Mutual Information Based Initialization of Forward-Backward Search for Feature Selection in Regression Problems

Alberto Guillén[1], Antti Sorjamaa[2], Gines Rubio[3],
Amaury Lendasse[2], and Ingacio Rojas[3]

[1] Department of Informatics
University of Jaen, Spain
[2] Department of Computer Architecture and Technology
University of Granada, Spain
[3] Department of Information and Computer Science
Helsinki University of Technology, Finland

Abstract. Pure feature selection, where variables are chosen or not to be in the training data set, still remains as an unsolved problem, especially when the dimensionality is high. Recently, the Forward-Backward Search algorithm using the Delta Test to evaluate a possible solution was presented, showing a good performance. However, due to the locality of the search procedure, the initial starting point of the search becomes crucial in order to obtain good results. This paper presents new heuristics to find a more adequate starting point that could lead to a better solution. The heuristic is based on the sorting of the variables using the Mutual Information criterion, and then performing parallel local searches. These local searches provide an initial starting point for the actual parallel Forward-Backward algorithm.

1 Intoduction

Input selection is a crucial part when building an approximator. Too many input variables increase the calculation time and model complexity and even lead to suboptimal results. On the other hand too few variables might not contain all the relevant information for an accurate approximation.

In many cases, the approximator cannot be used to test all possible combinations of variables in order to find the optimal one. That can be due to the huge number of combinations, which increases exponentially with respect to the number of variables, or due to the fact that not all approximators can distinguish the relevant inputs from the bogus ones.

In this paper, a greedy selection methodology, called Forward-Backward Search, is used to select the variables. It relies on the Delta Test estimation methodology [1]. Even though Forward-Backward is not going through all possible solutions and does not guarantee the optimality of the final selection, it always finds a local optimal one.

Because of the locality of the Forward-Backward, a good initialization is crucial, and this paper presents several new heuristics for the initialization. The Forward-Backward methodology is also deterministic with respect to the initialization; the same initial selection of variables provides the same final solution.

The rest of the paper is organized as follows: Section 2 presents the Forward-Backward Search algorithm and the theoretical background of the Delta Test. Then, Section 3 introduces the new improvements incorporated to enhance the variable selection. Afterwards, Section 4 shows an experimental result, where the heuristics are briefly compared.

2 Forward-Backward Search

Forward-Backward Search (FBS) is an algorithm that results from the joining of two methodologies: Forward and Backward selections [2]. Both the Forward Selection and the Backward Elimination (or Pruning) methods suffer from an incomplete search. The FBS offers the flexibility to reconsider input variables previously discarded and *vice versa*, to discard input variables previously selected. It can start from any initial input set, including empty, full or randomly initialized input set.

Let us suppose a set of inputs $X^i,\ i = 1, 2, \cdots, d$ and output Y, the procedure of the Forward-Backward Search is summarized in Figure 1. In the procedure example the k-Nearest Neighbors (kNN) criterion [3] is used as an example criterion for evaluating the input set, but the criterion can be almost any criteria or a suitable approximator.

1. (Initialization)
   Let set S be the selected input set, which can contain any input variables, and set F be the unselected input set containing the inputs, which are not in set S. Compute $k\mathrm{NN}(S, Y)$ error.
2. (Forward-Backward Search)
   Find:
   $$X^s = \arg\max_{X^i, X^j} \{k\mathrm{NN}(\{S, X^j\}, Y)\} \cup \{k\mathrm{NN}(S \backslash X^i, Y)\}, X^i \in S, X^j \in F.$$
   If the previously computed kNN error is smaller than the new kNN error , stop; otherwise, update set S and save the new kNN error, repeat step 2 until S is equal to any former selected S.
3. (Result)
   The selection result is in set S.

Fig. 1. Forward-Backward Search Strategy.

It is noted that the selection result depends on the initialization of the input set. In this paper, several options are considered and the options are discussed more deeply in Section 3.

In the course of FBS procedure, the number of evaluated input sets varies and is dependent on the initialization of the input set, the stopping criteria and the nature of the problem. Still, it is not guaranteed that in all cases this selection method finds the global optimal input set.

2.1 The Delta Test

The Delta Test (DT), introduced by Pi and Peterson for time series [4] and proposed for variable selection in [1], is a technique to estimate the variance of the noise, or the mean squared error (MSE), that can be achieved without overfitting. Given N input-output pairs $(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, the relationship between $\mathbf{x}_i$ and y_i can be expressed as

$$y_i = f(\mathbf{x}_i) + r_i, \qquad i = 1, ..., N \tag{1}$$

where f is an unknown function and r is the noise. The DT estimates the variance of the noise r.

The DT is useful for evaluating the nonlinear correlation between two random variables, namely, input and output pairs. The DT can also be applied to input variable selection: the set of input variables that minimizes the DT is the one that is selected. Indeed, according to the DT, the selected set of input variables is the one that represents the relationship between input variables and the output variable in the most deterministic way.

The DT is based on a hypothesis coming from the continuity of the regression function. If two points $\mathbf{x}$ and $\mathbf{x}'$ are close in the input space, the continuity of the regression function implies that the outputs $f(\mathbf{x})$ and $f(\mathbf{x}')$ are also close enough in the output space. Alternatively, if the corresponding output values are not close in the output space, this is due to the influence of the noise.

The DT can be interpreted as a particularization of the Gamma Test [5] considering only the first nearest neighbor. Let us denote the first nearest neighbor of a point $\mathbf{x}_i$ in the $\mathbb{R}^d$ space as $\mathbf{x}_{NN(i)}$. The nearest neighbor formulation of the DT estimates Var$[r]$ by

$$\mathrm{Var}[r] \approx \delta = \frac{1}{2N} \sum_{i=1}^{N} (y_i - y_{NN(i)})^2, \text{with } \mathrm{Var}[\delta] \to 0 \text{ for } N \to \infty \tag{2}$$

where $y_{NN(i)}$ is the output of $\mathbf{x}_{NN(i)}$.

3 New Initialization Heuristics for the Forward-Backward Search

The following subsections describe the different approaches proposed to define an adequate starting point. The original FBS has been parallelized in order to take advantage of the architectures available nowadays. The search for the best solution is distributed to several computers in order to have more solutions in less time.

The parallel implementation is quite straightforward and consists of the division of the generated subsets of variables from the first iteration of the FBS. This division is shown in Figure 2.

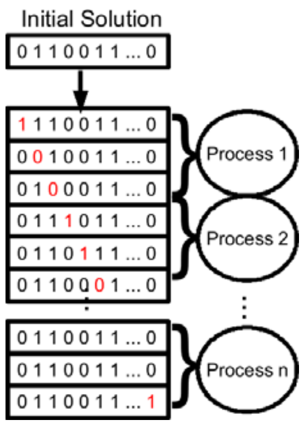

Fig. 2. Parallel scheme for the Forward-Backward Search.

Once each process has a part of the subset, they proceed as the original FBS. The algorithm stops when all the processes have converged to a solution. Then, the best solution is found among the final solutions of the individual processes.

3.1 Simple Mutual Information Based Initialization

Let $X^l = \{\boldsymbol{x}_m^l\}$ with $l \in 1, ..., d$ (i.e. X^l is the l-th input variable) and $Y = \{y_m\}$ with $\{m = 1...M\}$. The Mutual Information (MI) between X^l and Y can be defined as the amount of information that X^l provides about Y, and can be expressed as:

$$I(X^l, Y) = \sum_{y \in Y} \sum_{x \in X} \mu_{X^l,Y}(x, y) \log \frac{\mu_{X^l,Y}(x, y)}{\mu_X^l(x)\mu_Y(y)}. \qquad (3)$$

$\mu_{X^l,Y}$ is the joint probability distribution function of X^l and Y, and $\mu_X^l(x)$ and $\mu_Y(y)$ are the marginal probability distribution functions of X^l and Y respectively.

Therefore, in order to obtain the MI between X^l and Y, only the estimate of the joint probability density function is needed. This value can be computed using several techniques based on histograms, kernels or the kNN. In this paper, the one based on the kNN is used [6].

For each input variable, the MI between that variable and the output is computed and, once finished, it is possible to rank all the input variables according the values of MI. Then, the initial solution for the FBS is defined as a number of $first$ variables in the ranking. The problem now is to determine the actual number of variables, since the value obtained by the MI is not enough to perform this selection. Unfortunately, the only chance is to set this value manually.

Another issue is to determine the value of k for the kNN algorithm that computes the MI. Although there is a possibility of guessing an adequate value: compute the MI values using several k and select the one that provides the highest values.

3.2 RaVI: Ranked Variables Initialization

Being aware of the two significant drawbacks of the used MI heuristic (the determination of the k and the final number of variables to be chosen), another heuristic is considered. This new heuristic requires the definition of only one parameter. This value can again be set manually as in the MI, or as a function of the available computational resources making the heuristic more flexible when executed in different computer architectures or systems.

The heuristic works as follows: it performs a division of the original input vector into subvectors of a smaller dimension. Then, the local search is applied to each subvector.

When the search is focused on the subvector, there are several possibilities in handling the rest of the inputs. In this paper, we are considering four starting alternatives: 1) all zeros, 2) subvectors ones, the other inputs zeros, 3) all ones, 4) subvectors zeros, the other inputs ones.

The RaVI scheme is summarized in Figure 3.

In the Figure, the initial solutions are divided into slices of size 3 (depicted as x) and the remaining values are not changed (depicted as -) during the first FBS. This first FBS is done sequentially and separately in each process. Once all the processes have converged to a local optima, the processes perform a collective communication and share the results found by the other processes. Then, the initial starting point for the parallel FBS (named as middle solution in Figure 3) is computed by concatenating all local solutions. Finally, the parallel FBS, presented in Figure 2, is performed starting from the middle solution.

The new aspect of this method is the sorting of the variables before performing the slice division and the following local searches. Since the variables are going to be analyzed with their neighbors in a local manner, it is convenient to rank the variables with high MI values close to each other when performing the local searches. When performing the MI, in this case, it is not so crucial to select the k for the MI as the most optimal one, because each of the slices will contain similar variables in terms of their MI values. When the local results are concatenated, the middle solution includes inputs with a wide range of MI values, only the optimality inside each slice is emphasized.

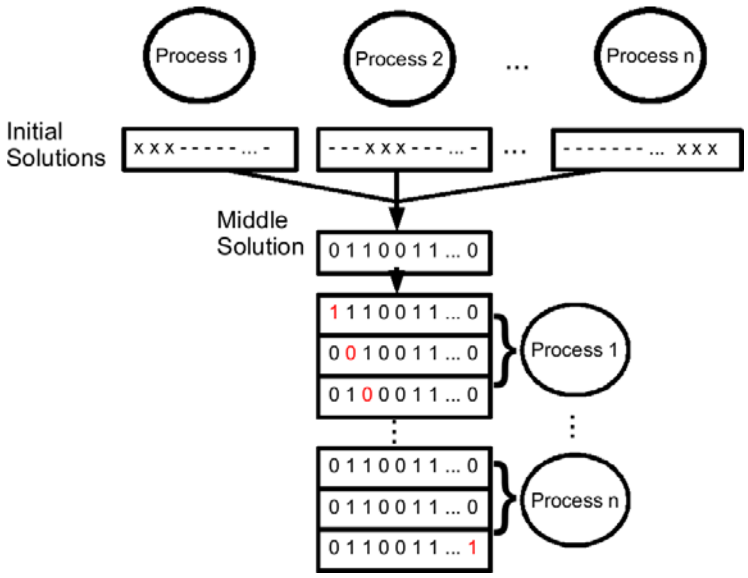

Fig. 3. Scheme of the algorithm using the slice division heuristic to initialize the starting point for the FBS.

Here, we use two different sorting schemes based on the MI criterion. The first one sorts the variables in a descending order, starting from the variable with the largest MI value and ending up with the one with the lowest.

The second sorting scheme, called RaVI Mix, is aiming to bunch together variables with high and low values of MI. This sorting gives more chances to the sublocal searches to select good variables, whether they have large MI value or not. The second sorting scheme is visualized in the following:

$$\left[X^{\mathrm{MI}(1)}\ X^{\mathrm{MI}(d)}\ X^{\mathrm{MI}(2)}\ X^{\mathrm{MI}(d-1)} \cdots X^{\mathrm{MI}(m)}\right], \tag{4}$$

where MI denotes the ranking of all d inputs in the dataset, MI(1) denotes the input with the highest MI value and MI(d) the one with the lowest. m is the middlemost input in the ranking.

After the sorting, the input space is divided into sublocal search spaces, or slices, as demonstrated in Figure 3.

4 Experiments

In this paper, we use a dataset from the recently organized ESTSP 2007 conference. The ESTSP dataset presents a weekly sea temperature for roughly 17 years and contains 874 values and it is shown in Figure 4.

The dataset is transformed into a regressor of 55 input variables, and one output variable, and a total of 819 samples using a sliding window over the

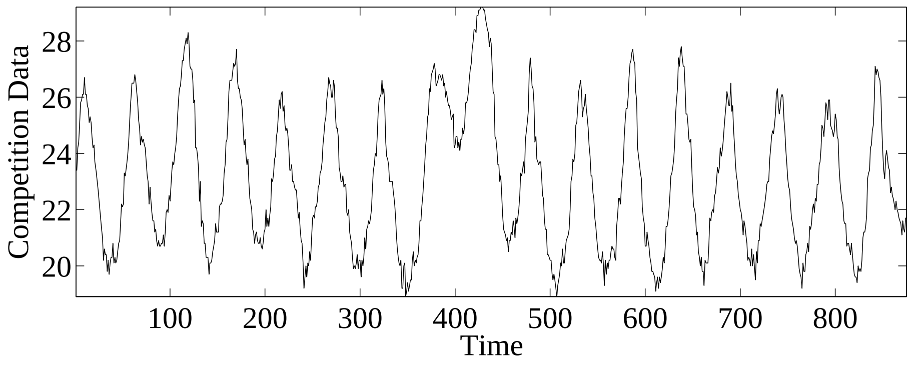

Fig. 4. The ESTSP 2007 Competition dataset.

whole dataset. We have a time series prediction problem of one step ahead and we want to do the input variable selection in order to decrease the amount of input variables. All the samples are used in the variable selection part.

We used 8 processes (or processors) with each method, which set the sizes of the slices to 6 or 7 (7 slices with 7 variables and one with 6 variables). As already mentioned, the number of processes can be defined manually or with respect to the computational resources available. The parallel implementation was performed using Matlab software and recently developed MPI implementation [7].

In the preliminary tests performed before the actual comparison of the methods, a well-known Housing dataset was used to verify the correctness of the methods. Because Housing dataset includes only 13 inputs, it is possible to use exhaustive search to compute the global optimum according to the Delta Test. Using any of the presented heuristics with any presented initialization, the global optimum was always found.

Table 1 summarizes the results of all methods using the ESTSP 2007 Competition dataset.

From Table 1 we can see that the RaVI Mix methodology obtains the lowest Delta Test value and, therefore, has selected the best set of inputs. However, there are no big differences among the methods, even though the Delta Test value with all variables is 1.1765. It means that each of the presented heuristics have done their job adequately.

The Exhaustive Search of all possible combinations in the 55 dimensional space is clearly unfeasible task and one must use some sort of heuristics to ease the search process. The Table 1 shows that the MI alone is not enough to guide the FBS toward more optimal selection of input variables. Even though the Simple MI heuristic searched through the largest amount of input combinations, it was not able to find better solution than RaVI Mix heuristic.

Furthermore, we observed that although there is a possibility to end up in the same solutions when using the parallel implementation of the FBS, only

Table 1. Results of all methods using the ESTSP 2007 Competition dataset.

Heuristic	Starting Point	# Variables	Delta Test	Solutions	Time
pFBS	All	46	0.0284	3856	248
	None	18	0.0299	4640	168
Simple MI	10 best MI	15	0.0272	8056	222
RaVI	All	22	0.0269	4243	194
	None	28	0.0277	6629	282
	Ones and Zeros	19	0.0267	3541	145
	Zeros and Ones	31	0.0283	5398	236
RaVI Mix	All	21	0.0284	3755	156
	None	32	**0.0264**	5445	289
	Ones and Zeros	25	0.0269	4077	148
	Zeros and Ones	28	0.0293	6358	273

few searches ended up with the same local minima. Roughly only one percent of the solutions searched through were already evaluated by another parallel computation thread.

Figure 5 shows the selected variables using the RaVI Mix selection scheme and some inputs with high and low MI values for comparison.

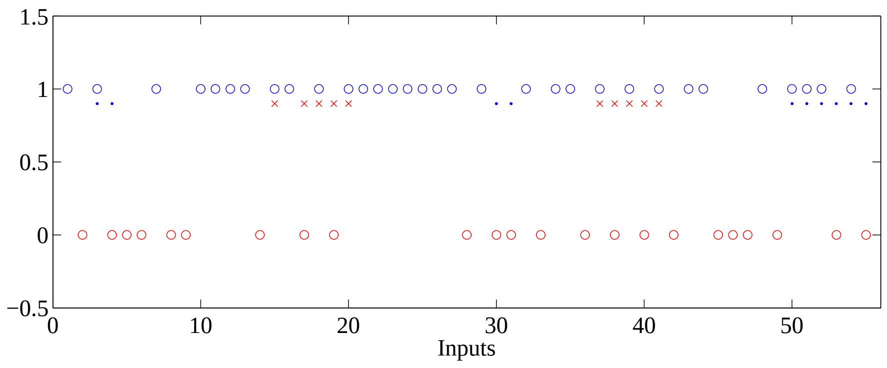

Fig. 5. Selected inputs using the RaVI Mix method starting from all zeros. The input variables are on the horizontal axis, red circles at zero depict not selected variables and blue circles at one depict selected variables. Blue dots at 0.9 denote the variables with 10 highest MI values and red crosses the ones with the 10 lowest values.

From Figure 5 we can see that the selection is not selecting all variables with the highest MI values, but also the ones with very low value. For example, variables 30 and 31 are among the variables with the highest MI values, but none of them is selected. On the other hand, variables from 37 to 41 are among the variables with the lowest MI values, but some of them are chosen by the RaVI Mix.

This suggests that the MI value alone is not able to give a clear justification to use the variable, and that also the variables with low MI value can be useful in the approximation.

5 Conclusions

The problem of finding a good subset of variables for any kind of regression model is still remaining as an unsolved problem. Due to the high dimensionality of the real-life problems, it is not possible to apply an exhaustive search that would provide the global optimum.

Within this context, this paper presents several heuristics to improve the behavior of a previously published algorithm, the Forward-Backward Search. These new heuristics rely on the theoretical basis provided by the Mutual Information. The search starts from a point that could be closer to adequate local minimum.

Another relevant aspect of one of the heuristics is the possibility to analyze the relationships between variables, defining neighbor relationships. This aspect can be further studied using different metrics, since this last heuristic provided the best results in a complex input selection problem.

For further work, other ranking criteria are tested and the effect of using different sizes of local searches are quantified and compared. Also other means of local estimation of a good selection of inputs are tried.

Acknowledgment

Part of the work of Antti Sorjamaa is supported by a grant from Nokia Foundation, Finland.

References

1. Eirola, E., Liitiäinen, E., Lendasse, A., Corona, F., Verleysen, M.: Using the delta test for variable selection. In: ESANN 2008, European Symposium on Artificial Neural Networks, Bruges (Belgium). (April 2008)
2. Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., Lendasse, A.: Methodology for long-term prediction of time series. Neurocomputing **70**(16-18) (October 2007) 2861–2869
3. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
4. Pi, H., Peterson, C.: Finding the embedding dimension and variable dependencies in time series. Neural Computation **6**(3) (1994) 509–520
5. Jones, A.J.: New tools in non-linear modelling and prediction. Computational Management Science **1**(2) (2004) 109–149
6. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Phys. Rev. **69** (2004) 66–138
7. Guillen, A., Rojas, I., Rubio, G., Pomares, H., Herrera, L.J., Gonzalez, J.: A new interface for MPI in matlab and its application over a genetic algorithm. In Lendasse, A., ed.: Proceedings of the European Symposium on Time Series Prediction. (2008) 37–46 Download: http://atc.ugr.es/∼aguillen.