

RBF Network with Genetic Algorithm for Feature Selection

Jasmina Novakovic

Abstract – The aim of this paper is to show the possible improvement of the reliability of classification of RBF networks using genetic algorithms for attribute selection. A disadvantage of RBF networks is that they cannot deal effectively with irrelevant features. Genetic search may filter features leading to reduce dimensionality of the feature space. In our experiments, genetic search improves classification accuracy of RBF network.

Keywords - classification accuracy, feature selection, genetic algorithm, RBF network.

I. INTRODUCTION

Feature selection is an active field in computer science. It has been a fertile field of research and development since 1970's in statistical pattern recognition [1]–[3], machine learning and data mining [4]–[9].

Feature selection is a fundamental problem in many different areas, especially in forecasting, document classification, bioinformatics, and object recognition or in modelling of complex technological processes. For some problems, all features may be important, but for some target concept, only a small subset of features is usually relevant.

Feature selection reduces the dimensionality of feature space, removes redundant, irrelevant, or noisy data. It brings the immediate effects for application: speeding up a data mining algorithm, improving the data quality and thereof the performance of data mining, and increasing the comprehensibility of the mining results.

Feature selection can be defined as a process that chooses a minimum subset of M features from the original set of N features, so that the feature space is optimally reduced according to a certain evaluation criterion. As the dimensionality of a domain expands, the number of feature N increases. Finding the best feature subset is usually intractable [10] and many problem related to feature selection have been shown to be NP-hard [11].

Jasmina Novakovic, Faculty of Computer Science, Megatrend University, Bulevar Umetnosti 29, 11070 Beograd, Serbia, (e-mail: jnovakovic@megatrend.edu.rs).

Feature selection algorithms may be divided into filters [12], [13], wrappers [10] and embedded approaches [4]. Filters methods evaluate quality of selected features, independently from the classification algorithm, while wrapper methods require application of a classifier (which should be trained on a given feature subset) to evaluate this quality. Embedded methods perform feature selection during learning of optimal parameters (for example, neural network weights between the input and the hidden layer).

Some classification algorithms have inherited ability to focus on relevant features and ignore irrelevant ones. Decision trees are primary example of a class of such algorithms [14], [15], but also multi-layer perceptron (MLP) neural networks with strong regularization of the input layer may exclude the irrelevant features in an automatic way [16]. Such methods may also benefit from independent feature selection. On the other hand, some algorithms have no provisions for feature selection. The k -nearest neighbor algorithm is one family of such methods that classify novel examples by retrieving the nearest training example, strongly relying on feature selection methods to remove noisy features.

Researchers have studied various aspects of feature selection. Search is a key topic in the study of feature selection [17] such as search starting points, search directions, and search strategies. Another important aspect is how to measure the goodness of a feature subset [17]. According to class information availability in data, there are supervised feature selection approaches [18], [5] as well as unsupervised feature selection approaches [19], [20], [21], [6].

Section II presents genetic algorithms. Section III describes RBF network. Section IV describes the experiments and results. Section V concludes and gives future investigations.

II. GENETIC ALGORITHMS

Genetic algorithms are adaptive search techniques based on the principles of natural selection in biology [22]. Genetic algorithms employ a population of competing

solutions—evolved over time—to converge to an optimal solution. The solution space is searched in parallel, which helps in avoiding local optima. Solution for feature selection is typically a fixed length binary string representing a feature subset—the value of each position in the string represents the presence or absence of a particular feature. The algorithm is an iterative process where each successive generation is produced by applying genetic operators such as crossover and mutation to the members of the current generation. Crossover combines different features from a pair of subsets into a new subset. Mutation changes some of the values (thus adding or deleting features) in a subset randomly. The application of genetic operators to population members is determined by their fitness. For feature selection, it means how good a feature subset is with respect to an evaluation strategy. If feature subsets are better than they have a greater chance of being selected to form a new subset through crossover or mutation.

An outline of the genetic algorithm is shown in Fig. 1.

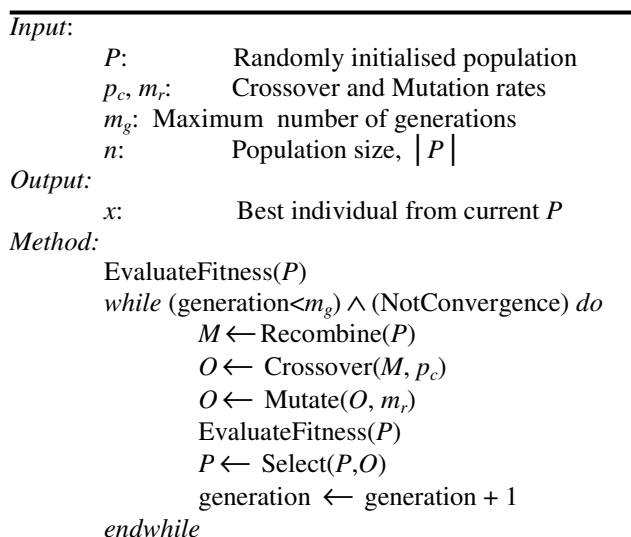


Fig. 1. Genetic algorithm

III. RBF NETWORK

The radial basis function (RBF) network as the supervised learning algorithm is adopted here to build models. This section gives a brief overview of this algorithm.

A popular type of feed forward network is RBF network. RBF network has two layers, not counting the input layer. Each hidden unit essentially represents a particular point in input space, and its output, or activation, for a given instance depends on the distance between its point and the instance—which is just another point. Intuitively, the closer these two points, the stronger the activation. This is achieved by using a nonlinear transformation function to convert the distance into a similarity measure. A bell-shaped Gaussian activation function, whose width may be different for each hidden unit, is commonly used for this purpose. The hidden units are called RBFs because the points in instance space for which a given hidden unit produces the same activation form a hypersphere or hyperellipsoid.

The output layer of an RBF network takes a linear combination of the outputs of the hidden units and—in classification problems—pipes it through the sigmoid function. The parameters that such a network learns are (a) the centers and widths of the RBFs and (b) the weights used to form the linear combination of the outputs obtained from the hidden layer.

One way to determine the first set of parameters is to use clustering, without looking at the class labels of the training instances at all. The simple k-means clustering algorithm can be applied, clustering each class independently to obtain k basis functions for each class. Intuitively, the resulting RBFs represent prototype instances. Then the second set of parameters can be learned, keeping the first parameters fixed. This involves learning a linear model using one of the techniques such as, linear or logistic regression. If there are far fewer hidden units than training instances, this can be done very quickly.

A disadvantage of RBF networks is that they give every feature the same weight because all are treated equally in the distance computation. Hence, they cannot deal effectively with irrelevant features.

IV. EXPERIMENTS AND RESULTS

Five natural domains and three artificial Monk's domains [23] were used for evaluating genetic search with machine learning algorithms, taken from the UCI repository of machine learning databases [24]. These domains were chosen because of (a) their predominance in the literature, and (b) the prevalence of nominal features, thus reducing the need to discretize feature values.

The characteristics of these domains are summarised on Table 1.

TABLE 1: DOMAIN CHARACTERISTICS [24].

Domain	Instances	Features	%	
			Missing	Accuracy
mu	8124	22	1.3	51.8
vote	435	16	5.3	61.4
cr	690	15	0.6	55.5
ly	148	18	0.0	54.7
bc	286	9	0.3	70.3
M1	432	6	0.0	50.0
M2	432	6	0.0	67.1
M3	432	6	0.0	52.8

On Table 1 data sets above the horizontal line are natural domains, those below are artificial. The default accuracy is the accuracy of always predicting the majority class on the whole data set. The % Missing column shows what percentage of the data set's entries (number of features \times number of instances) have missing values. The following is a brief description of the data sets.

Mushroom (mu) This is a large data set containing 8124 instances which includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family [25]. The task is to distinguish edible from poisonous mushrooms on the basis of 22 nominal attributes describing characteristics of the

mushrooms such as the shape of the cap, odour, and gill spacing.

Vote This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key issues such as education spending and immigration. In the original data, there are lists with nine different types of votes. There are 435 (267 democrats, 168 republicans) instances and all features are binary.

Australian credit screening (cr) This file concerns credit card applications. The task is to distinguish credit-worthy from non credit-worthy customers. Data set characteristics is multivariate; feature characteristics are categorical, integer and real.

Lymphography (ly) This is a small medical data set containing 148 instances with 18 nominal features. The task is to distinguish healthy patients from those with metastases or malignant lymphoma. The values for class attribute are normal find, metastases, malign lymph and fibrosis. This is the one of three medical domains (the others being Primary Tumour and Breast Cancer) provided by the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia.

Breast Cancer (bc) The task is to predict whether cancer will recur in patients. There are 9 nominal attributes describing characteristics such as tumour size and location. 286 examples are provided.

Monk's problems The Monk's problems are three artificial domains, each using the same representation, that have been used to compare machine learning algorithms [23]. Monk's domains contain instances of robots described by six nominal features:

- Head-shape \in {round, square, octagon}
- Body-shape \in {round, square, octagon}
- Is-smiling \in {yes, no}
- Holding \in {sword, balloon, flag}
- Jacket-colour \in {red, yellow, green, blue}
- Has-tie \in {yes, no}

There are three Monk's problems. The domains for all Monk's problems are the same with 432 instances. For each problem, the domain has been partitioned into a train and test set.

Monk1 (M1) The concept is:

(head-shape = body-shape) or (jacket-colour = red)

This problem is difficult due to the interaction between the first two features. But, only one value of the jacket-colour feature is useful.

Monk2 (M2) The concept is:

Exactly two of the features have their first value.

This is a hard problem due to the pairwise feature interactions and the fact that only one value of each feature is useful. Note that all six features are relevant to the concept.

Monk3 (M3) The concept is:

(jacket-colour = green and holding = sword) or
(jacket-colour \neq blue and body-shape \neq octagon)

In M3 5% class noise added to the training set. This is the only Monk's problem that is with noise. It is possible to achieve approximately 97% accuracy using only the (jacket-colour \neq blue and body-shape \neq octagon) disjunct.

A. Artificial Domains

The purpose of the experiments described in this section is to empirically test the claim that genetic algorithms can improve the accuracy of classification algorithms. Classification accuracy was estimated using ten-fold crossvalidation with each algorithm on each data set.

The Monk's problems are challenging artificial domains that have been used to compare the performance of machine learning algorithms. This section tests genetic search on the three Monk's problems with domains which are involve irrelevant features, noise, and high degrees of feature interaction. Each problem uses the same representation and has six features: three relevant features in M1 and M3; M2 uses all six features.

TABLE 2: CLASSIFICATION ACCURACY OF RBF NETWORK WITH FULL DATA SETS AND GENETIC APPROACH.

<i>Data set</i>	full	Genetic search
mu	98.5	99.0
vote	94.5	95.4
ly	80.4	79.7
cr	79.1	82.0
bc	71	73.1
M1	44.9	74.1
M2	67.1	65.7
M3	50.9	99.1

Genetic search is able to improve the accuracy of RBF dramatically on M1 and M3 (Table 2). On M1, 74.1% accuracy is achieved with just the jacketcolour feature. On M2, removing features degrades the performance of RBF. On M3, accuracy of RBF significantly improves.

TABLE 3: SELECTED ATTRIBUTES WITH GENETIC SEARCH.

<i>Data set</i>	<i>Genetic search - selected attributes</i> ($m_g = 20$, $n = 20$, $p_c = 0.60$, $m_r = 0.033$)
mu	5,7,8,12,13,18,20
vote	3,4,10,11
ly	1,2,7,8,9,11,13,15,16,18
cr	4,6,8,9,11,14,15
bc	3,4,5,6,9
M1	5
M2	5
M3	2

Table 3 shows the number of features selected by genetic search on the Monk's problems. Genetic search are unable to select all the relevant features for M1, M2 and M3 due to the high order feature interactions. The jacket-colour feature is selected for M1 and is one of the three relevant features in this concept. All six features are relevant for M2 and all interact, but genetic search selects the jacket-colour feature. On M3, genetic search choose body-shape, one of the feature which give the second conjunction of the concept (jacket-colour \neq blue and body-shape \neq octagon).

B. Natural Domains

The results of testing genetic search on five natural domains are described in this section. The purpose of the experiments described in this section is to empirically test the claim that genetic algorithms can improve the accuracy of classification algorithms. The performance of learning algorithms with and without feature selection is taken as an indication of genetic search success in selecting useful features, because the relevant features are often not known in advance for natural domains. Classification accuracy was estimated using ten-fold crossvalidation with each algorithm on each data set.

Table 2 shows for each machine learning algorithm, how many natural domains accuracy was improved and degraded by genetic search with following values of parameters: $m_g = 20$, $n = 20$, $p_c = 0.60$, $m_r = 0.033$. In our experiment, we empirically chose standard parameters for the genetic algorithm, on the basis that in most cases, these parameters give the best results. For RBF genetic search maintains or improves accuracy for four data sets and degrades for one. Genetic search have difficulty on domains with the highest number of classes.

Table 3 shows features selected by genetic search for each data set. On half data sets, genetic search has reduced the number of features by more than half. Evaluation of selecting features is fast.

In our experiment, we use the standard parameters of genetic algorithm, we have selected based on of those parameters that generated the best results in most cases. However, in some cases, we can get better results with different values of these parameters. If we change these parameters, classification accuracy are changed.

The experiments presented in this article show that genetic search's ability to select useful features does carry over from artificial to natural domains.

A disadvantage of RBF networks is that they cannot deal effectively with irrelevant features. For RBF genetic search improves accuracy for six data sets, and only for two data sets degrades accuracy.

V. CONCLUSIONS

Genetic search may filter features leading to reduce dimensionality of the feature space. This is especially effective for classification methods that do not have any inherent feature selections build in, such as the nearest neighbor methods or some types of neural networks. Genetic search have been used for feature selection, evaluated and compared using RBF network as classifier on five real and three artificial benchmark data.

There are many questions and issues that remain to be addressed and that we intend to investigate in future work. Some improvements of the selecting methods presented here are possible. One of these improvements, which will be able to do is choosing the best parameters for each data set separately, to get the greater reliability of classification of RBF network.

REFERENCES

- [1] N. Wyse, R. Dubes and A.K. Jain, "A critical evaluation of intrinsic dimensionality algorithms", In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice*, Morgan Kaufmann Publishers, Inc., 1980, pp. 415-425.
- [2] M. Ben-Bassat, "Pattern recognition and reduction of dimensionality", In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of statistics-II*, North Holland, 1982, pp. 773-791.
- [3] W. Siedlecki and J. Sklansky, "On automatic feature selection", *International Journal of Pattern Recognition and Artificial Intelligence*, 2:197-220, 1988.
- [4] A.I. Blum and P. Langley, "Selection of relevant features and examples in machine learning", *Artificial Intelligence*, vol 97, 1997, pp. 245-271.
- [5] M. Dash and H. Liu, "Feature selection methods for classifications", *Intelligent Data Analysis: An International Journal*, 1(3), 1997. <http://www-east.elsevier.com/ida/free.htm>.
- [6] J. G. Dy and C. E. Brodley, "Feature subset selection and order identification for unsupervised learning". In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 247-254.
- [7] Y. Kim, W. Street and F. Menczer, "Feature selection for unsupervised learning via evolutionary search", In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, pp. 365-369.
- [8] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection", In *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.
- [9] P. Mitra, C. A. Murthy and S. K. Pal, "Unsupervised feature selection using feature similarity", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):301-312, 2002.
- [10] R. Kohavi and G.H. John, "Wrappers for feature Subset Selection", *Artificial Intelligence*, vol. 97, 1997, pp. 273-324.
- [11] A.L. Blum and R.L. Rivest, "Training a 3-node neural networks is NP-complete", *Neural Networks*, 5:117-127, 1992.
- [12] H. Almuallim and T.G. Dietterich, "Learning with many irrelevant features", In: *Proc. AAAI-91*, Anaheim, CA, 1991, pp. 547-552.
- [13] K. Kira and L.A. Rendell, "The feature selection problem: traditional methods and a new algorithm", In: *Proc. AAAI-92*, San Jose, CA, 1992, pp. 122-126.
- [14] L. Breiman, J.H. Friedman, R.H. Olshen and C.J. Stone, *Classification and Regression Trees*, Wadsworth and Brooks, Monterey, CA, 1984.
- [15] J.R. Quinlan, *C4.5: Programs for machine learning*, San Mateo, Morgan Kaufman, 1993.
- [16] W. Duch, R. Adamczak and K. Grabczewski, "A new methodology of extraction, optimization and application of crisp and fuzzy logical rules", *IEEE Transactions on Neural Networks*, vol. 12, 2001, pp. 277-306.
- [17] J. Doak, "An evaluation of feature selection methods and their application to computer security", Technical report, Davis CA: University of California, Department of Computer Science, 1992.
- [18] S.M. Weiss and C.A. Kulikowski, *Computer Systems That Learn*, Morgan Kaufmann Publishers, San Mateo, California, 1991.
- [19] M. Dash, H. Liu and J. Yao, "Dimensionality reduction of unsupervised data", In *Proceedings of the Ninth IEEE International Conference on Tools with AI (ICTAI'97)*, November, 1997, Newport Beach, California, 1997, IEEE Computer Society, pp. 532-539.
- [20] M. Dash and H. Liu, "Handling large unsupervised data via dimensionality reduction", In *Proceedings of 1999 SIGMOD Research Issues in Data Mining and Knowledge Discovery (DMKD-99) Workshop*, 1999.
- [21] L. Talavera, "Feature selection as a preprocessing step for hierarchical clustering", In *Proceedings of International Conference on Machine Learning (ICML'99)*, 1999.
- [22] J. H. Holland, *Adaption in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
- [23] S. B. Thrun, Bala, J. Bloedorn, E. Bratko, I. Cestnik, B. Cheng, J. De Jong, K. Dzeroski, S. Fahlman, S.E. Fisher, D. Hamann, R. Kaufman, K. Keller, S. Kononenko, I. Kreuziger, J. Michalski, R. S. Mitchell, T. Pachowicz, P. Reich, Y. Vafaie, H. Van de Welde, W. Wenzel, W. J. Wnek and J. Zhang, "The MONK's problems: A performance comparison of different learning algorithms", Technical Report CMU-CS-91-197, Carnegie Mellon University, 1991.
- [24] C. J. Merz and P. M. Murphy, *UCI Repository of machine learning databases*, <http://www.ics.uci.edu/mllearn/MLRepository.html>, 1998.
- [25] G. H. Lincoff, *The Audubon Society Field Guide to North American Mushrooms*, Alfred A. Knopf, New York, 1981.