Theory and Applications of Mathematics & Computer Science

(ISSN 2067-2764)
http://www.uav.ro/applications/se/journal/index.php/tamcs

Theory and Applications of Mathematics & Computer Science 1 (2) (2011) 31–41

Wrapper Approach for Feature Selections in RBF Network Classifier

Jasmina Novakovic[a,*]

[a]*Faculty of Public Administration, Megatrend University, Belgrade, Serbia.*

Abstract

In this paper we investigate the impact of wrapper approach on classification accuracy and performance of RBF network. Wrapper approach used six rule induction algorithms for evaluators on supervised learning algorithms RBF network and tested using eight real and three artificial benchmark data sets. Classification accuracy and performance of RBF network depends on evaluators. Our experimental results indicate that every rule induction algorithms in wrapper approach maintains or improves the accuracy of RBF network for more than half data sets. Evaluation of selecting features with wrappers approach is not so fast compare with filters approach.

Keywords: classification accuracy, feature selection, RBF network, rule induction algorithm, wrapper approach.
2000 MSC: 68T01, 97P20, 97R40.

1. Introduction

Feature selection has been a fertile field of research and development since 1970's in statistical pattern recognition, machine learning and data mining. It is a fundamental problem in many different areas, especially in forecasting, document classification, bioinformatics, object recognition or in modelling of complex technological processes. Datasets with thousands of features are not uncommon in such applications. 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.

The main idea of feature selection is to choose a subset of input variables by eliminating features with little or no predictive information. We try to avoid selecting too many or too few features than necessary. If insuficient features are selected, the information content to keep the concept of the data is degraded. If too many features are selected, including redundant or irrelevant features, the classification accuracies may be lower due to the interference of irrelevant information. The goal of feature selection is to find those features

*Corresponding author
Email address: jnovakovic@megatrend.edu.rs (Jasmina Novakovic)

that may neither affect the target in any way (called irrelevant features) nor add anything new to the target (called redundant features) and exclude them.

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 (Kohavi & John, 1997) and many problem related to feature selection have been shown to be NP-hard (Blum & Rivest, 1992).

Algorithms for feature selection may be divided into filters (Almuallim & Dietterich, 1991), (Kira & Rendell, 1992), wrappers (Kohavi & John, 1997) and embedded approaches. 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. The weakness of the filter approach lies in that the selected feature subset may not lead to high performance in induction systems, such as the classification system. The wrapper approach combines data dimension reduction with induction algorithms, but high computational cost is a heavy burden. 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 (Breiman *et al.*, 1984), (Quinlan, 1993), but also multi-layer perceptron (MLP), neural networks with strong regularization of the input layer may exclude the irrelevant features in an automatic way (Duch *et al.*, 2001). 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 relaying on feature selection methods to remove noisy features.

Our research interest includes wrapper approaches for feature selections. Wrapper methods require application of a classifier; in our experiment we use rule induction algorithms. We chose the radial basis function (RBF) network, because it cannot deal effectively with irrelevant features. A disadvantage of RBF network is that it gives every feature the same weight because all are treated equally in the distance computation. The main aim of this paper is to experimentally verify, on benchmark data sets, the impact on performance and classification accuracy on RBF network with wrapper approaches.

This paper is organized as follows. In the next section we briefly describe the wrapper approach. Section 3 gives a brief overview of RBF network as classification algorithm that we use in our experiment. Section 4 presents experimental evaluation. Final section contains discussion of the obtained results and some closing remarks.

2. Wrapper Approach for Feature Selections

Wrapper approach uses the method of classification itself to measure the importance of features set; hence the feature selected depends on the classifier model used. Wrapper methods generally result in better performance than filter methods because the feature selection process is optimized for the classification algorithm to be used. The fact is, wrapper methods are too expensive for large dimensional database in terms of computational complexity and time since each feature set considered must be evaluated with the classifier algorithm used (Dash & Liu, 1997),(Doraisamy *et al.*, 2008), (Saeys *et al.*, 2007).

Kohavi et al. were the first to introduce the wrapper approach to the mainstream data mining community. They successfully used the wrapper approach to search for an optimal feature subset customized to a specific induction learning algorithm and domain. The idea behind the wrapper approach is very simple and is shown

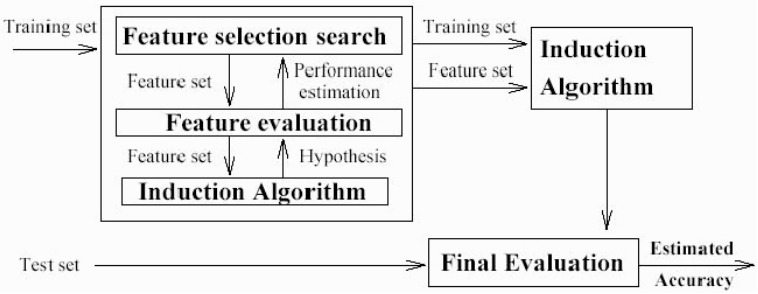


Figure 1: Wrapper approach to variable subset selection based on the incorporation of the learning algorithm (Almuallim & Dietterich, 1991).

in Figure 1. Some performance measure is used to evaluate the classifier built on each feature subset using a set aside distinct portion of the dataset, and the feature subset with the highest evaluation is used as the final set to build the final classifier on all the data instances in the training set. The resulting classifier can then be evaluated on an independent test set that is not used during the search process to assess the efficacy of the wrapper approach in selecting the feature subset.

After Kohavi et al. many researchers experimented with the wrapper approach in various contexts. The wrapper approach in selecting the features for a Nave-Bayes classifier was used by Langley and Sage (Langley & Sage, 1994). Pazzani (Pazzani, 1995) created super-features by combining the base features for a Nave-Bayes classifier by using the wrapper approach and demonstrated that it really was able to find the correct combination of features when they interacted. Singh and Provan (Singh & Provan, 1995) significantly improved the original K2 algorithm when they selected the features for Bayesian networks using the wrapper approach. Kohavi and John (Kohavi & John, 1997) again demonstrated the use of the wrapper with other search methods using probabilistic estimates for feature subset selection.

The wrapper approach has been used for many other problems except than feature selection. The wrapper approach for tweaking the parameters of C4.5 for maximal performance were applied by Kohavi and John (Kohavi & John, 1995). Skalak (Skalak, 1994) used the wrapper approach in an interesting fashion to select the training instances instead of the features in connection with nearest-neighbor classifiers.

3. RBF Network Classifier

RBF network as supervised learning algorithms is adopted here to build models. This section gives a brief overview of this algorithm. This network emerged as variant of artificial neural network in late 80's. RBF network is a popular artificial neural network architecture that has found wide applications in diverse fields of engineering. It is used in function approximation, time series prediction, and control.

RBF network is an artificial neural network that uses radial basis functions as activation functions. This network is a special class of neural networks in which the activation of a hidden neuron is determined by the distance between the input vector and a prototype vector. Prototype vectors refer to centers of clusters obtained during RBF training. Usually, in RBF network three kinds of distance metrics can be used: Euclidean, Manhattan, and Mahalanobis distances.

In Figure 2 is presented architecture of RBF network. An input vector x is used as input to all radial basis functions, each with different parameters. The output of the network is a linear combination of the outputs from radial basis functions.

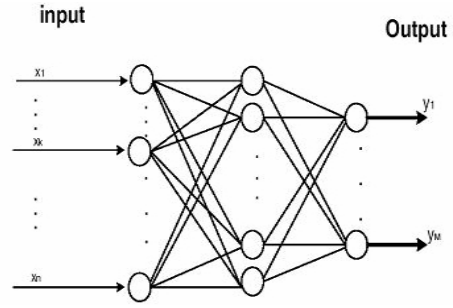


Figure 2. Architecture of RBF network.

RBF network has three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer. The output,$\phi : \mathbb{R}^n \to \mathbb{R}$, of the network is thus

$$\phi(\mathbf{x}) = \sum_{i=1}^{N} a_i \rho(\|\mathbf{x} - \mathbf{c}_i\|) \tag{3.1}$$

where N is the number of neurons in the hidden layer, $\mathbf{c}_i$ is the center vector for neuron i, and a_i are the weights of the linear output neuron. In RBF network, in the basic form, all inputs are connected to each hidden neuron. Typically, the norm is taken to be the Euclidean distance and the basis function is taken to be Gaussian

$$\rho(\|\mathbf{x} - \mathbf{c}_i\|) = \exp\left[-\beta \|\mathbf{x} - \mathbf{c}_i\|^2\right]. \tag{3.2}$$

In RBF network the Gaussian basis functions are local in the sense that

$$\lim_{\|\mathbf{x}\| \to \infty} \rho(\|\mathbf{x} - \mathbf{c}_i\|) = 0 \tag{3.3}$$

i.e. changing parameters of one neuron has only a small effect for input values that are far away from the center of that neuron.

RBF network is universal approximators on a compact subset of $\mathbb{R}^n$, which means that a RBF network with enough hidden neurons can approximate any continuous function with arbitrary precision. The weights a_i, c_i, and β are determined in a manner that optimizes the fit between ϕ and the data.

In training phase, there are three types of parameters that need to be chosen to adapt RBF network for a particular task: the center vectors c_i, the output weights w_i, and the RBF width parameters β_i. In the sequential training of the weights are updated at each time step as data streams in. In RBF network for some tasks it makes sense to define an objective function and select the parameter values that minimize its value. The least squares function is the most common objective function

$$K(\mathbf{w}) \stackrel{def}{=} \sum_{t=1}^{\infty} K_t(\mathbf{w}) \tag{3.4}$$

where

$$K_t(\mathbf{w}) \stackrel{def}{=} [\mathbf{y}(t) - \phi(\mathbf{x}(t), \mathbf{w})]^2 \tag{3.5}$$

The dependence on the weights is explicitly included. Minimization of the least squares objective function by optimal choice of weights optimizes accuracy of fit.

In some cases, multiple objectives, such as smoothness as well as accuracy, must be optimized. If it is, it is useful to optimize a regularized objective function such as

$$H(\mathbf{w}) \stackrel{def}{=} K(\mathbf{w}) + \lambda S(\mathbf{w}) \stackrel{def}{=} \sum_{t=1}^{\infty} H_t(\mathbf{w}) \tag{3.6}$$

where

$$S(\mathbf{w}) \stackrel{def}{=} \sum_{t=1}^{\infty} S_t(\mathbf{w}) \tag{3.7}$$

and

$$H_t(\mathbf{w}) \stackrel{def}{=} K_t(\mathbf{w}) + \lambda S_t(\mathbf{w}) \tag{3.8}$$

where optimization of S maximizes smoothness and λ is known as a regularization parameter.

Three training schemes for RBF network are: one-stage training, two-stage training and three-stage training. In one-stage training procedure, only the weights connecting the hidden layer and the output layer are adjusted through some kind of supervised methods, e.g., minimizing the squared difference between the RBF network's output and the target output. The centers of hidden neurons are subsampled from the set of input vectors (or all data points are used as centers) and, typically, all scaling parameters of hidden neurons are fixed at a predefined real value. Usually, two-stage training is used for constructing RBF network. At the first stage, the hidden layer is constructed by selecting the center and the width for each hidden neuron using various clustering algorithms. At the second stage, the weights between hidden neurons and output neurons are determined, for example by using the linear least square method. In a three-stage training procedure RBF network is adjusted through a further optimization after being trained using a two-stage learning scheme.

In function approximation and classification tasks, generalization and the learning abilities are important issues. If RBF network has as many hidden neurons as the training patterns, RBF network can attain no errors for a given training data set. In that case, the size of the network may be too large when tackling large data sets and the generalization ability of such a large RBF network may be poor. Smaller RBF networks may have better generalization ability; but, too small RBF network will perform poorly on both training and test data sets. It is recommendable to determine a training method which takes the learning ability and the generalization ability into consideration at the same time.

The problem is how to optimally determine the key parameters of RBF classifier. Determination the so-called 'sufficient number of hidden units' is required prior knowledge. Though the number of the training patterns is known in advance, it is not the only element which affects the number of hidden units. The data distribution is another element affecting the architecture of RBF network.

The performance of RBF network depends heavily on the network structure especially the input and hidden neurons. Incorrect input neurons or poorly located RBF centers will induce bias to the fitted network model. The main difficulty in operating RBF network concerns the optimization of the hidden layer. When a dynamic architecture is preferred to a predefined one, the optimization method often consists in a gradient descent algorithm which can get trapped in local minima. Furthermore, the activation function has an influence on the final state of the optimization process.

4. Experimental Results

Natural and artificial domains were used for evaluating wrapper approach with RBF network, taken from the UCI repository of machine learning databases. 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. Table 1 is shown the characteristics of these domains.

Table 1. Domain characteristics.

Domain	Instances	Features	% Missing	Average # Feature Vals	Max/Min # Feature Vals	Default Class Vals	Accuracy
mu	8124	22	1.3	5.3	12/1	2	51.8
vote	435	16	5.3	2.0	2/2	2	61.4
cr	690	15	0.6	4.4	14/2	2	55.5
ly	148	18	0.0	2.9	8/2	4	54.7
pt	339	17	3.7	2.2	3/2	22	24.8
bc	286	9	0.3	4.6	11/2	2	70.3
au	226	69	2.0	2.2	6/2	24	25.2
sb	683	35	9.5	2.8	7/2	19	13.5
M1	432	6	0.0	2.8	4/2	2	50.0
M2	432	6	0.0	2.8	4/2	2	67.1
M3	432	6	0.0	2.8	4/2	2	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 X number of instances) have missing values. Average # Feature Vals and Max/Min # Feature Vals are calculated from the nominal features present in the data sets. The following is a brief description of the data sets.

Mushroom (mu) 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. 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 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) data set 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. Number of instances is 690, number of features is 15, and there are missing values.

Lymphography (ly) 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.

Primary Tumor (pt) data set involves predicting the location of a tumor in the body of a patient on the basis of 17 nominal features. There are 339 instances. There are 22 values for class attribute corresponding to body locations: lung, head & neck, esophasus, thyroid, stomach, duoden & sm.int, colon, rectum, anus, salivary glands, pancreas, gallblader, liver, kidney, bladder, testis, prostate, ovary, corpus uteri, cervix uteri, vagina and breast.

Breast Cancer (bc) data set involves predicting whether cancer will recur in patients. There are 9 nominal attributes describing characteristics such as tumor size and location with 286 examples.

Audiology (au) data set containing 226 instances described by 69 nominal features. The task is to diagnose ear dysfunctions. There are 24 values for classe attribure.

Soybean-large (sb) data set containing 683 instances described by 35 nominal features. The task is to diagnose diseases in soybean plants. Features measure properties of leaves and various plant abnormalities. There are 19 values for class attribute (diseases).

Monk's problems domains are the same for all Monk's problems with 432 instances. There are three Monk's problems. Monk's domains contain instances of robots described by six nominal features:

$Head - shape \in \{round, square, octagon\}$

$Body - shape \in \{round, square, octagon\}$

$Is - smiling \in \{yes, no\}$

$Holding \in \{sword, balloon, flag\}$

$Jacket - colour \in \{red, yellow, green, blue\}$

$Has - tie \in \{yes, no\}$

The concept of Monk1 (M1) 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.

The concept of Monk2 (M2) 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.

The concept of Monk3 (M3) is:

(jacket-colour = green and holding = sword) or

(jacket-colour ≠ blue and body-shape ≠ 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 ≠ blue and body-shape ≠ octagon) disjunct.

The typical goal of supervised learning algorithms is to maximize classification accuracy on unseen test set, so we have adopted this as our goal in guiding the feature subset selection.

In our experiment a normalized Gaussian radial basis function network is used. It uses the k-means clustering algorithm to provide the basis functions and learns either a logistic regression (discrete class problems) or linear regression (numeric class problems) on top of that. In RBF network symmetric multivariate Gaussians are fit to the data from each cluster. If the class is nominal it uses the given number of clusters per class. RBF network standardizes all numeric attributes to zero mean and unit variance. We set RBF network in following way:

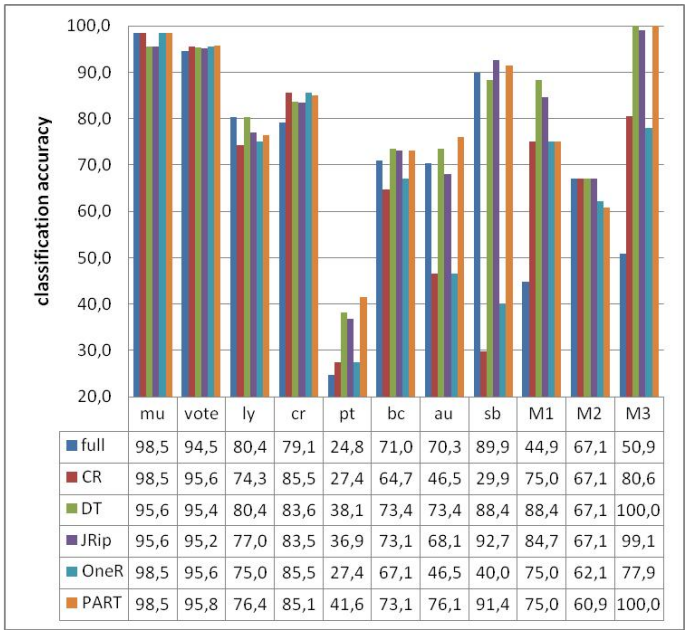
- The random seed to pass on to k-means is set on value 1.
- Maximum number of iterations for the logistic regression to perform is set on value -1.
- The minimum standard deviation for the clusters is set on value 0.1.

- The number of clusters for k-means to generate is set on value 2.
- The ridge value for the logistic or linear regression is set on value 1.0E-8.

Wrapper approach evaluates attribute sets by using a learning scheme. Cross validation is used to estimate the accuracy of the learning scheme for a set of attributes. Five rule induction algorithms are used as classifiers for estimating the accuracy of subsets. These algorithms are: ConjuctiveRule (CR), DecisionTable (DT), JRip, OneR and PART.

CR algorithm implements a single conjunctive rule learner that can predict for numeric and nominal class labels. A rule consists of antecedents "AND"ed together and the consequent (class value) for the classification/regression. In this case, the consequent is the distribution of the available classes (or mean for a numeric value) in the dataset. DT algorithm builds a decision rule using a simple decision table majority classifier. It summarizes the dataset with a 'decision table' which contains the same number of attributes as the original dataset. JRip implements a propositional rule learner - Repeated Incremental Pruning to Produce Error Reduction (Ripper). Ripper builds a ruleset by repeatedly adding rules to an empty ruleset until all positive examples are covered. Rules are formed by greedily adding conditions to the antecedent of a rule (starting with empty antecendent) until no negative examples are covered. After a ruleset is constructed, an optimization postpass massages the ruleset so as to reduce its size and improve its fit to the training data. OneR is a simple algorithm, it builds one rule for each attribute in the training data and then selects the rule with the smallest error rate as its 'one rule'. PART is a separate-and-conquer rule learner, producing sets of rules called 'decision lists' which are ordered set of rules. This algorithm builds a partial C4.5 decision tree in each iteration and makes the "best" leaf into a rule. PART is a combination of C4.5 and Ripper rule learning.

In our experiment, rule induction algorithms are used following settings: number of xval folds to use when estimating subset accuracy is five; seed to use for randomly generating xval splits is set on 1; threshold is set on 0.01.



	mu	vote	ly	cr	pt	bc	au	sb	M1	M2	M3
full	98,5	94,5	80,4	79,1	24,8	71,0	70,3	89,9	44,9	67,1	50,9
CR	98,5	95,6	74,3	85,5	27,4	64,7	46,5	29,9	75,0	67,1	80,6
DT	95,6	95,4	80,4	83,6	38,1	73,4	73,4	88,4	88,4	67,1	100,0
JRip	95,6	95,2	77,0	83,5	36,9	73,1	68,1	92,7	84,7	67,1	99,1
OneR	98,5	95,6	75,0	85,5	27,4	67,1	46,5	40,0	75,0	62,1	77,9
PART	98,5	95,8	76,4	85,1	41,6	73,1	76,1	91,4	75,0	60,9	100,0

Figure 3. Classification acccuracy of RBF network with wrapper approach.

We chose these values of parameters for algorithms, on the basis of those parameters that generated the best results in most cases. But, in some cases, we can get better results with different values of these parameters. If we change these parameters, classification accuracy are changed.

The results of testing wrapper approach on eight natural domains and three artificial domains are described in this section. The purpose of the experiments described in this section is to empirically test the claim that wrapper approach can improve the accuracy of RBF network. The performance of learning algorithm with and without feature selection is taken as an indication of wrapper approach 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 cross validation on each data set.

Table 2. Features selections by wrapper approach.

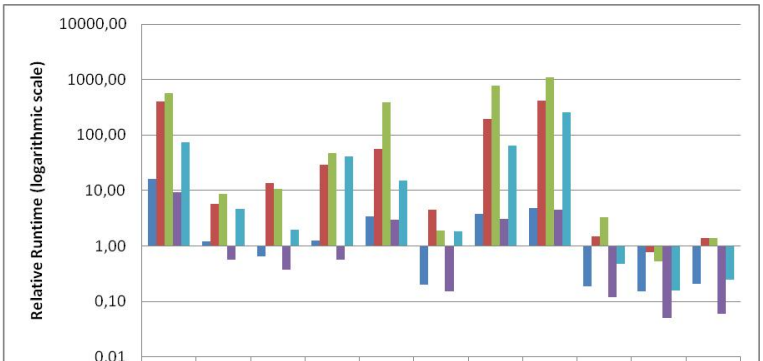
Data set	*CR*	*DT*	*JRip*	*OneR*	*PART*
mu	5	2,3,5,12,20	2,3,5,12,20	5	3,5,8,12,20
vote	4	4	3,4,7,8,16	4	3,4,7,9,11
ly	13	7,12,13,17	1,7,9,13,17	13	7,13,17
cr	9	4,9,10,12	3,9,10,11,14,15	9	2,3,4,5,7,9,10,11,13
pt	15	2,3,5,7,10,13,15	1,2,7,9,10,13,14,15	15	2,3,5,6,8,9,13,15,16
bc	-	5,6	5,6	5	1,5,6
au	1	1,11,15,17,66	1,2,6,7,8,10,11,13,14,15,17,26,27, 39,40,54,55,57,58,63,65,66	1	1,2,6,7,11,15,51,65,66
sb	13	3,12,14,15,16,17, 18,22,28,29	1,3,9,15,17,18,19,22,23,24,26,28, 29,30,31,32,35	29	1,3,9,10,13,14,15,17,18,20,22, 23,26,28,29,35
M1	6	2,3,6	2,3,6,7	6	3
M2	-	-	-	-	-
M3	3	3,5,6	2,3,5,6,7	3	3,5,6

Wrapper approach with CR maintains or improves the accuracy of RBF network for seven data sets and degrades its accuracy for four. For DT wrapper approach maintains or improves accuracy for none data sets and degrades for two. For JRip wrapper approach maintains or improves accuracy for eight data sets and degrades for three. For OneR wrapper approach maintains or improves accuracy for six data sets and degrades for five. For PART wrapper approach maintains or improves accuracy for nine data sets and degrades for two. The best results we have with DT and PART. Wrapper approach is able to improve the accuracy of RBF network dramatically on M1 and M3.

Table 2 shows features selected by wrapper approach for each data set. Rule induction algorithms CR and OneR reduced the number of features the most, for each data set they selected one feature. Only for two natural and two artificial domains some of the implemented rule induction algorithms have not reduced the number of features by more than half.

The experiments presented in this article show that wrapper approach's ability to select useful features does carry over from artificial to natural domains. Analysis of the results on the natural domains has revealed a weakness with RBF network. RBF network with huge data set size, many instances and number of classes, required the long time for processing, with or without feature selection.

The computational efficiency of different wrapper approach is measured in seconds of running time. The experiment was run on AMD Phenom (tm) 9650 Quard-Core Pocessor 2.31 GHz with 4GB RAM. Each configuration is run 10 times, and the average running time is used as a result. The Figure 4 shows the relative runtime (logarithmic scale). As we can see, RBF network had difficulty on domains with the highest number of classes (especially pt, au and sb). For four data sets, with huge data set size, many instances and number of classes (au, sb, pt and mu), very long time was nessesery to calculate the results, with or without feature selection.



	mu	vote	ly	cr	pt	bc	au	sb	M1	M2	M3
Time CR	15,95	1,23	0,65	1,26	3,36	0,20	3,77	4,82	0,19	0,15	0,21
Time DT	402,90	5,74	13,77	29,64	55,55	4,44	197,66	422,74	1,50	0,76	1,40
Time JRip	576,64	8,62	10,78	46,93	388,71	1,91	768,45	1086,8	3,31	0,53	1,40
Time OneR	9,18	0,57	0,37	0,57	3,01	0,15	3,05	4,54	0,12	0,05	0,06
Time PART	75,00	4,68	1,94	41,65	15,22	1,86	63,56	253,33	0,48	0,16	0,25

Figure 4. Time taken to build model (seconds).

5. Conclusions

A disadvantage of RBF networks is that they cannot deal effectively with irrelevant features. Wrapper approach may filter features leading to reduce dimensionality of the feature space. Wrapper approach with different rule induction algorithms have been used for feature selection, evaluated and compared using RBF network as classifiers on eight real and three artificial benchmark data.

Every rule induction algorithms in wrapper approach maintains or improves the accuracy of RBF network for more than half data sets. The best results we have with DT and PART, they maintains or improves the accuracy of RBF network for nine data sets, and only degrades for two. Wrapper approach is able to improve the accuracy of RBF network dramatically on M1 and M3. But, evaluation of selecting features is not fast, compare to filter approach.

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. The algorithms and data sets will be selected according to precise criteria: classify algorithms and several data sets, either real or artificial, with nominal, binary and continuous features. These conclusions and recommendations will be tested on larger data sets using various classification algorithms in the near future.

References

Almuallim, Hussein and Thomas G. Dietterich (1991). Learning with many irrelevant features. In: *In Proceedings of the Ninth National Conference on Artificial Intelligence*. AAAI Press. pp. 547–552.

Blum, A. L. and R. L. Rivest (1992). Training a 3-node neural networks is np-complete. *Neural Networks* **5**, 117–127.

Breiman, L., J. Friedman, R. Olshen and C. Stone (1984). *Classification and Regression Trees*. Wadsworth and Brooks. Monterey, CA.

Dash, M. and H. Liu (1997). Feature selection for classification. *Intelligent Data Analysis* **1**, 131–156.

Doraisamy, Shyamala, Shahram Golzari, Noris Mohd. Norowi, Md Nasir Sulaiman and Nur Izura Udzir (2008). A study on feature selection and classification techniques for automatic genre classification of traditional malay music.. In: *ISMIR* (Juan Pablo Bello, Elaine Chew and Douglas Turnbull, Eds.). pp. 331–336.

Duch, W., R. Adamczak and K. Grabczewski (2001). A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. *IEEE Transactions on Neural Networks* **12**(2), 277 – 306.

Kira, Kenji and Larry A. Rendell (1992). The Feature Selection Problem: Traditional Methods and a New Algorithm.. In: *AAAI*. AAAI Press and MIT Press. Cambridge, MA, USA. pp. 129–134.

Kohavi, R. and G. H. John (1997). Wrappers for feature subset selection. *Artificial Intelligence* **97**, 273–324.

Kohavi, Ron and George H. John (1995). Automatic parameter selection by minimizing estimated error. In: *In Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann. pp. 304–312.

Langley, Pat and Stephanie Sage (1994). Induction of selective bayesian classifiers. In: *Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann. pp. 399–406.

Pazzani, M. (1995). Searching for attribute dependencies in bayesian classifiers. In: *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*. pp. 112–128.

Quinlan, J. Ross (1993). *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. 1 ed.. Morgan Kaufmann.

Saeys, Yvan, Iñaki Inza and Pedro Larrañaga (2007). A review of feature selection techniques in bioinformatics.. *Bioinformatics (Oxford, England)* **23**(19), 2507–2517.

Singh, Moninder and Gregory M. Provan (1995). A comparison of induction algorithms for selective and non-selective bayesian classifiers. In: *Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann. pp. 497–505.

Skalak, David B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann. pp. 293–301.