Performance Evaluation of Feature Subset Selection Approaches on Rule-Based Learning Algorithms

Ali Ozturk[1,2]
[1] *KTO Karatay University, Konya, Turkey*
[2] *Havelsan Inc., Ankara, Turkey*

***Abstract*— There are two main approaches for feature subset selection, i.e., wrapper and filter based. In wrapper based approach, which is a supervised method, the feature subset selection algorithm acts as a wrapper around an induction algorithm. The induction algorithm is actually a black-box for the feature subset selection algorithm and is mostly the classifier itself. The filter approach is an unsupervised method and attempts to assess the merits of features from the data while ignoring the performance of the induction algorithm. In this study, the effects of the feature subset selection approaches on the classification performance of rule-based learning algorithms, i.e., C4.5, RIPPER, PART, BFTree were investigated. These algorithms are fast in case of wrapper based approach. For various datasets, significant accuracy improvements were achieved with the wrapper based feature subset selection method. Other algorithms like Multilayer Perceptron (MLP) and Random Forests (RF) were also applied on the same datasets for the purpose of accuracy comparison. These two algorithms were very inefficient in terms of time when they were used in wrapper approach.**

***Keywords*— Rule-based learning, feature extraction, wrapper, filtering.**

I. INTRODUCTION

There are many benefits of feature selection in machine learning, i.e., reduced amount of data to be learned, improved prediction accuracy, more compact and understandable knowledge representation and reduced execution time. The prediction accuracy of the machine learning algorithms degrades when many features that are irrelevant for prediction are used [1]. Some of the algorithms may be robust to irrelevant features but their prediction performance degrades if there are correlated features even if they are relevant [2]. Another problem with many irrelevant features is the increase in model building time and prediction time of the algorithms.

The feature subset selection problem is defined as finding a subset of existing features on which running an induction algorithm gives a classifier with the highest possible accuracy. However, since the datasets are not infinite in practical we don't know the underlying distribution of the data and therefore the accuracy estimation of the classifier is restricted to our data set [3]. There are two different feature subset selection approaches: wrapper based and filter based. In wrapper approach, the feature subset selection algorithm is used for searching a relevant subset using the classifier itself as the feature subset evaluator. This approach does not require the knowledge of internal implementation of the induction algorithm, only the interface is necessary. In filter approach, the feature subset selection is done as a pre-processing step before training the classifier. The main drawback here is ignoring the performance of the learning algorithm on the selected subset. The filtering methods attempt to evaluate a relevant subset of features by using heuristics based on the merits of data.

Forman [4] evaluated twelve different filter feature selection methods for text classification using a linear Support Vector Machine (SVM) classifier. The information gain was found the best method among twelve methods. Liu et al. [5] tested five different filter-based feature selection methods on different datasets of documents and proposed an iterative feature selection method using expectation maximization. Mustra et al. [6] compared five different wrapper feature selection methods with three different classifiers on mammographic images. The best-first search with forward selection and backward selection methods were found the best ones among the five. The results were improved as much as 12 percent after feature selection. Abusamra et al. [7] analysed the classification performance of three different classifiers on microarray gene expression data by applying eight different filter feature selection methods. They found that the best filter method changed depending on the data set and classifier and some of the filter methods degraded the classification accuracy. Liu et al. [8] applied various feature selection approaches to improve the accuracy of fault diagnosis in industrial applications. They extracted the feature set by obtaining geometrical similarity measures and mutual information from the data of numerous sensors. They used SVM and neural networks as wrappers, then compared the results with distance-based and entropy-based feature selection. The wrapper-based approach gave the best result.

In this study, the effects of the feature subset selection approaches on the classification performance of C4.5, RIPPER, PART and BFTree algorithms were investigated. Significant accuracy improvements were achieved with the wrapper based feature subset selection method on various data sets obtained from UCLA data repository [9]. For comparison purposes, Multilayer Perceptron and Random Forests algorithms were also applied on the same datasets.

Manuscript received November 27, 2018; accepted December 21, 2018.
*Corresponding author: ali.ozturk@karatay.edu.tr

II. MATERIALS AND METHODS

The rule-based learners allow the knowledge extracted from a dataset to be represented as if-then rules that are easy for domain experts to understand. By this way, there is more chance to analyse and validate the extracted knowledge. The general strategy of rule-based learning algorithms is generating a decision tree, transforming it into a rule set and then simplifying the rules to find a more accurate tree [10]. The C4.5, RIPPER and BFTree perform a global optimization process on the set of initially induced rules. BFTree use reduced-error pruning process [11] which is complex and time consuming. For C4.5, reduced-error pruning or its own default pruning strategy can be used. In RIPPER [12], a rule is immediately pruned after growing which is an incremental pruning scheme to reduce error. On the other hand, PART [10] performs rule-induction by producing accurate and compact rule set without using making global optimization. It adopts the idea of building a partial tree rather than a fully explored one. A single rule is extracted from a partial tree whenever it is built. Then, the best leaf of a small set of subtrees is identified. The leaves of these subtrees correspond to different rules. The best leaf is determined by choosing the one which covers the greatest number of instances or the one with the lowest error rate.

A. C4.5

Five distinct stages are necessary in C4.5 algorithm [13]. After creating a decision tree based on information gain criteria, the C4.5 algorithm first transforms the decision tree into a rule set. Then, the rules are simplified by deleting their conditions to obtain minimum error rate. Then, a rule subset is found using minimum description length (MDL) principle with the current rules [14]. The resulting rule set may contain overlapping rules and may not cover all classes. Therefore, when a class is covered more than one rule, a decision is made on which rule to apply and a default rule is added for uncovered classes. Finally, rules are deleted from the resulting rule set with a greedy algorithm in order to decrease the error on training data.

B. RIPPER

RIPPER implements a divide-and-conquer strategy by generating rules one by one. Then, it removes the instances covered by those rules and induces further rules for the remaining instances in an iterated manner [12]. RIPPER employs pruning by putting some training data aside to determine when to drop the tail of a rule. The stopping criterion is based on the heuristic of MDL principle as in C4.5. In the rule induction step, the rules are replaced or revised based on the error of the new rule set on the pruning data. RIPPER has a drawback called hasty generalization [10] which happens due to the interaction of the pruning with the heuristic used in the algorithm.

C. PART

Unlike C4.5 and RIPPER, PART algorithm does not perform global optimization to produce the reduced rule sets. The basic idea is building a partial tree for obtaining a single rule rather than building a full tree which contains lots of rules. The partial tree is a stable subtree which can't be simplified any further. The algorithm still employs divide-and-conquer strategy, however a pruned decision tree is built for current data set to obtain a single rule. A rule is made by the leaf which covers most portion of the data set and then the currently constructed tree is discarded [10].

D. BFTree

In BFTree algorithm, the best node is expanded first as compared to other depth-first decision tree algorithms discussed above [15]. The best node is the one which leads to the maximum information gain among all nodes. Although the building order is different, the resulting tree is the same. The BFTree used in this study uses reduced-error post-pruning method to avoid overfitting.

E. Multilayer Perceptron and Random Forests

Multilayer Perceptron Neural Networks with back propagation are feed-forward networks where modification on the weights is based on the difference between the computed and actual values of the output nodes. The main idea is to minimize the mean squared error between the actual and computed output values in an iterative manner. Stopping criterion is the reduction of the total network error to a predefined level. They are easy to program, well-suited to most of the classification and regression problems, robust against noise [16]. However, they generally require too much time to converge. For this reason, they show very poor performance when used as an induction algorithm in wrapper based feature subset selection.

Random Forests algorithm builds an ensemble of decision trees and generally trained with bagging method which increases the overall performance [17]. The algorithm adds additional randomness while growing the tree. When it is deciding to split a node, it searches for the best feature among a subset of features which are chosen randomly rather than searching the most important feature. This mechanism generally gives better results. Random Forests generally avoids overfitting by creating random subset of features and building smaller trees using these subsets. Then, the algorithm combines these subtrees into a forest. However, the computation may be slower depending on how many subtrees the algorithm builds. This is why using random forests as an inducer in wrapper based feature subset selection is not feasible.

F. Principal Component Analysis (PCA)

The methods such as PCA, Linear Discriminant Analysis (LDA) and Multidimensional Scaling transform the original feature set into a new set to discover more meaningful information [18]. PCA is a mathematical method that transforms a number of possibly correlated variables into a smaller subset of uncorrelated variables while still preserving most of the information [19]. PCA is the most common unsupervised dimensionality reduction and feature extraction technique [20] and is also known as Karhunen-Loeve transform [21]. It is also used for data compression and data visualization. PCA seeks a linear combination of variables where maximum variance is extracted from the original variables. This process is repeated continuously by removing currently obtained variance and seeking another linear combination which corresponds to the maximum remaining variance. The finally obtained principal components are linear combinations of the original variables weighted by their contribution to the variance in a particular

orthogonal dimension.

PCA first finds the covariance matrix of the data set. Then the eigenvectors and their eigenvalues are calculated from the obtained covariance matrix. After sorting the eigenvectors in descending order according to their eigenvalues, first k eigenvectors are chosen as the new k dimensions for the data set. By this way, the original n dimensional data set is transformed into k dimensions.

G. The Data Sets

The datasets used in this study are labor, sonar and ionosphere data which were obtained from UCLA data repository [9].

TABLE I. THE DATASETS AND THEIR PROPERTIES AS USED IN THIS STUDY.

Dataset	Num. of instances	Numeric attributes	Nominal Attributes	Classes
Ionosphere	351	34	0	2
Labor	57	8	8	2
Sonar	208	60	0	2

The properties of these datasets are given in Table 1. The ionosphere and sonar data contains all numeric attributes while labor data has equal number of numeric and nominal attributes. The instances in the data sets belong to one of two classes.

III. EXPERIMENTAL RESULTS

In this study, Weka software package [22] has been used for the evaluation of the feature subset selection approach on the classification algorithms mentioned in the previous section.

We used the default pruning strategy of the C4.5 algorithm instead of reduced-error pruning due to its better performance. The confidence factor used for pruning was 0.25 which affects the amount of pruning. Smaller values mean more pruning while bigger values degrade the performance of the resultant decision tree. The minimum number of instances for each leaf was used as 2.

The first hyper parameter of the RIPPER algorithm determines the total number of the folds to be used for pruning. We defined the fold number as 3 which means one third of the data was used for pruning and the remaining was used for growing the decision tree. The second parameter which was set as 2.0 in this study defines the minimum total weight of the examples of the data set in a rule. RIPPER performs optimization runs based on MDL principle after obtaining the rule set by incremental growing-and-pruning strategy. Two optimization runs were made for each data set in the study.

For PART implementation, the minimum number of instances to produce a rule was set as 2 and the confidence factor was used as 0.25. The C4.5 pruning strategy was chosen instead of reduce-error pruning.

We used the minimum 2 number of instances for the leaf nodes in BFTree implementation. The Gini index [22] was used instead of information as the splitting criterion in the nodes while growing the tree. There were 5 number of folds for internal cross validation during pruning and the post-pruning strategy was used instead of pre-pruning.

In the study, Multilayer Perceptron with single hidden layer was used with learning rate 0.3 and momentum 0.2. The total number of hidden layer neurons was 18, 14 and 32 for ionosphere data, labor data and sonar data, respectively. The number of epochs used for training was 500.

In Random Forests implementation the bag size was set as to use the whole training set. The out-of-bag error calculations were ignored to improve the response time. No limitation was set for the depth of the tree while building the forest and the total number of iterations was 100.

The classification performance of rule-based classifiers along with MLP and RF without feature subset selection is given in Table 2. All the algorithms were evaluated with 10-fold cross validation. According to the results, although RF outperformed the other algorithms for Ionosphere data, the performance of C4.5 and PART were better than MLP. There is equal number of nominal and numeric attributes in Labor data, which leads degradation in the performance of the algorithms. For Labor data RF and MLP outperformed the rule-based classifiers, where RF was better than MLP. The accuracies of MLP, RF and PART are close for Sonar data and higher than the remaining methods, where MLP was the best one among these three algorithms. Since, the number of features is high in Sonar data, the performance of all algorithms degrades.

TABLE II. THE PERFORMANCE OF THE CLASSIFIERS WITHOUT FEATURE SUBSET SELECTION.

Dataset	C4.5	RIPPER	PART	BFTree	MLP	RF
Ionosphere	91.5%	89.7%	91.7%	90.0%	91.2%	92.9%
Labor	73.7%	77.1%	78.9%	78.9%	85.9%	89.5%
Sonar	71.2%	73.1%	80.3%	71.6%	82.2%	81.3%

The results given in Table 2 can also be seen in Figure 1, visually.

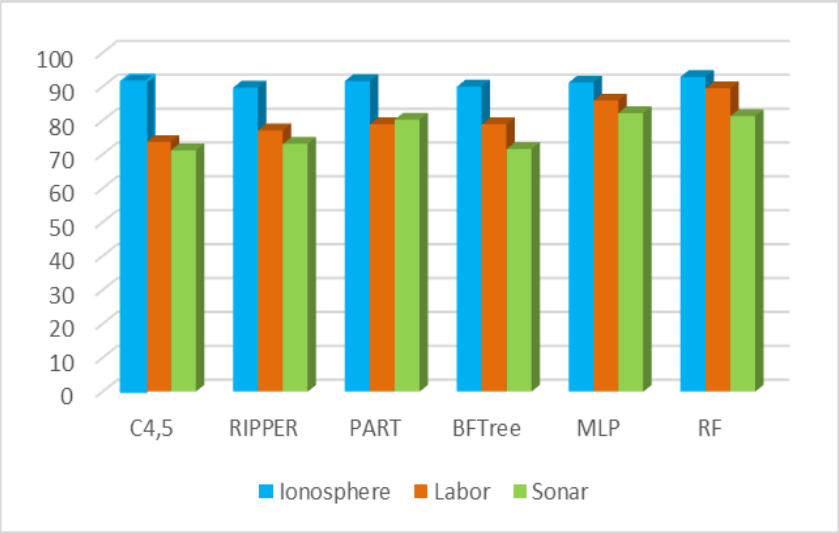

Fig. 1. Visual representation of the performance of the classifiers without feature subset selection.

Table 3 summarizes the classification performance of rule-based classifiers with wrapper-based feature subset selection. The usage of MLP and RF as induction algorithm was infeasible in wrapper-based approach, therefore they were excluded from the results. As can be seen from the table, the performance of PART algorithm was degraded only for sonar data. The performance of other algorithms increased obviously for all data sets. For Ionosphere data, all of the rule-based classifiers outperformed RF and MLP after wrapper-based feature selection had been applied. RIPPER and BFTree reached the performance of RF for Labor data. In case of Sonar data, although the performance of the rule-based classifiers significantly improved except PART, they could not catch RF and MLP. Only C4.5 came close to their performance. This is due to the high number of features in Sonar data.

TABLE III. THE PERFORMANCE OF RULE-BASED CLASSIFIERS WITH WRAPPER-BASED FEATURE SUBSET SELECTION.

Dataset	C4.5	RIPPER	PART	BFTree
Ionosphere	94.0%	91.2%	93.7%	93.5%
Labor	80.7%	89.5%	87.7%	89.5%
Sonar	79.3%	77.4%	71.2%	75.0%

The results given in Table 3 can also be seen in Figure 2, visually.

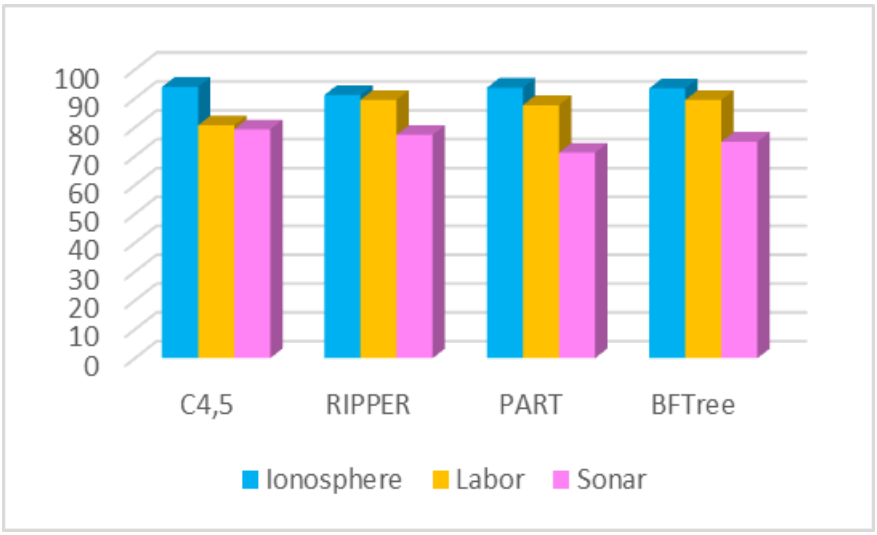

Fig. 2. Visual representation of the performance of rule-based classifiers with wrapper-based feature subset selection.

The performance of rule-based classifiers along with MLP and RF after PCA feature subset selection is given in Table 4.

TABLE IV. THE PERFORMANCE OF THE CLASSIFIERS AFTER PCA FEATURE SUBSET SELECTION.

Dataset	C4.5	RIPPER	PART	BFTree	MLP	RF
Ionosphere	90%	89.2%	89.2%	90%	90.3%	93.7%
Labor	93%	93%	93%	91.2%	84.2%	89.5%
Sonar	73.6%	72.1%	69.7%	75%	75%	77.4%

As can be seen from the Table 4, the performance of PART and MLP severely degrades for Sonar data after application of PCA. The performance of RF and RIPPER also degrades although they are not as bad as the former ones. Figure 3 gives a visual representation of the results given in Table 4.

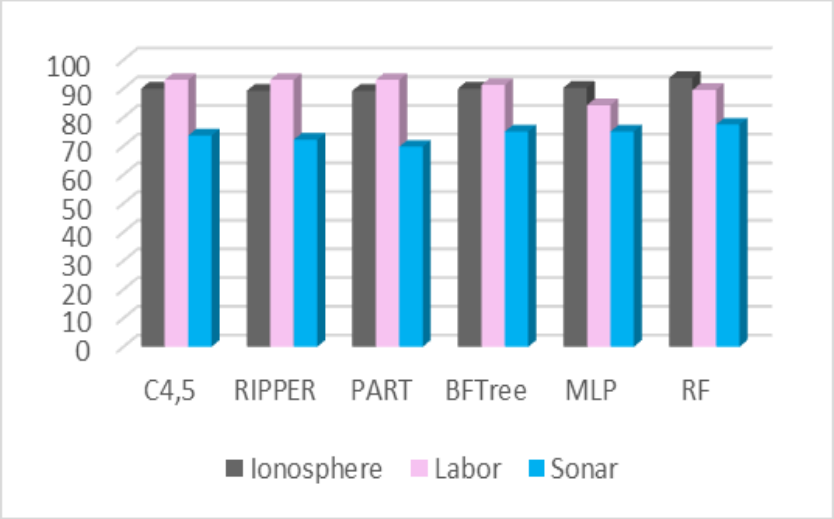

Fig. 3. Visual representation of the performance of the classifiers after PCA feature subset selection.

Wrapper-based feature selection gave better performance than PCA for rule-based learners generally. Only the performance of BFTree remained the same for each type of feature selection approach. For Labor data, PCA outperformed wrapper-based approach when it is used with rule-based classifiers. MLP performance degraded after PCA application and RF was not affected. PCA showed poor performance in Ionosphere data for rule-based classifiers than wrapper-based approach. Moreover, it has negative effect on the accuracy of these algorithms except BFTree. After the application of PCA there is also degradation in the performance of MLP, while RF performed better.

REFERENCES

[1] H. Almuallim and T.G. Dietterich, “Learning Boolean concepts in the presence of many irrelevant features”, *Artificial Intelligence*, vol. 69, pp. 279-306, 1994.

[2] G. John, R. Kohavi and K. Pfleger, “Irrelevant features and the subset selection problem”, in *Proc. 5th International Conference on Machine Learning*, New Brunswick, NJ, 1994, pp. 121-129.

[3] R. Kohavi and G. John, “Wrappers for feature subset selection”, *Artificial Intelligence*, vol. 97, pp. 273-324, 1997.

[4] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *J. Mach. Learn. Res*., vol. 3, pp. 1289–1305, 2003.

[5] T. Liu, S. Liu, and Z. Chen, "An evaluation on feature selection for text clustering", in Proc. *20th International Conference on Machine Learning (ICML-2003)*, Washington DC, USA, AAAI Press, pp. 488–495, 2003.

[6] M. Mustra, M. Grgic, and K. Delac, "Breast density classification using multiple feature selection", *Automatika*, vol. 53, pp. 1289–1305, 2012.

[7] H. Abusamra, "A comparative study of feature selection and classification methods for gene expression data of glioma", *Procedia Computer Science*, vol. 23, pp. 5–14, 2013.

[8] C. Liu, D. Jiang, and W. Yang, "Global geometric similarity scheme for feature selection in fault diagnosis", *Expert Systems with Applications*, vol. 41, issue 8, pp. 3585–3595, 2014.

[9] M. Lichman, “UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science”, 2013.

[10] E. Frank and H. I. Witten, “Generating Accurate Rule Sets Without Global Optimization”, in Proc. *15th Int. Conf. on Machine Learning*, pp. 144-151.

[11] J.R. Quinlan, “Simplifying decision trees”, *Int. Journal of Man-Machine Studies*, vol. 12, pp. 221-234, 1987.

[12] W.W. Cohen, “Fast Effective Rule Induction”, in *Proc. 12th Int. Conf. on Machine Learning*, 1995, pp. 115-123.

[13] J.R. Quinlan, “C4.5: Programs for Machine Learning”, *Machine Learning*, vol. 16, pp. 235-240, 1994.

[14] J.R. Quinlan, “MDL and categorical theories (continued)”, in *Proc. 12th Int. Conf. on Machine Learning*, 1995, pp. 464-470.

[15] J. Friedman, T. Hastie and R. Tibshirani, “Additive logistic regression: A statistical view of boosting”, *Annals of Statistics*, vol. 28(2), pp. 337-407, 2000.

[16] A. Ozturk and R. Seherli, “Nonlinear Short-term Prediction of Aluminum Foil Thickness via Global Regressor Combination”, *Applied Artificial Intelligence*, vol. 31(7-8), pp. 568-592, 2017.

[17] L. Breiman, “Random Forests”, *Machine Learning*, vol. 45(1), pp. 5-32, 2001.

[18] J. Tang, S. Alelyani, and H. Liu, "Feature Selection for Classification: A Review," in: C. Aggarwal (ed.), Data Classification: Algorithms and Applications. CRC Press, 2014.

[19] I.T. Jolliffe, *Principal Component Analysis*, John Wiley & Sons, 2002.

[20] A. Tharwat, “Principal component analysis - a tutorial”, *Int. J. App. Pattern Recognition*, vol. 3, pp. 197-238, 2016.

[21] C.M. Bishop, *Pattern Recognition and Machine Learning*, (Singapore – Springer), 2006.

[22] Eibe Frank, Mark A. Hall, and Ian H. Witten. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, 4th Edition, 2016.

[23] L. Rokach and O. Maimon, “Decision Trees”, *The Data Mining and Knowledge Discovery Handbook*, 2005, ch. 9, pp 165-192.