Fast and robust selection of highly-correlated features in regression problems

Andreas Maier
Pattern Recognition Lab
FAU Erlangen-Nürnberg
Martensstr. 3, Erlangen, Germany
andreas.maier@fau.de

Dalia Rodríguez-Salas
Pattern Recognition Lab
FAU Erlangen-Nürnberg
Martensstr. 3, Erlangen, Germany
dalia.rodriguez@fau.de

Abstract

Feature selection for regression problems can be highly beneficial in terms of robustness and execution speed. The Correlation-based Feature Selection (CFS) algorithm, in the attempt to find the best feature subset, evaluates different subsets and selects the one with the highest "goodness". Such goodness is based on the correlation between the addition of all features in the subset with the output variable. However, such a simple addition assumes that all features have the same weight in the output variable. This, in turn, assumes that all features are uncorrelated with each other. By considering an optimal weighting instead, a more robust measurement of the goodness can be obtained; however, such weighting is computationally expensive. In this paper, a feature selection algorithm named "R-fast" which considers the $\boldsymbol{\beta}$ coefficients in the standardized linear regression model as optimal weights is proposed. We also present a technique to quickly estimate approximations to the $\boldsymbol{\beta}$ coefficients. Our algorithm was evaluated using Multiple Regression Analysis (MRA) over 8 synthetic and 10 real-world datasets. Results show that, when selecting features with our proposed algorithm, MRA's performance is better than or equal to those obtained when selecting features with others well-known filter algorithms and when no selection is performed.

1 Introduction

Data used to train regression algorithms is usually not free of redundant and useless features. Performance of these algorithms can be affected when processing such features, not only in terms of time needed to process redundant and useless information, but in some cases, in terms of accuracy when predicting the value of objects which regression algorithms were not trained with. Feature selection attempts to remove the highest amount of redundant and useless features as possible, and thus, reducing dimensionality which will allow regression algorithms a faster execution and an easier interpretation of the resulting model.

The three main approaches to feature selection algorithms are wrapper, filter, and embedded [4]. For each subset to be evaluated, wrappers train a given target algorithm and evaluate its performance. A *goodness* associated with such performance is subsequently assigned to the subset. Using the target regression algorithm to evaluate the *goodness* of feature subsets grants wrapper approaches to obtain a better feature subset than its filter and embedded counterparts. Nonetheless, a re-sampling technique is needed in order to train and evaluate the performance of the target algorithm. This situation leads to a repeated calling of the target algorithm, making the use of wrapper algorithms unsuitable for high-dimensional regression problems, specially when the target algorithm has a high number of parameters to be optimized. Regression algorithms that embed the feature selection do not need re-training the algorithm, and they are commonly faster than wrappers. However, there are not embedded feature selection techniques for all available regression algorithms. Moreover, wrapper and embedded algorithms need to be run again if it is decided to change the target algorithm. In contrast, filters are independent of the regression algorithm.

There are two different types of filter algorithms, those which only evaluate the *goodness* of single-features and those which evaluate the *goodness* of multiple-features (feature subsets). The former selects those features with a *goodness* higher than a given threshold, thus, only removing useless but non-redundant features. The latter selects the feature subset with the highest *goodness*, considering the removal of both useless and redundant features. Filter algorithms which evaluate feature subsets, as well as wrappers, commonly do not evaluate all-possible subsets. This is due to the fact that there are 2^m possible subsets for a problem with m features, thus, performing an entire space evaluation is only feasible when the number of features is small. Instead, searching strategies are employed in order to generate feature subsets to be evaluated. Such generation can be in a forward direction, backwards direction, bi-directional, and/or in a random manner [10]. These strategies have to set a stopping criteria in order to avoid exploring the entire search-space.

The CFS algorithm [6] is a well-known filter algorithm where the *goodness* of a feature subset is equal to the linear correlation of the addition of the features with the output variable. This assumes that all features contribute equally to the output variable, but in most real-world regression problems, features interact amongst each other and differentially contribute to the output variable. Therefore, by differentially weighting the features, a more reliable *goodness* measurement can be achieved.

The main contributions of this paper are:

1. A filter feature selection algorithm that, in order to measure the *goodness* of a feature subset, calculates the correlation of the output variable with the addition of the differentially-weighted features using $\boldsymbol{\beta}$ coefficients.
2. We propose to use accurate as well as approximated $\boldsymbol{\beta}$ coefficients for weighting features.
3. We also propose to use the Stochastic Gradient Descent (SGD) as an option to approximate the $\boldsymbol{\beta}$ coefficients with a reduced amount of epochs.

The rest of this paper is organized as follows. First we present related work in Section 2, following the definition of our proposed algorithm in Section 3. Evaluations of our proposal are presented in Section 4 and finally our conclusions are drawn in Section 5.

2 Related Work

It is important to highlight that related work reviewed in this section is focused in feature selection for regression problems; interested readers in feature selection for general machine learning problems may wish to refer to [7].

Feature selection algorithms based on the embedded approach penalize the cost function of linear regression algorithms by adding a term in function of the norm of the feature coefficients. This penalty introduces shrinkage in the coefficients towards zero, and those features with an exact-zero coefficient can be removed from the resulting model. The LASSO penalty [14] uses the l_1-norm and it is the base of several other penalty functions [15].

There are many criteria under the wrapper approach to measure the *goodness* of feature subsets, based on the performance of the target regression algorithm, that are widely used to evaluate the *goodness* of feature subsets, such as the mean absolute error, Mallow's C_p criterion, Akaike information criterion, Bayesian information criterion, and several criteria based on the correlation coefficient [17]. This is not the case for the filter approach since most of the criteria to evaluate the *goodness* of features are addressed to select features in domains where the output variable has nominal values (classification problems). In this section, we will only consider the two most common algorithms.

The RReliefF algorithm [13], evaluates the *goodness* of single-features based on Bayes' theorem and differences among object pairs in the training set. Each of these pairs contains one of q randomly sampled objects and one of its p nearest objects. Its main drawback is that it has parameters to be fixed that strongly depend on the training data. In addition, given that RReliefF is a filter algorithm that evaluates single-features, it only considers the removal of useless features.

The CFS algorithm [6] evaluates multiple-features under the filter approach. This algorithm defines the *goodness* G^S of a feature subset S as in Eq. (1), where $\overline{r}$ is the mean of Pearson's correlations, k is the number of features in S, $\boldsymbol{y}$ is the output variable and $\boldsymbol{x}_j$ is the j-th feature. G^S_{CFS} depends on the average correlation between the output variable $\boldsymbol{y}$ and every feature $\boldsymbol{x}_j$ ($\overline{r_{\boldsymbol{y}\boldsymbol{x}_j}}$), and the average correlation of every pair of different features $\boldsymbol{x}_j$ ($\overline{r_{\boldsymbol{x}_j\boldsymbol{x}_{j'}}}$). Eq. (1) is derived by estimating the Pearson's correlation between the output variable and a composite variable which is the result of adding all features given in standardized values in [3].

$$G^S_{\text{CFS}} = \frac{k\overline{r_{\boldsymbol{y}\boldsymbol{x}_j}}}{\sqrt{k + k(k-1)\overline{r_{\boldsymbol{x}_j\boldsymbol{x}_{j'}}}}} = r_{\boldsymbol{y}(\boldsymbol{x}_1+\boldsymbol{x}_2+\cdots+\boldsymbol{x}_k)} \quad (1)$$

CFS' heuristic is biased to select those features highly correlated with the output variable but uncorrelated with each other. Nonetheless, as pointed out by Guyon et.al. in [4]: "*A variable that is completely useless by itself can provide a significant performance improvement when taken with others*". Variables (features) which meet the conditions from [4] are known as suppressors. Due to the fact that suppressor features are low correlated with the output, but correlated with others, CFS's heuristic tends to ignore them.

3 Correlation-based Feature subset evaluation by the addition of weighted features

Central to our research is the idea that the more correlated $\boldsymbol{y}$ and $w_1\boldsymbol{x}_1 + w_2\boldsymbol{x}_2 + \cdots + w_k\boldsymbol{x}_k$ are, the more suitable are the k features to use them to predict the values on $\boldsymbol{y}$. By estimating those weights $(w_1, w_2, \ldots, w_k)$ considering the interactions of the features with the output and the interactions among features, a more reliable measure of the correlation between the features in a subset and the output can be obtained. $\boldsymbol{\beta}$ coefficients in the Standardized Linear Regression Model (SLRM) take into account such interactions and are known to maximize the correlation of the weighted addition of the features which is given by the SLRM's correlation coefficient R^{SLRM}.

The computation of the $\boldsymbol{\beta}$ coefficients involves the computation of the pseudo-inverse of a matrix, which is commonly solved by means of QR decomposition using the Householder reflector [1]. This process has a theoretical complexity of $\mathcal{O}^{\text{SLRM}}(M^2 \times N)$, where M is the number of features and N the number of observations. However, fast but good approximations to the $\boldsymbol{\beta}$ coefficients ($\hat{\beta}$) can be achieved with the Stochastic Gradient Descent (SGD) algorithm, where the objective is to reduce the sum of the squared error ($\epsilon(\hat{\boldsymbol{\beta}})$) while changing $\hat{\beta}_j$ in function of the partial derivative of $\epsilon(\hat{\boldsymbol{\beta}})$ with respect to $\hat{\beta}_j$ for $j = 1$ to M. SGD's theoretical complexity is $\mathcal{O}^{SGD}(N \times M \times 3 \times C)$, where C is the number of epochs.

Note that for each feature subset to be evaluated, the computation of its respective weights must be performed. In order to perform this evaluation quickly, the SGD can be used every time it is more time-suitable than using the SLRM. Thus, as soon as $\mathcal{O}^{\text{SLRM}} > \mathcal{O}^{\text{SGD}}$ (or $M > 3 \times C$ after factorizing $M \times N$), the feature subset evaluation should be performed with the SGD method in order to speed up the feature selection process. The *goodness* measure $G^S_{\text{R-fast}}$ of a subset S with k features, is given by Eq. (2).

$$G^S_{\text{R-fast}} = \begin{cases} R^{\text{SGD}}, & \text{if } k > 3 \times C \\ R^{\text{SLRM}}, & \text{otherwise} \end{cases} \quad (2)$$

From Eq. (2), it can be seen that the *goodness* of subsets with $3 \times C$ or less features is given by R^{SLRM} and for subsets with more features is given by the SGD's correlation coefficient, R^{SGD}. Thus, the $\boldsymbol{\beta}$ coefficients are calculated for subsets with up to $3 \times C$ features, and only the approximations $\hat{\boldsymbol{\beta}}$ for subsets with more than $3 \times C$ features. Since smaller subsets are evaluated with the accurate weights, we recommend to use a forward direction to generate subsets on the strategy search. However, it is important to highlight that R-fast can be used to evaluate the *goodness* of generated subsets given for search strategies following a forward direction, backwards direction, bi-directionally or in a random manner.

When the generation of subsets is given for search strategies in a forward direction, *small* values of C (epochs) can be set for the SGD algorithm. This is due to that under this condition, when the search strategy starts generating subsets with size $3 \times C + 1$, it has already selected the subset with the $3 \times C$ features which maximizes the correlation with the output and the weighted addition of the features on it.

4 Evaluation

In this section we will show that the performance of a regression algorithm when selecting features with our proposed technique does not decrease significantly. We choose the Multiple Regression Analysis (MRA) as regression method being the most standard regres-

sion method. We also selected CFS and RReliefF to compare our proposed algorithm, due to that both are well-known filter algorithms.

Experimental setup: We implemented our approach in Java using Weka [5]. The experiments were carried out on a 4th Gen Intel© Core™ i7-4720HQ processor at 2.60GHz and 8.0GB PC3L-12800 DDR3L SDRAM at 1600 MHz memory running Windows 10 Home version 1607.

Let "R-acc" be the feature selection algorithm which only uses accurate $\boldsymbol{\beta}$ coefficients as weights. Our experimentation will be addressed to three cases in order to show that: A) R-fast is a considerably faster option than R-acc and makes the regression algorithm to have a performance as good as that using R-acc, B) differentially weighting the features leads to a more robust goodness measure than using equally weighting, and C) R-fast leads to a more robust feature selection than others well-known filter algorithms.

As in [11], the statistic paired t-test with a significance level of 0.05 is applied in both experiments, in order to compare the MRA's performance. Whenever a statistically significant improvement or degradation is present in the compared results, it is indicated with symbols "$*$" and "$\circ$", respectively. On one hand, when performing feature selection with R-fast and CFS, the generation of feature subsets is determined in accordance to the best first search algorithm [12], and with a forward direction and termination after 5 generations of feature subsets without any improvement in the *goodness* value. On the other hand, when performing feature selection with RReliefF, the ranking of the features is calculated considering 10 objects as neighbors and with 0.0 and 0.01 as selection threshold. Either performing feature selection or not, MRA is built and evaluated by means of a 10-folds cross-validation scheme. Synthetic and real-world datasets were used in the experiments. Different correlation settings were considered to generate 8 synthetic datasets. Each dataset has three features and 1000 objects, the output variable and the features are random vectors chosen from multivariate normal distributions with mean $(0, 0, 0, 0)$ and different variances given in accordance to the desired correlation among vectors.

The real-world datasets were taken from different sources [2, 8, 9, 16] with different amounts of objects and features: *ailerons* 13750 and 40, *cpu-act* 8192 and 21, *housing* 506 and 13, *hungarian* 294 and 13, *pbc* 418 and 18, *pl35* 35 and 150, *pol* 15000 and 48, *pyrim* 74 and 27, *triazines* 186 and 60, *wisconsin* 194 and 32, respectively.

4.1 Exact vs approximated β coefficients

MRA's performance when employing R-acc is compared with that obtained when using R-fast. Results when using the real-world datasets are shown in Table 1, where it can be seen that in any dataset the MRA's correlation coefficient showed a significant degradation. From the last row, it can be seen that there is an average increment in the MRA's correlation coefficient. We attribute this situation to the fact that by taking a small number of epochs for the SGD algorithm, an over fitting of the data is avoided.

The average selection time is reported in Table 2, where R-fast requires a smaller amount of time to perform the selection, furthermore, in most cases the difference is statistically significant.

The amount of selected features between R-acc and R-fast are presented in Table 3. As seen, R-fast tends to select a statistically significant smaller amount of features. Since the correlation coefficient is maximized when using the exact weights, the use of approximated weights leads to smaller values in the correlation coefficient. Thus, $R^{SGD} \leq R^{SLRM}$; therefore, when R-fast changes from R^{SLRM} to R^{SGD}, the greater the number of epochs is, the *harder* it is for R^{SGD} to reach the last subset evaluation obtained with R^{SLRM}. This explains the situation presented in the corresponding columns for R-fast with 2,3,4 and 5 epochs, where the number of selected features is equal to the amount of features evaluated using R^{SLRM} $(3 \times C)$ in most cases.

		R-fast				
Dataset	R-acc	$C=1$	$C=2$	$C=3$	$C=4$	$C=5$
ailerons	0.90	0.90	0.90	0.90	0.90	0.90
pyrim	0.46	0.71	0.63	0.65	0.67	0.60
pol	0.68	0.68	0.68	0.68	0.68	0.68
pbc	0.60	0.61	0.60	0.62	0.60	0.60
hungarian	0.71	0.71	0.71	0.71	0.71	0.71
housing	0.85	0.85	0.85	0.85	0.85	0.85
triazines	0.44	0.48	0.48	0.41	0.39	0.43
pl35	0.72	0.88	0.89	0.82	0.79	0.80
cpu-act	0.85	0.84	0.85	0.85	0.85	0.85
wisconsin	0.32	0.37	0.28	0.31	0.30	0.30
Average	0.65	0.70	0.69	0.68	0.67	0.67

Table 1: MRA's average correlation coefficient when using R-acc and R-fast with different number of epochs (C).

		R-fast				
Dataset	R-acc	$C=1$	$C=2$	$C=3$	$C=4$	$C=5$
ailerons	28110.2	14343.0$*$	12762.8$*$	15719.3$*$	18275.6$*$	20511.9$*$
pyrim	49.1	33.3$*$	30.5$*$	34.5$*$	39.6$*$	42.5$*$
pol	50604.5	31094.3$*$	24897.7$*$	29288.5$*$	34336.2$*$	38892.2$*$
pbc	73.6	62.0$*$	64.0$*$	66.6$*$	70.7	72.4
hungarian	28.3	21.3	22.6	23.9	27.9	23.7
housing	41.7	35.6	37.4	39.1	40.0	40.2
triazines	1202.1	691.9$*$	794.8$*$	644.0$*$	695.0$*$	681.4$*$
pl35	3326.8	1504.2$*$	1228.3$*$	1548.2$*$	1852.2$*$	2168.9$*$
cpu-act	2185.4	1330.9$*$	1585.0$*$	1846.0$*$	2031.7$*$	2097.5$*$
wisconsin	172.2	126.9$*$	126.4$*$	111.5$*$	127.6$*$	140.7$*$
Average	8579.4	4924.3	4155.0	4932.2	5749.6	6467.1

Table 2: Average selection time for R-acc and R-fast with different number of epochs (C), expressed in milliseconds.

		R-fast				
Dataset	R-acc	$C=1$	$C=2$	$C=3$	$C=4$	$C=5$
ailerons	32.3	8.6$*$	6.0 $*$	9.0 $*$	12.0 $*$	15.0 $*$
pyrim	25.6	9.0$*$	6.5 $*$	9.0 $*$	12.0 $*$	15.0 $*$
pol	23.9	11.2$*$	6.4 $*$	9.0 $*$	12.0 $*$	15.0 $*$
pbc	17.1	11.5$*$	10.6 $*$	9.0 $*$	12.0 $*$	15.0 $*$
hungarian	11.3	9.7$*$	8.6 $*$	9.0 $*$	11.3	11.3
housing	12.5	11.4	11.1	11.0	11.9 $*$	12.5
triazines	42.9	18.1$*$	22.7 $*$	15.5 $*$	16.5 $*$	15.0 $*$
pl35	25.0	8.4$*$	6.0 $*$	9.0 $*$	12.0 $*$	15.0 $*$
cpu-act	20.1	4.9$*$	6.0 $*$	9.0 $*$	12.0 $*$	15.0 $*$
wisconsin	31.2	15.6$*$	14.2 $*$	9.0 $*$	12.0 $*$	15.0 $*$
Average	24.2	10.8	9.8	9.8	12.3	14.4

Table 3: Average number of selected features for R-acc and R-fast with different number of epochs (C).

4.2 Differentially vs equally weighting

The synthetic datasets are employed to compare MRA's performance when using R-fast after one epoch, against those obtained with CFS. Results in Table 4 show that the number of selected features by CFS shows a significant improvement in 6 out of 8 datasets. Nonetheless, MRA's correlation coefficient gets significantly worse in 4 out of those 6 datasets. As seen in Table 4, the correlation of the output $\boldsymbol{y}$ with $\boldsymbol{x}_1$ is 0 in the first 4 datasets. However, $\boldsymbol{x}_1$ does have a correlation greater than 0 with $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ for 21-H, 21-M and 21-L, thus, in those 3 cases $\boldsymbol{x}_1$ can be considered as a potential suppressor variable. In those datasets with a potential suppressor feature, MRA's performance showed a clear degradation when using CFS. The last four cases consider a mid-correlation with the output, but different feature-feature correlations. Even though the dataset 03-H does not have a

potential suppressor variable, the MRA's performance also shows a degradation, this can be due to the high interaction among the three features.

Dataset description					Correlation coefficient		No. of selected features	
ID	$r_{x_j x_{j'}}$	r_{yx_1}	r_{yx_2}	r_{yx_3}	R-fast	CFS	R-fast	CFS
21-H	0.75	0.00	0.66	0.25	1.00	0.67 ∘	3.00	1.00 ∗
21-M	0.50	0.00	0.75	0.25	0.87	0.74 ∘	2.20	1.00 ∗
21-L	0.25	0.00	0.75	0.25	0.77	0.73 ∘	3.00	1.00 ∗
21-N	0.00	0.00	0.75	0.25	0.61	0.59 ∘	3.00	1.00 ∗
03-H	0.75	0.40	0.50	0.60	0.60	0.59	3.00	1.00 ∗
03-M	0.50	0.40	0.50	0.60	0.64	0.64	3.00	2.00 ∗
03-L	0.25	0.40	0.50	0.60	0.71	0.71	3.00	3.00
03-N	0.00	0.40	0.50	0.60	0.87	0.87	3.00	3.00

Table 4: Synthetic datasets description, MRA's average correlation coefficient, and average number of selected features when performing feature selection using R-fast and CFS.

4.3 R-fast vs well-known filter algorithms

The real-world datasets are employed to compare the performance of the MRA without feature selection (*none*) against using R-fast, after one epoch, and by using CFS and RReliefF. As shown in Table 5, while MRA's performance does not decrease significantly when using R-fast, it does decrease significantly in four datasets when using CFS. Although MRA shows a good performance when using RReliefF with 0.0 as threshold, a small increment (0.01) in the selection threshold changes completely the MRA's performance, passing from one to six datasets with a significant decrement, respectively.

Dataset	*none*	R-fast	CFS	RReliefF (0.0)	RReliefF (0.01)
ailerons	0.90	0.90	0.87 ∘	0.90	0.77 ∘
pyrim	0.46	0.71	0.70	0.44	0.50
pol	0.68	0.68	0.48 ∘	0.61 ∘	0.00 ∘
pbc	0.60	0.61	0.54 ∘	0.57	0.47 ∘
hungarian	0.71	0.71	0.72	0.67	0.64
housing	0.85	0.85	0.82 ∘	0.84	0.79 ∘
triazines	0.44	0.48	0.42	0.46	0.45
pl35	0.83	0.88	0.77	0.86	0.82
cpu-act	0.85	0.84	0.84	0.85	0.68 ∘
wisconsin	0.32	0.37	0.37	0.36	0.00 ∘
Average	0.66	0.70	0.65	0.66	0.51

Table 5: MRA's average correlation coefficient without feature selection (*none*), R-fast, CFS and RReliefF (threshold 0.0 and 0.01) over real-world datasets.

Results in Table 6 show that the amount of selected features clearly differs from one algorithm to another in most cases. Regardless the good MRA's performance when using RReliefF with 0.0 as threshold, the reduction in the amount of features dropped notably in comparison with R-fast and CFS.

Dataset	*none*	R-fast	CFS	RReliefF (0.0)	RReliefF (0.01)
ailerons	40.0	8.6 ∗	20.0 ∗	39.5	2.0 ∗
pyrim	27.0	9.0 ∗	6.5 ∗	18.3 ∗	16.0 ∗
pol	48.0	11.2 ∗	4.0 ∗	10.5 ∗	0.0 ∗
pbc	18.0	11.5 ∗	5.2 ∗	15.8 ∗	11.1 ∗
hungarian	13.0	9.7 ∗	8.0 ∗	8.1 ∗	3.2 ∗
housing	13.0	11.4 ∗	4.0 ∗	9.6 ∗	3.3 ∗
triazines	60.0	18.1 ∗	7.5 ∗	27.0 ∗	13.7 ∗
pl35	150.0	8.4 ∗	25.9 ∗	99.2 ∗	79.0 ∗
cpu-act	21.0	4.9 ∗	10.6 ∗	21.0	2.0 ∗
wisconsin	32.0	15.6 ∗	15.2 ∗	10.0 ∗	0.0 ∗
Average	42.2	10.8	10.7	25.9	13.0

Table 6: MRA's average number of features without feature selection (*none*), R-fast, CFS and RReliefF (threshold 0.0 and 0.01) over real-world datasets.

5 Conclusions

In this paper was empirically shown that when the correlation of the weighted addition of the features with the output variable is used as *goodness* measure, using the $\boldsymbol{\beta}$ coefficients as weights is more reliable than using equal weights. Furthermore, if the approximations of the $\boldsymbol{\beta}$ coefficients are obtained with the SGD algorithm, when it is more time feasible than using the accurate ones, the selection time significantly improves and the number of selected features tends to decrease.

By selecting features with R-fast, the regression algorithm does not significantly decrease its performance. In addition, it was shown that, different to the CFS algorithm, R-fast is capable of selecting those features that by themselves are useless but, when taken with others can provide a significant performance improvement on the regression algorithm.

References

[1] Åke Björck. *Linear Least Squares Problems*, pages 211–430. Springer International Publishing, Cham, 2015.

[2] Thomas R. Fleming and David P. Harrington. Counting processes and survival analysis. 169, 2011.

[3] Edwin Ernest Ghiselli. *Theory of psychological measurement*, pages 175 – 182. McGraw-Hill, 1964.

[4] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[5] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[6] Mark A Hall. Correlation-based feature selection of discrete and numeric class machine learning. 2000.

[7] Alan Jović, Karla Brkić, and Nikola Bogunović. A review of feature selection methods with applications. In *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205. IEEE, 2015.

[8] M. Lichman. UCI machine learning repository, 2013. URL: `http://archive.ics.uci.edu/ml`.

[9] Andreas Maier. *Speech of children with cleft lip and palate: Automatic assessment*. Logos-Verlag, Berlin, 2009.

[10] Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. Feature selection algorithms: a survey and experimental evaluation. In *Proceedings of the International Conference on Data Mining (ICDM'03)*, pages 306–313. IEEE, 2002.

[11] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine Learning*, 52(3):239–281, 2003.

[12] Elaine Rich and Kevin Knight. Artificial intelligence. *McGraw-Hill, New*, 1991.

[13] Marko Robnik-Šikonja and Igor Kononenko. An adaptation of relief for attribute estimation in regression. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, pages 296–304, 1997.

[14] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[15] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B*, 73(3):273–282, 2011.

[16] Luís Torgo. Regression data sets. 2014. URL: `http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html`.

[17] Javier Trejos, Mario A. Villalobos-Arias, and Jose Luis Espinoza. Variable selection in multiple linear regression using a genetic algorithm. *Handbook of Research on Modern Optimization Algorithms and Applications in Engineering and Economics*, page 133, 2016.