Using Early-Stopping to Avoid Overfitting in Wrapper-Based Feature Selection Employing Stochastic Search

John Loughrey JOHN.LOUGHREY@CS.TCD.IE
Pádraig Cunningham PADRAIG.CUNNINGHAM@CS.TCD.IE
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. In this paper we show that two stochastic search techniques (Simulated Annealing and Genetic Algorithms) that can be used for wrapper-based feature selection are susceptible to overfitting in this way. However, because of their stochastic nature, these algorithms can be stopped early to prevent overfitting. We present a framework that implements early-stopping for both of these stochastic search techniques and we show that this is successful in reducing the effects of overfitting and in increasing generalisation accuracy in most cases.

1. Introduction

The benefits of wrapper-based techniques for feature selection are well established (Kohavi & Sommerfield, 1995). 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. Furthermore, the extent of the overfitting is related to the depth of the search. Reunanen (2003) shows that, whereas Sequential Forward Floating Selection (SFFS) beats Sequential Forward Selection (SFS) on the data used in the training process, the reverse is true on hold-out data. He argues that this is because SFFS is a more intensive search process i.e. it explores more states.

In this paper we show that this tendency to overfit can be quite acute in stochastic search algorithms such as Genetic Algorithms (GA) and Simulated Annealing (SA) as these algorithms are able to intensively explore the search space. We show that early-stopping is an effective strategy for preventing overfitting in feature selection using SA or GA. 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 SFFS and SFS strategies evaluated by Reunanen (2003) or the standard Backward Elimination strategy that is popular in wrapper-based feature selection.

In (Loughrey & Cunningham, 2004) we approach this problem using a modified genetic algorithm (GA) that stops the search early in order to avoid overfitting, and we find that the results we get are favourable. In this paper we show that SA is amenable to a neat form of early-stopping. Optimisation using SA is analogous to the cooling of metals and we show how the SA can be *quenched* so that the search *freezes* before overfitting can occur. In section 3.2 we show how SA can be speeded up to avoid overfitting and in section 3.3 we show how to *calibrate* this process using cross-validation.

The paper is organised as follows. We begin in section 2 with a discussion of the wrapper-based approach to feature selection and an illustration of the potential overfitting problem. The early-stopping solution to overfitting is described in section 3. The approach is evaluated on SA and GA in section 4 and the paper concludes with some suggestions for future work in section 5.

2. Feature Selection

Feature selection is defined as the selection of a subset of features to describe a phenomenon from a larger set that may contain irrelevant or redundant features. Feature selection attempts to identify and eliminate unnecessary features, thereby reducing the dimensionality of the data and reducing the model variance (van der Putten & van Someren, 2004). Improving classifier performance and accuracy are usually the motivating factors. The accuracy of Nearest Neighbour classifiers (k-NN) is particularly degraded by the presence of these irrelevant features. The evaluations presented in this paper are on k-NN classifiers.

The two alternative approaches to feature selection are the use of filters and the wrapper-based method. Filtering techniques attempt to identify features that are related to or predictive of the outcome of interest and they operate independently of the learning algorithm. An example is Information Gain, which was originally introduced to Machine Learning research by Quinlan as a criterion for building concise decision trees (Quinlan, 1993) but it is now widely used for feature selection in general.

The wrapper approach differs in that it evaluates subsets based upon the accuracy estimates provided by a classifier built with that feature subset. Thus wrappers are much more computationally expensive than filters but can produce better results because they take the *bias* of the classifier into account and evaluate features in context. The wrapper search then uses some heuristic such as Backward Elimination or Forward Selection to traverse the feature subset space in search of an optimal subset. The big issue with the wrapper approach is the computational cost since the search is directed by an assessment of the accuracy attributable to the feature mask (feature subset). This assessment needs to be as accurate as possible so it is common to use cross-validation as is described in section 3.3.

2.1. Overfitting in Wrapper-Based Feature Selection

In the original publications on wrapper-based feature selection Kohavi and John (1997) mentioned the problem of overfitting but illustrated that it was not a problem on the datasets they examined. As with all machine learning algorithms this is true if the data available adequately covers the phenomenon. The problem is that sample size is often limited in many real world applications, especially in medical and financial applications, in these situations overfitting in wrapper-based feature selection is a real problem.

Overfitting in feature selection 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. In (Kohavi & Sommerfield, 1995) (Reunanen, 2003) this problem is described, although little is said on how it can be addressed. However, we believe that limiting the extent of search will help combat overfitting. Kohavi et al. (1997) describe the feature weighting algorithm DIET, in which the set of possible feature weights can be restricted. Their experiments show that when DIET is restricted to two non-zero weights the resultant models perform better than when the algorithm allows for a larger set of feature weights, in situations when the training data is limited. This restriction on the possible set of values in turn restricts the extent to which the algorithm can search, and therefore constrains the representational power of the model.

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 case base during training restricts the models representational power on the underlying data (Koistinen & Holmström, 1991), while limiting the number of hidden units in a neural network will have a similar effect. However, in feature selection we only have two possible weights, a feature can only have a value of '1' or '0' i.e. be turned 'on' or 'off', so we cannot restrict this aspect any further. In a stochastic search we can constrain the intensity of the search through early-stopping.

In Figure 1 we illustrate the idea that the more states visited in the search the more likely the search is to overfit to the training data. The right hand axis shows the number of nodes visited during the search, while the left hand axis shows the accuracies obtained. The best generalisation accuracy is achieved by the Backward Elimination (be) search while the GA search overfits more and more as the number of generations increases from 100 to 200 to 400. Forward Selection (fs) is also economic in the number of nodes it searches but still manages to overfit the data. The relatively poor performance of Forward Selection compared to Backward Elimination has been documented previously (Aha & Bankert, 1994).

2.2. Overfitting and the Bias-Variance Decomposition of Error

The bias-variance decomposition (Kohavi & Wolpert, 1996) makes available the components of the error measure and this enables us to see if the errors we are getting are due to model variance or due to model bias.

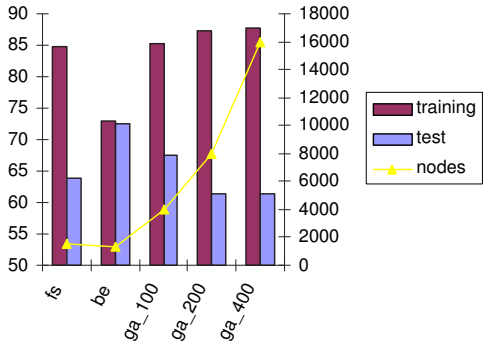

Figure 1. The Figure shows the effect of the depth/intensity of the wrapper search on the 'spectf' data set, where the generalization accuracy is reduced as more nodes are evaluated.

Errors due to overfitting should show up as model variance error. Thus measures to combat overfitting should reduce the variance component of error if effective. A high bias suggests that the learning method is not correct for the domain, and represents the shortcomings of the learning method in modelling the data and is generally not related to a lack of data or overfitting.

Successful feature selection should result in a reduction in model variance, but we expect that this measure will increase once again when we overfit during the wrapper search.

3. 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 3), an outer level to assess generalisation accuracy and an inner level to determine the early-stopping point.

As was emphasised 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.

3.1. Genetic Algorithms

Genetic Algorithms have been inspired by the biological process of evolution and attempt to capture the concept of the 'survival of the fittest'. In the GA search we maintain a fixed population of possible solutions and these individuals evolve as the search progresses in an attempt to find an optimal solution. Evolutionary strategies such as crossover and mutation are used to maintain quality and diversity of the population.

A GA is an attractive search policy for wrapper-based feature selection as crossover and mutation are straightforward. Each solution in the population is represented as a feature mask and crossover is achieved by splicing two masks. Mutation simply involves flipping bits on or off in the masks.

The results reported here use a basic GA algorithm that uses Roulette Wheel selection based on the cross-validation accuracy of the feature masks. In fact the log of the accuracy is used to slow down the convergence. We used a two-point crossover and a mutation rate of 0.05. The population size for each of the data sets was fixed to 40, and it was allowed to search for 120 generations.

After the inner cross-validation layer of the framework we estimate at which generation overfitting was likely to begin.

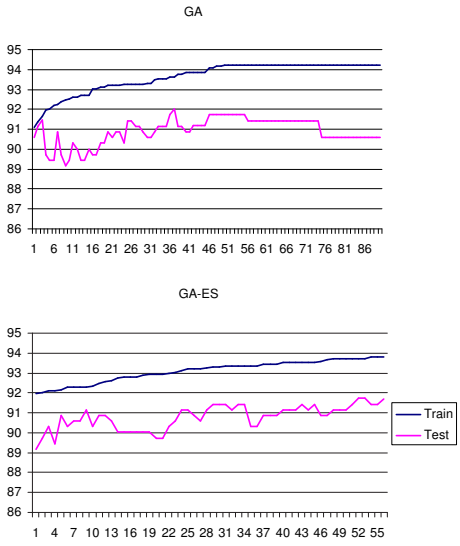

Figure 2. Analysis of the GA and GA-ES search progression

The GA-ES is modified in that we stop the search at an earlier generation, the generation at which overfitting occurred, as shown in Figure 2. We have already shown that this is a reasonably effective strategy for combatting overfitting in GAs (Loughrey & Cunningham, 2004).

3.2. 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:

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

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 3. 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.

Figure 4 describes the idea behind SA-ES. In the normal run, we identify the number of nodes that we evaluate in the iterations before overfitting starts to occur, then in the modified algorithm we stretch these attempts out so that they cover the entire cooling phase of the modified search. This is achieved by reducing K by the appropriate proportion. In the example in Figure 4: $K \leftarrow K \times 20/65$.

```
T = T * 0.9 /* Reduce the temperature */
NTries = 0; NSucc = 0
while(NTries < TryLim * 10 * K) and (NSucc < SuccLim * K)
    MDash = PerturbMask(M)
    if AcceptNewMask?(MDash,M)
        NSucc = Nsucc + 1
        M = MDash
    endif
    NTries = NTries + 1
endwhile
```

Figure 3. The core of the SA algorithm

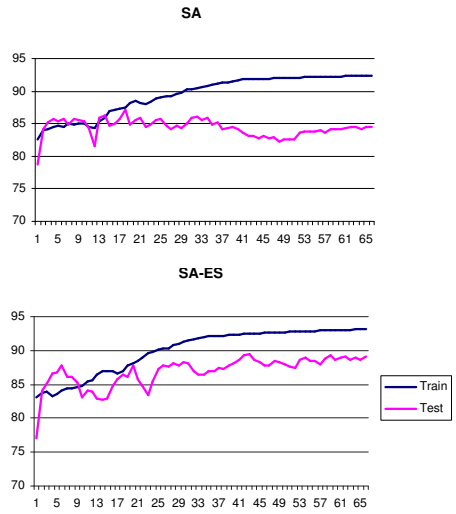

Figure 4. Analysis of the SA and SA-ES search progression

3.3. Stochastic Search with Early-Stopping - (SS-ES)

So the basic principle for both SA and GA's 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 and Kulikowski (1991). In each *fold* of the outer cross-validation, the original data source is divided into two in a 90:10 split. This 10% will be the outer test set that will be used to evaluate the generalisation accuracy of the resulting feature set. The 90% 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 data again into a 90:10 split. 90% of this data is used to build a classifier and the 10% is used to estimate the validation accuracy. This is repeated 10 times. Therefore in the inner cross-validation we have 81% of the

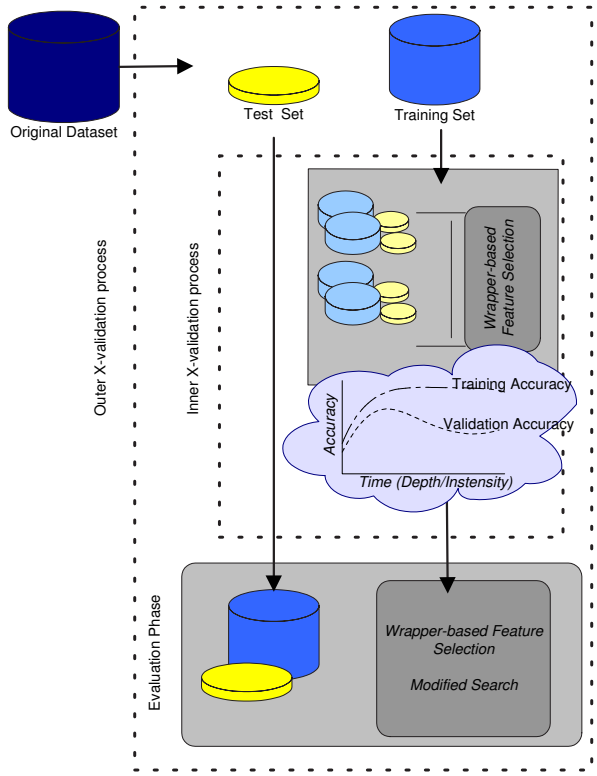

Figure 5. SS-ES Framework

	GA		GA-ES	
	Train	Test	Train	Test
wpbc	83.33	**77.39**	82.30	75.76
spectf	81.25	72.50	73.06	72.50
sonar	91.78	86.55	91.72	**90.36**
ionosphere	94.21	**90.59**	93.89	89.44
glass	80.23	74.19	80.32	**76.71**
diabetes	74.96	70.17	74.81	**71.49**
bronc	67.16	**59.89**	64.07	58.86
colon	93.67	80.08	92.36	**84.21**

Table 1. Comparison of results across eight data sets using GA and GA-ES

	SA		SA-ES	
	Train	Test	Train	Test
wpbc	81.50	73.29	81.10	**73.82**
spectf	83.75	65.00	80.00	**73.75**
sonar	92.52	85.07	93.27	**88.43**
ionosphere	94.43	**92.86**	94.14	92.00
glass	80.11	73.79	80.06	**76.60**
diabetes	74.44	69.01	73.71	**72.40**
bronc	80.32	56.62	77.38	**59.04**
colon	93.79	83.17	87.88	**83.65**

Table 2. Comparison of results across eight data sets using SA and SA-ES

original data to train the classifier in each fold and 9% for estimating the validation accuracy of that classifier (repeated over 10 folds).

4. Evaluation

As Kohavi and 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 six datasets from the UCI collection (Blake & Merz, 1998) and two other datasets; the Colon dataset described in (Alon et al., 1999) and the Bronchiolitis daset described in (Walsh et al., 2004). These are data-sets that proved to exhibit overfitting in our preliminary analysis.

The results on the GA-ES are shown in Table 1 and the results on the SA-ES are shown in Table 2. The GA-ES results are not very encouraging with the GA-ES winning in 4, loosing in 3 and drawing in one. This is due to difficulties with automatically identifying the early-stopping point in the cross-validation process. This could be improved by making this a manual (interactive) process; however, the SA-ES strategy shows more promise.

The SA-ES improves on the simple SA in 7 of the 8 datasets. This is probably due to the fact that the SA-ES strategy is more robust than the GA-ES strategy. The practice of quenching the cooling process more rapidly so that the SA freezes before overfitting is more robust than the GA-ES strategy which is subject to variability because the GA is being stopped before convergence.

When perform multiple runs of the feature selection process using the SA and SA-ES strategies and decompose the error on the resulting feature sets into their bias-variance components we have a better understanding of what effect the early-stopping has. The results of this are shown in Figure 6 and it is clear that SA-ES generally reduces the variance component of error. This supports the hypothesis that SA-ES reduces overfitting.

5. Conclusions and Future Work

In this paper we have proposed early-stopping as a policy for preventing overfitting in wrapper-based feature subset selection that uses stochastic search. We have

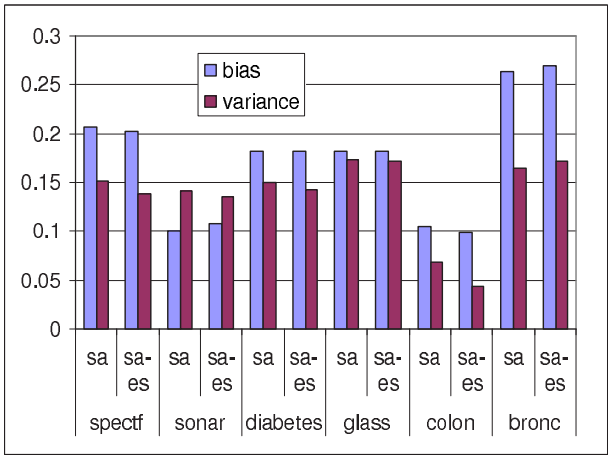

Figure 6. A comparison of the Bias-Variance decomposition of error for the SA and SA-ES strategies

described implementations of this idea for Genetic Algorithms and Simulated Annealing. Our evaluation shows that the strategy is effective in general being most successful with Simulated Annealing.

The main problem with this approach is the instability of the stochastic search which can lead to the early-stopping strategy returning poor feature subsets from time to time. So far we have tried to fully automate the early-stopping process, our next objective is to develop an interactive workbench where the user will be shown the overfitting graphs and will be allowed to evaluate a range of early-stopping alternatives.

References

Aha, D., & Bankert, R. (1994). Feature selection for case-based classification of cloud types: An empirical comparison. *AAAI 1994 Workshop on Case-Based Reasoning*. AAAI Press.

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.

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*.

Kohavi, R., & Sommerfield, D. (1995). Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. *KDD* (pp. 192–197).

Kohavi, R., & Wolpert, D. (1996). Bias plus variance decomposition for zero-one loss functions. *Machine Learning: Proceedings of the Thirteenth International Conference* (pp. 275–283). Morgan Kaufmann.

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.

Quinlan, J. (1993). *C4.5: Programs for machine learning*. San Francisco: Morgan Kaufmann.

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

van der Putten, P., & van Someren, M. (2004). A bias-variance analysis of a real world learning problem: The coil challenge 2000. *Machine Learning*, *57*, 177–195.

Walsh, P., Cunningham, P., Rothenberg, S. J., O'Doherty, S., Hoey, H., & Healy, R. (2004). An artificial neural network ensemble to predict disposition and length of stay in children presenting with bronchiolitis. *European Journal of Emergency Medicine*, *11*, 259–264.

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.