Abstract—An experimental study of several decision issues for wrapper Feature Selection with Multi-Layer Perceptrons is presented, namely the stopping criterion, the data set where the saliency is measured and the network retraining before computing the saliency. Experimental results with the Sequential Backward Selection procedure indicate that the increase in the computational cost associated with retraining the network with every feature temporarily removed before computing the saliency is rewarded with a significant performance improvement. Despite being quite intuitive, this idea has been hardly used in practice. Regarding the stopping criterion and the data set where the saliency is measured, the procedure profits from measuring the saliency in a validation set, as reasonably expected. A somehow non-intuitive conclusion can be drawn by looking at the stopping criterion, where it is suggested that forcing overtraining may be as useful as early stopping.

I. INTRODUCTION

Feature Selection (FS) plays an important role in many Supervised Machine Learning problems. In addition of reducing the storage requirements and the computational cost, FS may lead to the improvement of the generalization performance [7]. The problem of FS can be defined as follows: given a set of \( N_f \) features, select a subset that performs the best under a certain evaluation criterion. This definition leads to a search problem in a space of \( 2^{N_f} \) elements. Therefore, two components must be specified: the feature subset evaluation criterion and the search procedure through the space of feature subsets.

We will focus on FS with Multi-Layer Perceptrons (MLPs) within the wrapper approach [5]. Many FS algorithms for MLPs use, as feature subset evaluation criterion, different variations of the concept of saliency defined in some pruning methods [12]. A feature is considered more important whenever its saliency is larger, so that input units with small saliencies can be eliminated. Following the wrapper approach, the most commonly used saliency is the value of the loss function, usually the sum-of-squares error (SSE). Regarding the search procedure, most FS algorithms for MLPs use the Sequential Forward Selection (SBS) algorithm. SBS is a top-down process. Starting from the complete set of available features, one feature is deleted at every step of the algorithm, chosen on the basis of which of the available candidates gives rise, together with the remaining features, to the best value of the evaluation criterion. Ideally, the performance of the system is expected to improve until a subset of features remains for which the elimination of further variables results in performance degradation. In general, SBS helps to detect irrelevant\(^1\) variables in the first steps [7].

FS for MLPs with the SBS procedure and using the SSE as evaluation criterion involves taking a number of decisions, for which there are neither a commonly accepted criterion nor comparative studies. First, the stopping criterion of the training phase. Usually, networks are trained until a local minimum for the training set is found, although there are several exceptions, where an early stopping procedure is performed (see [17], [1], [19], for example). Second, the data set where the SSE should be measured. Many existing methods only use the training set to that end. Several exceptions use a validation or test set to compute the saliency (see [9], [4], [17], [19], [14], for example). Finally, whether or not the network should be retrained at every step with every feature temporarily removed before computing the saliency. To the best of our knowledge, the only models that retrain the network at every step with every feature temporarily removed/added before computing the saliency are those described in [16], [10], [14]. Among them, only the model presented in [14] is a pure SBS procedure.

An experimental study of the aforementioned decision issues when performing FS with MLPs and the SBS procedure is presented in this work. Experimental results indicate that the increase in the computational cost associated with retraining the network with every feature temporarily removed is rewarded with a significant performance improvement. This issue is shown to be critical, although, as previously mentioned, it has been hardly used in practice. Regarding the data set where the value of the SSE is measured, the SBS procedure for MLPs profits from measuring the SSE in a validation set, which is quite an intuitive idea. Instead, a somehow non-intuitive conclusion is drawn by looking at the stopping criterion, where forcing overtraining is shown to be potentially as useful as early stopping.

A significant improvement in the overall results with respect to learning with the whole set of variables is observed, which

---

\(^1\)There is no commonly accepted definition of the relevance of a variable (see [3], [7], for example). Given a data set, we consider that a variable is irrelevant for a Supervised Machine Learning system when its optimal performance is not affected negatively by the absence of that variable (17, page 29). Note that this is a dynamic definition, since the relevance of a variable may be affected by the presence or absence of other ones.
A feature can be computed following two approaches:

1) First, the network is trained with the whole set of features temporarily removed and the SSE is computed. The saliency of a feature is presented in figure 1. The outer loop selects the variable to eliminate: every feature is permanently eliminated at every step. The inner loop selects the variable to eliminate: every feature is temporarily removed, the network is optionally retrained until a certain stopping criterion is met, and then the value of the SSE is computed (on a certain data set). The feature corresponding to the lowest value of the SSE is permanently eliminated. The algorithm in figure 1 involves three decision issues, as explained next.

The first decision issue is the stopping criterion in the training phase. Two different stopping criteria were tested. The first one is to stop where a minimum of the SSE for a validation set is achieved. The second one is to train until a minimum for the training set is obtained. Suppose that the properties of the data set allow the negative effect of overfitting to appear. It seems that performing early stopping with a validation set could be the most promising idea. But it could also be argued that overtraining the network until a local minimum of the SSE for the training set forces the system to use all the available variables as much as possible. In this situation, irrelevant variables could be more outstanding when the system is not allowed to use them [14].

The second decision issue is the data set where the SSE is measured. The measurement of the SSE in a validation set is, probably, the most reasonable choice, since it can be considered as an estimator of the generalization error. But selecting the minimum number of features that allows to fit the training set as well as the whole set of variables does could also be thought as a quite reasonable way to obtain a good feature subset. In this case, the SSE should be measured in the training set. Both schemes were tested.

The third decision issue involves whether the network is retrained or not after the feature is temporarily removed and before computing the saliency. With this idea, the saliency of a feature can be computed following two approaches:

1) First, the network is trained with the whole set of available features. Then, every feature is temporarily removed and the SSE is computed. The saliency of every feature is computed in the same trained network. This procedure involves training \( N_f - 1 \) networks.

2) For every feature, the network is retrained with that feature temporarily removed. For every trained network, the SSE is computed. This procedure involves training \( N_f(N_f + 1)/2 \) networks (in this case, the training prior to the inner loop can be omitted).

Algorithm

```plaintext
Algorithm
Let \( V_1 \) the whole set of \( N_f \) features
for \( N = 1 \) up to \( N_f - 1 \) do
  Train the network with \( V_N \) until a certain stopping criterion is satisfied, and keep its generalization performance (decision issue, see text for details)
for each \( v \in V_N \) do
  Set \( V = V_N - \{ v \} \)
  Optionally, train the network with the features in \( V \) (decision issue, see text for details)
  Obtain the saliency of \( v \) by computing the value of the sum-of-squares error function \( E_v \) on a certain data set (decision issue, see text for details)
end for
Set \( V_{N+1} = V_N - \{ v^* \} \), where \( v^* \) corresponds to the lowest value of \( E_v \) in the previous loop
end for
Return \( V_{N^*} \), where \( N^* \) corresponds to the best generalization performance of the network at any step of the previous loop
end Algorithm
```

Fig. 1. A basic SBS procedure for MLPs and the SSE as the saliency.

III. EXPERIMENTS

Some experiments on both artificial and benchmark classification data sets were performed. For every data set, the six aforementioned configurations were tested with the SBS procedure for MLPs described in figure 1.
A. Experimental Setting

All the experiments were performed with stratified Cross-Validation (CV). Previously to every CV, the examples in the data set were randomly shuffled. For the Training/Training configuration, 5 runs of a 5-fold CV were conducted (the folds that are not in the training set were used as test data), giving a total of 25 runs for each training step. For the Training/Validation and Validation/Validation configurations, 5 runs of a double 5-4-fold CV [13] were performed as follows. A 5-fold CV (the outer CV) is performed to obtain 5 folds (4 folds to "learn" and 1 fold to test). Then, the 4 folds of the "learning set" of the outer CV were used as follows: 3 folds to train and 1 fold to validate, as in a 4-fold CV (the inner CV). Therefore, the number of trained models in every double 5-4-fold CV was 20, giving a total of 100 runs for each training step.

A computational cost as small as possible for the whole process was required. As pointed out in [15], MLPs using sine activation functions (and an appropriate choice of initial parameters) usually need less hidden units and learn faster than MLPs with sigmoid functions when both types are trained with Back-Propagation (BP). Therefore, we used MLPs with one hidden layer of sinusoidal units, the hyperbolic tangent in the output layer, and trained with standard BP in pattern mode.

In order to introduce the least external variability, all the configurations were tested with the same (although different for every data set) network architecture and parameters.

B. The Augmented Two Spirals Data Set

An augmented version of the well-known Two Spirals data set was constructed, where 13 irrelevant features were artificially added to the two original variables. Some of them were noisy. The whole set of variables was defined as

\[
\{x_1, x_2, x_1^2, x_2^2, x_1 \cdot x_2, x_1 + x_2, x_1 - x_2, \\
x_1 + \mathcal{N}(0, 1), x_2^2 + \mathcal{N}(0, 1), x_1 \cdot x_2 + \mathcal{N}(0, 1), \\
x_1 + x_2 + \mathcal{N}(0, 1), x_1 - x_2 + \mathcal{N}(0, 1), \\
\mathcal{U}(0, 1), \mathcal{N}(0, 1), \mathcal{N}(0, 5)\}
\]

where \(x_1, x_2\) are the original features in the data set, and \(\mathcal{N}\) and \(\mathcal{U}\) are the normal and the uniform distributions, respectively. Each original training, validation and test sets comprise 194 two-dimensional points with balanced classes. In order to perform the experiments with CV, the three data sets were joined into a single data set.

The results are shown in table I (column 'Test') as the average percentage of correctly classified patterns on the respective test sets in the following trained networks:

1) For the Training/Training configuration, the networks with minimum test set error among the networks with minimum training set error after every variable is permanently eliminated.

### Table I

<table>
<thead>
<tr>
<th>Retrain</th>
<th>Stopping Criterion/SSE Measurement</th>
<th>Test</th>
<th>MSE</th>
<th>SVar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Training/Validation</td>
<td>99.89%</td>
<td>0.01</td>
<td>(x_1, x_6, x_7)</td>
</tr>
<tr>
<td>Yes</td>
<td>Validation/Validation</td>
<td>99.66%</td>
<td>0.02</td>
<td>(x_6, x_1, x_2)</td>
</tr>
<tr>
<td>Yes</td>
<td>Training/Training</td>
<td>93.76%</td>
<td>0.19</td>
<td>(x_4, x_5, x_6, x_2)</td>
</tr>
<tr>
<td>No</td>
<td>Training/Validation</td>
<td>92.30%</td>
<td>0.25</td>
<td>(x_7, x_6, x_7, x_4, x_8, x_3)</td>
</tr>
<tr>
<td>No</td>
<td>Training/Training</td>
<td>92.10%</td>
<td>0.24</td>
<td>(x_6, x_2, x_7, x_8, x_5, x_4, x_3)</td>
</tr>
<tr>
<td>No</td>
<td>Validation/Validation</td>
<td>87.10%</td>
<td>0.39</td>
<td>(x_6, x_7, x_4, x_8, x_3)</td>
</tr>
</tbody>
</table>

Fig. 2. Percentage of correct examples for the Augmented Two Spirals data set in the training and test sets with respect to the number of eliminated variables in the SBS procedure for MLPs with retraining (top) and without it (bottom).
2) For the Training/Validation and Validation/Validation configurations, the networks with minimum test set error among the networks with minimum validation set error after every variable is permanently eliminated.

The mean SSE on the test set (column 'MSE') and the variables that allowed to obtain these results (column 'SV ar') are also shown. Values in table I are computed as the mean over the different folds in the respective CV. Figure 2 shows, for every configuration, the evolution of the percentage of correct examples in the training and test sets with respect to the number of eliminated variables.

As expected, the addition of irrelevant features affects very negatively the performance of sinusoidal MLPs in this problem, even if overfitting is tried to be controlled (see figure 2). The information needed to learn the problem is present, but the system is not able to use it in a proper way. The reason for this fact may be the relatively small number of examples in the data set, that did not allow to filter this kind of features. As far as variables are eliminated, performance improves. However, many variables must be eliminated to obtain good performance.

Retraining the network with every feature temporarily removed before computing the saliency has a positive effect on the SBS procedure for MLPs, both for the number of selected features and the overall performance. When retraining is present, it seems that the SBS procedure uses the validation set to construct a better subset of variables than if only the training set is used, although there is no significant difference regarding the stopping criterion (note that the Training/Validation configuration with retraining obtains similar results to the Validation/Validation one). The observed results can be explained by looking at the behavior of every configuration during the SBS procedure, as explained next.

Without retraining, the temporary elimination of any variable leads to large errors in the training set when compared to the training error with the whole set of features. This happens because the obtained solution after the training process uses all the variables in a significant way. In addition, noise-free variables (from $x_1$ to $x_7$) do not seem to be used more than the rest in order to learn the data set. Therefore, the elimination of a feature is decided in quite a random way.

When retraining is present,  

1) In the Training/Training configuration, the training set can be almost perfectly learned independently of the temporarily eliminated variable. Therefore, no significant difference can be stated among the variables. This behavior was observed for the first 9 or 10 steps in all the experiments performed. Therefore, and similar to the case of absence of retraining, the permanent elimination of a variable is decided in quite a random way during (too) many steps.

2) For the Training/Validation and Validation/Validation configurations, in contrast, the variables are much more clearly differentiated from the beginning of the SBS procedure. The criterion to eliminate a variable permanently does not seem random.

Surprisingly, the evolution of the training set error when retraining is present is also better with a validation set than without it (see figure 2). A common aspect of the configurations that do not obtain satisfactory results is the fact that they consider variables $x_3$ and $x_4$ (the squared of the original variables) as important. These variables do not seem the most promising ones for this problem. Although they allow, together with other ones, to fit the training set, those feature subsets are not good for generalization purposes.

C. Experiments on Benchmark Data Sets

In this section, the experiments on several benchmark data sets with the SBS procedure for MLPs described in figure
TABLE IV
Test set results and number of selected variables for the Hepatitis data set for different configurations of retraining/stopping criterion/SSE measurement data set in the SBS procedure.

<table>
<thead>
<tr>
<th>Retrain</th>
<th>Stopping Criterion/SSE Measurement</th>
<th>Test</th>
<th>Mse</th>
<th>NVar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Training/Validation</td>
<td>93.90%</td>
<td>0.24</td>
<td>3</td>
</tr>
<tr>
<td>Yes</td>
<td>Validation/Validation</td>
<td>93.77%</td>
<td>0.25</td>
<td>3</td>
</tr>
<tr>
<td>Yes</td>
<td>Training/Training</td>
<td>92.26%</td>
<td>0.25</td>
<td>3</td>
</tr>
<tr>
<td>No</td>
<td>Training/Validation</td>
<td>92.13%</td>
<td>0.26</td>
<td>3</td>
</tr>
<tr>
<td>No</td>
<td>Validation/Validation</td>
<td>88.97%</td>
<td>0.40</td>
<td>1</td>
</tr>
<tr>
<td>No</td>
<td>Training/Validation</td>
<td>88.26%</td>
<td>0.36</td>
<td>3</td>
</tr>
</tbody>
</table>

TABLE V
Test set results and number of selected variables for the Ionosphere data set for different configurations of retraining/stopping criterion/SSE measurement data set in the SBS procedure.

<table>
<thead>
<tr>
<th>Retrain</th>
<th>Stopping Criterion/SSE Measurement</th>
<th>Test</th>
<th>Mse</th>
<th>NVar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Training/Validation</td>
<td>93.61%</td>
<td>0.22</td>
<td>5</td>
</tr>
<tr>
<td>No</td>
<td>Validation/Validation</td>
<td>92.77%</td>
<td>0.24</td>
<td>5</td>
</tr>
<tr>
<td>Yes</td>
<td>Validation/Validation</td>
<td>92.73%</td>
<td>0.24</td>
<td>5</td>
</tr>
<tr>
<td>No</td>
<td>Training/Validation</td>
<td>92.57%</td>
<td>0.24</td>
<td>5</td>
</tr>
<tr>
<td>No</td>
<td>Training/Validation</td>
<td>92.40%</td>
<td>0.25</td>
<td>5</td>
</tr>
<tr>
<td>Yes</td>
<td>Training/Training</td>
<td>90.51%</td>
<td>0.33</td>
<td>3</td>
</tr>
</tbody>
</table>

TABLE VI
Test set results and number of selected variables for the Sonar data set for different configurations of retraining/stopping criterion/SSE measurement data set in the SBS procedure.

<table>
<thead>
<tr>
<th>Retrain</th>
<th>Stopping Criterion/SSE Measurement</th>
<th>Test</th>
<th>Mse</th>
<th>NVar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Validation/Validation</td>
<td>89.73%</td>
<td>0.33</td>
<td>14</td>
</tr>
<tr>
<td>Yes</td>
<td>Training/Training</td>
<td>88.59%</td>
<td>0.37</td>
<td>7</td>
</tr>
<tr>
<td>Yes</td>
<td>Training/Validation</td>
<td>87.95%</td>
<td>0.36</td>
<td>11</td>
</tr>
<tr>
<td>No</td>
<td>Training/Training</td>
<td>87.41%</td>
<td>0.40</td>
<td>57</td>
</tr>
<tr>
<td>No</td>
<td>Validation/Training</td>
<td>85.02%</td>
<td>0.46</td>
<td>46</td>
</tr>
<tr>
<td>No</td>
<td>Validation/Validation</td>
<td>84.49%</td>
<td>0.47</td>
<td>50</td>
</tr>
</tbody>
</table>

1 are shown. Three data sets from the UCI repository [2] were selected, namely Hepatitis, Ionosphere and Sonar. A brief description of these data sets can be found in table II, together with several results found in the literature from settings in which the whole set of features was used. In table II, MLP+BP means “Multi-Layer Perceptrons trained with Back-Propagation” and DistAl is a constructive learning algorithm for Neural Networks specific for classification problems [20]. The key idea behind DistAl is to add hidden units with a hyperspherical Radial Basis Function (RBF) based on a greedy strategy which ensures that the new hidden unit correctly classifies a maximal subset of training patterns belonging to the same class. Additionally, table III shows several results found in the literature after the application of an FS procedure on these data sets. We did not find more references (for wrapper approaches and with a similar experimental setting to that performed in this work) than those showed in the table. However, given the good results of DistAl with the whole set of features (see table II) and the good behavior of genetic algorithms for FS [6], the results in table III can be considered as a useful reference for comparison purposes. The search procedure used in [20] was a genetic algorithm where the fitness function for a given feature subset is computed as the mean percentage of correctly classified patterns on the test sets of a 10-fold CV trained with DistAl.

Our results are shown in tables IV to VI. Similar to the Augmented Two Spirals data set, the best results are always obtained retraining the network with every feature temporarily removed before computing the saliency. Therefore, the increase in the computational cost associated with this scheme is rewarded with a significant performance improvement. This issue is shown to be critical, although, as previously mentioned, it has been hardly used in practice.

Regarding the stopping criterion/SSE measurement data set, the SBS procedure seems to profit from measuring the SSE in a validation set, although it is unclear which is the best stopping criterion. For the Sonar problem the Validation/Validation strategy appears to work best. For the Ionosphere problem, in contrast, the Training/Validation strategy selects a better subset of variables. For the Hepatitis problem both configurations can be considered as equivalent. The goodness of the Val-
idation/Validation configuration can be intuitively explained, since it tries to obtain the best possible generalization results at every step. The Training/Validation configuration, in contrast, improves performance by forcing overtraining (and measuring the SSE in a validation set). This is a non-intuitive result. The explanation pointed out in [14] is that forcing the system to use all the available features as much as possible helps to detect irrelevant variables.

Although in a different scale, a similar behavior to that of the the Augmented Two Spirals data set was observed. First, the training set can be fitted with a much smaller subset of features than the original one. Regarding the test set, performance improves until a subset of features remains for which the elimination of further variables results in performance degradation. This behavior seems to reveal the existence of irrelevant variables that the SBS procedure has detected and eliminated. However, the differences among the different configurations suggest that, as in the Augmented Two Spirals data set, there are several variables that allow to fit the training set but they do not provide good generalization. The number of examples may not be large enough to filter these variables in some cases.

Finally, we can appreciate an important improvement in the overall results with respect to learning with the whole set of variables (see Table II) and compared with existing FS wrappers in the literature (see Table III). An important reduction in the final number of selected variables is also observed. The good results obtained with the Validation/Validation and Training/Validation with retraining configurations are mainly due, in our opinion, to a proper detection of irrelevant variables.

IV. Future Work

The main drawback of the SBS procedure for MLPs presented in this work is its computational cost, particularly when retraining is performed. Training algorithms faster than BP may obviously be used, but BP was not the main source of the computational cost in our experiments. The first steps of the algorithm, when probably many irrelevant variables still remain, take most of the computational time. Several heuristics could be designed to eliminate the most clearly irrelevant variables with a low computational cost. Then, when a reasonable number of features remains, the whole procedure would start.

The results provided in this work were obtained with standard BP and sinusoidal hidden units, but the basic scheme presented in this work can be tested within any other framework which can be adjusted to the required specifications. In particular, the SBS procedure in figure 1 could be performed with Support Vector Machines [18] using some function of the margin as saliency and different hardness of the margin as the stopping criterion.

3The results obtained for the Ionosphere data set could be seen as unsatisfactory when compared with those obtained in [20]. In contrast, they can be considered as very satisfactory when compared with those obtained by MLP models. It is worth noting that RBF networks allow to obtain better solutions than MLP ones for this problem (see Table II). Therefore, FS models based on RBF units, such as DistAl, are expected to obtain excellent results.

ACKNOWLEDGEMENTS

This work was supported by Consejo Interministerial de Ciencia y Tecnología (CICYT), under projects DPI2002-03225 and CGL2004-04702-C02-02.

REFERENCES