Speeding Up the Wrapper Feature Subset Selection in Regression by Mutual Information Relevance and Redundancy Analysis

Gert Van Dijck and Marc M. Van Hulle

Computational Neuroscience Research Group, Laboratorium voor Neuro-en Psychofysiologie, K.U. Leuven, B-3000 Leuven, Belgium
{gert, marc}@neuro.kuleuven.ac.be

Abstract. A hybrid filter/wrapper feature subset selection algorithm for regression is proposed. First, features are filtered by means of a relevance and redundancy filter using mutual information between regression and target variables. We introduce permutation tests to find statistically significant relevant and redundant features. Second, a wrapper searches for good candidate feature subsets by taking the regression model into account. The advantage of a hybrid approach is threefold. First, the filter provides interesting features independently from the regression model and, hence, allows for an easier interpretation. Secondly, because the filter part is computationally less expensive, the global algorithm will faster provide good candidate subsets compared to a stand-alone wrapper approach. Finally, the wrapper takes the bias of the regression model into account, because the regression model guides the search for optimal features. Results are shown for the 'Boston housing' and 'orange juice' benchmarks based on the multilayer perceptron regression model.

1 Introduction

Feature selection and feature construction have been addressed by many researchers in statistics and machine learning, see [1] for a recent overview. Feature construction constructs new features from the original inputs in a linear or non-linear way. Most feature construction techniques are developed for classification problems. However, they are easily adapted for regression problems by first discretizing the continuous target values using class-blind discretization algorithms [2], hence, artificially creating class labels. Feature selection on the other hand considers a selection from the original inputs, without constructing new ones. Both feature construction and feature selection help tackling the curse of dimensionality. In reducing the number of inputs one searches for the optimal bias-variance trade-off: a large number of inputs imply that more parameters need to be estimated and this causes a larger variance, however a too small number of inputs increases the bias. Feature construction has the disadvantage that it does not preserve the semantics of the inputs: combining inputs in a linear or non-linear way, makes the new features hard to interpret and hence makes an understanding of the nature of the problem difficult. Another huge disadvantage is

S. Kollias et al. (Eds.): ICANN 2006, Part I, LNCS 4131, pp. 31 – 40, 2006.

that feature construction does not decrease the measuring cost: all inputs still need to be measured, by possibly very expensive sensors, even when they are non-informative.

Therefore we adopt a feature subset selection approach in this article. Feature selection can be separated in two approaches: the *filter* approach and the *wrapper* approach [3]. In the filter approach the feature subset selection is performed independently of the training of the regression model. In this case feature subset selection is considered as a preprocessing step to induction. This is computationally more efficient, but ignores the fact that an optimal selection of features is dependent on the regression model. As stated before the performance of the regression model is strongly dependent on the size of the feature subset. On the other hand the wrapper approach is computationally more involved, but takes the interactions of the feature subset and the regression model into account. The term 'wrapper' stems from the fact that the feature selection is wrapped around the regression model which is considered as a black-box. In this article we propose a hybrid solution: first irrelevant and largely redundant features are removed, subsequently a search with a wrapper is performed among the features that passed the filter.

2 Filter Preprocessing

In this section we investigate an information-theoretic measure in order to determine irrelevant and redundant features. We use the 'Boston housing' and the 'orange juice' datasets for illustrative purposes. The proposed methods are inherited from [4] where a hybrid approach is proposed for pattern recognition (classification), instead of regression.

2.1 Irrelevance Determination by Permutation

As explained before, a wrapper approach takes the limitations of the particular regression model into account. In the search for optimal feature subsets we need to estimate the performance of the feature sets found so far. This requires the training of the regression model based on the selected features for a chosen training set. The accuracy of the model is then estimated by simulating the trained regression model on a test set. It is common that a lot of features are included in the feature subset search that do not contain any information about the target variable. This information can be described by the concept of mutual information between the regression variable F_i and the target variable T:

$$I(F_i,T) = \int_{f_i}\int_{t} P(f_i,t)\log_2\left(\frac{P(f_i,t)}{P(f_i)P(t)}\right)df_i dt \ . \tag{1}$$

The use of the mutual information in regression is largely motivated by the data inequality theorem, which states that [5]:

$$I(F_i,T) \geq I(g(F_i),T) \, . \tag{2}$$

Hence, a function of the variable F_i cannot increase the information about the target T. If we can show that the original variable T is not dependent on F_i (F_i is not informative about the target T), which implies the mutual information in (1) is equal to 0, we can discard F_i, because any further processing can not increase the information about the target.

In practice we face the problem that we do not know the joint distribution between target variable and input variables, hence, the mutual information needs to be estimated from the data. This finite sample estimate is likely to be different from 0 and in general will depend on the sample size, parameter settings of the estimator and the distributions in (1). Thus, looking whether the estimated mutual information is exactly equal to 0 is not satisfactory. However, we can easily circumvent this problem in the following way. We define a hypothesis test where the null hypothesis H_0 tests the assumption that the feature variable and the target variable are independent. We can easily obtain the distribution of the mutual information conditioned under the particular sample distributions. Therefore, we randomly permute the ordering of the samples of the target variable, hence, removing the dependencies between the target variable and the input variable, relative to the feature samples. Performing this permutation N times provides us with N samples of a sample distribution of the mutual information under the H_0 hypothesis. Note that this strategy contains some resemblance with the creation of surrogate time series in time series analysis [6]: a 'ground-truth' or reference is created by e.g. randomly permuting the phase of the signals under the given sample distribution of the frequency spectrum.

Further on, we will estimate the mutual information with the $I^{(1)}$ estimator of Kraskov et al. [7], which estimates mutual information directly from a K-Nearest Neighbour method. Figure 1 shows the sample distribution of the mutual information between input variable F_5 (nitric oxides concentration, NOX) and the target variable (median value of owner-occupied homes, MEDV) of the 'Boston housing' data set for 1000 permutations. We note that under the H_0 hypothesis the mean (0.1106) of the mutual information is considerably different from 0. This divergence from 0 can be partly explained by the fact that the $I^{(1)}$ is designed for continuous distributed features and target variables. However, the NOX variable appears to have a discrete nature (although in the accompanying housing.names file it is considered as a continuous feature). Based on the sample distribution we can define a threshold for which a feature (when larger than the threshold) can be considered statistically relevant. For the example in figure 1 we have $P_{0.01} = 0.1369$, the actual MI (without performing the permutation) is equal to 0.1920. When we perform this analysis for all 13 features in the 'Boston housing' dataset we find that all features are statistically relevant, except for input variable F_4 (Charles River dummy variable CHAS, $P_{0.01} = 0.02$ and actual MI equal to 0.0163). We remark that the CHAS variable is discrete and therefore we did not use the $I^{(1)}$ estimator, but estimated the mutual information by means of the marginal entropy estimator (marginal and conditional entropies estimated from formula (20) in [7]):

$$\hat{I}(F,T) = \hat{H}(T) - \sum_{i=0,..,C} \hat{H}(T \mid F = i).\hat{p}(F = i) \; . \tag{3}$$

In formula (3) the discrete feature F (CHAS for the 'Boston housing' data set) takes different category values: 0,…, C (0 and 1 in this case).

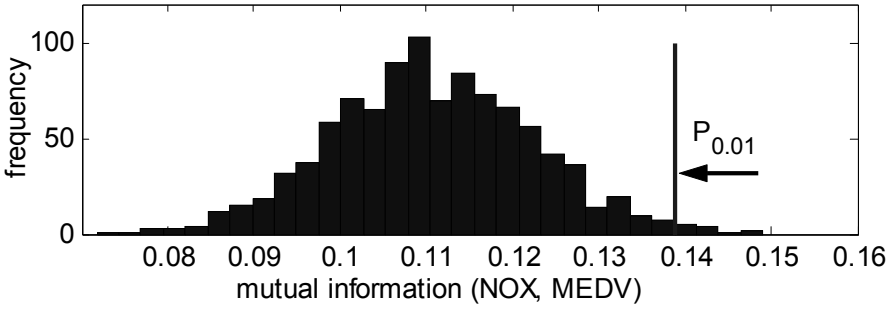

Fig. 1. Sample distribution of the mutual information between input variable NOX and target variable MEDV. The distribution was obtained under null hypothesis (independent input and target variable) by randomly permuting (1000 permutations) the samples. Note that the mean differs considerably from 0. The actual MI is equal to 0.1920, this is larger than $P_{0.01}$ ($P_{0.01}$ = 0.1389) and hence NOX can be considered as a relevant feature for target variable MEDV.

In figure 2 we show the mutual information of all features of the 'orange juice' database and the $P_{0.01}$ thresholds from 100 permutations. Where the MI exceeds the threshold the features are statistically relevant.

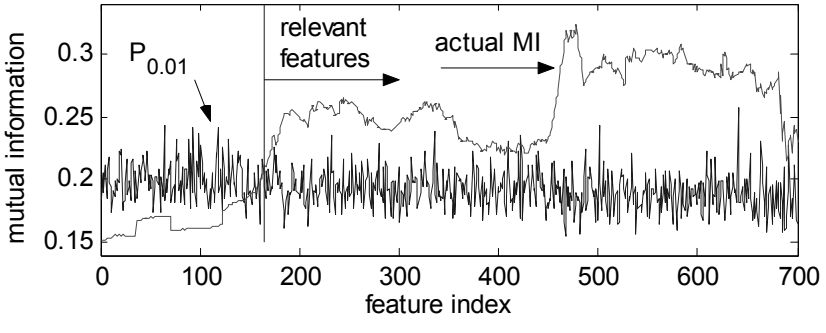

Fig. 2. Feature relevance as determined by the $P_{0.01}$ value of the permutation test (using 100 permutations). The lower noisy curve shows the $P_{0.01}$ value determined from the permutations. The upper curve shows the actual MI (without permutations). Note that starting from approximately feature 165 all features are considered as statistically significant.

Finally, we remark that permutation testing for feature relevance analysis has been described independently in [8].

2.2 Redundancy Detection

Features that are individually relevant might, however contain overlapping information considering the target variable. Therefore in literature [9] the distinction is made

between strongly relevant and weakly relevant features. A strongly relevant feature F_i is defined as:

$$P(T \mid F_i, G_i) \neq P(T \mid G_i) \tag{4}$$

$$G_i = F - \{F_i\},$$

with F the complete feature set.

A weakly relevant feature F_i is a feature for which (4) holds for at least one strict subset G_i' of G_i. So weakly relevant features need to be interpreted as relevant features, but for which redundant, i.e. strongly correlated, features or feature sets exist.

A redundancy filter tries to detect and remove the redundant features of the weakly relevant features. Thus, the redundancy filter needs to filter out redundant feature subsets, but needs to retain a representative feature for the redundant subset. From formula (4) it is clear that we need to rely on heuristics for the identification of weakly relevant features:

- We do not dispose of the real underlying distributions in (4),
- It requires that we find at least one subset of G_i for which the inequality in (4) holds, in a worst case scenario this requires considering all possible subsets of G_i. This is of almost the same complexity as solving the FSS problem itself, because this would require finding the smallest possible subset of the complete feature set for which the equality in (4) holds.

The heuristic approach is taken where redundant features are assembled in a cluster and a representative feature is taken out of the cluster. The feature closest to the cluster centroid can act as a representative feature for all features in the cluster. We have following requirements for the clustering procedure:

- A first requirement for the clustering procedure is: strongly relevant features must form a cluster on themselves. Therefore in the clustering procedure it is sound to consider every feature initially as a separate cluster.
- A second requirement is that the maximum distance between any features in a cluster should be limited in order for the feature closest to the centroid to be representative.

In order to achieve these goals clusters are iteratively merged starting from the initial features as seeds. In order to obtain compact clusters, when merging, the distance between 2 clusters D_i and D_j is defined as the maximum distance between any features:

$$d_{\max}(D_i, D_j) = \max_{\substack{\mathbf{F}_i \in D_i \\ \mathbf{F}_j \in D_j}} \left\| \mathbf{F}_i - \mathbf{F}_j \right\|. \tag{5}$$

As a distance measure between features we propose 1 minus the normalized MI between features, this leads to 0 distance for the distance between the same features and a distance of 1between independent features.

Cluster merging is stopped when distance between any clusters exceeds a predefined threshold τ (maximal number of clusters to be formed or a maximal distance that may not be exceeded when merging clusters). The described clustering procedure is known as hierarchical agglomerative clustering with a 'complete' merging strategy of the clusters [10]. The threshold τ heuristically defines the non-redundant features which are represented by the cluster centroids.

A wrapper search is then performed on the features that pass both the relevance and redundancy filter. If we apply this redundancy filtering strategy to the 'orange juice data' and set τ equal to 5 clusters we get following result in figure 3.

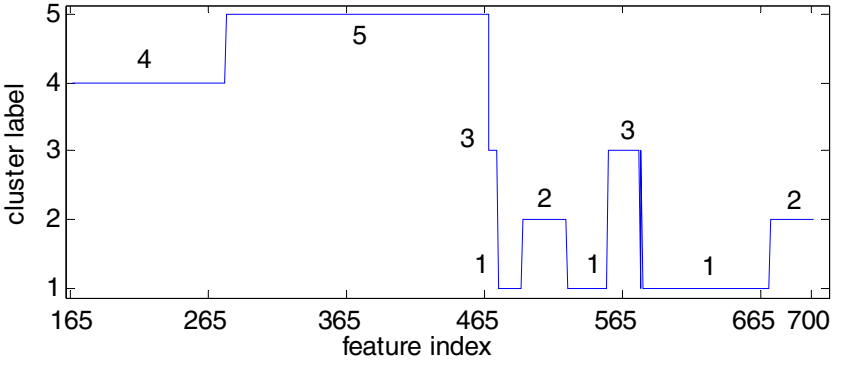

Fig. 3. Redundancy analysis on spectral 'orange juice' data. The figure shows which features are assigned to which cluster if we set τ equal to 5 clusters in the redundancy analysis. It is interesting to observe that contiguous features tend to end up in the same cluster. This could be expected, while small differences in spectral components tend to give rise to redundant features. As a distance measure 1-nMI (normalized mutual information) was used.

From figure 3 we observe the interesting (but expected) result that contiguous features tend to be assigned to the same cluster, hence, features that are obtained from small differences in spectra tend to be redundant. We obtained a similar result for features computed from the continuous wavelet transform in [4]: features obtained for small changes in scale coefficients tend to be strongly dependent and therefore can be approximated by the cluster centroid [4]. Note that this redundancy analysis can be considered as a strategy of sampling from the initial feature space. The sampling strategy has the advantage that where features are strongly redundant we need only a few representative features, while where features are not redundant we need more feature samples.

2.3 Wrapper Search

A supervised search is performed on the features that pass both the relevance and redundancy filter. Given the strong dependency of the regression model on the curse of dimensionality and the assumptions made in the regression model to map input variables to a target variable, these interactions need to be taken into account to achieve optimal performance. By applying filter techniques the wrapper is focused on

strongly relevant features. By applying the filtering techniques the wrapper can be applied with decreased computational cost. In the wrapper approach 2 choices need to be made: the regression model and the search among the possible subsets. We opted for the following choices:

- Regression model: we used a widely accepted model for regression: a Multi-layer Perceptron (MLP) neural network [11]. Such MLP models are capable of approximating any function on a finite interval, provided the number of hidden neurons and the training data set are large. The input layer is defined by the number of inputs (D), for the hidden layer we choose 5 sigmoid neurons, the output layer is determined by the number of targets and consists of 1 linear neuron. The Levenberg-Marquardt algorithm was used in batch-mode to train the parameters of the network. To compute the performance of the feature subset, the data set was divided in 3 parts: a training set, a validation set and a test set. The validation set was used to avoid overtraining of the network, hence, when the error on the validation set increased the training was stopped. The intermediate performance of the feature set was then estimated on the test-set. This was repeated 10 times by using a 10-fold cross-validation procedure, the final performance of the test set was obtained from the averages of the intermediate performances.

- Search procedure: several search procedures have been proposed to the feature subset selection problem, although most often research has been focusing on pattern classification. Among the best well-known search procedures in feature selection for pattern classification are: exhaustive search, branch and bound [12], sequential search algorithms (SSA's) [13] and more recently Genetic Algorithms (GA's) [4], [14], [15]. We focus on GA's, because in a comparative study [15] it was shown that GA's can compete with the best search algorithms (SSA's) for feature subset selection and even outperform SSA's for larger feature sets (typically when the number of features is larger than 50). The 'roulette wheel' selection strategy was chosen, where the fitness function was determined by:

$$\text{fitness}(\{F_i\}) = \begin{cases} (\text{Var}(T)-\text{MSE}(\{F_i\}))^n & \text{, if MSE}(\{F_i\}) < \text{Var}(T) \\ 0 & \text{, if MSE}(\{F_i\}) \geq \text{Var}(T). \end{cases} \quad (6)$$

From (6) we observe that any feature subset (F_i) with a mean square error performance (MSE) smaller than the variance of the target variable, gets rewarded. Parameter n controls 'selective pressure' [14]: a higher n will reward good solutions disproportionally. We set n equal to 2. Finally, we used following settings in the GA: the probability of cross-over between individuals (an individual is a particular feature subset) p_c is equal to 0.3, the probability of mutation p_m of a feature within every individual equal to 0.01, the number of individuals per population equal to 30 and the number of populations equal to 100.

Finally, in figure 4 a schematic overview of the overall feature subset selection strategy is presented.

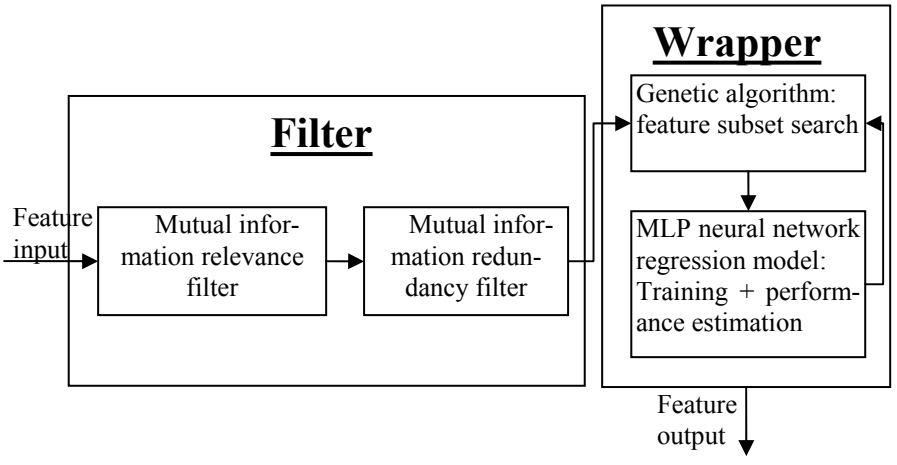

Fig. 4. Schematic overview of the overall feature subset selection strategy for regression. First, irrelevant and redundant features are removed in the filter. Second, the wrapper approach focuses on the smaller set of interesting features.

3 Results

3.1 Boston Housing

We summarize the application of the FSS strategy of figure 4 in table 1 for the 'Boston housing' data for feature subset sizes 1 to 5.

We remark that the relevance analysis showed that only feature 1 is irrelevant (CHAS feature) and that the smallest distance between any features is equal to 0.461 (features RAD: index of accessibility to radial highways and TAX: full-value property-tax rate). We performed further simulations with feature subsets up to all features (13). The best feature subset obtained contained 12 features (feature 4 not included) and has an MSE of 14.83, however, none of the results obtained with more than 3 features could be proven to be statistically significant compared to the result with 3 features. Feature 4 was never included in the smaller subset sizes and thus could have been successfully removed by the relevance filter.

Table 1. Performance of the MLP feature subset strategy on the 'Boston housing' data

Feature subset size	*Feature list (1-13) Best solution*	*Performance (MSE)*
1	[13]	28.08 ± 0.75
2	[6 13]	21.66 ± 1.25
3	[3 6 13]	18.81 ± 1.80
4	[2 5 6 13]	18.96 ± 1.63
5	[6 8 9 11 13]	16.03 ± 1.13

3.2 Orange Juice

Table 2 presents the results of the MLP FSS strategy on the 'orange juice' data set. This data set has been made available by the BNUT unit of the UCL (Université Catholique de Louvain). In performance1 the results of the algorithm of figure 4 without the filter and in performance2 the results with filter (with τ equal to 25 clusters) are tabulated.

Table 2. Performance of the MLP feature subset strategy on the 'orange juice' data

Feature subset size	*Performance1*		*performance2*	
	MSE	*time*	*MSE*	*Time*
5	49.64	301	55.64	254
6	50.91	299	58.36	280
7	69.63	352	53.78	279
8	57.25	360	59.45	325
9	54.76	410	60.57	386
10	57.94	485	46.48	461
11	51.73	641	56.98	520
12	54.71	534	47.28	545
13	38.04	680	49.81	416
14	49.14	750	46.98	463
15	52.34	647	48.66	489

MSE is the performance in 'mean square error' of the best feature subset found, *time* is the total number of times a 10-fold cross-validation procedure (this means: training, validation and testing) was needed over 100 populations to estimate the performance of a feature subset (one has a maximum of 30*100 evaluations). Once this performance for a subset is computed, it can be stored and thus it does not need to be recomputed if the feature subset reappears in future populations. Reappearance of performing subsets is very likely (and expected), due to the fitness selection strategy. The increased performance in speed (lower time) in table 2 can be explained by the reduction of the 700 features to 25 features used in the wrapper: crossover and mutation are more likely to generate previous occurring individuals. Hence, while an increase in speed for a hybrid approach would be evident for search strategies such as: exhaustive search, SSA's, greedy search and so on, it is less evident for GA's, when keeping the number of individuals per population and the number of populations fixed. Furthermore, a paired t-test on the MSE's (mean square errors) shows that the performance of the 2 approaches is equivalent; on the other hand a paired t test shows that the difference in number of evaluations needed is statistically significant. The cost of the filter preprocessing can be ignored if a limited number of permutations are performed.

4 Conclusions

We have shown that relevance and redundancy analysis helps interpreting the data under study. Permutation tests are used to find statistically motivated thresholds that determine statistically relevant features. Finally, it was shown that the filter preprocessing increases the speed of the wrapper approach in the feature subset search.

Acknowledgements

This first author was supported by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT Vlaanderen). The second author is funded by the Belgian Fund for Research -- Flanders (G.0248.03, G.0234.04).

References

1. Guyon, I. Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3 (2003) 1157-1182.
2. Kurgan, L.A., Cios, K.J.: CAIM Discretization Algorithm. IEEE Transactions on Knowledge and Data Engineering 16 (2004) 145-153.
3. Kohavi, R., John G. H.: Wrappers for Feature Subset Selection. Artificial Intelligence 97 (1997) 273-324.
4. Van Dijck G., Van Hulle M. M., Wevers, M.: Hierarchical Feature Subset Selection for Features Computed from the Continuous Wavelet Transform. 2005 IEEE Workshop on Machine Learning for Signal Processing (2005) 81-86.
5. Cover, T. M., Thomas, J. A.: Elements of information theory. John Wiley & Sons, New York (1991).
6. Schreiber, T., Schmitz, A.: Surrogate Time Series, Physica D 142 (2000) 346 – 382.
7. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating Mutual Information. Phys. Rev. E. 69 (2004) 066138.
8. Francois, D., Wertz, V., Verleysen, M.: The Permutation Test for Feature Selection by Mutual Information. European Symposium on Artificial Neural Networks (2006) 239-244.
9. John, G., Kohavi, R. Pfleger, K.: Irrelevant Features and the Subset Selection Problem. In Proc. of the Eleventh Int. Conf. on Machine Learning, (1994) 121-129.
10. Duda, R.O., Hart, P.E., Stork, D. G. *Pattern Classification*, John Wiley & Sons Inc., New York (2001).
11. Bishop, C.M. Neural Networks for Pattern Recognition. Oxford University Press Inc., New York (1997).
12. Narendra, P. M., Fukunaga, K.: A Branch and Bound Algorithm for Feature Subset Selection. IEEE Trans. Computers 26 (1977) 917-922.
13. Pudil, P., Novovicova, J., Kittler, J., Floating Search Methods in Feature Selection. Pattern Recognition Letters 15 (1994) 1119-1125.
14. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996).
15. Kudo, M., Sklansky, J., Comparison of Algorithms that Select Features for Pattern Recognition. Pattern Recognition 33 (2000) 25-41.