A Weight Analysis-based Wrapper Approach to Neural Nets Feature Subset Selection

Dietrich Schuschel
Longbow Apache Software
Boeing Company,
5000 E. McDowell, B347
Mesa, AZ 85205, USA
dietrich.schuschel@boeing.com
(602) 891-7223

Chun-Nan Hsu
Department of Computer Science & Eng.
Arizona State University
P.O. Box 875406
Tempe, AZ 85287, USA
chunnan@isi.edu
(602) 965-1757

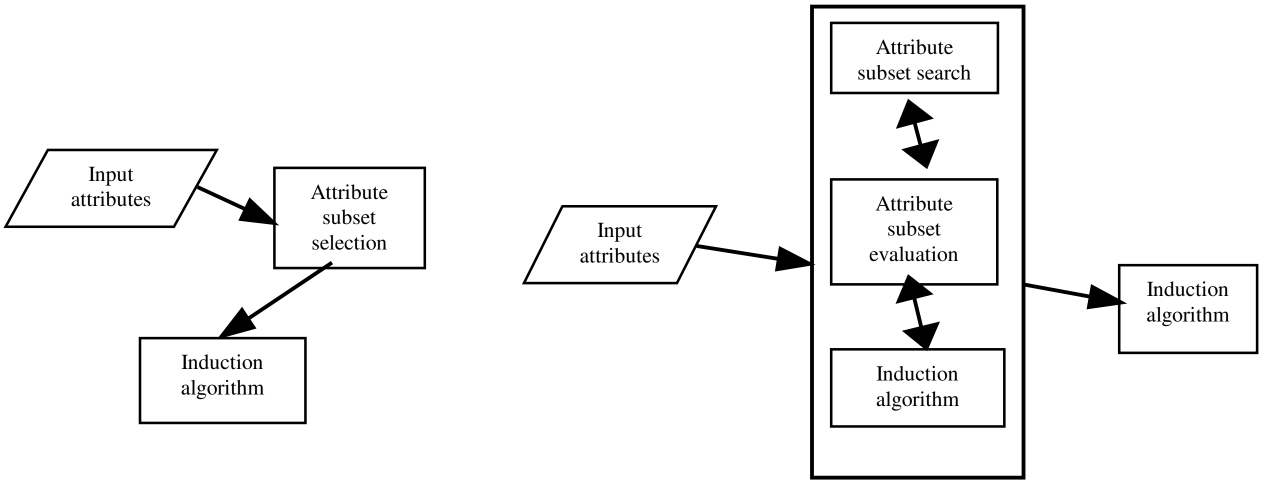
Abstract

This paper presents a novel attribute selection approach for backprop neural networks. Previously, an attribute selection technique known as *the wrapper model* was shown effective for decision trees induction. However, it is prohibitively expensive when applied to real-world neural net training characterized by large volumes of data and many attribute choices. Our approach incorporates a weight analysis based heuristic called ANNIGMA to direct the search in the wrapper model and allows effective attribute selection feasible for neural net applications. Experimental results on standard data sets show that this approach can efficiently reduce the number of inputs while maintaining or even improving the accuracy. We also report two sucessful applications of our approach in the helicopter maintenance applications.

Subject Areas: Neural networks, Machine Learning

1 Introduction

Selection of relevant attributes (features) is of primary importance to the success of a neural net. The goal is to find the minimum subset of attributes that yield the highest accuracy. This problem is especially severe when real-world applications are attempted and human selection of attributes is not available, desirable, or dependable. There are two models of feature subset selection. In *the filter model*, the attributes are filtered independently of the induction algorithm. This filtering is done as a preprocessing step as shown in Figure 1(a). In contrast, *the wrapper model* (John, Kohavi, Pfleger, 1994) wraps around the induction algorithm, searching the attribute subset space guided by the performance of the induction algorithm as illustrated in Figure 1(b).



(a) The Filter model (b) The wrapper model

Figure 1 Filter model v.s. wrapper model of feature subset selection

Since the filtering model ignores the effect of the attribute subset on the performance of the induction algorithm, many researchers have pointed out that it may not be as effective and general as the wrapper model (John, Kohavi, Pfleger, 94; Kohavi, Sommerfield 95; Caruana, Freitag 94). They make the point that attribute subset selection must take into account the biases of the induction algorithm in order to perform well. However, since in the wrapper model, a large number of training is required to search for the best performing attribute subset, it can be prohibitively expensive for neural net applications. Many search strategies were proposed to speed up the search, including hill climbing (Caruana, Freitag 94), compound operators (Kohavi, Sommerfield 95), randomized algorithms (Liu, Setiono 96a), et cetra. However, when applied to neural net application, at each branching point of the search, these approaches still need to train m neural nets with cross validation to select the next attribute subset, where m is the branching factor. This can be prohibitively expensive in real-world applications. For example, in one of our applications, one training cycle takes 200 hours to process. For k = fifty attributes, one branch point would take 200*50 = 10,000 hours.

A directed attribute subset search is needed for neural nets. In this paper's approach, the attribute subset search is accelerated by a heuristic called Artificial Neural Net Input Gain Measurement Approximation (ANNIGMA). ANNIGMA ranks neural net attributes by relevance. Following each training, the ANNIGMA heuristic ranks the attributes by relevance. This makes it unnecessary to train *m* neural nets for each branching point. A huge improvement in speed is now realized. The improvement is of the order linear versus exponential time required when compared to the base wrapper model.

The algorithm has been implemented into a system in C and MATLAB and tested on standard datasets. The results show that the ANNIGMA-wrapper approach can select relevant attributes for artificial datasets and smaller subsets for real-world datasets compared to previous work. We also report two successful applications of the ANNIGMA-wrapper approach in helicopter maintenance, ultrasonic pulse-echo signal classification and helicopter rotor blade stress prediction. This paper also reports two successful real-world applications of our approach in helicopter maintenance. The first application is improvement of an ultrasonic pulse-echo signal classification system that will be fielded by the US Army to monitor health of composite laminate helicopter structures. The second application is development of a neural net-based system that predicts helicopter rotor blade strain. The reduced list of inputs is particularly beneficial to the second application where expensive physical sensors must be designed and installed.

The remainder of this paper is organized as follows. Section 2 surveys related work in the field of machine learning and neural nets. Section 3 describes the ANNIGMA heuristic and how the ANNIGMA heuristic is used with the wrapper model. Section 4 reports the results of the experiments on standard data sets. In section 5, we report the experimental results for the pulse-echo domain and the experimental results for the strain prediction domain. Conclusions and future work are discussed in section 7.

2 Related Work

This section reviews closely related work. We first review research into the attribute subset selection problem. Next, the development of the wrapper model, designed to improve attribute subset selection is reviewed. Finally, we review subset space search strategies for use with the wrapper model.

2.1 The optimal attribute subset and the wrapper model

Relevant feature identfication is first discussed in (Almuallim, Dietterich, 92). The selection of optimal attribute subsets has some interesting recent developments in the machine learning. In (Caruana, Freitag, 94), the authors state that excess attributes are detrimental to a learning system in two ways. They can contribute noise and can interact counterproductively to the learning algorithm's bias. (John, Kohavi, Pfleger, 94) points out that to measure the relevance of a given attribute, one must take the existence and relevance of other attributes into account.

In follow up work, (Kohavi, Sommerfield, 95) considers that the optimality of an attribute subset depends on both the specific induction algorithm and the training data at hand. Therefore, an optimal attribute subset for a given induction algorithm and a given training set is defined as a subset such that the induction algorithm can generate a hypothesis with the highest predictive accuracy in the training set. The problem with this definition is that for real-world applications, the expected real-world performance of the selected subset should also be considered. To remedy this problem, we estimate real-world performance against a reserved set of data exclusively for testing in all of our experiments.

The definition of the optimal attribute subset leads to *the wrapper model* (John, Kohavi, Pfleger, 94; Kohavi, Sommerfield, 95) in contrast to *the filter model* applied in previous work.

2.2 Searching the attribute subset space

(John, Kohavi, Pfleger, 94) suggests many searche strategies to search for the optimal attribute subset. One simple algorithm, called *backward elimination*, starts with all the attributes and greedily removes the one that most improves performance (or degrades it the least). Another algorithm, *forward selection*, is also suggested. This algorithm starts with an empty set of attributes, and greedily adds attributes. (Caruana, Freitag, 94) describes five hillclimbing methods in the context of the wrapper model. The five methods are *forward selection* (FS), *backward elimination* (BE), *forward stepwise selection* (FSS), *backward stepwise elimination* (BSE), and BSE–SLASH. The term *stepwise* implies that attributes can be added or removed at each step. BSE-SLASH eliminates (slashes) any attributes that were not used during the last wrapper cycle. These hillclimbing methods improve on the base wrapper model by providing direction to the search, but still require a large amount of computing resources. Results indicate that the bi-directional hillclimbers (FSS, BSE, and BSE-SLASH) outperform both FS and BE. The bi-directional methods also allow stopping and restarting to find better answers. In addition, any selection method is better than no selection method.

(Kohavi, Sommerfield 95) describes the use of compound operators is suggested as a way to increase attribute search efficiency. Compound operators imply making more than one change to the attribute subset after all the child-nodes are evaluated. Although these approaches improve performance, they still require *m* training passes per wrapper cycle.

2.3 Recent Advances

Recently, more clever search strategies have been studied to improve the efficiency of the feature selection. In (Liu, Setiono, 96a), a probabilistic wrapper algorithm is proposed. This algorithm is called Las Vegas Wrapper (LVW). This algorithm randomly selects an attribute subset, tests it, then compares it with the current best attribute subset. This algorithm ensures that for sufficiently large counter stopping criteria, an optimal subset will be found. This method is good for small attribute sets, and small datasets, but becomes impractical for larger searches.

In (Liu, Setiono, 96b), a probabilistic filter approach is proposed called Las Vegas Filter (LVF). In this algorithm, random attribute subsets are selected based on *inconsistency* criterion. This method is simple, fast, and unaffected by any learning bias. However, noisy data requires that a minimum allowable inconsistency rate be adjusted from a priori knowledge for best results. This method does not work with continuous data values. Also, this criteria will always select unique identifiers like serial numbers or social security numbers.

A Neural Network Feature Selector (NNFS) is proposed in (Setiono, Liu, 97). This method adds a penalty term to the error function used to train the network. For each attribute in the network, the accuracy is computed with all weights associated with that attribute set to 0. The attribute that gives the smallest decrease in network accuracy is removed. The network is then retrained and the process is repeated. This method is very similar to the ANNIGMA-wrapper method. However, it requires modification to the neural network training algorithm, which may not be generally available with neural network packages.

In (Yang, Hanovar, 97), experimental results are presented for neural networks constructed with *DistAL* (Yang, Parekh, Hanovar, 97). DistAL is a constructive neural network learning algorithm. DistAl adds hidden neurons, one at a time, until a network can completely classify a training set. Our experience with a similar induction process that adds neurons until accuracy is achieved (radial-basis neural nets), the generalization capabilities were generally poorer than standard sigmoidal-based feed-forward neural nets.

3 The ANNIGMA-Wrapper Approach

The heuristic called ANNIGMA ranks attributes by relevance. The reasoning behind this heuristic is based on the back-propagation algorithm. Neural net weights can be viewed as representing the gain of the input signal to the output node. The back-propagation algorithm adjusts these weights such that the output of the net tracks the target signal (Haykin 94). The back-propagation algorithm works by minimizing the error of the net's output with the target signal. Input signals that are noisy or irrelevant to the output will have a high error rate if the neural net attempts to rely on them. Therefore, these signals will have their gains reduced to the extent that the signal is irrelevant and/or noisy and does not contribute to the output. In a similar manner, the weights of relevant and noise-free signals will be increased.

3.1 An Approximate Metric for Total Gain

The equation for a two layer neural net with the first layer having a logistic activation function, $\varphi(x) = \frac{1}{1+\exp(-x)}$, and the second layer having a linear output, is: $O_k = L_k * \sum_j \varphi\left(\sum_i A_i * W_{ij}\right) * W_{jk}$,

where i, j, k are the input, hidden, and output layers node indices, respectively,

L are the second layer linear multiplier values, A are the input nodes (attributes), O are the output nodes, and W are the weights between the layers.

The output as a function of a single input A_i, can be expressed as

$$O_k = L_k * \sum_j \varphi\left(A_i * W_{ij} + C_{ij}\right) * W_{jk} \tag{1}$$

where C_{ij} represents the constant value of all the other inputs, including biases. C_{ij} here acts as a 'setpoint' on the logistic function curve. This equation, with all of the inputs processed through the logistic function, is too complex to analyze directly. An approximation can be made of the total relative gain, G, of a particular input node i to a particular output node j. The approximation substitutes a linear factor for the logistic activation function. The approximation's error is reduced when the inputs are all in the same range. If we substitute a linear factor, F, for the activation function, we have

$$O_k = L_k * \sum_j F * \left(A_i * W_{ij} + C_{ij}\right) * W_{jk}\ . \tag{2}$$

Local Gain, LG, is defined to be

$$LG_{ik} = \left|\frac{\Delta O_i}{\Delta A_k}\right|. \tag{3}$$

The ANNIGMA value is the local gain, LG, normalized to a scale of 100:

$$ANNIGMA_{ik} = \frac{LG_{ik}}{\max\left(LG_k\right)} * 100\,. \tag{4}$$

Removing constants and common factors, LG can be calculated from (2) and (3) as

$$LG_{ik} = \sum_j \left|W_{ij} * W_{jk}\right|. \tag{5}$$

When the input attributes are scaled in the same range, the weights will give an approximation of the relative gain of each of these attributes. Although a set of weights can be contrived to contradict this heuristic, a backpropagation trained net has not yet exhibited such behavior in the author's experiments.

3.2 An illustrative example of the ANNIGMA Heuristic

We synthesized a small dataset to test and illustrate the predictive power of ANNIGMA heuristic. The dataset for this example has binary valued attributes. These binary values satisfy the normalization requirement of the ANNIGMA heuristic. The dataset contains six columns of input attributes (A1-A6) and one column of target outputs (O1). The network is trained using all six input attributes. The ANNIGMA heuristic then identifies attribute relevance.

Three meta-signals are represented in this dataset, RAND, XOR and SUM<2. The first meta-signal, RAND, denoted as the first column (A1), is a random signal from {0,1} representing noise and is totally irrelevant to the target output O1. The next meta-signal, XOR, is the next two columns, relating to the target through the XOR function (O1 = XOR (A2, A3)). This tests the algorithm's ability to classify linearly attributes as relevant. The third meta-signal, SUM<2, is the next three columns, relating to the target through their sum. (O1 = (A4 + A5 + A6 < 2)) (Table 1). SUM<2 is linearly separable and an

easy concept to learn; a decision tree induction algorithm will always choose these attributes as decision points as opposed to the linearly inseparable XOR function

Table 1.Table 1 Sample records of synthesized data set

Random	XOR		Sum<2			Target
A1	A2	A3	A4	A5	A6	O1
Rand(0,1)	1	1	1	1	1	0
Rand(0,1)	0	0	1	1	0	0
Rand(0,1)	1	0	0	0	1	1
Rand(0,1)	0	1	0	1	0	1
Rand(0,1)	0	1	1	0	0	1

Under uncorrupted conditions, the target output O1 and the meta-signals XOR and SUM<2 must agree, and other combinations of attributes are not possible (i.e., XOR(A2,A3) = (A4 + A5 + A6 < 2) = O1). This redundancy makes it easy for a neural net to learn the target concept in a noise-free environment. However, to simulate large-scale, real-world situations, where noise exists along with redundancy, the fourth attribute (A4) was corrupted by flipping the bit with a twenty percent probability. (i.e., P[A4 <= not(A4)] = 0.20) This tests the ability of the heuristic to select the noise-free attributes.

The ANNIGMA output is given in (Figure 2). For each of the six attributes (Attribute 1-6), there are six bars representing the gain, G, for each of six neural net training iterations. The rightmost (black) bar of each attribute reflects a fully trained net and is the ANNIGMA output.

Figure 2 shows that within a few training iterations, ANNIGMA correctly suppressed the noisy inputs, A1 and A4. ANNIGMA correctly identified the XOR signal (A2 and A3) as the most relevant in thirty of thirty different neural net initializations. The wrapper algorithm can use the ANNIGMA output to reduce the attribute subset size to two: columns A2 and A3, which are the minimum attributes needed to correctly identify O1 with 100% accuracy.

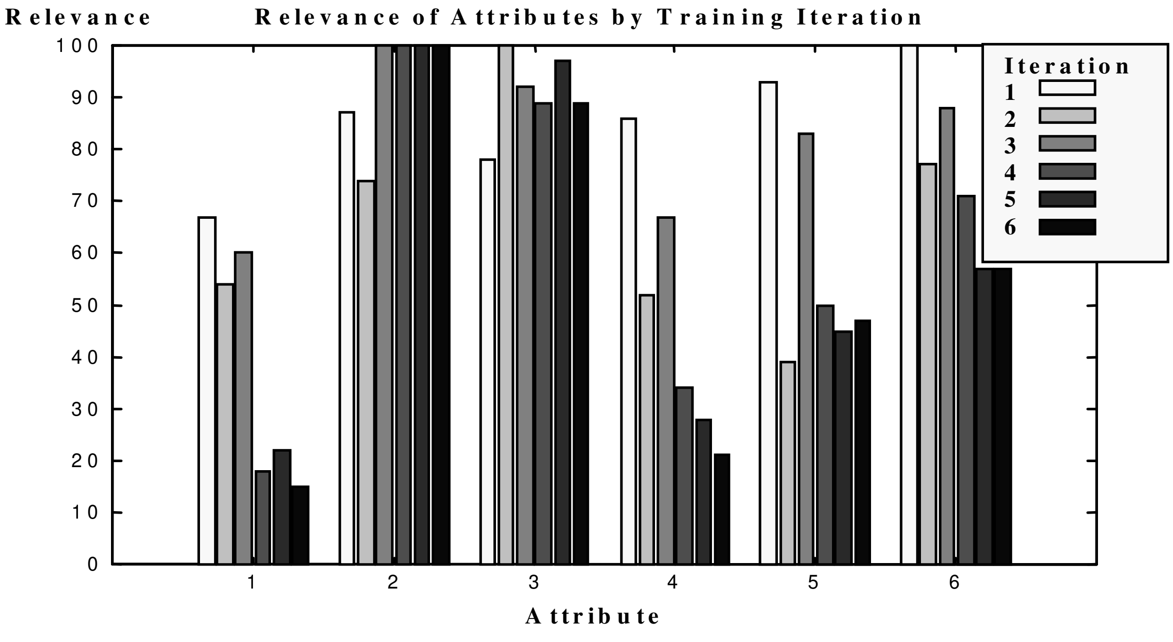


Figure 2 ANNIGMA output by training iteration

3.3 Integrating ANNIGMA into the Wrapper Algorithm

We now describe how to integrate the ANNIGMA heuristic inot the wrapper model. In particular, we describe **Procedure ANNIGMA-wrapper**. The input to this procedure is the set of attributes, and their associated training data. On the top level, this procedure consists of two nested loops. The outter loop, called *the wrapper cycle*, selects the next subset to evaluate based on the ANNIGMA evaluation of each attribute and the performance of the neural network using this subset of attribute in terms of its accuracy.

The accuracy and the ANNIGMA evaluation are obtained in the inner loop called *the training cycle.* The accuracy is evaluated using multiple initializations and n-fold cross-validation. Once a training cycle is completed, the ANNIGMA heuristic is evaluated as described in section 3. The attribute subset selection strategy, **Procedure Subset-select**, selects the subset of attributes for the next wrapper cycle. This procedure can be instantiated by any backword selection strategy, such as a BE or a BSE strategy (Caruana, Freitag, 94). The advantage of backward selection is that it starts with all the attributes on the first cycle, which allows an initial ANNIGMA evaluation of all attributes to quickly eliminate the least relevant ones. Another benefit is the relative linearity of the resulting attribute subset search. In the next section, we will show that for small data sets, BE is sufficient to select the relevant attributes. We will discuss a variation of BSE to deal with large real world datasets in Section 5. This section focuses on **Procedure ANNIGMA-wrapper**, which is listed as follows:

```
Procedure ANNIGMA-wrapper
1. Input: D = an initial set of N attributes
2. Let E = D
3. For i = 1 to I wrapper cycles
4.    For j = 1 to J cross-validations
5.          Let F = neural net trained on E
6.          Let Error(i,j) = test results of F
7.          Let A(i,j,E) = ANNIGMA(F) weighted by Error(i,j)
8.    call Subset-select
```

Procedure ANNIGMA-wrapper sets up the programming loops around the induction algorithm. In line 2, *E*, our current attribute subset, is initialized to the entire set of attributes available, *D*. In line 3, a For statement is used, indicating a finite number of wrapper cycles. Other stopping criteria are used in practical applications, including clock time, or performance on test data. In line 4, the variable *J* represents the total number of cross-validation subsets used. In line 5, *F* represents the weights and biases of a trained neural net. Here, the training takes place on the data represented by the E attributes. In line 6, the neural net is tested on the portion of the data left out due to cross-validation. There is an additional set of data for final performance testing which is not used here. In line 7, the ANNIGMA values are weighted by their performance. This ensures that the better performing neural nets have proportionately greater influence on the final ranking of attributes.

Note that a large amount of neural net training takes place during each cycle. It is important to adjust the neural net training parameters such that time is not wasted overtraining the nets. It has been found that only a few (<10) epochs (dataset presentations) can generate useable ANNIGMA results. In line 8, after all cross-validation training and testing is complete, **procedure Subset-select,** the attribute subset selection heuristic, is called.

4 Tests Against Standard Datasets

This section reports experiments exploring the results of using the ANNIGMA-wrapper algorithm against standard artificial and real-world datasets obtained from the UCI Machine-Learning Repository (Merz, Murphy, 98). These datasets were chosen to maximize the comparability of the ANNIGMA-wrapper approach with other published approaches to attribute subset selection, especially those concerning neural network induction (Liu, Setiono, 96a), (Liu, Setiono, 96b), (Setiono, Liu, 97), (Yang, Hanovar, 97).

4.1 Standard Datasets

Artificial Datasets

The optimal attribute subset is known for artificial datasets.

CorrAL This is the same dataset first described in (John, Kohavi and Pfleger 94). There are six binary attributes, a_1-a_6. Attribute a_5 is irrelevant, and attribute a_6 is correlated to the class label 75% of the time. The target concept is (a_1 and a_2) or (a_3 and a_4). Both ID3 and C4.5 choose attribute a_6 as the root. The dataset used consisted of 128 generated records, half of which were randomly chosen and exclusively reserved for testing. A optimal result would be selection of $a_{1\text{-}4}$ and 0% error rate. This dataset is very similar to the one used in section 3.2.

Monk3 This dataset has six attributes and 432 samples and 6 attributes, a1-a6. The target concept is (a5 = 3 and a4 = 1) or (a5 /= 4 and a2 /= 3). 5% class noise is added to the training set. The ML Repository has partitioned the data into training and test sets. Two different trials were run on this dataset. In the first, Monk3a, the attributes were used as-is. In the second, Monk3b, the 6 features in the database were expanded to 15 binary attributes where each binary attribute is value of each multi-valued attribute. For example, an attribute that can take 3 values becomes 3 attributes where 2 of the 3 will always have the value 0, the other one having the value 1. This was done so that some comparison could be made to the technique presented in (Setiono, Liu, 97), where it is used to enhance the effectiveness of attribute. An optimal result is the selection of attributes a5, a4, and a2, and an error rate of less than 5%.

Real-world Datasets

Vote This dataset consists of the voting records of 435 congressmen on 16 issues in the 1984 congress, 2nd session. The votes are classified into yea, nay, and unknown. The classification problem is to predict, for each congressmen, whether they are republican or democrat. 218 training instances were randomly selected from the entire dataset as in (Setiono, Liu, 97). The dataset was digitized by setting nay to 0, yea to 1, and unknown to 0.5.

Credit This dataset contains 690 records of 15 attributes. The records are classified into 2 categories. The dataset was divided into 490 training and 200 test instances. 37 records have one or more missing values. Several of the attributes consisted of multiple values. The multi-valued attributes have each of their values set to digital values corresponding to their position in the *creditnames.txt* file accompanying the dataset, starting with 0. Unknown values are set to 0.5.

4.2 Experimental Procedure

To estimate the accuracy of the attribute subsets, 10-fold cross validation on the training set was used. The test data was presented to the final attribute subset's neural net only after the final attribute subset was selected and associated neural net was trained. Backwards Elimination (BE) was chosen as the attribute subset search strategy. BE is a simple and linear running-time heuristic that will stress the ability of the ANNIGMA-wrapper algorithm to correctly eliminate the least relevant attribute at each step. BE guarantees that the run-time will be limited to fewer wrapper cycles than the number of attributes. Additional stopping criteria includes detection of strongly increasing (>10%) error rate. 30 applications of the ANNIGMA-wrapper were completed for each dataset. The results of the 30 trials were averaged, and the standard deviation is also reported.

4.3 Results

Table 2 shows that the ANNIGMA-wrapper procedure is very competitive with other published methods of attribute subset selection. The top section contains data pertinent to this experiment. After the first row identifies the name of the dataset, the second row identifies the transfer function used in the hidden layer of the neural net. The linear transfer function resembles a = bx + y. The tansig transfer function is a sigmoidal function based on the tangent function. The tansig transfer function might be considered more complex and powerful in it's modeling ability. The transfer function of the output layer is always linear. The third line in the table, 'Hidden', is the number of hidden units used, and the fourth line is the number of neural net training cycles used for training the reference net (all attributes) and all training in the ANNIGMA-wrapper procedure. The fifth line in the table is the number of initial attributes for the ANNIGMA-wrapper procedure, and the number of inputs for the reference neural nets. The sixth line, 'Neural net', gives the reference net's error rate in percent. All performance statistics are rounded to 2 significant digits near the decimal point.

The next 2 rows, ANNIGMA-wrapper Att and Err, give the performance of the ANNIGMA-wrapper algorithm in Attributes selected and Error rate percent, respectively. This format is repeated in the following rows for other published results as listed in the notes following the table. The results for (John, Kohavi, Pfleger, 94) were obtained from (Liu, Setiono 97a), interpolated from the original's bar charts.

These results were achieved by examining a minimal number of attribute subsets. The backwards elimination (BE) heuristic removes one attribute with each wrapper cycle. For each dataset, the number of attribute subsets searched was less than the number of attributes. The ANNIGMA heuristic points out a direct route from the entire attribute set to an optimal or near-optimal attribute subset. In the next section, our results on the real-world application show that performance is further improved for more complicated datasets by allowing attributes to be re-introduced as in BSE.

For CorrAL and Monk3, the ANNIGMA-wrapper algorithm selected the correct attributes. For CorrAL, the correct 4 were always chosen, occasionally adding an irrelevant one. ANNIGMA-wrapper had the best performance of those algorithms tested against this dataset, improving significantly over the base neural net's performance.

For Monk3a, using attributes as-is, the correct attributes 2, 4, and 5 were always selected. For Monk3b, the correct attributes were chosen, but never the minimum subset. Note that when using the expanded representation, there are aliases for each attribute (a4 = 1 is equal to a4 /= 2 and a4 /= 3). Comparing Monk3a and Monk3b, we see that ANNIGMA-wrapper can identify relevant attributes even when they are multiple-valued. Since the Monk3 training dataset has 5% class noise, and the test dataset contains the training dataset, it is expected that the best achievable results would be less than 100%. The 12 noisy records out of a total of 432 records results in an expected best error rate of 2.78%. This is exactly the error rate achieved by ANNIGMA-wrapper is most of it's cases. It is unclear how some of the other induction algorithms were able to achieve a better error rate.

The vote dataset appears to be a linearly separable problem. ANNIGMA-wrapper was able to employ a completely linear neural net architecture to solve it. This also resulted in excellent performance, equal to the best reported among those that did not train on the entire dataset. This linear seperability can explain why the decision tree algorithms performed very well on this problem.

The credit dataset is perhaps the most difficult of all of these standard datasets. It generates the highest error rates for any induction algorithm. The ANNIGMA-wrapper algorithm required a large number of hidden nodes and use of the non-linear tansig transfer function. In earlier trials, a smaller number of hidden nodes and cycles yielded a 16% error rate. Attribute expanding was also attempted here, but results, note reported here, were time-consuming and disappointing. The resulting dataset consisted of 49 attributes, resulting in a even larger weight matrix. The lesson learned from this is that attribute expansion can have a negative effect when it does not offset the cost of adding another attribute. The final configuration of 10 hidden nodes was tried after a setting of 6 nodes produced inconsistent results. This researchers experience has been that as the error variance among the 30 trials is reduced, the best available error rate is achieved. This can be incorporated into a node-size heuristic in future work. The error rate for this dataset achieved by DistAL appears to be lower than ANNIGMA-wrapper. We want to note that their result is the training error rate rather than based on a separate test set (Yang, Hanovar, 97).

Table 2 Comparing ANNIGMA-wrapper with published results on standard data sets

Dataset		Corral	Monk3a	Monk3b	Vote	Credit
Transfer		Tansig	Tansig	Linear	Linear	Tansig
Hidden		10	8	3	3	10
Cycles		8	15	8	8	10
Attributes	Att	6	6	15[1]	16	15
Neural net	Err	2.3 (4)	10 (5.2)	2.8 (0)	3.2 (0)	14 (2.1)
ANNIGMA – wrapper	Att	4 (0.2)	2.3 (0.7)	2.2 (0.4)	3.3 (1.9)	6.8 (2.8)
	Err	0 (0.2)	2.9 (0.8)	2.8 (0)	3.1 (0.2)	12 (1.1)
SL – NNFS	Att	N/A	N/A	3.9 (1.8)	2 (0.2)	N/A
	Err	N/A	N/A	1.6 (1.7)	4.2(1.6)	N/A
YH – DistAL [2]	Att	N/A	N/A	N/A	8.9 (1.8)	8 (2.1)
	Err	N/A	N/A	N/A	1.2 (1.2)	8.5 (2.8)
JKP - C4.5 - FW	Att	2	3	N/A	1	3
	Err	19	2	N/A	3	19
JKP - C4.5 – BW	Att	5	2	N/A	15	14
	Err	0	1	N/A	3	21
LSa – C4.5 – LVF	Att	4	3	N/A	8	5
	Err	20	1.1	N/A	5.5	15.2
LSb – C4.5 – LVW	Att	4	3	N/A	3	6
	Err	6.2	2.8	N/A	4	15

1. 6 discrete attributes in the database expanded into 15 binary attributes for each value.
2. 10-fold cross-validation performance on training set.
JKF - (John, Kohavi, and Pfleger, 94) LSa – (Liu, Setiono, 96a) LSb - (Liu, Setiono, 96b)
SL – (Setiono, Liu, 97) YH – (Yang, Hanovar, 97) LVF – Las Vegas Wrapper LVF – Las Vegas Filter
NNFS – Neural Network Feature Selector C4.5 – (Quinlan 93) DistAL – Genetic neural net generation

In summary, these experimental results show that ANNIGMA-wrapper selects the optimal attribute subset in trivial cases, and it selects at least a near-optimal and better attribute subset than the other methods compared to in this paper.

5 Applications in the Helicopter Industry

This section reports two successful applications of the ANNIGMA-wrapper approach in the helicopter industry. The first application is a classification system for ultrasonic pulse-echo signals. This system classifies fixed-length, real-valued records into a small number of categories by a feed-forward neural network. The second application is the development of a system that predicts the strain on helicopter rotor blades. In this application, identification of signals that are relevant to blade strain prediction is a critical task since sensors need to be developed and manufactured. The resulting system would be part of a larger system called Health and Usage Monitoring System (HUMS). HUMS are currently a critical industry-wide topic that promises to reduce helicopter-operating costs while increasing reliability. The result show that the ANNIGMA-wrapper algorithm with BSE reduces the requirement for subject-matter experts during neural net development in both applications.

5.1 Backward Stepwise Elimination

The datasets in the helicopter applications are very large in terms of their size and the number of attributes. As a result, it is not practical to apply BE to select the subset in each wrapper cycle because in each wrapper cycle, BE eliminates only one attribute. We present a variation of Backward Stepwise Elimination (BSE) as the search strategy, which may eliminate or add more than one attributes in each wrapper cycle. This strategy is given as follows. It will be invoked as the subset selection procedure by **Procedure ANNIGMA-wrapper** when applied to the two real-world applications.

Procedure BSE-Subset-Select keeps the best performing (or adds previously discarded) attributes for the following wrapper cycle. This is adjusted based on the performance of the previous cycle. When the performance degrades, the best of the discarded attributes are brought back into the training set.

```
Procedure BSE-Subset-Select
Sort E by their ANNIGMA values, averaged over all cross-validations
A(i,J,E)
Let H = the sorted history-averaged ANNIGMA rank of the discarded
attributes
Error(i) = Average Error(i,J)
if i < 4
   Let E = top 75% of E
elseif Error(i) <= minimum previous Error
   Let E = top 85% of E
elseif Error(i) <= mean of previous Error
   Let E = top 90% of E plus the highest H
elseif Error(i) <= maximum of previous Error
   Let E = top 90% of E plus the highest 2 H
else
   Let E = top 50% of E plus the highest size(E)/2 H
end
```

5.2 Pulse-Echo Experiment

The ANNIGMA-wrapper algorithm was applied to reduced the number of attributes in a neural network system which examines helicopter parts formed of laminated composite materials. In this system, the inspection process consists of sending a short ultrasonic pulse into the material under test. Material failures are associated with echo signal shapes. The system must classify these digitized waveforms into ‘good’ and ‘bad’ categories (see Figure 3 and 4). The raw echo signals are collected and digitized by an A/D converter. After several preprocessing and cleanup steps, each record of the signal contains 192 normalized attributes. These records are ready for the neural net to perform classification.

The ANNIGMA-wrapper algorithm uses all 192 attributes as it’s initial attribute set. The perfomance of each subset is evaluated on 150 pulse-echo waveform that were not used in training. After 48 wrapper cycle, 10 attributes are selected. This subset achieves 99% accuracy. Since no subset is found to improve the accuracy and a subset is selected as the final result.

Previously, we have used a wavelet-compression approach to attribute selection and select the top 25 maximum valued attributes. That system achieved 97 % accuracy. The ANNIGMA-wrapper algorithm, starting with the same wavelet coefficients, used only 10 attributes to achieve 99% accuracy. The intersection of the two sets contains only two attributes. It can therefore be concluded that the set of

ANNIGMA-wrapper selected coefficients is more relevant to the classification of pulse-echo waveform than those selected by wavelet compression.

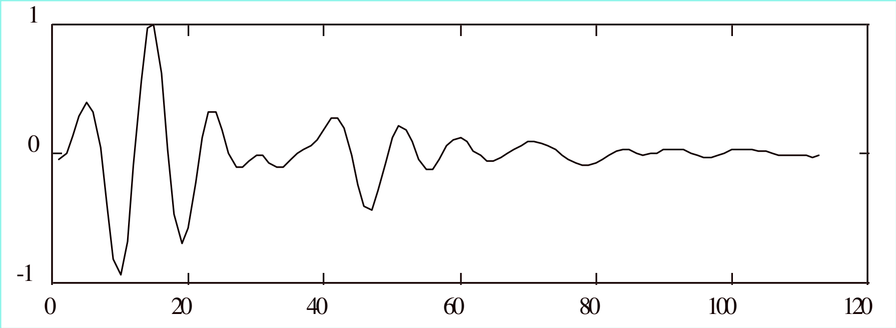


Figure 3 Good waveform

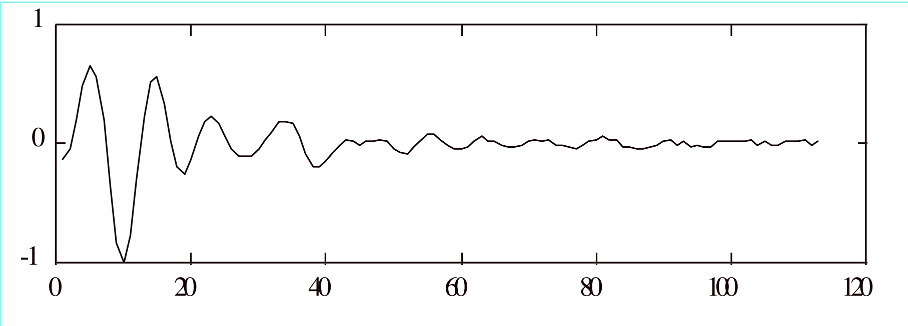


Figure 4 Waveform representing impact damage

5.3 Strain signal prediction

Helicopter rotor blade repair and replacement are major cost factors in helicopter operations. However, they are essential to assure helicopter safety. Current industry standard to measure the fatigue life is based on a fixed number of operating hours assuming most severe operation. This is an inefficient because perfect blades may be mistakenly scrapped. A proposed solution is to track the strains on the blades to calculate fatigue damage (Downing, S. 1982, ASTM E 1049-85, among others). However, it is not feasible to directly measure strains on production helicopters due to the high expense of a sensor/monitoring system, and the unreliability of a strain sensor mounted on a rotor blade. We attempt to solve this problem through a neural net trained to predict strain from other easily obtainable helicopter signals. The ANNIGMA-wrapper algorithm can help us to identify necessary sensors and signals.

The neural nets in this application is to approximate the strain gauge reading mounted on the helicopter blades as a function of the given sensory inputs. Initially, forty-one sensors from among thousands available on the helicopter were selected by flight dynamics experts. The selected sensors include accelerometers, strain, temperature, and pressure measurement devices. The sensory inputs are collected in real-world helicopter flight maneuvers at a 1000 samples /second rate. We have 197 sets of training datasets (1G Bytes large total). Each set contains the sensory data collected from 3 to 19 seconds periods.

Due to very large size of the datasets, the first wrapper cycle takes 200 hours of exclusive run-time on a 200 MHz Pentium Pro computer. The initial application of the ANNIGMA-wrapper algorithm generated the results in Table 3.

Table 3 Strain Prediction ANNIGMA-wrapper Results by Cycle

Wrapper Cycle	Number of Attributes	Average Absolute error
1	41	0.173
2	28	0.169
3	19	0.163
4	13	0.774

The The implication of these results is that the relevant information content in the 19 selected attributes is equal to that in the 41 attributes. Therefore, the 19 attributes have more specific relevance and are recommended for further system development. Additionally, the first wrapper cycle was re-run to check the repeatability of the ANNIGMA heuristic. It was found that the high-ranking attributes were within two ranks, and the maximum difference at the lower ranks was five.

6 Conclusion

We have shown that ANNIGMA, a weight analysis based heuristic, can accurately rank attributes by relevance to a target output. The ANNIGMA-wrapper performs equally well or even better on standard dataset compared with the previous work on feature subset selection, while the ANNIGMA-wrapper approach achieved this performance with much less training effort. Our results on the real-world helicopter applications show that the wrapper model attribute subset selection becomes practical when accelerated by the ANNIGMA heuristic.

Identified limitations of the ANNIGMA-wrapper algorithm include the requirement that attributes be normalized, which can be difficult to achieve in some problem domains. This problem is similar to the problem of extrapolation in any predictive algorithm. Re-normalization and repeating the ANNIGMA-wrapper algorithm is indicated when input ranges change significantly. Areas for further research include optimizing the degree to which a neural net needs to be trained before the ANNIGMA heuristic can be applied. Refining the attribute selection strategy is another area of development.

References

(Almuallim, Dietterich, 92) Almuallim, H., and Dietterich, T.G. (1992). Efficient algorithms for identifying relevant features. In Proceedings of the Ninth Canadian Conference on Artificial Intelligence, pp. 38-45. Vancouver, BC: Morgan Kaufmann.

(ASTM, 85) ASTM E 1049-85 1985, Standard Practices for Cycle Counting in Fatigue Analysis*, American Society for Testing and Materials*, Revision 1985.

(Caruana, Freitag, 94) Caruana, R., and Freitag, D. 1994, Greedy Attribute Selection*, Machine Learning: Proceedings of the Eleventh International Conference,* Morgan Kaufman, San Francisco, CA.

(Domingos, 96) Domingos, P. (1996). Context-Sensitive Feature Selection for Lazy Learners. Artificial Intelligence Review, 11, 227-253, 1997. (Special issue on lazy learning, edited by David Aha.)

(Downing, Socie, 82) Downing, S., and Socie, D. 1982, Simple Rainflow counting algorithms, *International Journal on Fatigue*, January, 1982.

(John, Kohavi, and Pfleger, 94) John, G. H., Kohavi R., and Pfleger, K. 1994, Irrelevant features and the Subset Selection Problem*, Machine Learning: Proceedings of the Eleventh International Conference*, Morgan Kaufman, San Francisco, CA.

(Haykin, 94) Haykin, Simon, 1994, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company, New York, NY.

(Kohavi, Sommerfield, 95) Kohavi, R., Sommerfield, D. 1995, Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology, *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA.

(Liu, Setiono, 96a) Liu, H., and Setiono, R., (1996). Feature selection and classification - a Probabilistic Wrapper Approach. In Proceedings of the 9th International Conference on Industrial and Engineering Applications of AI and ES.

(Liu, Setiono, 96b) Liu, H., and Setiono, R., (1996). A Probabilistic Approach to Feature Selection - A Filter Solution. In 13th International Conference on Machine Learning (ICML'96), July 1996, pp. 319-327. Bari, Italy.

(Merz, Murphy, 98) Merz, C.J., & Murphy, P.M. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

(Opitz, 97) Opitz, David W (1997). The Effective Size of a Neural network: A Principal Component Approach. Fourteenth International Conference on Machine Learning, (pp. 263-271), Nashville, TN. [http://www.cs.umt.edu/CS/FAC/OPITZ] University of Montana, Department of Computer Science.

(Quinlan, 93) Qunlan, J. Russ (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann, Los Altos, CA.

(Skalak, 94) Skalak, D. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. Proceedings of the Eleventh International Machine Learning Conference (ICML-94), pp. 293-301, New Brunswick, NJ: Morgan Kauffmann.

(Setiono, Liu, 97) Setiono, R., and Liu, H. (1997). Neural network feature selector. IEEE Transactions on Neural Networks, 8, (3), 654-662.

(Yang, Hanovar, 97) Yang, J., and Honavar, V. (1997). Feature subset selection using a genetic algorithm. IEEE Expert (Special Issue on Feature Transformation and Subset Selection). In press.

(Yang, Parekh, Hanovar, 97) Yang, J., Parekh, R. and Honavar, V. (1997). DistAl: An inter-pattern distance-based constructive learning algorithm. Technical Report ISU-CS-TR 97-05, Iowa State University.