2

Review of Input Variable Selection Methods for Artificial Neural Networks

Robert May[1], Graeme Dandy[2] and Holger Maier[3]
[1]*Veolia Water, University of Adelaide*
[2,3]*University of Adelaide*
Australia

1. Introduction

The choice of input variables is a fundamental, and yet crucial consideration in identifying the optimal functional form of statistical models. The task of selecting input variables is common to the development of all statistical models, and is largely dependent on the discovery of relationships within the available data to identify suitable predictors of the model output. In the case of parametric, or semi-parametric empirical models, the difficulty of the input variable selection task is somewhat alleviated by the *a priori* assumption of the functional form of the model, which is based on some physical interpretation of the underlying system or process being modelled. However, in the case of artificial neural networks (ANNs), and other similarly data-driven statistical modelling approaches, there is no such assumption made regarding the structure of the model. Instead, the input variables are selected from the available data, and the model is developed subsequently. The difficulty of selecting input variables arises due to (i) the number of available variables, which may be very large; (ii) correlations between potential input variables, which creates redundancy; and (iii) variables that have little or no predictive power.

Variable subset selection has been a longstanding issue in fields of applied statistics dealing with inference and linear regression (Miller, 1984), and the advent of ANN models has only served to create new challenges in this field. The non-linearity, inherent complexity and non-parametric nature of ANN regression make it difficult to apply many existing analytical variable selection methods. The difficulty of selecting input variables is further exacerbated during ANN development, since the task of selecting inputs is often delegated to the ANN during the learning phase of development. A popular notion is that an ANN is adequately capable of identifying redundant and noise variables during training, and that the trained network will use only the salient input variables. ANN architectures can be built with arbitrary flexibility and can be successfully trained using any combination of input variables (assuming they are good predictors). Consequently, allowances are often made for a large number of input variables, with the belief that the ability to incorporate such flexibility and redundancy creates a more robust model. Such pragmatism is perhaps symptomatic of the popularisation of ANN models through machine learning, rather than statistical learning theory. ANN models are too often developed without due consideration given to the effect that the choice of input variables has on model complexity, learning difficulty, and performance of the subsequently trained ANN.

Recently, ANN modellers have become increasingly aware of the need to undertake input variable selection (IVS), and a myriad of methods employed to undertake the IVS task are described within reported ANN applications—some more suited to ANN development than others. This review therefore serves to provide some guidance to ANN modellers, by highlighting some of the key issues surrounding variable selection within the context of ANN development, and survey some the alternative strategies that can be adopted within a general framework, and provide some examples with discussion on the benefits and disadvantges in each case.

2. The input variable selection problem

Recall that for an unknown, steady-state input-output process, the development of an ANN provides the non-linear transfer function

$$Y = F(X) + \varepsilon, \tag{1}$$

where the model output Y is some variable of interest, X is a k-dimensional input vector, whose component variables are denoted by $X_i (i = 1, \ldots, k)$, and ε is some small random noise. Let C denote the set of d variables that are available to construct the ANN model. The I_{d-k} problem of input variable selection (IVS) is to choose a set of k variables from C to form X (Battiti, 1994; Kwak & Choi, 2002) that leads to the optimal form of the model, F.

Dynamic processes will require the development of an ANN to provide a time-series model of the general form

$$Y(t+k) = F(Y(t), \ldots, Y(t-p), X(t), \ldots, X(t-p)) + \varepsilon(t). \tag{2}$$

Here, the output variable is predicted at some future time $t + k$, as a function of past values of both input X and output Y. Past observations of each variable are referred to as *lags*, and the model order p defines the maximum lag of the model. The model order reflects the persistence of dynamics within the system. In comparison to the steady-state model formulation, the number of variables in the candidate set C is now multiplied by the model order. Consequently, for systems with strong persistence, the number of candidate variables is often quite large.

ANN models may be specified with insufficient, or uninformative input variables (under-specified); or more inputs than is strictly necessary (over-specified), due to the inclusion of superfluous variables that are uninformative, weakly informative, or redundant. Defining what constitutes an optimal set of ANN input variables first requires some consideration of the impact that the choice of input variables has on model performance. The following arguments summarise the key considerations:

- *Relevance.* Arguably the most obvious concern is that too few variables are selected, or that the selected set of input variables is not sufficiently informative. In this case, the outcome is a poorly performing model, since some of the behaviour of the output remains unexplained by the selected input variables. In most cases, it is reasonable to assume that a modeller will have some expert knowledge of the system under consideration; will have surveyed the available data, and will have arrived at a reasonable set of candidate input variables. The *a priori* assumption of model development is that at least one or more of the available candidate variables is capable of describing some, if not all, of the output behaviour, and that it is the nature and relative strength of these relationships that is unknown (which is, of course, the motivation behind the development of non-parametric

models). Should it happen that none of the available candidates are good predictors, then the problem of model development is intractable, and it may be necessary to reconsider the available data and the choice of model output, and to undertake further measurements or observations before revisiting the task of model development.

- *Computational Effort.* The immediately obvious effect of including a greater number of input variables is that the size of an ANN increases, which increases the computational burden associated with querying the network—a significant influence in determining the speed of training. In the case of the multilayer perceptron (MLP), the input layer will have an increased number of incoming connection weights. In the case of kernel-based generalised regression neural network (GRNN) and radial basis function (RBF) networks, the computation of distance to prototype vectors is more expensive due to higher dimensionality. Furthermore, additional variables place an increased burden on any data pre-processing steps that may be undertaken during ANN development.
- *Training difficulty.* The task of training an ANN becomes more difficult due to the inclusion of redundant and irrelevant input variables. The effect of redundant variables is to increase the number of local optima in the error function that is projected over the parameter space of the model, since there are more combinations of parameters that can yield locally optimal error values. Algorithms such as the back-propagation algorithm, which are based on gradient descent, are therefore more likely to converge to a local optimum resulting in poor generalisation performance. Training of the network is also slower because the relationship between redundant parameters and the error is more difficult to map. Irrelevant variables add noise into the model, which also hinders the learning process. The training algorithm may expend resources adjusting weights that have no bearing on the output variable, or the noise may mask the important input-output relationships. Consequently, many more iterations of the training algorithm may be required to determine a near-global optimum error, which adds to the computational burden of model development.
- *Dimensionality.* The so-called *curse of dimensionality* (Bellman, 1961) is that, as the dimensionality of a model increases linearly, the total volume of the modelling problem domain increases exponentially. Hence, in order to map a given function over the model parameter space with sufficient confidence, an exponentially increasing number of samples is required (Scott, 1992). Alternatively, where a finite number of data are available (as is generally the case in real-world applications), it can be said that the confidence or certainty that the true mapping has been found will diminish. ANN architectures like the MLP are particularly susceptible to the curse due to the rapid growth in the number of connection weights as input variables are added. Table 1 illustrates the growth in the sample size required to maintain a constant error associated with estimates of the input probability, as determined by the pattern layer of a GRNN. Some ANN architectures can also circumvent the curse of dimensionality through their handling of redundancy and their ability to simply ignore irrelevant variables (Sarle, 1997). Others, such as RBF networks and GRNN architectures, are unable to achieve this without significant modifications to the behaviour of their kernel functions, and are particularly sensitive to increasing dimensionality (Specht, 1991).
- *Comprehensibility.* In many applications, such as in the case of ANN transfer functions for process modelling, it will often suffice to regard an ANN as a "black-box"' model. However, ANN modellers are increasingly concerned with the development of ANN

Dimension, d	Sample size, N
1	4
2	19
3	67
4	223
5	768
6	2790
7	10 700
8	43 700
9	180 700
10	842 000

Table 1. Growth of sample size with increasing dimensionality required to maintain a constant standard error of the probability of an input estimated in the GRNN pattern layer (Silverman, 1986).

models for knowledge discovery from data (KDD) and data mining (Craven & Shavlik, 1998). The goal of KDD is to train an ANN based on observations of a process, and then interrogate the ANN to gain further understanding of the process behaviour it has learned. Rule-extraction from ANN models can be useful for a number of purposes, including: (i) defining input domains that produce certain ANN outputs, which can be useful knowledge in itself; (ii) validation of the ANN behaviour (e.g. verifying that input-output response trends make sense), which increases confidence in the ANN predictions; and (iii) the discovery of new relationships, which reveals previously unknown insights into the underlying physical process (Craven & Shavlik, 1998; Darbari, 2000). Reducing the complexity of the ANN architecture, by minimising redundancy and the size of the network, can significantly improve the performance of data mining and rule extraction algorithms.

Based on the arguments presented, a desirable input variable is a highly informative explanatory variable (i.e a good predictor) that is dissimilar to other input variables (i.e. independent). Consequently, the optimal input variable set will contain the fewest input variables required to describe the behaviour of the output variable, with a minimum degree of redundancy and with no uninformative (noise) variables. Identification of an optimal set of input variables will lead to a more accurate, efficient, cost-effective and more easily interpretible ANN model.

The fundamental importance of the IVS issue is evident from the depth of literature surrounding the development and discussion of IVS algorithms in fields such as classification, machine learning, statistical learning theory, and many other fields where ANN models are applied. In a broad context, reviews of IVS approaches have been presented by Kohavi & John (1997), Blum & Langley (1997) and more recently, by Guyon & Elisseeff (2003). However, in many examples of the application of ANNs to modelling and data analysis applications, the importance of IVS is often understated. In other cases, the task is given only marginal consideration and this often results in the application of *ad hoc* or inappropriate methods. Reviews by Maier & Dandy (2000) and Bowden (2003) examined the IVS methods that have been applied to ANN applications in engineering and concluded that there was a need for a more considered approach to the IVS task. Certainly, no consensus has been reached regarding

suitable methods for undertaking the IVS task in the development of ANN regression or time-series forecasting models (Bowden, 2003).

3. Taxonomy of algorithms

Figure 1 presents a taxonomy, which provides some examples of the various approaches that have been proposed within ANN literature. IVS algorithms can be broadly classified into three main classes: *wrapper*, *embedded* or *filter* algorithms (Blum & Langley, 1997; Guyon & Elisseeff, 2003; Kohavi & John, 1997), as shown in Figure 1. These different conceptual approaches to IVS algorithm design are illustrated in Figure 2. Wrapper algorithms, as shown in Figure 2(a), approach the IVS task as part of the optimisation of model architecture. The optimisation searches through the set, or a subset, of all possible combinations of input variables, and selects the set that yields the optimal generalisation performance of the trained ANN. As the name indicates, embedded algorithms (Figure 2(b)) for IVS are directly incorporated into the ANN training algorithm, such that the adjustment of input weights considers the impact of each input on the performance of the model, with irrelevant and/or redundant weights progressively removed as training proceeds. In contrast, IVS filters (Figure 2(c)) distinctly separate the IVS task from ANN training and instead adopt an auxiliary statistical analysis technique to measure the relevance of individual, or combinations of, input variables.
Given the general basis for the formulation of both IVS wrapper and filter designs, the diversity of implementations that can possibly be conceived is immediately apparent. However, designs for wrappers and filters share the same overall components, in that, in addition to a measure of the informativeness of input variables, each class of selection algorithms requires:

1. a criterion or test to determine the influence of the selected input variable(s), and
2. a strategy for searching among the combinations of candidate input variables.

3.1 Optimality Criteria

The optimality criterion defines the interpretation of the arguments presented in Section 2 into an expression for the optimal size k and composition of the input vector, X. Optimality criteria for wrapper selection algorithms are derived from, or are exactly the same as, criteria that are ultimately used to assess the predictive performance of the trained ANN. Essentially, the wrapper approach treats the IVS task as a model selection exercise, where each model corresponds to a unique combination of input variables. Recall that the most commonly adopted measure of predictive performance for ANNs is the mean squared error (MSE), which is given by

$$\text{MSE} = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2 \tag{3}$$

where y_j and $\hat{y}_j$ are the actual and predicted outputs, which correspond to a set of test data. Following the development of m models, a simple strategy is to select the model that corresponds to the minimum MSE. However, the drawback of this criterion is that the "best" performing model, in terms of the MSE, is not necessarily the "optimal" model, since models with a large number of input variables tend to be biased as a result of over-fitting. Consequently, it is more common to adopt an optimality criterion such as Mallows' C_p (Mallows, 1973), or the Akaike information criterion (AIC) (Akaike, 1974), which penalise overfitting. Both Mallows' C_p and the AIC determine the optimal number of input variables

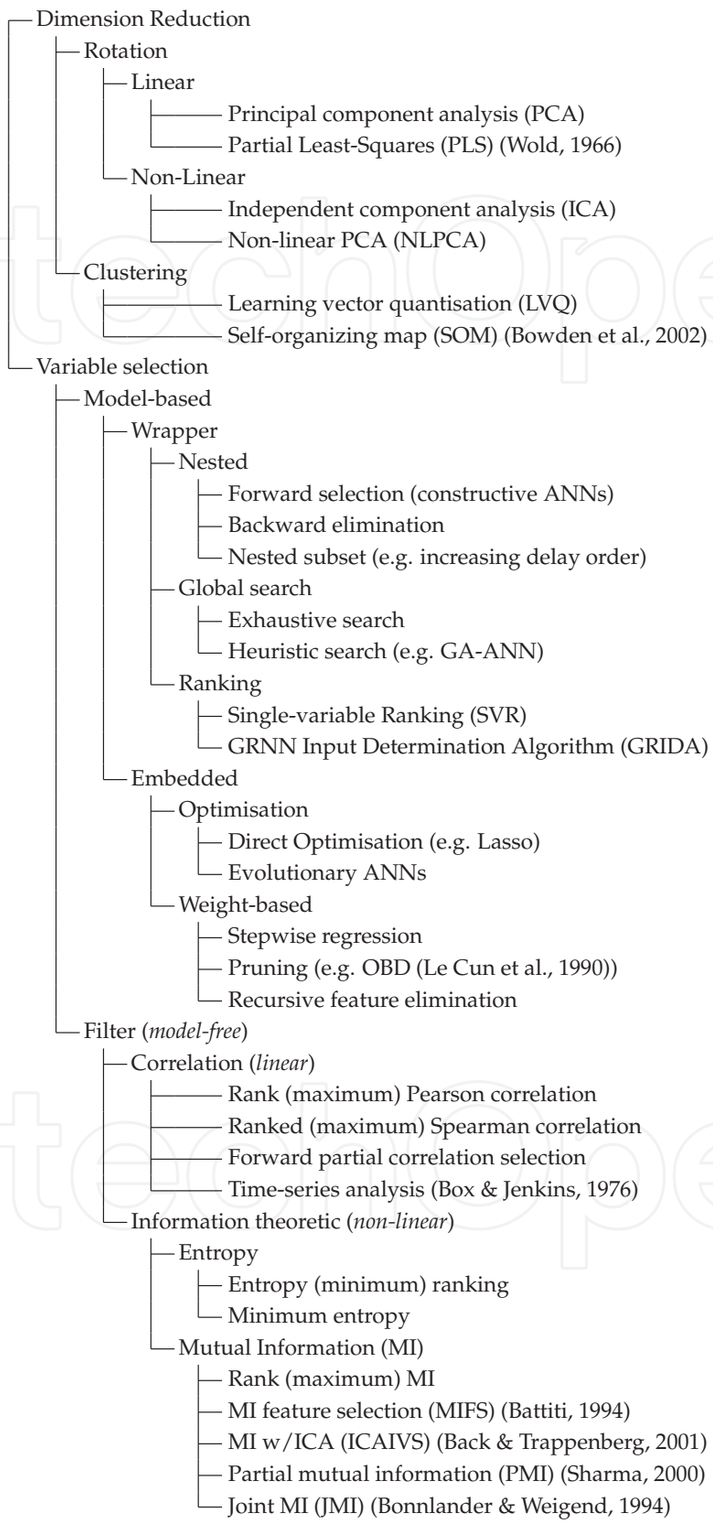
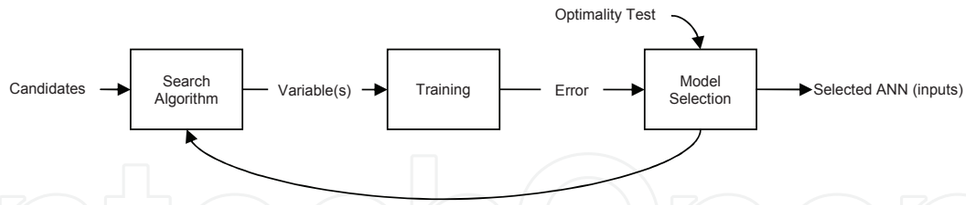
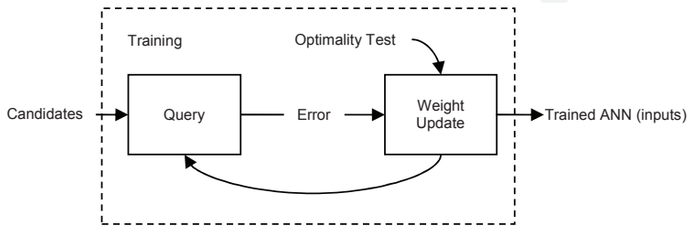


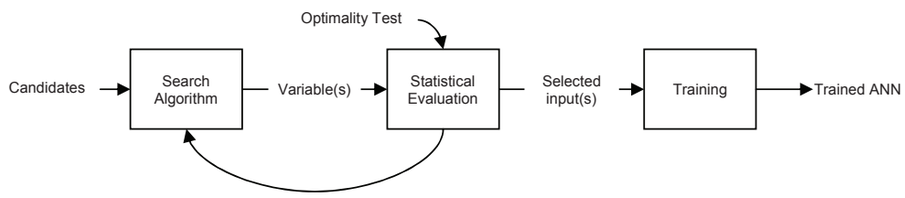
Fig. 1. Taxonomy of IVS Strategies and Algorithms



(a) Wrapper



(b) Embedded



(c) Filter

Fig. 2. Conceptual IVS approach using a (a) wrapper, (c) embedded, or (b) filter algorithm.

by defining the optimal trade-off between model size and accuracy by penalising models with an increasing number of parameters. In fact, the C_p criterion is considered to be a special case of the AIC.
Mallows' C_p is is defined as

$$C_p = \frac{\sum_{j=1}^{n} \left(y_j - \hat{y}_j(k) \right)^2}{\sigma_d^2} - n + 2p, \tag{4}$$

where $y_j(k)$ are the outputs generated by a model using p parameters, and σ_d^2 are residuals for a full model trained using all d possible input variables. C_p measures the relative bias and variance of a model with p variables. The theoretical value of C_p for an unbiased (optimal) model will be p, and in model selection, the model with the C_p value that is closest to p is selected.

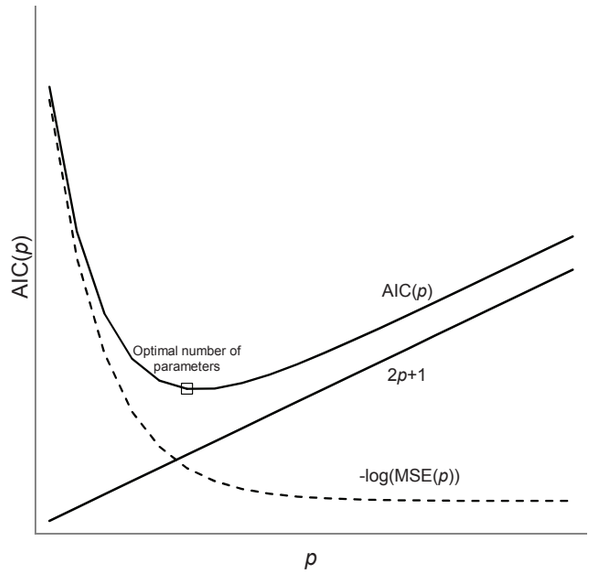


Fig. 3. The Akaike Information Criterion determines the optimum trade-off between model error and size

The AIC is defined as

$$\mathrm{AIC} = -n \log \frac{\sum_{j=1}^{n} \left(y_j - \hat{y}_j(k) \right)^2}{n} + 2(p+1). \tag{5}$$

Here, the accuracy is determined by the log-likelihood, which is a function of the MSE. The complexity of the model is determined by the term $p+1$, where p is the number of model parameters. Typically, the regression error decreases with increasing p, but since the model is more likely to be over-fit for a fixed sample size, the increasing complexity is penalised. At some point an optimal AIC is determined, which represents the optimal trade-off between model accuracy and model complexity. The optimum model is determined by minimising the AIC with respect to the number of model parameters, p. This is illustrated in Figure 3.
Other model selection criteria have also been similarly derived, such as the Bayesian information criterion (BIC) (Schwarz, 1978), which is similar to the AIC, although it applies a more severe penalty of $(k \ln n)$ to the number of model parameters. The expression for the AIC in (5) assumes a linear regression model, but can be extended to non-linear regression. However, it should be noted that in this case, $p+1$ no longer sufficiently describes the complexity of the model and other measures are required. Such measures include the *effective number of parameters*, or Vapnik-Chernovenkis dimension. The values of these measures are a function of the class of regression model that is estimated and the training data. The effective number of parameters, d can be determined by trace(S), where S is a matrix defined by the expression

$$\hat{y} = Sy. \tag{6}$$

For kernel regression, the hat matrix, S, is equal to $K^T K$, where the elements of K correspond to each $K_j(x,h)$, and the complexity is therefore given by trace$(K^T K)$. Factors affecting

complexity include the number of data, the dimension of the data, and the number of basis functions. The VC-dimension is similarly defined as the number of data points that can be *shattered* by the model (i.e. how many points in space can be uniquely separated by the regression function). However, calculating the VC-dimension of complex regression functions can be difficult (Hastie et al., 2001). For MLP architectures, the VC-dimension is related to the number of connection weights, and for RBF networks the VC-dimension depends on the number of basis functions and their respective bandwidths, if different value are used for each basis function. Both the effective number of parameters and the VC-dimension revert to the value of $p + 1$ for linear models.

In filter algorithm designs, the optimality criterion is embedded in the statistical analysis of candidate variables, which defines the interpretation of "good" input variables. In general, selection filters search amongst the candidate variables and identify suitable input variables according to the following criteria:

1. maximum relevance (MR),
2. minimum redundancy (mR), and
3. minimum redundancy–maximum Relevance (mRMR).

The criterion of maximum relevance ensures that the selected input variables are highly informative by searching for variables that have a high degree of correlation with the output variable. Input ranking schemes are a prime example of MR techniques, in which the relevance is determined for each input variable with the output variable. Greedy selection can be applied to select the k most relevant variables, or a threshold value can be applied to select inputs that are relevant, and reject those which are not.

The issue with MR criteria is that the selection of the k most relevant candidate variables does not strictly yield an optimal ANN. Here, Kohavi & John (1997) make the distinction between relevance and usefulness by observing that redundancy between variables can render highly relevant variables useless as predictors. Consequently, a criterion of minimum redundancy aims to find inputs that are maximally dissimilar from one another, in order to select the most useful set of relevant variables. The application of an additional mR criterion with the existing MR criterion leads to mRMR selection criteria, where input variables are evaluated with the dual consideration of relevance, with respect to the output variable; and independence (dissimilarity), with respect to the other candidate variables (Ding & Peng, 2005).

Embedded IVS considers regularisation (reducing the size or number) of the weights of a regression to minimise the complexity, while maintaining predictive performance. This involves the formulation of a training algorithm that simultaneously finds the minimum model error and model complexity, somewhat analogous to finding the optimum the AIC. If the model architecture is linear-in-the-parameters, the resulting expression can be solved directly. Depending on the model complexity term, this approach gives rise to various embedded selection algorithms, such as the Lasso (Tibrishani, 1996). However, the non-linear and non-parametric nature of ANN regression does not lend itself to this approach (Guyon & Elisseeff, 2003; Tikka, 2008). Instead, embedded selection is typically applied in ANN model development in the form of a *pruning* strategy, where the connection weights of a network are assessed, and insignificant weights are removed from the network. Pruning algorithms were originally developed to address the computational burden associated with training fully connected networks, given that many of the weights may be only marginally important due to redundancy within the ANN architecture. However, the strategy also offers the means of selectively removing inputs, since an input variable is eliminated by eliminating all connection

weights between an input and the first hidden layer (Tikka, 2008). A criterion is required to identify which connection weights should be pruned, and several different approaches can be used to determine how weights are removed (Guyon & Elisseeff, 2003):

1. Analysis of sensitivity of training error to elimination of weights, or
2. Elimination of variables based on weight magnitude.

Where the first approach has been used, different expressions for the sensitivity of the error to the weights have led to various different algorithms. The use of derivatives of error functions or transfer functions at hidden nodes with respect to the weights are common strategies, and lead to examples of pruning algorithms such as the optimal brain damage (OBD) algorithm (Le Cun et al., 1990).

3.2 Search strategies

Search strategies applied to IVS algorithms seek to provide an efficient method for searching through the many possible combinations of input variables and determining an optimal, or near optimal set, while working within computational constraints. Searches may be global, and consider many combinations; or local methods, which begin at a start location and move through the search space incrementally. The latter are also commonly referred to as *nested subset* techniques, since the region they explore comprises overlapping (i.e. nested) sets by incrementally adding variables.

3.2.1 Exhaustive search

Exhaustive search simply evaluates all of the possible combinations of input variables and selects the best set according to a predetermined optimality criteria. The method is the only selection technique that is guaranteed to determine the optimal set of input variables for a given ANN model (Bonnlander & Weigend, 1994). Given the combinatorial nature of the IVS problem, the number of possible subsets that form the search space is equal to 2^d, with subsets ranging in size from single input variables, to the set of all available input variables. Exhaustive evaluation of all of these possible combinations may be feasible when the dimensionality of the candidate set is low, but quickly becomes infeasible as dimensionality increases.

3.2.2 Forward selection

Forward selection is a linear incremental search strategy that selects individual candidate variables one at a time. In the case of wrappers, the method starts by training d single-variable ANN models and selecting the input variable that maximises the model performance-based optimality criterion. Selection then continues by iteratively training $d - 1$ bivariate ANN models, in each case adding a remaining candidate to the previously selected input variable. Selection is terminated when the addition of another input variable fails to improve the performance of the ANN model. In filter designs, the single most relevant candidate variable is selected first, and then forward selection proceeds by iteratively identifying the next most relevant candidate and evaluating whether the variable should be selected, until the optimality criterion is satisfied.

The approach is computationally efficient overall, and tends to result in the selection of relatively small input variable sets, since it considers the smallest possible models, and trials increasingly larger input variable sets until the optimal set is reached. However, because forward selection does not consider all of the possible combinations, and only searches a

small subset, it is possible that the algorithm may encounter a locally optimum set of input variables and terminate prematurely. Also, due to the incremental nature of the forward search, the algorithm may ignore highly informative combinations of input variables that are only marginally relevant individually (Guyon & Elisseeff, 2003).

3.2.3 Step-wise selection

Forward selection is said to have *fidelity*, in that once an input variable is selected, the selection can not be undone. Step-wise selection is an extension of the forward selection approach, where input variables may also be removed at any subsequent iteration. The formulation of the step-wise approach is aimed at handling redundancy between candidate variables. For example, a variable X_a may be selected initially due to high relevance, but is later found to be inferior to the combination of two other variables, X_b and X_c, which only arises at a subsequent iteration. The initially selected input variable X_a is now redundant, and can be removed in favour of the pair X_b and X_c.

A common example of this approach is step-wise regression, which is widely used for the development of linear regression models. In this wrapper approach, linear models are iteratively constructed by adding an input variable to the model, and re-estimating the model coefficients. Input variables are retained based on analysis of the coefficients of the newly developed model. The selection process continues until the model satisfies some optimality criterion, such as the AIC (see Section 3.1), that is, when $k + 1$ input variables are no better than the preceding k variables.

3.2.4 Backward elimination

Backward elimination is essentially the reverse of the forward selection approach. In this case, all d input variables are initially selected, and then the most unimportant variables are eliminated one-by-one. In wrapper selection strategies, the relative importance of an input variable may be determined by removing an input variable X_i and evaluating the effect on the model that is retrained without it; or, by examining the influence of each of the input variables on the output y through some sensitivity analysis. In filter strategies, the least relevant candidates are iteratively removed until the optimality criterion is satisfied.

In general, backward elimination is inefficient in comparison with forward selection, as it can require the development and evaluation of many large ANN models before reaching the optimal model. Since all input variables are initially included, it may be more difficult to determine the relative importance of an individual input variable than in forward selection, which starts with a single input variable. Also, wrapper algorithms based on backward elimination may potentially be biased by overfitting of large models.

3.2.5 Heuristic search

Heuristic search techniques are widely used in optimisation problems where the search space is large. Heuristic search algorithms are particularly adept at efficiently finding global, or near-global optimum solutions within large search spaces by exploiting the common attributes of good solutions. In general, the various algorithms each implement a search that combines random evaluation of solutions throughout the entire search space, with a mechanism to increase the focus of the search in regions that lead to good solutions. Examples of heuristic search algorithms applied to IVS include evolutionary algorithms (EAs), such as genetic algorithms (GAs) (Bowden, 2003) and ant colony optimization (ACO) (Izrailev & Agrafiotis, 2002; Marcoulides & Drezner, 2003; Shen et al., 2005).

The application of heuristic optimisation techniques to IVS overcomes the significant computational requirement of exhaustive search, while maintaining the desirable characteristic of providing a global (or, near-global) optimum. Moreover, EA-based IVS wrappers are an attractive option because they can also be included as part of evolutionary ANN training algorithms, which also seek to determine optimal ANN parameter values by minimising the ANN cross-validation error. However, the application of heuristic search techniques requires calibration of search algorithm parameters, which is itself not a trivial task. In general, setting the search parameters involves a trade-off between the amount the search space that is explored, and the rate at which the algorithm converges to a final solution. Finally, heuristic algorithms retain a certain degree of randomness, and although they search more solutions in comparison to sequential selection algorithms, there is still no guarantee that the sub-space explored will include the globally optimal solution.

4. Dimensionality reduction

The taxonomical classification in Figure 1 includes dimensionality reduction algorithms as a class of algorithms reducing the number of variables within a dataset. Dimensionality reduction is often performed in order to reduce the computational effort associated with data processing, or to identify a suitable subset of variables to include in the analysis. Although the goal of dimension reduction differs from that of variable selection, it is a closely related area and is a regularly employed data pre-processing step in many multivariate data analysis applications, and it is worth including some discussion since many dimensionality reduction techniques are employed for IVS. The following considers some common examples of dimension reduction algorithms. Comprehensive surveys of dimensionality reduction techniques can be found in Carreira-Perpinan (1997) and Fodor (2002).

4.1 Principal component analysis

Principal component analysis (PCA) is a commonly adopted technique for reducing the dimensionality of a dataset X. PCA achieves dimensionality reduction by expressing the p variables $(x_1, \ldots, x_p)$ as d feature vectors (or, *principal components* (PCs)), where $d < p$. The PCs are a set of orthogonal, linear combinations of the original variables within the dataset. Essentially, PCA can be considered a data pre-processing algorithm that determines an optimal rotational transformation of the dataset, X, that maximises the amount of variance of the output Y that is explained by the PCs (Fodor, 2002).
Considering a given dataset X, PCA is performed as follows:

i. Subtract the mean value of each variable, to ensure that $\bar{x}_i = 0$ for each $x_i \in X$.
ii. Find the covariance matrix $\Sigma = \mathrm{Cov}(X) = X^T X$.
iii. Determine the unit eigenvectors $e_1, \ldots, e_p$ of Σ.
iv. Determine the corresponding eigenvalues $\lambda_1, \ldots, \lambda_p$.
v. Rank the eigenvectors according to their eigenvalues.
vi. Select the d PCs according to their eigenvalues.

Selection of PCs is based on examining the eigenvalues of each PC, which correspond to the amount of variance explained by each PC, and thereby including only the significant PCs as input features. A common selection method is to rank the PCs and select all PCs whose eigenvalues exceed some threshold λ_0, or generate a plot of the cumulative eigenvalue as a function of the number of PCs, k, to ensure the selected components explain the desired

amount of variance of Y. Another technique is to use and generate a *scree* plot of the percentage contribution of each k^th PC and to visually identify an optimal value of k (Fodor, 2002).

PCA has been used as the basis for IVS for the development of ANN models (see, for example, Olsson et al. (2004), Gibbs et al. (2006), and Bowden (2003)). However, the mixing of input variables is assumed to be linear, as is the relationship between principal components and the output. Consequently, the application of PCA in this case is flawed, since it will fail to identify any non-linear relationships within the data. Although non-linear versions of the PCA algorithm exist, the transformations of the data can be highly complex, and interpretation of the PCs is much more difficult. An additional disadvantage of PCA is that the algorithm identifies important component vectors, rather than variables. Consequently, although PCA may be useful in removing noise from the data, it is not possible to distinguish the unique contributions of individual variables to the variance in the output.

4.2 Independent component analysis

Independent component analysis (ICA) seeks to determine a set of d independent component vectors within a dataset X. The approach is conceptually similar to PCA, although it relaxes the orthogonality constraint on component vectors. Furthermore, where PCA determines the optimal transformation of the data by considering covariance and identifying uncorrelated PCs based on covariance, ICA considers statistically independent combinations of variables where the order of the statistic that is used can be arbitrary (Fodor, 2002). ICA is therefore not restricted to linear correlations, and is more widely applicable to non-linear datasets (Back & Trappenberg, 2001). However, like PCA, ICA cannot discriminate unique variables as predictors, and is restricted to determining independent feature vectors.

4.3 Vector quantization

Vector quantization (VQ) refers to techniques that describe a larger set of n vectors by c codebook, or prototype vectors. VQ is closely associated with data clustering and is more commonly associated with algorithms for data compression, in terms of length n. Bowden (2003) demonstrates the potential for the self-organising map (SOM) to perform vector quantisation as an alternative to PCA for data dimensionality reduction. In this case, the d vectors of the candidate set are represented by the prototype vectors of the SOM. Similar candidate variables will be identified by the formation of groups, which have the closest proximity (defined by some distance measure) to the same prototype vector. However, the distance metric needs to be carefully selected, since the Euclidean distance used in SOM applications is not strictly a measure of correlation. Using linear correlation or covariance will cluster based on the strength of linear dependence between variables. Other measures, such as entropy or mutual information (MI), would be more suitable for clustering ANN variables, since they will measure non-linear dependence.

5. Wrappers

Wrapper algorithms are the first of the three main classes of variable selection algorithm shown according to Figure 1. Wrapper algorithms are the simplest IVS algorithm to formulate. Essentially, the algorithm that results is defined by the choice of the induction algorithm (i.e. model architecture). The efficiency of a wrapper algorithm will depend on the ability of the model to represent relationships within the data; and how efficiently trial models can be constructed and evaluated.

5.1 Single variable regression (SVR)

The notion of ranking individual candidate variables according to correlation can be extended by implementing a wrapper approach in order to relax the assumption of linearity in correlation analysis (Guyon & Elisseeff, 2003). In this approach, a single variable regression[1] (SVR) is constructed using each candidate variable, which is then ranked according to the model performance-based optimality criterion, such as the cross-validation error. In comparison to ranking filters, SVR can potentially suffer from overfitting due to the additional flexibility in the regression model.

The GRNN input determination algorithm (GRIDA) (Bowden et al., 2006) is a recent example of an SVR wrapper for input variable ranking, which proceeds as follows:

i. Let $X \rightarrow C$. (Initialisation)
ii. For each $x \in X$,
iii. Train a GRNN and determine MSE_x.
iv. For $b = 1$ to 100, (Bootstrap)
v. Randomly shuffle $x \rightarrow \varepsilon$.
vi. Estimate $\text{MSE}_{\varepsilon,b}$.
vii. Estimate $\text{MSE}_{\varepsilon}^{(95)}$.
viii. If $\text{MSE}_x > \text{MSE}_{\varepsilon}^{(95)}$ or $\text{MSE}_x > \Theta$ (Selection),
ix. Remove x from X.
x. Return X.

where $\text{MSE}_{\varepsilon}^{(95)}$ is the 95^{th} percentile, and Θ is some threshold value.

Considering each variable in turn, a GRNN is trained, and then the MSE of the model is determined for a set of test data. However, rather than greedy selection of the k best variables, each variable is compared to a bootstrap estimate of a confidence bound for the randomised model error, $\text{MSE}_{\varepsilon}^{(95)}$. A variable is rejected immediately if the model error exceeds the randomised error, since it is no better predictor than a random noise variable. Further strictness on selections is imposed through the heuristic error threshold, Θ. However, a suitable value for Θ needs to be determined first. The number of variables selected for a given value of Θ will be dependent on several factors, including the degree of noise in the data, the error function used, and the distribution of the error over the candidate variables. Consequently, optimal values for Θ can only be determined for each dataset by trial and error. The estimation of the confidence bound on the error for each SVR is a significant computational requirement. Given the assumed constraint $0 < \Theta < \text{MSE}_{\varepsilon}^{(95)}$, the estimation of the bootstrap may not even be necessary to perform IVS. However, the method does provide useful information in discriminating noise variables from weakly informative ones.

SVR does not consider interractions between variables, and may select redundant variables. In order to overcome this, dimensionality reduction is required as a pre-processing step, in order to obtain an independent set of candidate variables, prior to variable selection. Such an approach was used in the example of the GRNN-based SVR IVS algorithm (Bowden et al., 2006), where a SOM was used to achieve dimension reduction.

[1] The term has been adapted from the term single variable classifier (SVC), which is more often referred to within literature due to its application in classification

5.2 GA-ANN

Heuristic search techniques are ideally suited to IVS wrapper designs, since they provide an efficient search of the combinations of candidate input variables, with the simplicity of the black-box wrapper approach. Bowden et al. (2005) utilised an evolutionary wrapper strategy for IVS that combined a genetic algorithm (GA) optimisation with a generalised regression neural network (GRNN). The method exploits the fast GRNN training times, and the fixed architecture of the GRNN, which avoids the need to optimise the internal architecture and training algorithm parameters. These are required for the development of other architectures, such as the MLP. A simple binary GA (a GA with decisions encoded as 1 or 0 within a binary string) was utilised, with the objective of minimising the MSE obtained by hold-out validation on a set of test data. In order to overcome the inability of the wrapper methodology to detect interractions between candidate variables, as with GRIDA, Bowden et al. (2005) adopted SOM-based dimensionality reduction as a pre-processing stage to reduce the candidate variables to a subset of independent variables.

6. Embedded algorithms

Embedded algorithms are a distinct class of IVS algorithm where the selection of variables is *embedded* within the ANN training algorithm. The distinction between wrapper and embedded algorithms is not always obvious, since embedded selection also involves iterative update and evaluation of the model parameters based on model performance. The key difference is that the evaluation of input variables occurs within the training algorithm, and only a single model is trained. Embedded algorithms also consider the impact of each individual variable on the performance of the model, while wrapper algorithms simply consider model performance for a given set of input variables as a whole.

6.1 Recursive feature elimination

Recursive feature elimination (RFE) is an implementation of backward-elimination as an embedded IVS algorithm (Guyon & Elisseeff, 2003). The RFE approach involves an iterative process of training an ANN model, initially using all candidate input variables, and then removing one or more input variables at each iteration based on the rank magnitude of the weights corresponding to each input variable. The technique has been developed and successfully applied using support vector machines (SVMs), but is extensible to other ANN architectures by providing a suitable expression for the overall connection weight for a given input variable.

6.2 Evolutionary ANNs

Evolutionary ANNs (EANNs) utilise an evolutionary algorithm to determine the optimal set of weights, where the optimisation is formulated with ANN weights as the decision variables, and an objective function incorporating model error, with penalties for model complexity. Due to the complexity of the ANN error surface that is projected over the weight-space, evolutionary algorithms have been found to be a good alternative to gradient descent algorithms. EAs are robust and able to more reliably find a near-globally optimum solution, even for highly non-linear functions, with many multiple local optima; whereas gradient descent algorithms have greater potential to converge prematurely at a local minimum. By applying penalty terms to avoid large weights, or by using a model sparsity term within the objective function, the optimisation can effectively determine the optimum trade-off between model error and model complexity during training (Tikka, 2008).

IVS is embedded within the EANN approach, since the optimisation will implicitly exclude input variables by setting the input connection weight values close to, or equal to zero. Alternatively, the approach may also be made more explicit by extending the optimisation formulation to include the inclusion/exclusion of input variables within the ANN architecture as a separate binary decision variable. This approach is therefore similar to the GA-ANN wrapper approach, but extends the optimisation to the weight parameters.

7. Filters

In the taxonomy shown in Figure 1, filter algorithms represent the second sub-class of variable selection algorithms and represent an alternative to the wrapper approach. The design of filter algorithms is typically defined by the measure of relevance that is used to distinguish the important input variables, as well as the optimality criteria, as they have been previously defined for filters in Section 3.1. Incremental search strategies tend to dominate filter designs, since the relevance measure is usually a bivariate statistic, which necessitates evaluating each candidate-output relationship. Currently, two broad classes of filters have been considered: those based on linear correlation; and those based on information theoretic measures, such as mutual information.

7.1 Rank correlation

Arguably the most commonly used relevance measure in multivariate statistics is the Pearson correlation. The Pearson correlation (also called *linear correlation*, or *cross-correlation*), R, is defined by

$$R_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{7}$$

where R_{XY} is the short-hand notation for $R(X,Y)$. In (7), the numerator is simply the sample covariance, $Var(X,Y)$; and the two terms in the denominator are the square-root of the sample variances, $Var(X)$ and $Var(Y)$. The application of correlation analysis to variable selection originates from linear regression analysis. The squared correlation, R^2_{XY}, is the coefficient of determination, and if X and Y have been standardised to have a zero mean, R^2 is the equivalent to the coefficient of a linear fit between X and Y.

Input variable ranking based on the Pearson correlation is one of the most widely used IVS methods. The selection of candidate variables that are sorted by order of decreasing correlation is based either on greedy selection of the first k variables, or upon all variables for which the correlation is significantly different from zero. The significance of the Pearson correlation can be determined directly, since the error associated with estimation of correlation from a sample is defined by the t-distribution. A rule of thumb (for large n) is that variables with an absolute correlation greater than $2/\sqrt{n}$ are significant.

Identification of significant correlations is a common technique in data mining applications, such as gene expression analysis, where the goal is simply to mark potentially important genes for further investigation. However, in terms of IVS algorithms, the method is classed as an MR filter, and does not consider interactions between variables. Redundancy is particularly problematic for multivariate time-series forecasting, which considers lagged values that are often highly correlated (auto-correlated).

7.2 Partial correlation

In the case where candidate variables are themselves correlated, redundancy becomes an important issue. In such cases, a correlation ranking approach is likely to select too many variables, since many candidates will each provide the same information regarding the output variable. Given three variables X, Y and Z, the partial correlation, $R'(X, Y|Z)$ measures the correlation between X and Y after the relationship betwen Y and Z has been discounted. The partial correlation can be determined from the Pearson correlation using the equation:

$$R_{XY \cdot Z} \frac{R_{XY} - R_{XZ}R_{YZ}}{\sqrt{(1 - R_{XZ}^2)(1 - R_{YZ}^2)}} \tag{8}$$

where $R_{XY \cdot Z}$ and R_{XY} etc. are the short-hand notation for $R'(X, Y|Z)$ and R_{XY} etc.
Partial correlation is similar to stepwise multiple linear regression. The subtle difference is that in stepwise MLR, successive models are fitted with additional input variables, and variables are selected (or later rejected) based on the estimated model coefficients. However, in partial correlation analysis, the magnitude of R' for each variable is not necessarily equal to the regression coefficients for a fitted MLR model, since redundancy between variables means that the solution to the MLR parameter estimation is a line (two redundant coefficients) or a surface, that is, there will be infinite combinations of equivalent model coefficients. The partial correlations obtained are in fact one specific solution to the MLR parameter estimation. Another difference is that forward selection is used in partial correlation analysis, because once the most salient variable has been selected, it will not be rejected later, and the partial correlations of subsequent variables will be dependent on those already selected.

7.3 Box-Jenkins

Box-Jenkins time-series analysis (Box & Jenkins, 1976), which considers the development of linear auto-regressive, moving-average (ARMA) models to represent dynamic processes, is the most common approach to the development of time-series and process transfer functions. ARMA models are described by the general form

$$y(t+1) = \sum_{k=0}^{p} \alpha_k y(t-k) + \sum_{k=0}^{q} \beta_k u(t-k) \tag{9}$$

where α_k and β_k are coefficients and p and q denote the order of the autoregressive (AR) and moving-average (MA) components of the model, respectively. Identification of the optimal model parameters p and q forms the goal of Box-Jenkins model identification, and hence variable selection. The autocorrelation function (ACF), $R\left(Y(t-k), Y(t)\right)$, determines q and the partial autocorrelation function (PACF) determines p. The ACF is determined for a given time-series sample by

$$R_k = \frac{\sum_{i=1}^{n-k} (x_i - \bar{x})(x_{i-k} - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{10}$$

where R_k is the short-hand notation for the auto-correlation of a time-series with a delay of k. The PACF at a delay of k is denoted by ϕ_{kk}, and is estimated from the ACF based on the following series of equations

$$\phi_{11} = R_1 \tag{11}$$

$$\phi_{22} = \frac{R_2 - R_1^2}{1 - R_1^2} \tag{12}$$

$$\phi_{kj} = \phi_{k-1,j} - \phi_{kk}\phi_{k-1,k-j}, \text{ for } k \geq 2 \text{ and } j \geq 1, \tag{13}$$

$$\phi_{kk} = \frac{R_k - \sum_{j=1}^{k-1} \phi_{k-1,j} R_{k-j}}{1 - \sum_{j=1}^{k-1} \phi_{k-1,j} R_j}, \text{ for } k \geq 3. \tag{14}$$

The Box-Jenkins methodology can be used to similarly identify optimal linear autoregressive with exogenous inputs (ARX) models. In this case, the partial cross-correlation is used to identify the relevant lags of the exogenous variables.
Box-Jenkins and partial autocorrelation analysis have been used as the basis for IVS in the development of ANN models. In some examples, ANNs have been developed based on an optimal set determined for an ARX model. The ANNs were found to produce better predictions than the ARX model, and this has often provided the justification for ANN modelling in favour of conventional time-series techniques (Rodriguez et al., 1997). However, although this demonstrated the additional flexibility of ANN architectures to describe more complex behaviour, the ANN developed may not have been optimal, since the selection of inputs was based on the identification of a linear model. It may be the case that variables that are highly informative, but non-linearly correlated with the output variable, will be overlooked and excluded from the ANN model.

7.4 Mutual information

The limitations of linear correlation analysis have created interest in alternative statistical measures of dependence, which are more adept at identifying and quantifying dependence that may be chaotic or non-linear; and which may therefore be more suitable for the development of ANN models. Mutual information (MI) is a measure of dependence that is based on information theory and the notion of entropy Shannon (1948), and is determined by the equation

$$I(X;Y) = \iint p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \, dxdy, \tag{15}$$

where I denotes the MI between X and Y. MI measures the quantity of information about a variable Y that is provided by a second variable X. However, it is often convenient to simply regard MI as a more general measure of correlation, since despite originating from information theory, rather than statistics, MI is not entirely unrelated to Pearson correlation. In fact, it can be shown that in the case of noise-free, Gaussian data, MI will be related to linear correlation according the relationship:

$$I(X;Y) = \frac{1}{2} \log \left(1 - R_{XY}^2\right). \tag{16}$$

The advantage of MI over linear correlation is that MI is based solely on probability distributions within the data and is therefore an arbitrary measure, which makes no assumption regarding the structure of the dependence between variables. It has also been found to be robust due to its insensitivity to noise and data transformations (Battiti, 1994;

Darbellay, 1999; Soofi & Retzer, 2003). Consequently, MI has recently been found to be a more suitable measure of dependence for IVS during ANN development. Torkkola (2003) also discusses the merit of analysing MI, given that it provides an approximation to the Bayes error rate. Bayes' theorem, which gives the most general form of statistical inference, is given by

$$p(y|x \in X) = \frac{p(y \in Y)p(x \in X|Y)}{p(x \in X)} \tag{17}$$

Bayes' theorem can be used to determine the expectation $E(y|x \in X)$, assuming the probability distributions are known. MI provides an approximation to the error associated with the Bayes estimate of $E(y|X \in X)$, since it can be shown that the minimum error will be achieved for a maximal value of $I(X;Y)$. In effect, MI provides a generic estimation of the *modellability* of an output variable Y, which therefore makes MI an attractive measure of relevance in determining an optimal set of input variables, since we would seek the set of input variables that maximises the JMI, that is, the MI between the output and the input variable set.

The MIFS algorithm is a forward selection filter proposed by Battiti (1994) to address shortcomings with algorithms based on linear correlation. Considering the candidate set C and output variable Y, the MIFS algorithm proceeds as follows:

i. Let $X \rightarrow \phi$.
ii. While $|X| < k$,
iii. For each $c \in C$,
iv. Estimate $I(c, Y|X) = I(c, Y) - \beta \sum_{x \in X} I(c; x)$.
v. Find c_s that maximises $I(c, Y|X)$.
vi. Move c_s to X.
vii. Return X.

MIFS defines a MR filter and identifies suitable candidates according to the estimated bivariate MI between candidate variables and the output variable. The MI between the most salient candidate c_s and the already selected variables in X is estimated and subtracted from the relevance in order to achieve minimum redundancy. The heuristic weighting β determines the degree of redundancy checking within MIFS. If $\beta = 0$, then MIFS will neglect relationships between candidates and MIFS is reduced to a MI ranking filter. Increasing β increases the influence of candidate redundancy on selections, however if β is too large, then the redundancy is overstated and candidate interactions dominate the selection of variables, rather than the input-output relationships (Kwak & Choi, 2002). Battiti (1994) recommends that a weighting of 0.5–1.0 is appropriate. A criticism of the forward selection approach is that the JMI of the input variable set must be considered in order to correctly determine the optimality of the input variables (Bonnlander & Weigend, 1994). However, in MIFS, the forward selection procedure considers variables individually, and optimality of the JMI is inferred by the mRMR selection. The heuristic redundancy parameter β provides only an approximation to the conditional dependence and does not necessarily relate to the JMI.

7.5 Partial mutual information

Sharma (2000) proposed an IVS filter that is structured similarly to MIFS, but is based instead upon direct estimation of partial mutual information (PMI). The algorithm has been successfully applied to select predictors for hydrological models (Sharma, 2000) and ANN water quality forecasting models (Bowden et al., 2002; Kingston, 2006). The PMI-based filter also incorporates a mechanism for testing the significance of candidate variables, so that the termination point of the algorithm is optimally determined, which is an improvement over the greedy selection of k variables in MIFS. In this case, the termination criterion is based upon the distribution of the error in PMI estimation, which is numerically approximated by a bootstrap approach (Sharma, 2000). The significance of the most relevant candidate is determined by direct comparison to the upper confidence bound on the estimation error.
The details of the algorithm, are as follows (May et al., 2009a):

Let $\mathbf{X} \rightarrow \phi$ (Initialisation)
  While $C \neq \phi$ (Forward selection)
  Construct kernel regression estimator $\hat{m}_Y(\mathbf{X})$
  Calculate residual output $u = Y - \hat{m}_Y(\mathbf{X})$
  For each $c \in C$
    Construct kernel regression estimator $\hat{m}_c(\mathbf{X})$
    Calculate residual candidate $v = c - \hat{m}_c(\mathbf{X})$
    Estimate $I(v;u)$
  Find candidate c_s (and v_s) that maximises $I(v;u)$
   For b = 1 to B (Bootstrap)
     Randomly shuffle v_s to obtain v_s^*
     Estimate $I_b = I(v_s^*;u)$
  Find confidence bound $I_b^{(95)}$
  If $I(v_s,u) > I_b^{(95)}$ (Selection/termination)
    Move c_s to $\mathbf{X}$
  Else
    Break
Return selected input set $\mathbf{X}$.

Here, B is the bootstrap size; and $I_b^{(95)}$ denotes the 95$^{\text{th}}$ percentile bootstrap estimate of the randomised PMI, I_b. The algorithm is structured in a similar fashion to MIFS (Battiti, 1994), but has two advantages. First, PMIS inherently handles redundancy within the candidate set through the direct estimation of PMI, whereas MIFS approximates the effect of selected inputs by means of a heuristic weighting factor. Second, while MIFS uses greedy selection of a pre-specified number of input variables, PMIS includes a criterion that automatically determines the optimum point at which to terminate the selection procedure. The optimality of the input variable set is ensured because PMI is directly estimated, and the JMI can be

determined as a result of the MI chain-rule decomposition, which is given as (Cover & Thomas, 1991)

$$I(x_1, \ldots, x_p; y) = I(x_1; y) + I(x_2; y|x_1) + \cdots + I(x_p; y|x_1, \ldots, x_{p-1}). \tag{18}$$

Recall that in MIFS, the JMI cannot be directly approximated because redundancy is only approximated by a heuristic weighting factor. The termination criterion in PMIS automatically determines the optimal number of input variables, since the increase in JMI is additive, and once the contribution of an additional input variable is insignificant, the selection process terminates and the JMI will be maximised.

An additional benefit of the PMI-based approach is that the information yield during IVS provides a useful indication of the contribution of each input variable to the prediction of the output variable. Several methods for determining the usefulness of input variables based on analysis of the trained model have been described and range from sensitivity analysis, to aggregation of the weights associated with each input variable. However, the relative importance of an input variable can be determined statistically from the MI between each input and the output variable (Soofi & Retzer, 2003). The PMI estimated for a given variable can potentially also be used to classify input variables as informative, or weakly informative, as defined by Kohavi & John (1997), by considering the conditional relevance. Kingston (2006) considered several techniques for determining the relative importance (RI) of input variables and found that the method based on PMI yielded similar estimates of RI as methods based on analysis of the connection weights for a trained MLP. The method for estimating RI was based on the formula

$$\mathrm{RI}(i) = \frac{I'(x_i; y)}{\sum_{x \in X} I'(x; y)}, \tag{19}$$

where I' denotes the PMI estimated for candidate variable x during PMIS.

The usefulness of RI is that it provides an indication of the way in which the ANN generates predictions. Although it is assumed that the ANN is using all of the input variables, it may in fact only require some small subset of the available input variables to generate predictions. In this case, further refinements to the ANN can be made based on this interpretation. Such considerations might be important when considering the cost of data collection that is associated with ongoing deployment of ANN models. One might consider sacrificing model accuracy in favour of cost reductions in ANN maintenance by reducing the number of input variables even further, which would reduce data requirements. The RI of variables may also encourage increased efforts toward the development of measurement techniques to ensure data quality for important variables.

The main limitation of the PMI filter is the computational effort associated with the bootstrap estimation of MI. Even obtaining a single estimate of MI is an expensive computation due to the $O\{n^2\}$ density estimation, and computational efficiency is therefore influenced by the sample size, n. The use of a bootstrap-based test for significance further adds to the overall computational burden by increasing the number of estimations of MI required to implement IVS. Sharma (2000) restricts the size of the bootstrap to 100 in order to maintain reasonable analysis times. However, a small bootstrap of this size might compromise the accuracy of the termination criterion, since potentially the confidence bounds may be poorly estimated. May et al. (2009b) introduce several alternative termination criteria that eliminate the bootstrap significance-test from the algorithm. These were found to provide a significant improvement in both the accuracy and computational effort associated with the PMI-based algorithm.

Fernando et al. (2009) also present a fast implementation of the PMI-based filter design using an average-shifted histogram (ASH) (Silverman, 1986), rather than kernel density estimation, for the computation of PMI during selection.

7.6 ICAIVS

A hybrid ICA and IVS filter algorithm (ICAIVS) was proposed by Back & Trappenberg (2001), which consists of two main steps: (Trappenberg et al., 2006)

i. Produce a set of candidates which are as independent as possible (ICA).
ii. Measure relevance between the independent candidate variables and the desired output variables (IVS).

Here, the statistical analysis of input relevance is based on estimation of the joint dependence of combinations of input variables and considers all combinations from $c(x_1^p, y)$ through to $c(x_1^p, \ldots, x_n^p, y)$, where p denotes the order of the dependence that is measured. The IVS procedure then compares the relevance for each subset of variables, with respect to the average dependence for all subsets, and a subset is selected if the dependence exceeds some threshold value K.

A drawback of ICAIVS is that the algorithm does not scale well, given the large number $(3^n - 1)$ of statistical tests that must be performed, considering only second-order statistics. Recently, an improved version of ICAIVS was described that utilised MI as the statistical measure of dependence (Trappenberg et al., 2006). This reduced the number of statistical tests by considering only first order MI. However, the problem of specifying a suitable threshold value still remains. Both Back & Trappenberg (2001) and Trappenberg et al. (2006) used a value of 0.2 for this threshold. However, in this case the threshold value is heuristically determined, and a suitable value may vary depending on the dataset.

8. Summary

Input variable selection remains an important part of ANN model development, due to the negative impact that poor selection can have on the performance of ANNs during training and deployment post-development. The brief taxonomy of variable selections indicates the different classes of algorithms that have been employed to perform the IVS task, of which many more variations within each class are described within ANN literature. A summary of the attributes of different IVS algorithms is given in Table 2.

Several key considerations should guide a modeller in determining the most appropriate approach to take for IVS in any given circumstance. The first consideration is whether the algorithm is suitable for identifying non-linear relationships, which is a fundamental requirement for the development of ANN models. Many methods applied successfully within the context of linear regression are not suited to ANN development, where there is a presumption of non-linearity, and important relationships within the data may not be identified by a linear selection algorithm.

The choice between model-free and model-based IVS algorithms is also a key consideration. The restriction imposed by wrapper designs on the choice of model architecture, and the computational requirements associated with training multiple model instances, will dictate which approach to use. Where portability between ANN architectures is not required, the use of wrapper or embedded algorithms is likely to be straightforward to implement, and are a pragmatic approach to take. However, in all cases a wrapper that includes a regularisation

Algorithm	Criterion	Type	Non-linear	Redundancy	Search
Rank-R	Correlation	Filter	No	No	Greedy
Box-Jenkins	ACF/PACF	Filter	No	Yes	Greedy
R′	Correlation	Filter	No	Yes	Stepwise
SVR	Rank error	Wrapper	Yes	No	Greedy
Forward selection	Error	Wrapper	Yes	Yes	Nested
Backward selection	Error	Wrapper	Yes	Yes	Nested
GA-ANN	Error	Wrapper	Yes	Yes	Heuristic
MIFS	MI	Filter	Yes	Yes	Nested
PMIS	PMI	Filter	Yes	Yes	Nested
RFE	Weight size	Embedded	Yes	Yes	Nested
EANN	Error	Embedded	Yes	Yes	Heuristic

Table 2. Characteristics of input variable selection algorithms

term, or considers the impact of model size, should be used to achieve an optimal trade-off between model performance and size.

High computational requirements are the main drawback of model-based techniques. The training time for a single instance of an ANN will be a strong indication of the feasibility of a model-based approach. Popular methods for wrappers and embedded algorithms are typically applied where the number of training examples is relatively small, and the number of candidate input variables is large. Alternatively, fixed ANN architectures like the GRNN provide fast-training, which increases the suitability of model-based IVS. The search strategy employed in a wrapper approach also represents a balance between the number of unique solutions considered, and the computational effort expended. Backward elimination algorithms, such as RFE, train relatively few ANN models and can be highly efficient, but will restrict the search due to nesting behaviour. Brute force search is likely to be infeasible in most cases, but evolutionary search approaches can provide a suitable compromise that allows for greater coverage of combinations of input variables. If evolutionary optimisation is to be used for ANN training, than the embedded EANN approach may be appropriate, as an extension of weight optimisation.

In contrast to model-based IVS, filter algorithms offer a fast, model-free approach to variable selection, and are particularly suited to applications where independence from a specific ANN architecture is required; or, where computational efficiency is sought. A generic estimation of input variable importance and redundancy avoids the traps of model over-fitting, or idiosyncrasies of a specific ANN architecture. The ability to identify an optimal set of input variables prior to training an ANN eliminates the computational burden associated with training and model selection, which can reduce the overall effort of ANN development.

Most filter designs are based on bivariate statistics, since estimation of statistics for multivariate data is often too inaccurate to facilitate IVS due to finite-sample error, and typically use a ranking scheme or sequential forward selection scheme. Simple ranking schemes are greedy and select more input variables than necessary, by ignoring redundancy of candidates, and are not ideally suited to multivariate ANN regression. Forward selection provides an efficient search, and greediness can be overcome provided that adequate redundancy checking is incorporated into the statistical analysis of variables. This ensures that the approach selects the most informative input set with the smallest number of variables.

Mutual information provides a generic measure of the relevance of a candidate input variable and is highly suitable for ANN development because it measures the strength of both linear and non-linear relationships. MI is also less sensitive to data transformations than correlation, which makes it more reliable, even for linear data analysis applications. The PMI-based forward selection approach provides a forward selection strategy that both minimises redundancy and maximises relevance. The approach determines the optimal set of input variables by estimating the maximum joint mutual information. Estimation of MI is more computationally intensive than estimation of correlation, due to the need to estimate density functions of the data. However, several refinements related to estimation of critical values, and the use of faster methods for density estimation can significantly reduce the computational effort of this approach.

9. References

Akaike, H. (1974). A new look at the statistical model identification, *IEEE Transactions of Automatic Control* 19: 716–723.

Back, A. D. & Trappenberg, T. P. (2001). Selecting inputs for modeling using normalized higher order statistics and independent component analysis, *IEEE Transactions on Neural Networks* 12(3): 612–617.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning, *IEEE Transactions on Neural Networks* 5(4): 537–550.

Bellman, R. (1961). *Adaptive control processes: a guided tour*, Princeton University Press, New Jersey.

Blum, A. & Langley, P. (1997). Selection of relevant features and examples in machine learning, *Artificial Intelligence* 97(1–2): 245–271.

Bonnlander, B. V. & Weigend, A. S. (1994). Selecting input variables using mutual information and nonparametric density estimation, *International Symposium on Artificial Neural Networks*, Taiwan, pp. 42–50.

Bowden, G. J. (2003). *Forecasting water resources variables using artificial neural techniques*, Ph.d, University of Adelaide.

Bowden, G. J., Dandy, G. C. & Maier, H. R. (2005). Input determination for neural network models in water resources applications. part 1 - background and methodology, *Journal of Hydrology* 301(1-4): 75–92.

Bowden, G. J., Maier, H. R. & Dandy, G. C. (2002). Optimal division of data for neural network models in water resources applications, *Water Resources Research* 38(2): 1–11.

Bowden, G. J., Nixon, J. B., Dandy, G. C., Maier, H. R. & Holmes, M. (2006). Forecasting chlorine residuals in a water distribution system using a general regression neural network, *Mathematical and Computer Modelling* 44(5-6): 469–484.

Box, G. E. P. & Jenkins, G. M. (1976). *Time Series Analysis, Forecasting and Control*, Holden-Day Inc., San Francisco.

Carreira-Perpinan, M. A. (1997). A review of dimension reduction techniques, *Technical report*, Dept. of Computer Science, University of Sheffield.

Cover, T. M. & Thomas, J. A. (1991). *Elements of information theory*, Wiley series in telecommunications, John Wiley & Sons, Inc., New York.

Craven, M. W. & Shavlik, J. W. (1998). Using neural networks for data mining, *Future Generation Computer Systems* 13(2–3): 211–229.

Darbari, A. (2000). Rule extraction from trained ANN: A survey, *Technical report*, Institute of Artificial Intelligence, Dept. of Computer Science, TU Dresden.

Darbellay, G. A. (1999). An estimator of the mutual information based on a criterion for independence, *Computational Statistics & Data Analysis* 32: 1–17.

Ding, C. & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data, *Journal of Bioinformatics and Computational Biology* 3(2): 185–205.

Fernando, T. M. K. G., Maier, H. R. & Dandy, G. C. (2009). Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach., *Journal of Hydrology* 367(3–4): 165–176.

Fodor, I. K. (2002). A survey of dimension reduction techniques, *Technical report*, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory.

Gibbs, M. S., Morgan, N., Maier, H. R., Dandy, G. C., Nixon, J. B. & Holmes, M. (2006). Investigation into the relationship between chlorine decay and water distribution parameters using data driven methods, *Mathematical and Computer Modelling* 44(5-6): 485–498.

Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection, *The Journal of Machine Learning Reearch* 3: 1157–1182.

Hastie, T., Tibshirani, R. & Friedman, J. (2001). *The Elements of Statistical Learning. Data Mining, Inference and Prediction*, Springer Series in Statistics, Springer, New York.

Izrailev, S. & Agrafiotis, D. K. (2002). Variable selection for QSAR by artificial ant colony systems, *SAR and QSAR in Environmental Research* 13(3–4): 417–423.

Kingston, G. B. (2006). *Bayesian Artificial Neural Networks in Water Resources Engineering*, Ph.d., The University of Adelaide.

Kohavi, R. & John, G. (1997). Wrappers for feature selection, *Artificial Intelligence* 97(1–2): 273–324.

Kwak, N. & Choi, C.-H. (2002). Input feature selection for classification problems, *IEEE Transactions on Neural Networks* 13(1): 143–159.

Le Cun, Y., Denker, J. S. & Solla, S. A. (1990). Optimal brain damage, *Advances in Neural Information Processing Systems*, Morgan Kaufmann, pp. 598–605.

Maier, H. R. & Dandy, G. C. (2000). Application of neural networks to forecasting of surface water quality variables: Issues, applications and challenges, *Environmental Modelling & Software* 15(1): 101–124.

Mallows, C. L. (1973). Some comments on Cp, *Technometrics* 15: 661–675.

Marcoulides, G. A. & Drezner, Z. (2003). Model specification searches using ant colony optimization algorithms, *Structural Equation Modeling: A Multidisciplinary Journal* 10(1): 154–164.

May, R. J., Maier, H. R. & Dandy, G. C. (2009a). Data splitting for artificial neural networks using SOM-based stratified sampling, *Neural Networks* 23: 283–294.

May, R. J., Maier, H. R. & Dandy, G. C. (2009b). Development of artificial neural networks for water quality modelling and analysis, *in* G. Hanrahan (ed.), *Modelling of Pollutants in Complex Environmental Systems*, Vol. 1, ILM Publications, London, UK, pp. 27–62.

Miller, A. J. (1984). Selection of subsets of regression variables, *Journal of The Royal Statistical Society. Series A.* 147(3): 389–425.

Olsson, J., Uvo, C. B., Jinno, K., Kawamura, A., Nishiyama, K., Koreeda, N., Nakashima, T. & Morita, O. (2004). Neural networks for forecasting rainfall by atmospheric downscaling, *Journal of Hydraulic Engineering, ASCE* 9(1): 1–12.

Rodriguez, M. J., Serodes, J.-B. & Cote, P. (1997). Advanced chlorination control in drinking water systems using artificial neural networks, *Water Supply* 15(2): 159–168.

Sarle, W. (1997). Neural network FAQ. Periodic posting to the Usenet newsgroup comp.ai.neural-nets.

Schwarz, G. (1978). Estimating the dimension of a model, *Annals of Statistics* 6(2): 461–464.

Scott, D. W. (1992). *Multivariate density estimation: theory, practice and visualisation*, John Wiley and Sons, New York.

Shannon, C. E. (1948). A mathematical theory of communication, *Bell System Technical Journal* 27: 379–423.

Sharma, A. (2000). Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 - a strategy for system predictor identification, *Journal of Hydrology* 239: 232–239.

Shen, Q., Jiang, J.-H., Tao, J.-C., Shen, G.-L. & Yu, R.-Q. (2005). Modified ant colony optimization algorithm for variable selection in QSAR modeling : QSAR studies of cyclooxygenase inhibitors, *Journal of Chemical Information and Modeling* 45(4): 1024–1029.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*, Chapman and Hall, London.

Soofi, E. S. & Retzer, J. J. (2003). Information importance of explanatory variables, *IEE Conference in Honor of Arnold Zellner: Recent Developments in the Theory, Method and Application of Entropy Econometrics.*, Washington.

Specht, D. F. (1991). A general regression neural network, *IEEE Transactions on Neural Networks* 2(6): 568–576.

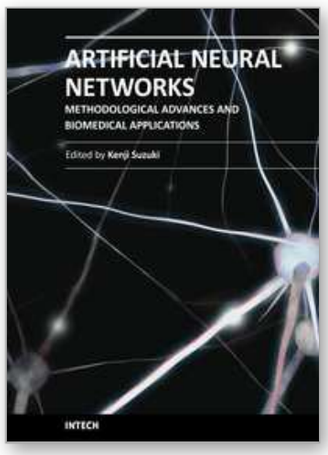
Tibrishani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B (Methodological)* 58(1): 267–288.

Tikka, J. (2008). *Input variable selection methods for construction of interpretable regression models*, Ph.d., Helsinki University of Technology.

Torkkola, K. (2003). Feature extraction by non-parametric mutual information maximization, *Journal of Machine Learning Research* 3: 1415–1438.

Trappenberg, T. P., Ouyang, J. & Back, A. D. (2006). Input variable selection: mutual information and linear mixing measures, *IEEE Transactions on Knowledge and Data Engineering* 18(1): 37–46.

Wold, H. (1966). Estimation of principal components and related models by iterative least squares, *in* P. Krishnaiah (ed.), *Multivariate Analysis*, Academic Press, New York.



Artificial Neural Networks - Methodological Advances and Biomedical Applications

Edited by Prof. Kenji Suzuki

ISBN 978-953-307-243-2
Hard cover, 362 pages
Publisher InTech
Published online 11, April, 2011
Published in print edition April, 2011

Artificial neural networks may probably be the single most successful technology in the last two decades which has been widely used in a large variety of applications in various areas. The purpose of this book is to provide recent advances of artificial neural networks in biomedical applications. The book begins with fundamentals of artificial neural networks, which cover an introduction, design, and optimization. Advanced architectures for biomedical applications, which offer improved performance and desirable properties, follow. Parts continue with biological applications such as gene, plant biology, and stem cell, medical applications such as skin diseases, sclerosis, anesthesia, and physiotherapy, and clinical and other applications such as clinical outcome, telecare, and pre-med student failure prediction. Thus, this book will be a fundamental source of recent advances and applications of artificial neural networks in biomedical areas. The target audience includes professors and students in engineering and medical schools, researchers and engineers in biomedical industries, medical doctors, and healthcare professionals.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Robert May, Graeme Dandy and Holger Maier (2011). Review of Input Variable Selection Methods for Artificial Neural Networks, Artificial Neural Networks - Methodological Advances and Biomedical Applications, Prof. Kenji Suzuki (Ed.), ISBN: 978-953-307-243-2, InTech, Available from: http://www.intechopen.com/books/artificial-neural-networks-methodological-advances-and-biomedical-applications/review-of-input-variable-selection-methods-for-artificial-neural-networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen