Using Early Stopping to Reduce Overfitting in Wrapper-Based Feature Weighting

John Loughrey and Pádraig Cunningham

Trinity College Dublin, College Green, Dublin 2, Ireland

Abstract. It is acknowledged that overfitting can occur in feature selection using the wrapper method when there is a limited amount of training data available. It has also been shown that the severity of overfitting is related to the intensity of the search algorithm used during this process. We demonstrate that the problem of overfitting in feature weighting can be exacerbated if the feature weighting is fine grained. With greater representational power we risk learning not only the signal, but also the idiosyncrasies of the training data. In this paper we show that both of these effects can be ameliorated by the early-stopping strategy we present. Using this strategy feature weighting will outperform feature selection in most cases.

1 Introduction

The benefits of wrapper-based techniques for feature selection are well established (Kohavi & John, 1997). However, it has recently been recognised that wrapper-based techniques have the potential to overfit the training data (Reunanen, 2003). That is, feature subsets that perform well on the training data may not perform as well on data not used in the training process. This has been identified in the context of wrapper based searches in (Kohavi et al., 1997) and (Reunanen, 2003).

In the feature weighting research, the DIET algorithm (Kohavi et al., 1997) displayed a greater tendency to overfit during the wrapper search when more weights were available. Traditionally researchers have avoided overfitting by reducing the representational power of the machine learning algorithm. However we believe that we can avoid overfitting while maintaining the representational power of the algorithm. We present a framework for feature weighting which harnesses the power of the wrapper-based approached along with a finer granularity of feature weights. Overfitting will be reduced by employing an early stopping procedure similar to that used in (Loughrey & Cunningham, 2005).

In this paper we show that this tendency to overfit can be quite acute in stochastic search algorithms such as Simulated Annealing (SA) as these algorithms are able to intensively explore the search space. It is worth noting that the applicability of early-stopping depends on the stochastic nature of the search. This idea would not be readily applicable in more directed search strategies such as the Sequential Forward Floating Selection (SFFS) and Sequential Forward Selection (SFS) strategies evaluated by (Reunanen, 2003) or the standard Backward Elimination strategy that is popular in wrapper-based feature selection.

In Loughrey and Cunningham (2004) and Loughrey and Cunningham (2005) we approach the overfitting problem in feature selection using the SS-ES framework and we find that the results we get are favorable. In this paper we attempt to apply a similar framework to feature weighting and we compare the performance of feature weighting and feature selection.

The paper is organized as follows. We introduce the concept of Feature Weighting briefly in Section 2. In Section 3 we outline the overfitting problem with regard to wrapper-based searches with limited training data. This is followed by a discussion in Section 4 where we introduce our SS-ES framework and the strategy that we employ in order to reduce the overfitting effect. This framework is evaluated in Section 5 on the SA algorithm comparing different weight sets and the paper concludes with some suggestions for future work in Section 6.

2 Feature Weighting

The k-NN classifier is built upon a distance function and as a result its performance is sensitive to noisy and irrelevant features. This is because in such cases these bad features will have the same influence on the classification as good, highly predictive features. By weighting features we attempt to identify irrelevant and redundant features and assign them a low weighting reducing their influence in the classifier.

Algorithms for feature weighting can be divided into two categories depending on whether they use feed-back from the classifier. There is the filter type, which use relationships within the data to rank the features and assign weights or the wrapper-based search method (Wettschereck et al., 1997). The wrapper approach differs in that it evaluates subsets based upon the accuracy estimates provided by a classifier built with that feature subset. The wrapper-based approaches are much more computationally expensive and time consuming than filters but can produce better results because they take the *bias* of the classifier into account and evaluate features in context. The main argument against the wrapper approach is its computational cost. Since the search is directed by an assessment of the accuracy attributable to the feature mask (feature subset) it is common to use cross-validation as is described in section 4.3 but this can be slow. However, it has been shown in Kohavi and John (1997) and Hall and Holmes (2003) that the wrapper approach generally outperforms filters, and as time is not an issue in our case it seems appropriate to use the wrapper.

During the search, weights can be determined from continuous values or they can be taken from a set of predetermined values of size w, where w will specify the number of non-zero values i.e. (0, $1/w$, $2/w$, ..., w/w). In the case where w=1, we are performing feature selection (a feature can either have a value of [0,1]). The weight space to be searched is in the order of w^f where f is the number of features. We use a stochastic search in this paper, to compare the effect of overfitting in both feature selection and feature weighting.

3 Overfitting in Wrapper-Based Searches

Overfitting is said to occur when the classifier start to learn aspects of the noise associated with small data sets. As shown in Figure 2 overfitting is characterized by the validation accuracy of the model peaking and then gradually deteriorating as training continues.

3.1 Overfitting in Feature Weighting (w>1)

Kohavi et al. (1997) describe their feature weighting algorithm DIET which uses a best-first search to explore the weight space where weights are assigned from a set of possible values (0, 1/w, 2/w,..., w). They claim that w=1 (feature selection) is difficult to outperform and furthermore as w increases so will the level of overfitting. We demonstrate this effect in Figure 1 where different values for w were used on an *artificial* generated using the WEKA toolkit[1]. This result was generated using a GA run for 50 generations on a population of 30 members. As the granularity of the weights increases so does the overfitting levels. This is consistent with what Kohavi et al. (1997) claim, that when when using a wrapper search on small data sets, decreasing the set of weights reduces the algorithm's variance thereby reducing the error rate.

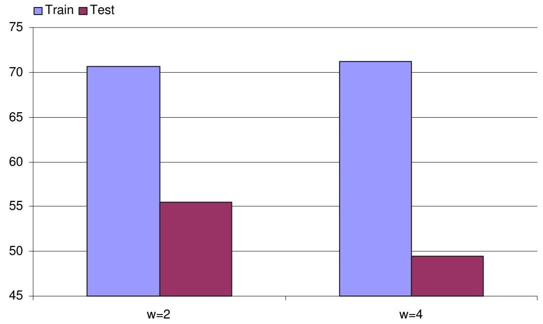

Fig. 1. The Figure shows the effect of the number of weights used in the wrapper search on the *artificial* data set. Overfitting is more apparent in the model that used more weights in the training phase.

[1] http://www.cs.waikato.ac.nz/ml/weka/

3.2 Overfitting in Feature Selection (w=1)

In feature selection (when w=1) the overfitting effect appears to be exacerbated by the intensity of the search since the more feature subsets that are examined the more likely the search is to find a subset that overfits. This effect is clearly present in Figure 2 which shows the search progression on our *artificial* data set with the Simulated Annealing search. In this example the trend-line for the validation accuracy is represented by the heavy line. It is clear that after iteration 25 we start to overfit to the noise in the training set, as the validation accuracy starts to deteriorate and doesn't recover while the training accuracy increases.

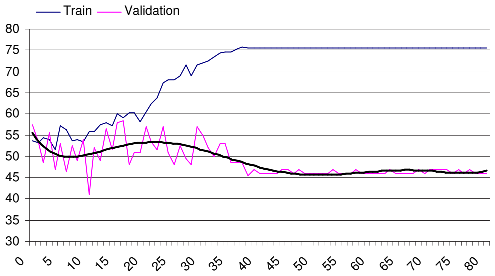

Fig. 2. The Figure shows the effect of the depth/intensity of the wrapper search (where w=1) on the *artificial* data set, where the generalization accuracy is reduced as more nodes are evaluated.

In the seminal publication on wrapper-based feature selection Kohavi and John (1997) mentioned the problem of overfitting but illustrated that it was not a problem on the data sets they examined. As with all machine learning algorithms this is true if the data available adequately covers the phenomenon. With sample size often limited in many real world applications, for example in medical and financial applications, overfitting in wrapper-based feature selection is a real problem. Overfitting in feature selection is raised again in (Reunanen, 2003). Although little is said on how it can be addressed, the danger of using intensive search strategies when data is limited is identified. Reunanen makes the comparison between the more simple SFS with the performance of the more intensive SFFS and states that a simpler search is less likely to overfit.

From this evidence one could expect that during wrapper-based feature weighting we run the risk of overfitting on two fronts. Firstly, as we increase the number of weights available we increase the model variance and therefore are likely to overfit more. Secondly the more intense the search, the more likely we are to overfit once again. Preliminary investigations provided support for this. In Figure 1 we can see that using w=4, performs less well on validation data than when w=2. However it is apparent that the use of more weights will not always be outperformed by simple feature selection. In fact using more weights may only result in the search being *prone* to overfitting. When looking at what happens early in the search there is evidence that feature weighting will outperform feature selection. Figure 3 compares these two situations. Using more weights increases the representational power and therefore it should be better able to capture the underlying concept, but as the search continues this additional power then starts to learn more about idiosyncrasies of the training set and as a result overfits to a greater extent than the search with the weight set restricted to w=2.

Obviously there is a trade-off to be reached here, between the representational power of the classifier and it's tendency to overfit during learning.

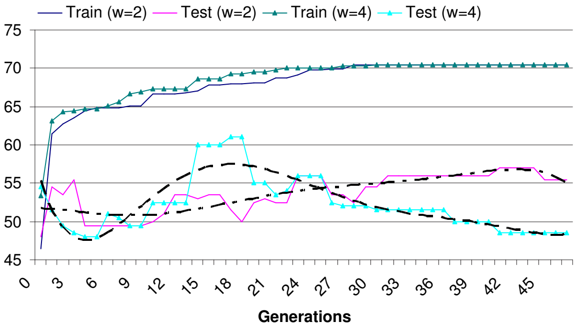

Fig. 3. The figure shows the effect of the depth/intensity of the wrapper search on the *artificial* data set, showing the different levels of overfitting in two searches with different weights (w=2, w=4)

4 Reducing Overfitting

There are many examples documented where constraining the representational power of an algorithm can lead to an increase in performance. The addition of noise to the training data restricts the potential to model the underlying data (Koistinen & Holmström, 1991), while limiting the number of hidden units in a neural network will have a similar effect.

In feature weighting we can restrict the number of weights we use for the features, which restricts the subspace size, but we can also restrict the intensity/depth of the search using early stopping which has been shown to work in the case of feature selection (Loughrey & Cunningham, 2004) and (Loughrey & Cunningham, 2005).

4.1 Early-Stopping in Stochastic Search

The motivation behind early-stopping is fairly straightforward - stop the search at the point that overfitting starts to happen. This is achieved by using a cross-validation analysis on the training data to determine when early-stopping starts to occur. Then a model is built with all the training data and the search is stopped at the appropriate point. While the idea is straightforward, it is awkward to evaluate the effectiveness of the process. This requires two nested levels of cross-validation (see section 4.1), an outer level to assess generalization accuracy and an inner level to determine the early-stopping point.

As was emphasized in the Introduction, this early-stopping strategy is only meaningful for wrapper-based feature selection where the search strategy is stochastic. It would not be sensible to stop a Forward Selection or Backward Elimination strategy as it would simply exclude some features from consideration. However, it does make sense to stop a GA or SA earlier on in the search process.

4.2 Simulated Annealing

Kirkpatrick et al. (1983) have shown that the models that describe the annealing of metals can be used to guide stochastic search. Simulated Annealing is similar to hill climbing search in that there is only one solution at a time under consideration. This solution is perturbed and the new solution is kept if it represents an improvement. The special feature of SA is that the new solution may still be kept even if it is poorer than the existing one. The probability of this is:

$$P(Accept) \propto e^{\frac{\Delta L}{T}} \tag{1}$$

In feature selection, ΔL would refer to the difference in accuracy between the old and new feature masks and T would be an artificial variable describing the ‘temperature’ of the system. The effect of this policy is that large deteriorations in accuracy are less likely to be accepted and any deterioration is less likely to be accepted as the temperature drops.

The core of an SA algorithm for feature subset selection is described in Figure 4. Initially, the system starts off at a high temperature and the search is allowed to proceed in a fairly random manner. The system cools in stages with the search staying at a given temperature until a number of perturbations have been explored or a number of successes have been achieved. Thus the rate of progress of the SA is determined by the cooling rate (0.9 in this example) and the factor K that determines how long is spent at each temperature level. For instance if K is halved the cooling will proceed twice as quickly. In terms of the original inspiration for SA, this might be described as *quenching* the system. So our early-stopping policy for SA still allows the system to *freeze*, it just spends less time at each temperature level.

```
T = T * 0.9 /* Reduce the temperature */
NTries = 0; NSucc = 0
while(NTries < TryLim * 10 * K) and (NSucc < SuccLim * K)
    M' = PerturbMask(M)
    if AcceptNewMask?(M',M)
        NSucc = Nsucc + 1
        M = M'
    endif
    NTries = NTries + 1
endwhile
```

Fig. 4. The core of the SA algorithm

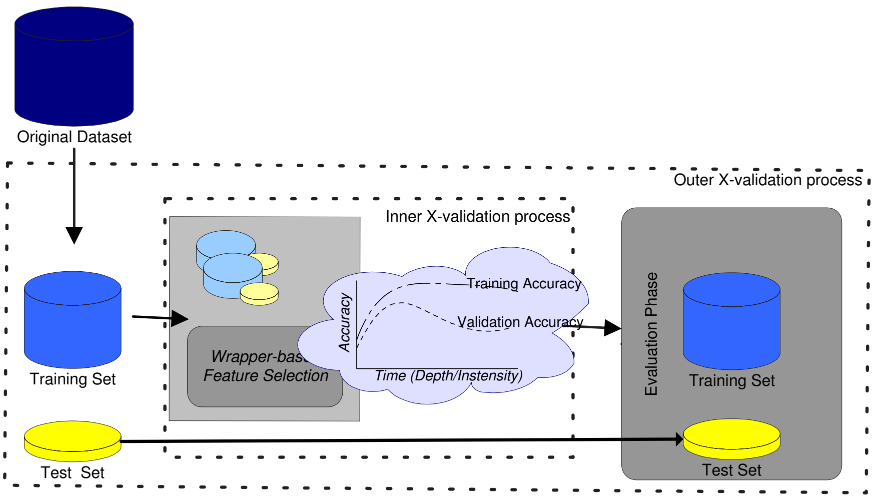

Fig. 5. SS-ES Framework

4.3 Stochastic Search with Early-Stopping - (SS-ES)

The basic principle for the SA is to modify the search algorithm so that it will reduce its intensity/depth of search depending on when overfitting was judged likely to occur.

Figure 5 shows the SS-ES Framework in which we evaluate the overfitting in the Wrapper-based subset selection process - this follows the principles outlined by (Weiss & Kulikowski, 1991). In each *fold* of the outer cross-validation, the original data source is divided into two in a 80:20 split. The 20% will be the outer test set that will be used to evaluate the generalization accuracy of the resulting feature set. The remaining 80% goes into our inner cross-validation which attempts to identify at which stage overfitting occurs in the wrapper search. The inner cross-validation divides the training data into a 90:10 split. 90% of this data is used to build a classifier and 10% is used to estimate the validation accuracy. This is repeated 10 times. Therefore, in the inner cross-validation we have 72% of the original data to train the classifier in each fold and 8% for estimating the validation accuracy of that classifier (repeated over 10 folds). An example of this is shown in Figure 6, where we identify overfitting to occur after iteration 32. So, in the evaluation phase we modify $K \leftarrow K \times 32/81$.

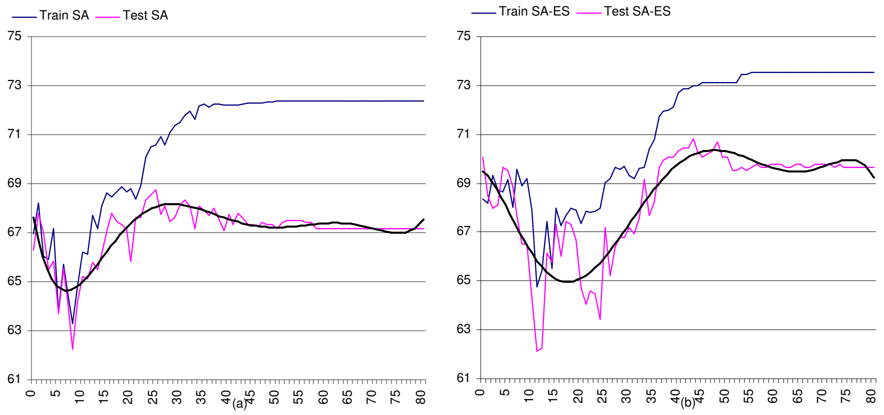

Fig. 6. (a) Shows an example of an inner evaluation in the SS-ES framework on the *diabetes* domain, where the model is trained on 72% of data available. To reduce the depth in the evaluation phase we speed up the search by a factor (81/32). (b) Shows the result, with the search reducing overfitting when trained on 80% of the data

5 Evaluation

As (Kohavi & John, 1997) point out, overfitting is often not a problem in wrapper-based feature subset selection, thus it will not show up in many feature selection tasks. In the evaluation we present here we work with five data sets from the UCI collection (Blake & Merz, 1998) and two other data sets; the *Colon* data set described in (Alon et al., 1999) and an *artificial* data set created using the WEKA toolkit. The *artificial* data set was created with 100 instances described by 50 features, 25 of which were deemed to be irrelevant. We selected these seven data-sets because they proved to exhibit overfitting in our preliminary analysis.

Our evaluation was carried out the FIONN workbench (Doyle et al., 2004) and consists of two experiments. Firstly, we would like to see if feature selection will consistently outperform feature weighting on the domains that we are studying. We compare the wrapper results when w=1 and w=4 and we would expect to see similar results as to those found in Figure 1, with w=1 outperforming the search with more weights.

Secondly, we will apply the SS-ES framework and we would expect this to yield an improvement on both searches. Intuitively, if we avoid overfitting successfully we should note an improvement in the generalization accuracy. Another characteristic of successful early-stopping is for the model to exhibit a lower training accuracy due to the shorter training time.

Figure 7 shows the results over all data sets when comparing the search over w=1 and w=4. In four out of six domains it seems to be the case that w=1 performs better than when w=4. Moreover, our results are in agreement with what (Kohavi et al., 1997) suggested with the DIET algorithm in that the more weights used the more likely one is to overfit.

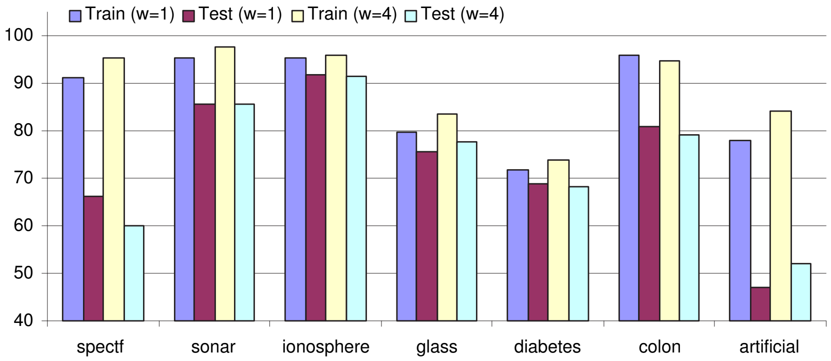

Fig. 7. Normal runs using SA. Clearly when using more that feature selection we appear to have more a risk of overfitting

However in Figure 8 which shows the SA with Early Stopping (SA-ES) results for both sets of weights, the results are a little different. In all but one domain (*colon*) the generalization accuracy for w=4 was better than that for w=1 when early-stopping was employed. The reason for this failure could be attributed to the quality of this data set.

Table 1 shows a summary of the results. The SA-ES approach identified when overfitting was likely to occur and successfully modified the search so that this was prevented in the final evaluation. The framework proved successful over both approaches but the improvement was more apparent when using the larger set of weights. The fact that when w=1 outperformed w=4 may not be that surprising. This has been stated before in (Kohavi et al., 1997) and is attributed to the restriction on the model variance. However, when we applied this early stopping procedure to both searches, w=4 outperformed w=1 in all but one.

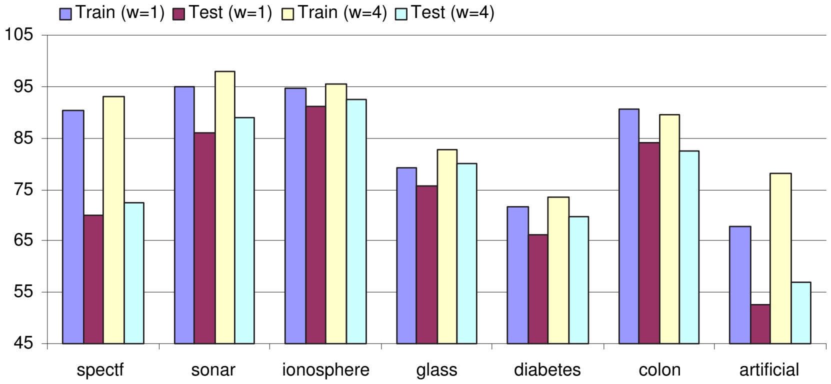

Fig. 8. Early-stopping results (SA-ES). The early-stopping yields more of an improvement when w=4.

	SA		SA-ES		SA		SA-ES	
	(w=1)		(w=1)		(w=4)		(w=4)	
	Train	Test	Train	Test	Train	Test	Train	Test
spectf	91.3	66.3	90.3	70.0	95.3	60.0	93.1	**72.5**
sonar	95.2	85.6	95.1	86.0	97.6	85.6	98.0	**88.9**
ionosphere	95.4	91.7	94.8	91.2	95.9	91.4	95.4	**92.6**
glass	79.7	75.7	79.2	75.7	83.6	77.6	82.7	**79.9**
diabetes	71.8	68.9	71.6	66.3	73.7	68.2	73.5	**69.7**
colon	96.0	80.9	90.7	**84.1**	94.8	79.2	89.5	82.4
artificial	78.0	47.0	67.9	52.5	84.1	52.0	78.1	**57.0**

Table 1. Comparison of results across seven data sets using SA and SA-ES

6 Conclusions and Future Work

In this paper we presented an early stopping framework for overfitting in wrapper based searches for feature weighting. We have shown that overfitting is a problem in these searches and stopping the search early will usually yield an increase in generalization accuracy when the amount of training data is limited. In the SS-ES framework it has also been shown that feature weighting will outperform feature selection in the cases studied here and we presented the reasons why we believe this is so. (Wettschereck et al., 1997) identified the need to design algorithms that explore the trade-off between searching the continuous weight space and a restricted set. We believe that using the early stopping procedure outlined here avoids this as we identify where overfitting begins. In this case, we suggest that the better performance is gained through feature weighting than through feature selection. We can also say that limiting the depth of search as shown here is more effective than restricting the set of weights available.

Future work would include performing these experiments over more data sets both artificial and real and should attempt to measure the effectiveness of the weighting process over feature relevance. The next phase of this research is to extend the experiments to search for an optimal set of weights, or to find when increasing the granularity of the weights no longer has any useful effect. Finally, an evaluation against the other wrapper based feature weighting algorithms would be needed in order to measure the effectiveness of the SA over other heuristics as described in Kohavi et al. (1997) and Jr. and Davis (1991).

Bibliography

Alon, U., Barai, N., Notterman, D., Gish, K., Ybarra, S., M. D., & Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci.*, *96*, 6745–6750.

Blake, C., & Merz, C. (1998). *UCI repository of machine learning databases* (Technical Report). University of California at Irvine, Department of Information and Computer Science, www.ics.uci.edu/ mlearn/MLRepository.html.

Doyle, D., Loughrey, J., Nugent, C., Coyle, L., & Cunningham, P. (2004). Fionn: A framework for developing cbr systems. To Appear in Expert Update.

Hall, M., & Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. *IEEE Trans. Knowl. Data Eng.*, *15*, 1437–1447.

Jr., J. D. K., & Davis, L. (1991). Hybridizing the genetic algorithm and the k nearest neighbors classification algorithm. *ICGA* (pp. 377–383).

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, *220, 4598*, 671–680.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*, 273–324.

Kohavi, R., Langley, P., & Yun, Y. (1997). The utility of feature weighting in nearest-neighbor algorithms. *Proceedings of the Ninth European Conference on Machine Learning.*

Koistinen, P., & Holmström, L. (1991). Kernel regression and backpropagation training with noise. *NIPS* (pp. 1033–1039).

Loughrey, J., & Cunningham, P. (2004). Overfitting in wrapper-based feature subset selection: The harder you try the worse it gets. *24th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-2004)* (pp. 33–43). Springer.

Loughrey, J., & Cunningham, P. (2005). *Using early-stopping to avoid overfitting in wrapper-based feature selection employing stochastic search* (Technical Report TCD-CS-2005-37). Department of Computer Science, Trinity College Dublin, Dublin, Ireland.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *J. Mach. Learn. Res.*, *3*, 1371–1382.

Weiss, S., & Kulikowski, C. (1991). *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems.* Morgan Kaufmann Publishers Inc.

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artif. Intell. Rev.*, *11*, 273–314.