME models of ANN.

Layer	Important part of ranks of features																				
0	7	10	2	9	6	3	8	5	1	4	0	0	0	0	0	0	0	0	0	0	0
1	2	7	10	8	6	3	9	5	1	4	25	22	28	37	44	45	0	0	0	0	0
Overall	7	2	10	9	6	3	8	5	1	4	25	22	28	37	44	45	0	0	0	0	0

4.2 Correct ranks from first layer

In this part we focused only on first layer from which ranks were computed. All configuration of NGA were tested on Hypercube data set.

Results for configuration 150 and are not displayed because of absolutely correct ranks(8 features in correct order and all others features were unused exept feature Nr. 9 in models 4, 5 and 6 which have correct order too). In table 4 there are first layer results from two configurations. Every row means ranks from first layer from mentioned model and depends on layer from previous model.

Tab. 4. Results of FeRaNGA-7 on the Hypercube data set with Default and 75 NGA configuration. All displayed ranks are from the first layer of each GAME model of AAN with specified configuration. Only first ten ranks are shown. For default NGA configuration in all models are most of ranks in wrong order and there are also some redundant features instead of relevant ones. Growing number of models from which ranks are computed causes improving of results. Configuration of NGA 75(nr. of epochs and individuals) is for 5, 6 and 7 models able to give correct results of ranks.

Model	NGA configuration																			
	Default										75									
1	1	2	4	34	6	8	26	39	44	0	1	4	2	5	3	6	7	10	8	9
2	1	4	2	26	34	3	5	6	8	10	1	2	4	3	5	6	7	8	10	9
3	1	4	3	5	6	26	44	0	0	0	1	2	4	3	5	6	7	8	0	0
4	1	2	4	3	5	6	44	8	26	0	1	2	4	3	5	6	7	8	9	0
5	1	2	3	4	5	6	8	44	0	0	1	2	3	4	5	6	7	8	0	0
6	1	4	2	3	5	6	44	8	37	0	1	2	3	4	5	6	7	8	9	0
7	1	4	2	3	5	6	8	44	0	0	1	2	3	4	5	6	7	8	9	0

Growing number improve results and in combination with strong configuration of NGA (higher number of epochs and/or individuals) is posible to reach the optimal results of ranks. First layer is more restrictive and give only most important results(features). Therefore results from first layer are suitable for cases when one need select only a few most important features. Penalty for accuracy is time which stronger configuration takes.

From table 4 are obvious dependencies between number of models from which FeRaNGA is computed and ranks correctness. Default NGA configuration of FeRaNGA-7 is less effective to obtain correct ranks without mistakes. On the other hand is this Default configuration more quick.

5 Changes of ranks between layers

Change of ranks importance between layers can be computed from percentual importance of every feature. In table 5 we present changes expressed relatively to kind of feature and to number of this features. Because the number of relevant and irrelevant features(10) is different from redundant features(30) must one respect it and recompute the changes among the sets of features by its number.

Tab. 5. Changes between first and second layer computed from 14 GAME models of ANN on Hypercube data set with 150 NGA configuration(150 epochs and individuals). The Relevant features have dominant position in the first layer. In the second layer irrelevant and redundat features are gaining importance at the expense of the relevant features.

	Gain / loss of importance relatively to kind and number of feature													
Features / Model	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Relevant	-0,35	-0,4	-0,32	-0,2	-0,02	-0,37	-0,32	-0,29	-0,15	-0,25	-0,25	-0,66	-0,43	-0,25
Irelevant	0,03	0,06	0,11	0,05	0,23	0,03	0,02	0,03	0,05	0,13	0,03	0,11	0,12	0,03
Redundant	0,1	0,11	0,07	0,05	0,13	0,11	0,1	0,08	0,03	0,04	0,07	0,18	0,1	0,07

If we compute average gain/loss from table 5 is the loss of importancy 0,3 on one relevant feature. On the other side gain of everyone redundant feature is 0,09 and everyone irrelevant feature is 0,07. Only in four cases ware gain of irrelevant features higher then gain by redundant features(models nr. 3, 5, 9 and 13). This observation demonstrate power of FeRaNGA method in first layer where only a few most important features is ranked and in every next layer this important features loss its importance on behalf of redundant and irrelevant features. Irrelevant features takes less part from importance of relevant features then redundant features.

6 Conclusion

In both cases of data sets there are obvious dependencies between number of ranked features and the order of layer in ANN. With growing number of layers the number of ranked features is also increasing. If there is a necessity to have only a few most significant features one can use results from the first layer of the ANN where only the most important features are ranked and the others are unused. On the other hand if we need ranks of all features is preferable to use results from the last layer of ANN or ranks computed over all layers in the model.

The FeRaNGA algorithm is much more restrictive in the first layer, but also more precise. In all next layers of learning process the NGA alows more processing elements to survive. In this case, also the less important ones.

Growing number of models from which FeRaNGA is computed and stronger configuration of NGA improves results as well. But configuration with higher number of epochs and/or individuals enlarge time necessary to obtain the results. The best way how to obtain correct results in short time is to use forceless or defautl configuration of NGA and compute ranks by using FeRaNGA-n only from first layers in *n* GAME models of ANN.

Acknowledgements

This research is partially supported by the grant Automated Knowledge Extraction (KJB201210701) of the Grant Agency of the Academy of Science of the Czech Republic and the research program "Transdisciplinary Research in the Area of Biomedical Engineering II" (MSM6840770012) sponsored by the Ministry of Education, Youth and Sports of the Czech Republic.

References

[1] P. Kordík. *Fully Automated Knowledge Extraction using Group of Adaptive Models Evolution*. PhD thesis, Czech Technical University in Prague, FEE, Dep. of Comp. Sci. and Computers, FEE, CTU Prague, Czech Republic, September 2006.

[2] H. Madala and A. Ivakhnenko. *Inductive Learning Algorithm for Complex System Modelling*. CRC Press, 1994. Boca Raton.

[3] A. Pilný, P. Kordík, and M. Šnorek. Feature ranking derived from data mining process. pages 889–898, Berlin, 2008. ICANN, Springer-Verlag Berlin Heidelberg 2008.

[4] S. Piramuthu. Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*, 156:483–494, 2004.

[5] M. Tesmer and P. Estevez. Amifs: adaptive feature selection by using mutual information. In *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks*, volume 1, page 308, Dept. of Electr. Eng., Chile Univ., Santiago, Chile, July 2004.