Feature Selection for Regression Problems

M. Karagiannopoulos, D. Anyfantis, S. B. Kotsiantis, P. E. Pintelas

Abstract-- **Feature subset selection is the process of identifying and removing from a training data set as much irrelevant and redundant features as possible. This reduces the dimensionality of the data and may enable regression algorithms to operate faster and more effectively. In some cases, correlation coefficient can be improved; in others, the result is a more compact, easily interpreted representation of the target concept. This paper compares five well-known wrapper feature selection methods. Experimental results are reported using four well known representative regression algorithms.**

Index terms: **supervised machine learning, feature selection, regression models**

I. Introduction

In this paper we consider the following regression setting. Data is generated from an unknown distribution P on some domain X and labeled according to an unknown function g. A learning algorithm receives a sample S = $\{(x_1, g(x_1)), \ldots, (x_m, g(x_m))\}$ and attempts to return a function f close to g on the domain X. Many regression problems involve an investigation of relationships between attributes in heterogeneous databases, where different prediction models can be more appropriate for different regions.

Many factors affect the success of machine learning on a given task. The representation and quality of the instance data is first and foremost [13]. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. In real-world data, the representation of data often uses too many features, but only a few of them may be related to the target concept. There may be redundancy, where certain features are correlated so that is not necessary to include all of them in modelling; and interdependence, where two or more features between them convey important information that is obscure if any of them is included on its own.

Generally, features are characterized [2] as:

1. Relevant: These are features which have an influence on the output and their role can not be assumed by the rest
2. Irrelevant: Irrelevant features are defined as those features not having any influence on the output, and whose values are generated at random for each example.
3. Redundant: A redundancy exists whenever a feature can take the role of another (perhaps the simplest way to model redundancy).

Feature selection algorithms in general have two components: a selection algorithm that generates proposed subsets of features and attempts to find an optimal subset; and an evaluation algorithm that determines how 'good' a proposed feature subset is, returning some measure of goodness to the selection algorithm. However, without a suitable stopping criterion the feature selection process may run exhaustively or forever through the space of subsets. Stopping criteria can be: (i) whether addition (or deletion) of any feature does not produce a better subset; and (ii) whether an optimal subset according to some evaluation function is obtained.

Educational Software Development Laboratory, Department of Mathematics, University of Patras, Greece,{mariosk, dany, sotos, pintelas}@math.upatras.gr

Ideally, feature selection methods search through the subsets of features, and try to find the best one among all the competing candidate subsets according to some evaluation function. However, this procedure is exhaustive as it tries to find only the best one. It may be too costly and practically prohibitive, even for a medium-sized feature set size. Other methods based on heuristic or random search methods, attempt to reduce computational complexity by compromising performance.

In [10] different feature selection methods are grouped into two broad groups (i.e., filter and wrapper), based on their dependence on the inductive algorithm that will finally use the selected subset. Filter methods are independent of the inductive algorithm, whereas wrapper methods use the inductive algorithm as the evaluation function. Wrapper methods wrap the feature selection around the induction algorithm to be used, using cross-validation to predict the benefits of adding or removing a feature from the feature subset used

A strong argument for wrapper methods is that the estimated correlation coefficient of the learning algorithm is the best available heuristic for measuring the values of features. Different learning algorithms may perform better with different feature sets, even if they are using the same training set [4]. The wrapper selection methods are able to improve performance of a given regression model, but they are expensive in terms of the computational effort. The existing filter algorithms are computationally cheaper, but they fail to identify and remove all redundant features. In addition, there is a danger that the features selected by a filter method can decrease the correlation coefficient of a learning algorithm.

The next section presents the most well known wrapper selection methods. In the sections III-VI, we compare five well-known wrapper feature selection methods. Experimental results are reported using four well known representative regression algorithms. The final section concludes this work.

II. Wrapper feature selection methods

Theoretically, having more features should result in more discriminating power. However, practical experience with machine learning algorithms has shown that this is not always the case, current machine learning toolkits are insufficiently equipped to deal with contemporary datasets and many algorithms are susceptible to exhibit poor complexity with respect to the number of features.

There are several wrapper selection algorithms that try to evaluate the different subsets of the features on the learning algorithm and keep the subsets that perform best. The simplest method is forward selection (FS). It starts with the empty set and greedily adds attributes one at a time. At each step FS adds the attribute that, when added to the current set, yields the learned structure that generalizes best. Once an attribute is added FS cannot later remove it. Backward stepwise selection (BS) starts with all attributes in the attribute set and greedily removes them one at a time, too. Like forward selection, backward selection removes at each step the attribute whose removal yields a set of attributes that yields best generalization. Also like FS, once BS removes an attribute, it cannot later add it back to the set [17].

A problem with forward selection is that it may fail to include attributes that are interdependent, as it adds variables one at a time. However, it may locate small effective subsets quite rapidly, as the early evaluations, involving relatively few variables, are fast. In contrast, in backward selection inter-dependencies are well handled, but early evaluations are relatively expensive [11]. In [5] the authors describe forward selection, backward selection and some variations with classification algorithms and conclude that any wrapper selection method is better than no selection method.

Sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS) are characterized by the changing number of features included or eliminated at different stages of the procedure [15]. A similar way is followed by the Best First search. The Best First search starts with an empty set of features and generates all possible single feature expansions [8]. The subset with the highest evaluation is chosen and is expanded in the same manner by adding single features. If expanding a subset results in no improvement, the search drops back to the next best unexpanded subset and continues from there. Given enough time a Best First search will explore the entire search space, so it is common to limit the number of subsets expanded that result in no improvement. The best subset found is returned when the search terminates. The Best First search can be combined with forward or backward selection.

Another way is to start the search from a randomly selected subset (i.e. Random Generation) and add or delete a feature at random. A more informed random feature selection is carried out with the use of genetic algorithms [18]. A solution 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. Mutation changes some of the values (thus adding or deleting features) in a subset randomly. Crossover combines different features from a pair of subsets into a new subset. The application of genetic operators to population members is determined by their fitness. Better feature subsets have a greater chance of being selected to form a new subset through crossover or mutation. An important aspect of the genetic algorithm is that it is explicitly designed to exploit epistasis (that is, interdependencies between bits in the string), and thus should be well-suited for this problem domain. However, genetic algorithms typically require a large number of evaluations to reach a minimum.

III. EXPERIMENTS WITH REPRESENTATIVES LEARNING TECHNIQUES

In this study, we used for our experiments the forward selection (FS), the backward selection (BS), the Best First forward selection (BFFS), the Best First backward selection (BFFS) and the genetic search selection (GS) with the combination of five common machine learning techniques. For our experiments, we used a representative algorithm from each machine learning technique namely: Regression Trees [17], Regression Rules [17], Instance-Based Learning Algorithms [1] and Support Vector Machines [16].

For the purpose of the present study, we used 12 well known dataset from the UCI repository [3]. In Table 1, there is a brief description of these data sets.

The most well known measure for the degree of fit for a regression model to a dataset is the correlation coefficient. If the actual target values are a1, a2, …an and the predicted target values are: p1, p2, … pn then the correlation coefficient is given by the formula:

$$R=\frac{S_{PA}}{\sqrt{S_P S_A}}$$

where

$$S_{PA}=\frac{\sum_i (p_i-\bar{p})(a_i-\bar{a})}{n-1},\quad S_P=\frac{\sum_i (p_i-\bar{p})^2}{n-1},$$

$$S_A=\frac{\sum_i (a_i-\bar{a})^2}{n-1}.$$

In order to calculate the models' correlation coefficient, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the regression model was trained on the union of all of the other subsets. The best features are selected according to the feature selection algorithm and the performance of the subset is measured by how well it predicts the values of the test instances. This cross validation procedure was run 10 times for each algorithm and the average value of the 10-cross validations was calculated. It must be mentioned that we used the free available source code for these algorithms by [17] for our experiments. We have tried to minimize the effect of any expert bias by not attempting to tune any of the algorithms to the specific data set. Wherever possible, default values of learning parameter were used. This naïve approach results in lower estimates of the true error rate, but it is a bias that affects all the learning algorithms equally.

Table 1. *Brief description of the used data sets*

Datasets	Instances	Categ. Attr.	Numer. Attr.
bodyfat	252	0	14
cleveland	303	7	6
cloud	108	2	4
Cpu	209	1	6
echoMonths	130	3	6
elusage	55	1	1
fishcatch	158	2	5
longlay	16	0	6
lowbwt	189	7	2
servo	167	4	0
veteran	137	4	3
vineyard	52	0	3
bodyfat	252	0	14

In the next sections, we present the experiment results for each learning algorithm. Generally, in the following tables, one can see the correlation coefficient of each algorithm. In following Tables, we represent as "v" that the specific algorithm performed statistically better than the simple method without applying feature selection (WS) according to t-test with $p<0.05$. Throughout, we speak of two results for a dataset as being "significant different" if the difference is statistical significant at the 5% level according to the corrected resampled t-test [14], with each pair of data points consisting of the estimates obtained in one of the 100 folds for the two learning methods being compared. On the other hand, "*" indicates that the simple method without applying feature selection performed statistically better than the specific algorithm according to t-test with $p<0.05$. In all the other cases, there is no significant statistical difference between the results (Draws). In the last row of the table one can also see the aggregated results in the form (α/b/c). In this notation "α" means that the simple method without applying feature selection is significantly less accurate than the compared algorithm in α out of 12 datasets, "c" means that the simple method without applying feature selection is significantly more accurate than the compared algorithm in c out of 12 datasets, while in the remaining cases (b), there is no significant statistical difference.

IV. Regression trees

Regression trees are binary decision trees with numerical values at the leaf nodes: thus they can represent any piecewise linear approximation to an unknown function. A regression tree is generated in two stages. The first builds an ordinary decision tree, using as splitting criterion the maximization of the intra-subset variation of the target value. The second prunes this tree back by replacing subtrees with a numerical value wherever this seems appropriate.

Regression trees are very unstable in this regard as small perturbations in the training data set can produce large differences in the structure (and predictions) of a model. REPTree [17] is a fast regression tree learner that uses information variance reduction and reduced-error pruning (with backfitting).

Table 2. Wrapper selection using RepTree

Datasets	WS	FS		BS		BFFS		BFBS		GS	
bodyfat	0.98	0.98		0.98		0.98		0.98		0.98	
cleveland	0.54	0.52		0.56		0.53		0.57	v	0.52	
cloud	0.86	0.9	v	0.89	v	0.88		0.88		0.88	
cpu	0.91	0.92		0.92		0.92		0.92		0.92	
echoMonths	0.73	0.73		0.72		0.73		0.72		0.73	
elusage	0.8	0.9	v	0.9	v	0.9	v	0.9	v	0.9	v
fishcatch	0.95	0.96		0.94		0.96		0.96		0.96	
longley	0.4	0.4		0.4		0.4		0.4		0.4	
lowbwt	0.78	0.77		0.77		0.77		0.77		0.77	
servo	0.85	0.84		0.84		0.84		0.84		0.84	
veteran	0.31	0.22		0.37		0.2		0.25		0.26	
vineyard	0.64	0.64		0.59		0.61		0.61		0.61	
Average correlation coefficient	0.72	0.73		0.74		0.73		0.73		0.73	
W-D-L		0/10/2		0/10/2		0/11/1		0/11/1		0/11/1	

Table 3. Wrapper selection using M5rules

Datasets	WS	FS		BS		BFFS		BFBS		GS	
bodyfat	0.99	0.99		0.99		0.99		0.99		0.99	
cleveland	0.7	0.72		0.73	v	0.72		0.72		0.73	v
cloud	0.92	0.93		0.94		0.94		0.94		0.94	
cpu	0.97	0.99		0.99		0.99		0.99		0.99	
echoMonths	0.71	0.71		0.72		0.71		0.72		0.72	
elusage	0.85	0.91	v	0.91	v	0.91	v	0.91	v	0.91	v
fishcatch	0.99	0.99		0.99		0.99		0.99		0.99	
longley	0.4	0.6	v	0.6	v	0.6	v	0.6	v	0.6	v
lowbwt	0.8	0.8		0.81		0.8		0.81		0.81	
servo	0.94	0.93		0.93		0.93		0.93		0.93	
veteran	0.55	0.54		0.58		0.56		0.58		0.56	
vineyard	0.64	0.69	v	0.69	v	0.69	v	0.69	v	0.69	v
Average correlation coefficient	0.79	0.82		0.83		0.82		0.83		0.83	
W-D-L		0/9/3		0/8/4		0/9/3		0/9/3		0/8/4	

As we have already mentioned, in Table 2, one can see the correlation coefficient of RepTree algorithm in each data set before and after the attribute selection procedures.

According to our experiments, BS is slightly better feature selection method for the RepTree algorithm; however, the forward selection search uses much fewer features for the induction and is less time consuming method.

Table 4.Wrapper selection using K*

Datasets	WS	FS		BS		BFFS		BFBS		GS	
bodyfat	0.88	0.99	v	0.99	v	0.99	v	0.99	v	0.99	v
cleveland	0.58	0.64		0.66	v	0.64		0.66	v	0.65	
cloud	0.83	0.91	v	0.91	v	0.91	v	0.91	v	0.91	v
cpu	0.99	0.98		0.98		0.98		0.98		0.98	
echoMonths	0.37	0.74	v	0.72	v	0.74	v	0.74	v	0.74	v
elusage	0.86	0.9	v	0.9	v	0.9	v	0.9	v	0.9	v
fishcatch	0.99	0.99		0.99		0.99		0.99		0.99	
longley	0.5	0.4		0.5		0.4		0.5		0.4	
lowbwt	0.61	0.78	v	0.79	v	0.79	v	0.78	v	0.78	v
servo	0.86	0.87		0.87		0.87		0.87		0.87	
veteran	0.28	0.49	v	0.45		0.49	v	0.5	v	0.49	v
vineyard	0.78	0.76		0.76		0.76		0.76		0.77	
Average correlation coefficient	0.71	0.79		0.8		0.79		0.8		0.79	
W-D-L		0/6/6		0/6/6		0/6/6		0/5/7		0/6/6	

Table 5.Wrapper selection using SMOreg

Datasets	WS	FS		BS		BFFS		BFBS		GS	
bodyfat	0.99	0.99		0.99		0.99		0.99		0.99	
cleveland	0.71	0.72		0.71		0.72		0.7		0.71	
cloud	0.94	0.94		0.94		0.94		0.94		0.94	
cpu	0.97	0.96		0.96		0.96		0.96		0.96	
echoMonths	0.69	0.71		0.71		0.7		0.71		0.7	
elusage	0.84	0.89	v	0.89	v	0.89	v	0.89	v	0.89	v
fishcatch	0.97	0.97		0.97		0.97		0.97		0.97	
longley	0.6	0.6		0.6		0.6		0.6		0.6	
lowbwt	0.78	0.79		0.78		0.78		0.78		0.78	
servo	0.84	0.83		0.84		0.84		0.84		0.84	
veteran	0.52	0.53		0.53		0.53		0.53		0.53	
vineyard	0.69	0.72		0.74	v	0.74	v	0.74	v	0.74	v
Average correlation coefficient	0.8	0.81		0.81		0.81		0.81		0.81	
W-D-L		0/11/1		0/10/2		0/10/2		0/10/2		0/10/2	

It must be mentioned that all the selection algorithms improve the correlation coefficient of RepTree algorithm.

Generally, the backward selection strategies are very inefficient for large-scale datasets, which may have hundreds of original attributes. The forward selection wrapper methods are less able to improve performance of a given regression model, but they are less expensive in terms of the computational effort and use fewer features for the induction.

Genetic selection typically requires a large number of evaluations to reach a minimum.

V. Regression rules

Inducing rules from a given training set is a well-studied topic in machine learning [17]. A regression rule is an IF-THEN rule that has as conditional part a set of conditions on the input features and as conclusion a regression model. M5rules algorithm produces propositional regression rules using routines for generating a decision list from M5´Model trees [17]. The algorithm is able to deal with both continuous and nominal variables, and obtains a piecewise linear model of the data.

In Table 3, one can see the correlation coefficient of M5rules in each data set before and after the attribute selection procedures.

According to our experiments, BS is slightly better feature selection method for the M5rules algorithm; however, the forward selection search uses much fewer features for the induction and is less time consuming method. It must be mentioned that all the selection algorithms improve the correlation coefficient of M5rules algorithm at least 4%.

VI. Instance-based learning

Instance-based learning algorithms belong in the category of lazy-learning algorithms [13], as they delay the induction until prediction is performed. One of the most straightforward instance-based learning algorithms is the nearest neighbour algorithm [1]. In Table 4, one can see the correlation coefficient of K* algorithm [6] in each data set before and after the attribute selection procedures.

According to our experiments, BS is slightly better feature selection method for the K* algorithm; however, the forward selection search uses much fewer features for the induction and is less time consuming method. It must be mentioned that all the selection algorithms improve the correlation coefficient of K* algorithm at least 11%.

Other scaling experiments showed that the nearest neighbour’s sample complexity (the number of training examples needed to reach a given correlation coefficient) increases exponentially with the number of irrelevant attributes present in the data [10].

VII. Support Vector Machines

The sequential minimal optimization algorithm (SMO) has been shown to be an effective method for training support vector machines (SVMs) on classification tasks defined on sparse data sets [16]. SMO differs from most SVM algorithms in that it does not require a quadratic programming solver. Shevade et al. [16] generalize SMO so that it can handle regression problems. This implementation globally replaces all missing values and transforms nominal attributes into binary ones.

In Table 5, one can see the correlation coefficient of SMOreg algorithm in each data set before and after the attribute selection procedures.

According to our experiments, BFFS is slightly better feature selection method for the SMOreg algorithm; however, the forward selection search uses much fewer features for the induction and is less time consuming method. It must be mentioned that all the selection algorithms improve the correlation coefficient of SMOreg algorithm.

VIII. Conclusions

Feature subset selection is the process of identifying and removing as much of the irrelevant and redundant information as possible. The role of feature selection in machine learning is (1) to speed up the prediction process, (2) to improve the correlation coefficient of a regression algorithm, and (3) to improve the comprehensibility of the learning results. Feature wrappers often achieve better results than filters due to the fact that they are tuned to the specific interaction between an induction algorithm and its training data.

However, they tend to be slower than feature filters because they must repeatedly call the induction algorithm and must be re-run when a different induction algorithm is used.

Generally, the backward selection strategies are very inefficient for large-scale datasets, which may have hundreds of original attributes. The forward selection wrapper methods are less able to improve performance of a given regression model, but they are less expensive in terms of the computational effort and use fewer features for the induction. Genetic selection typically requires a large number of evaluations to reach a minimum.

Naturally, none of the described feature selection algorithms is superior to others in all data sets; each algorithm has some advantages and disadvantages. Discussions of all the pros and cons of each individual selection algorithm are beyond the scope of this paper and will depend on the task at hand. However, we hope that this work can help the practitioners to avoid picking a wrong wrapper selection algorithm in combination with their favorite regression algorithm.

Finally, it must be mentioned that the problem of feature interaction can be addressed by constructing new features from the basic feature set (feature construction). This method reduces dimensionality by creating a new set of attributes that are mathematically related to the original set, but that contain significantly fewer attributes. Generally, transformed features generated by feature construction may provide a better discriminative ability than the best subset of given features, but these new features may not have a clear physical meaning [12].

In a future work, we will propose a hybrid feature selection technique which combines the advantages of both filter and wrapper selection techniques.

IX. References

[1] Aha, D. 1997. Lazy Learning. Dordrecht: Kluwer Academic Publishers.

[2] Bell D. and Wang H., A Formalism for Relevance and its Application in Feature Subset Selection. Machine Learning Vol. 41(2) (2000) 175–195.

[3] Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[4] Blum A. and Langley P., Selection of relevant features and examples in machine learning. Artificial Intelligence, 97:245–271, 1997.

[5] Caruana, R., and Freitag, D. 1994, Greedy Attribute Selection, Machine Learning: Proceedings of the Eleventh International Conference, Morgan Kaufman, San Francisco, CA.

[6] John, C. and Trigg, L., K*: An Instance- based Learner Using an Entropic Distance Measure", Proc. of the 12th International Conference on ML, (1995) 108-114.

[7] John, G.H., Kohavi, R., and Pfleger, K. 1994. Irrelevant features and the subset selection problem. In Proc. of the 11th International Conference on ML, W. Cohen and H. Hirsh (Eds.), CA: Morgan Kaufmann, pp. 121–129.

[8] Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. Artificial Intelligence, 97:273-324.

[9] Kudo M., Sklansky J. (2000), Comparison of algorithms that select features for pattern classifiers, Pattern Recognition 33: 25-41.

[10] Langley, P., Selection of relevant features in machine learning. In: Proceedings of the AAAI Fall Symposium on Relevance, 1–5, 1994.

[11] Liu H. and Motoda H., Feature Selection for Knowledge Discovery Data Mining. Boston: Kluwer Academic Publishers, 1998.

[12] Markovitch S. & Rosenstein D. (2002), Feature Generation Using General Constructor Functions, Machine Learning, 49, 59–98, 2002, Kluwer Academic Publishers.

[13] Mitchell, T. 1997. Machine Learning. McGraw Hill.

[14] Nadeau, C., Bengio, Y., Inference for the Generalization Error. Machine Learning, 52 (2003) 239-281.

[15] Pudil, P., Novovicova, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recognition Lett. 15, 1119–1125.

[16] Shevade, S., Keerthi, S., Bhattacharyya C., and Murthy, K. (2000). Improvements to the SMO algorithm for SVM regression. IEEE Transaction on Neural Networks, 11(5):1188-1183.

[17] Witten, I. and Frank E. (2000) Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Mateo, CA, 2000.

[18] Yang J, Honavar V. Feature subset selection using a genetic algorithm. IEEE Int Systems and their Applications 1998; 13(2): 44–49.