Engineering Applications of Neural Networks (2005)

Model and variable selection for nonlinear model design: some developments and methodology

Gérard DREYFUS
ESPCI-Paristech, Laboratoire d'Électronique
10 rue Vauquelin
75005 PARIS – FRANCE

ABSTRACT

The paper discusses the issues of model selection and variable (feature) selection for nonlinear modeling. Methods are described, which were designed to be simple and to involve little computational overhead, but are nevertheless generic, since they do not involve ad hoc heuristics. The principles of the methods are described, and pointers to the description of applications are provided.

1 INTRODUCTION

For many years, a large part of the research in numerical machine learning was devoted to the design of new learning machines (neural networks, radial basis function networks, wavelet networks, support vector machines, just to name a few) and training algorithms. However, crucial issues such as model selection and variable selection, which had been much investigated by statisticians, were considered only recently in numerical machine learning. Many reports on industrial applications of machine learning still overlook the importance of those issues. The present paper describes recently developed methods in variable selection and model selection, which were extensively tested both on academic and on industrial problems. It describes the basic ideas of the methods, and provides pointers to applications.

The first part of the paper is devoted to validation and cross-validation for model selection. Holdout has long been the most widely used method because of its simplicity, but it makes a poor use of the available data. Recent problems in bioinformatics, where data is sparse and candidate variables are very numerous, have spurred the quest for methods that make the best possible use of the data for training and for model selection.

The second part of the paper discusses variable selection separately from model selection; actually, variable selection can be viewed as a special instance of model selection, but that approach tends to lead to computer-intensive methods. We describe the random probe methods, which take into account the specific problems related to variable selection, and solve them as such.

Finally, the limitations of the above methods are described, and open questions are outlined.

2 SOME VARIATIONS ON VALIDATION AND CROSS-VALIDATION FOR MODEL SELECTION

In numerical machine learning, a model of the available data is sought, within a family of functions that are usually not derived from prior knowledge (physics, chemistry, economics, …), but which can approximate any reasonable nonlinear function. For instance, a model is sought within the family of neural networks, or of Radial Basis Function networks, wavelet networks, Gaussian kernel functions, etc. The chosen family of functions has parameters that are adjusted by a suitable training algorithm, and it has meta-parameters such as the number of hidden neurons, the width of the Gaussian kernel functions, etc. Therefore, model selection is mandatory for finding the most appropriate values of the meta-parameters, and, in the case of models that are nonlinear in their parameters, for finding the most appropriate minima of the non-convex cost function.

That is in contrast to statistical regression, whereby a model is derived from prior knowledge and is assumed to be unquestionably true. If that model has adjustable parameters, the latter are estimated from experimental data by a suitable algorithm, which is often very similar (if not identical) to training algorithms used in machine learning. In that context, the parameters usually have a specific meaning, and the validity of the estimation of the parameters must be assessed, e.g. by computing confidence intervals for the parameters.

Therefore, in machine learning, *the validity of the predictions* of the model is emphasized, whereas the *validity of the parameters* of the model is emphasized in traditional regression. Hence, the central question in machine learning is the generalization ability of the model: how does the model respond to situations that are not present in the training set? The generalization error can be estimated from a validation set, i.e. a set examples that have not been used for training the model; then the question that arises is that of the validity of that estimation, which can be expressed statistically in terms of a confidence interval on that estimation.

A confidence interval, with confidence threshold 1-α around the mean of a random variable Y, is an interval that contains the value of the expectation of Y with probability 1-α (see for instance (Mood 1974)). Therefore, a confidence interval ε on the difference between the estimate e_V of the generalization error (computed on a validation set), and the (unknown) true generalization error e is defined as:

$\Pr\left[\left(e_V - e\right) \geq \varepsilon\right] < 1 - \alpha$.

The solution to the problem of finding ε, given α and the number of examples, involves the Vapnik-Cervonenkis dimension of the model (Vapnik 1982). However, that principled approach leads to bounds on the generalization error that are usually too loose for practical applications.

2.1 Hold-out

The simplest technique for estimating the generalization error is *hold-out*: part of the available data (the training set) is used for training the model, and the rest of the data is hold out of the training set to serve as a validation set, from which the generalization error is estimated (Stone 1974). That very simple technique can be applied safely when the amount of available data is much larger than the number of parameters of the model.

In the context of regression, signal modeling or process modeling, it is assumed that the result y^k of the k-th measurement of the quantity of interest can be modeled as the sum of an unknown function $f(\mathbf{x}^k)$ and of a realization δ^k of a random variable Δ with zero expected value, which models all disturbances and measurement noise. Denoting the model by $g(\mathbf{x}, \boldsymbol{\theta})$, the purpose of training is to find a set of parameters that minimize the training root mean square error $TMSE = \sqrt{\frac{1}{N_T}\sum_{k=1}^{N_T} R_k^2}$, where N_T is the number of examples of the training set, and R_k is the modeling error on example k of the training set: $R_k = y^k - g\left(\mathbf{x}^k, \boldsymbol{\theta}\right)$. Therefore, the generalization ability of the model is estimated by the root mean square validation error $VMSE = \sqrt{\frac{1}{N_V}\sum_{k=1}^{N_V} R_k^2}$, where N_V is the number of examples in the validation set. The unbiased estimate of the standard deviation of the random variable Δ, estimated from a set of N measurements, is given by $\sigma_\Delta = \sqrt{\frac{1}{N-1}\left(y^k - f\left(\mathbf{x}^k\right)\right)^2}$. Therefore, if the model was perfect, i.e. if $f(\mathbf{x}^k) = g(\mathbf{x}^k, \boldsymbol{\theta})$ for all k, and if $N >> 1$, the training error and the validation error would be on the order of magnitude of the standard deviation of the disturbances acting on the data and the noise present in it. This makes sense since the model should not be expected to be more accurate than the data from which it is designed.

Since the training error can be made arbitrarily small by increasing the complexity of the model, the latter quantity should never be used for model selection. If the training error is substantially smaller than the validation error, *overfitting* has occurred: the complexity of the model is too high given the available data, so that the model has learnt the noise in addition to the deterministic part of the generating process of the data, hence performs poorly on fresh data.

2.2 Cross-validation

As mentioned above, holdout is an appropriate technique if data is abundant, so that the estimation of the generalization error on the validation set is reliable. If data is sparse, the number of elements in the validation set is small, so that the estimation of the generalization error on a single data set becomes unreliable. Then one must resort to cross-validation: the available data is split into D disjoint subsets, training is performed on D-1 subsets and the modeling error is computed for all examples of the validation set; the procedure is repeated D times with D different validation sets (D-fold cross-validation). Therefore, each example is present once and only once in the validation set, and the cross-validation root mean square error is computed from the validation data: $CVMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(R_k^V\right)^2}$, where N is the total number of examples and R_k^V is the modeling error on example k when the latter is in the validation subset. Therefore, cross-validation uses all examples for estimating the generalization error, while holdout uses only the elements of the holdout set. However, no unbiased estimator of the variance of cross-validation can be found, whereas such estimators exist for holdout (Bengio 2003).

2.3 Leave-one-out and virtual leave-one-out

When data is very sparse, the best estimate of the generalization error can be obtained by the leave-one-out technique. Leave-one-out is cross-validation with $D = 1$: each example is withdrawn in turn from the data set, training is performed with $N - 1$ examples, and the modeling error is computed on the left-out example. The leave-one-out score is subsequently computed as $E_t = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(R_i^{-i}\right)^2}$, where R_i^{-i} is the modeling error on example i when it is withdrawn from the training set. The leave-one-out score can be shown to be an unbiased estimate of the generalization error. However, it is a computer-intensive procedure.

Its computational burden can be alleviated to a considerable extent by the *virtual leave-one-out* procedure. First consider a linear model: $g(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{x}^T \boldsymbol{\theta}$, where the superscript T denotes matrix transposition, and $\boldsymbol{\theta}$ is the vector of parameters, of size p. Then the least squares solution is given by

$$\boldsymbol{\theta}_{LS} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \qquad (1)$$

where $\mathbf{y}$ is the N-vector of the observations, and $\mathbf{X}$ is the observation matrix, i.e. the (N, p) matrix whose columns are the observed values of the variables; hence matrix $\mathbf{X}$ has p columns, each of which has size N. If example i is left out of the training set, then a different model $\boldsymbol{\theta}_{LS}^{-i}$ is derived. We denote by $\mathbf{R}^{-i}$ the N-vector of the residuals (modeling errors) of the model trained without example i, N-1 of which pertain to the training set, and one of which, R_i^{-i}, is the residual of the left-out example. We show in the following that R_i^{-i} can be computed exactly from the modeling error of example i when that example is within the training set.

The cost function for the model computed with all example is $J = \mathbf{R}^T\mathbf{R}$, where $\mathbf{R}$ is the N-vector of its residuals. Since J is minimum for the least-squares solution, one has:

$$\mathbf{0} = \mathbf{X}^T\mathbf{R}. \qquad (2)$$

We denote by $\mathbf{R}^{-i}$ the N-vector of modeling errors on the full data set. Then the cost function for the model trained without example i is given by: $J^{-i} = \left(\mathbf{R}^{-i}\right)^T\mathbf{R}^{-i} - \left(R_i^{-i}\right)^2$; since it is minimum for the parameter vector $\boldsymbol{\theta}_{LS}^{-i}$, one has:

$$\mathbf{0} = \mathbf{X}^T\mathbf{R}^{-i} - R_i^{-i}\mathbf{x}_i \qquad (3)$$

where $\mathbf{x}_i$ is the p-vector of variables for example i ($\mathbf{x}_i^T$ is row i of matrix $\mathbf{X}$).

Moreover, one has

$$\mathbf{R}^{-i} = \mathbf{y} - \mathbf{X}\boldsymbol{\theta}_{LS}^{-i} = \mathbf{R} - \mathbf{X}\left(\boldsymbol{\theta}_{LS}^{-i} - \boldsymbol{\theta}_{LS}\right) \qquad (4)$$

Combining (2), (3) and (4), one obtains:

$$\mathbf{0} = -\mathbf{X}^T\mathbf{X}\left(\boldsymbol{\theta}_{LS}^{-i} - \boldsymbol{\theta}_{LS}\right) - R_i\mathbf{x}_i + R_i\mathbf{x}_i\mathbf{x}_i^T\left(\boldsymbol{\theta}_{LS}^{-i} - \boldsymbol{\theta}_{LS}\right) \qquad (5)$$

Therefore:

$$\boldsymbol{\theta}_{LS}^{-i} - \boldsymbol{\theta}_{LS} = -R_i\left(\mathbf{X}^T\mathbf{X} - \mathbf{x}_i\mathbf{x}_i^T\right)^{-1}\mathbf{x}_i, \qquad (6)$$

hence:

$$R_i^{-i} = R_i\left[1 + \mathbf{x}_i^T\left(\mathbf{X}^T\mathbf{X} - \mathbf{x}_i\mathbf{x}_i^T\right)^{-1}\mathbf{x}_i\right] \qquad (7)$$

From the Sherman-Morrison matrix inversion lemma, one has:

$$\left(\mathbf{X}^T\mathbf{X}-\mathbf{x}_i\mathbf{x}_i^T\right)^{-1}=\left(\mathbf{X}^T\mathbf{X}\right)^{-1}+\frac{\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{x}_i\mathbf{x}_i^T\left(\mathbf{X}^T\mathbf{X}\right)^{-1}}{1-\mathbf{x}_i^T\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{x}_i}. \tag{8}$$

Therefore, relation (7) can be written as:

$$R_i^{-i}=\frac{R_i}{1-h_{ii}} \tag{9}$$

where

$$h_{ii}=\mathbf{x}_i^T\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{x} \tag{10}$$

is the diagonal element of the orthogonal projection matrix (also termed “hat matrix”)

$$\mathbf{H}=\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T. \tag{11}$$

Therefore, *for linear models*, the leave-one-out score can be computed exactly; it is the square root of the PRESS (Predicted Residual Sum of Squares) statistic:

$$PRESS=\frac{1}{N}\sum_{i=1}^{N}\left(\frac{R_i}{1-h_{ii}}\right)^2 \tag{12}$$

For models that are nonlinear with respect to their parameters, a first-order Taylor expansion of the model with respect to its parameters can be written:

$$\mathbf{g}\left(\mathbf{x},\boldsymbol{\theta}^{-i}\right)\approx\mathbf{g}\left(\mathbf{x},\boldsymbol{\theta}\right)+\mathbf{Z}\left(\boldsymbol{\theta}^{-i}-\boldsymbol{\theta}\right) \tag{13}$$

under the assumption that the withdrawal of the example from the training set does not affect the model substantially. In relation (13), **Z** is the jacobian matrix of the model

$$Z_{ij}=\left(\frac{\partial\mathbf{g}\left(\mathbf{x},\boldsymbol{\theta}\right)}{\partial\theta_j}\right)_{\mathbf{x}=\mathbf{x}_i},$$

$\boldsymbol{\theta}$ is the vector of the parameters obtained after training from the whole data set: it is the vector of parameters corresponding to a (possibly local) minimum of the cost function, and $\boldsymbol{\theta}^{-i}$ is the vector of parameters after training the model from all examples except example *i*.

Based on relation (13), it can be shown (Monari and Dreyfus 2002) that an approximation of the leave-one-out score, termed virtual leave-one-out score, can be derived as:

$$E_p=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{R_i}{1-h_{ii}}\right)^2} \tag{14}$$

where h_{ii} is the *i*-th diagonal element of matrix

$$\mathbf{H}=\mathbf{Z}\left(\mathbf{Z}^T\mathbf{Z}\right)^{-1}\mathbf{Z}^T \tag{15}$$

Note the similarity with relation (11). Whether the model is linear or nonlinear in its parameters, the leverages, being diagonal elements of an orthogonal projection matrix, comply with the following relations:

$$0\le h_{ii}\le 1 \tag{16}$$

$$\sum_{i=1}^{N}h_{ii}=p \tag{17}$$

where *p* is the number of parameters of the model.

As a preliminary check, the rank of the jacobian matrix should be checked: a rank deficiency of the jacobian matrix means that there is a linear dependence between columns of the matrix, i.e. between the derivatives of the output with respect to the variables. For instance, if two columns are proportional, the derivatives of the model with respect to the two corresponding parameters are proportional; therefore, the number of parameters is too high given the complexity of the data; such a situation is very likely to give rise to overfitting.

When several candidate models, with different variables and/or different structures, have virtual leave-one-out scores that are on the same order of magnitude, i.e. when they differ by amounts that are on the order of magnitude of the standard deviation of the noise, the significance of the differences between the estimated performances of the models must be assessed; that can be done by statistical tests (see for instance (Anders 1999)). In such a case, an additional selection criterion may be used, which relies on the interpretation of the leverages.

2.4 Interpretation of the leverages, model selection and experimental planning

As mentioned in the previous section, the sum of the leverages is equal to the number of parameters of the model: $\sum_{i=1}^{N} h_{ii} = p$. Therefore, the leverage of example *i* can be regarded as the fraction of the degrees of freedom of the model that is used up by the model for fitting example *i*. Therefore, examples that have large leverages are examples that use up a large part of the parameters of the model, which is of course detrimental to the generalization error: the model is highly specialized on those points and is likely to generalize poorly. Clear examples of such situations can be found in (Oussar 2004).

Ideally, one would like all observations to have the same influence on the model; if that was possible, then one would have $h_{ii} = p/N$ for all *i*. Although that is generally impossible to achieve (unless the number of examples is equal to the number of parameters), it is desirable that the distribution of the leverages be as peaked as possible around their mean value *p*/*N*. That condition can be expressed by the normalized standard deviation of the leverages σ_n:

$$\sigma_n = \sqrt{\frac{N}{p(N-p)} \sum_{i=1}^{N} \left(h_{ii} - \frac{p}{N} \right)^2} .$$

$\sigma_n = 0$ if all leverages are equal to *p*/*N*, and $\sigma_n = 1$ in the worst case, where *p* examples have leverages equal to 1 and *N*-*p* leverages are equal to zero.

To summarize, selection among candidate models, having different features and/or different complexities, can be performed economically as follows:

- check the rank of the jacobian matrix of the model; discard any model whose jacobian matrix is rank deficient;
- compute the virtual leave-one-out scores of the candidate models; select the model(s) whose leave-one-out scores are smallest, and differ by less than the standard deviation of the noise, or which differ by quantities that are not considered statistically significant;
- among the latter models, select the model whose leverages are least scattered around the mean value.

As a final assessment of the quality of the predictions of the model, confidence interval can be computed, either analytically or by resampling (see for instance (Efron and Tibshirani 1993)). A confidence interval that uses the elements of the jacobian matrix, as defined above, can be convenient (Seber and Wild 1989); it is centered on the model prediction $g(\mathbf{x}, \theta_{LS})$, and its width *w* is given by $w \simeq 2t_\alpha^{N-p} s \sqrt{\mathbf{z}^T \left(\mathbf{Z}^T \mathbf{Z}\right)^{-1} \mathbf{z}}$, where t_α^{N-p} is the realization of a Student variable at level of significance α, and $s^2 = \frac{1}{N-1} \sum_{i=1}^{N} R_i^2$.

Figure 1 shows the training data, the prediction of a model trained from that data, and the confidence intervals for that model. As expected, the confidence intervals are very large where data is not present. Figure 2 illustrates the improvement that can be obtained by simply adding two examples at appropriate locations in input space. Discovering the optimal location of examples in input space in order to obtain the best model with the minimum number of experiments is known as *experimental planning*.

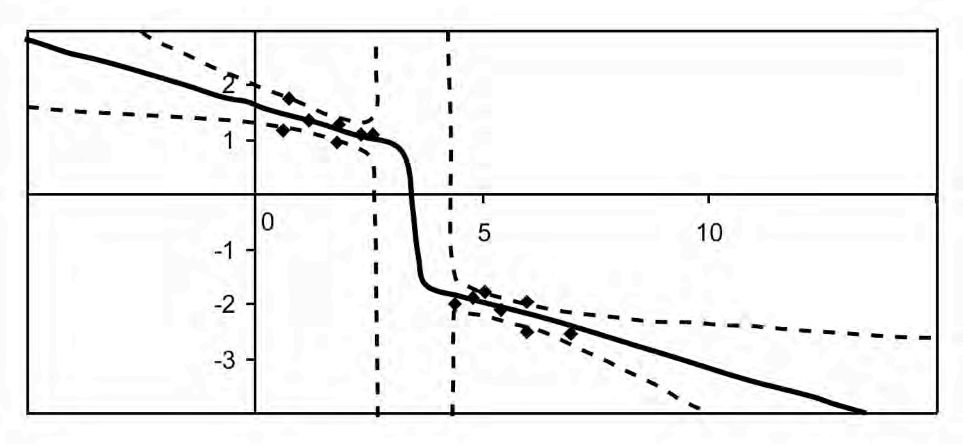

Figure 1
Solid line: prediction of the model; dashed lines: boundaries of the confidence interval on the prediction.

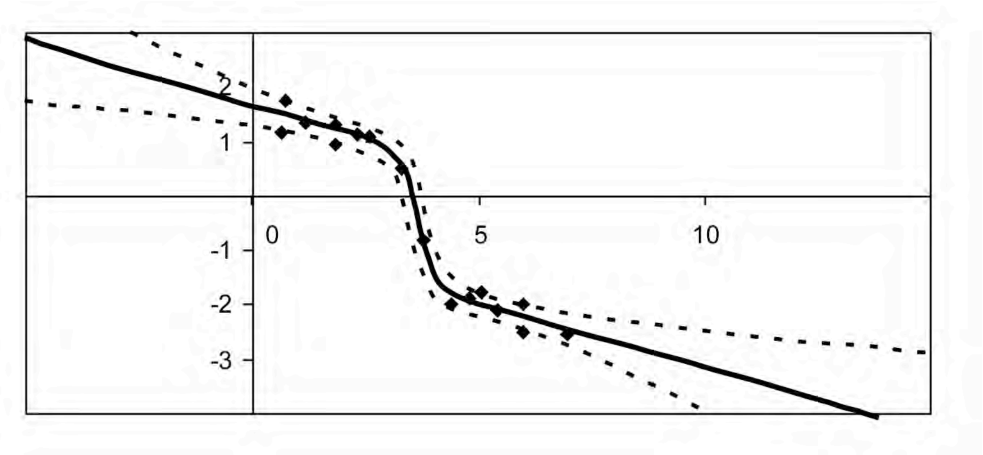

Figure 2
Solid line: prediction of the model; dashed lines: boundaries of the confidence interval on the prediction.

Note that, for models that are linear in their parameters, the leverages do not depend on the model: they depend only on the location of the examples in input space. That is evidenced from relation (10), which involves matrix **X** only. That is the reason why optimal experimental planning can be performed for linear models (*D*-optimal experimental planning) insofar as the number of parameters of the model, and their range of variation, are known.

By contrast, for models that are nonlinear with respect to their parameters, the jacobian matrix depends on the parameters of the model. Therefore, experimental planning must be performed in a iterative fashion: some experiments must be performed in order to build a first tentative model, which can be further refined by performing " optimal" experiments given that model, and so on iteratively until the budget of feasible experiments is exhausted. An industrial applications of *D*-optimal experimental planning of neural models in described in (Quach 2004).

The above techniques provide simple, computationally inexpensive means of assessing the generalization ability of a model with its selected variables. For a given model, they provide assessments of the suitability of different features, hence can serve for feature selection. In the next section, we describe the "random probe" methodology, which is more specifically designed for feature selection.

3 VARIABLE SELECTION AND RANDOM PROBES

Variable selection is a very important step in the design of a statistical model, for two reasons: first, the number of parameters of the model increases with the number of variables, which in turn increases the number of observations that are necessary for a reasonable estimation of the parameters; in addition, irrelevant variables just add noise to the model, which makes model design more difficult. Therefore, the model designer should strive to incorporate all influential variables in the model, but to discard all irrelevant variables, i.e. variables whose influence on the quantity to be modeled is smaller than the measurement noise of that quantity. A general introduction to the variable selection problem can be found in (Guyon 2003).

The first step of a principled variable selection method consists in defining a “relevance index”, i.e. some quantity that describes the relevance of the candidate variables for the prediction of the quantity to be modeled. Thus, the variable selection problem can be viewed as a classification problem: given a feature (the relevance index), assign each candidate variable to one of the two classes “relevant” or “irrelevant”. Therefore, the probability distribution functions (pdf’s) of the relevance index for relevant variables and for irrelevant variables should be as far apart as possible, so that a relevance threshold can be easily defined (Figure 3). Conversely, if the pdf’s have significant overlap, then “false negatives” (variables that are discarded although they are relevant) and “false positives” (variables that are selected although they are irrelevant nonzero probability) have nonzero probabilities (Figure 4); then a threshold r_0 must be chosen, which minimizes the total error risk, or one of them if all errors do not have the same cost. In contrast to statistical classification problems, both distributions are unknown, and no “example” of elements of the classes is available.

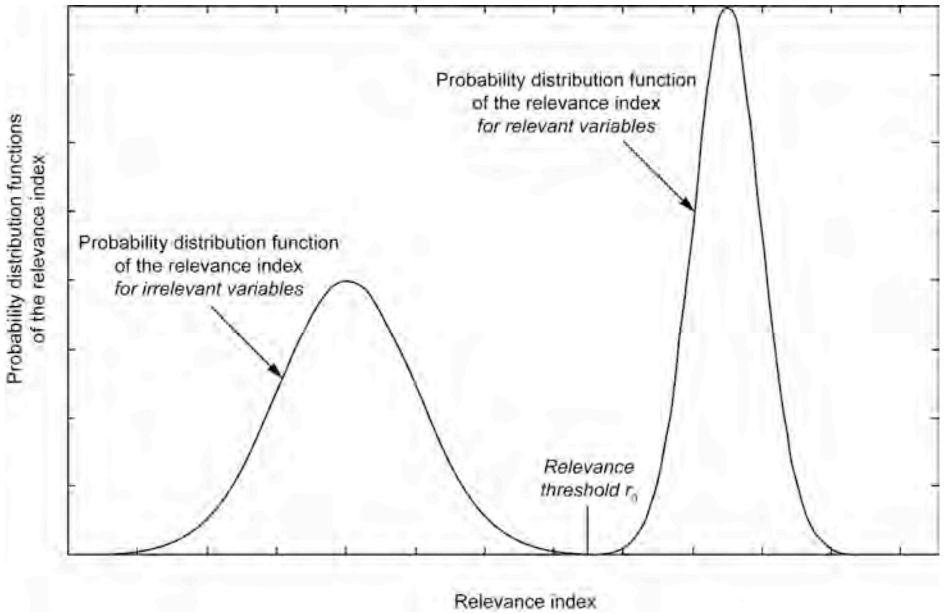

Figure 3
Probability distributions of relevant and irrelevant variables

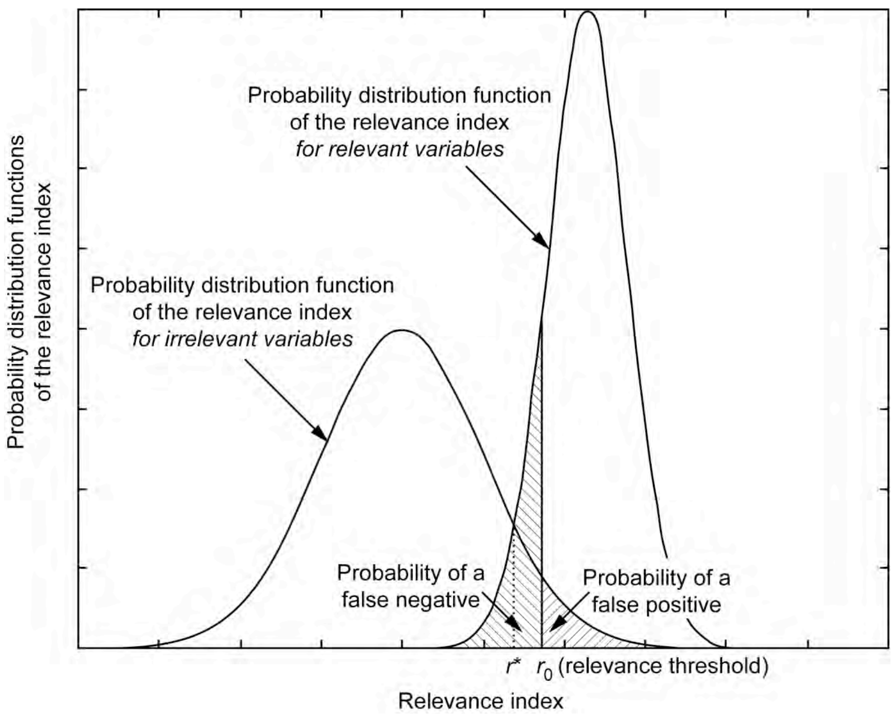

Figure 4

Probability distributions of relevant and irrelevant variables; if the relevance threshold is chosen equal to r^*, the probability of wrong selection is minimum

Basically, the random probe method consists in generating “fake” candidate variables, computing their relevance index and the probability distribution function thereof, and setting the relevance threshold accordingly.

3.1 Generation of the random probes

Random probes can be generated in a number of ways. Since the variables are usually scaled and centered, random features can be generated by simply shuffling randomly the components of candidate feature vectors (Bi 2003). Alternatively, random features can be generated from a distribution that is centered on zero and has variance equal to 1, e.g. a normal distribution (assuming that, according to sound practice, the candidate features are centered and normalized prior to any processing); the advantage of using the latter distribution will be explained in the next section.

3.2 Assessment of the relevance of the candidate features

Relevance assessment is most easily performed by constructing a model that is linear in its parameters, e.g. a polynomial model or a Support Vector Machine (Rakotomamonjy 2003). The features that are thus selected are then used for constructing a nonlinear-in-its-parameters model, if necessary.

In a first step, the set of candidate features, including the random probes, is defined. Since variable selection aims at designing an efficient model, a polynomial (typically quadratic) model is used for feature selection; therefore, the set of candidate variables contains the p primary candidate variables, and the $p(p+1)/2$ pairwise combinations of the primary variables, termed secondary variables.

The relevance of candidate variables can be assessed in a computationally efficient way by orthogonal forward regression (Chen et al. 1989): in observation space (of dimension N, where N is the number of observations in the training set), each candidate feature is depicted as a vector $\mathbf{x}$ whose components are the values of that input over the training set, and the quantity to be modeled is similarly depicted by an N-

vector **y**. If the angle φ_i between a candidate input and the output is zero, the model can be written as $\mathbf{y} = \lambda\, \mathbf{x}$, so that the variable **x** explains completely the output. Conversely, if the angle is π/2, feature **x** is completely irrelevant. Therefore, the dot product between each candidate input and the output, in observation space, can be a relevance index. In practice, it is more convenient to compute:

$$\cos^2 \varphi_i = \frac{(\mathbf{y} \cdot \mathbf{x}_i)^2}{\|\mathbf{y}\|^2 \, \|\mathbf{x}_i\|^2} \tag{18}$$

where $\mathbf{x}_i$ is the vector of observations of the i-th candidate variable.

In order to rank the inputs in order of decreasing relevance, the following orthogonalization procedure, which is similar to orthogonal forward regression (Chen et al. 1989) can be used:

- select the candidate variable that has the smallest angle (largest $\cos^2\varphi_i$) with the quantity to be modeled in observation space;
- project the output vector and all other candidate features onto the null space of the selected variable; compute the parameter pertaining to that input in the linear-in-its-parameters model;
- iterate in that subspace.

The orthogonalization step can be performed efficiently by the modified Gram-Schmidt procedure (Björck 1967).

The procedure terminates when all candidate features are ranked, or when a prescribed stopping condition, as described below, is met. If the model thus obtained is not satisfactory, a new model should be built, with higher degree if a polynomial is used.

3.3 Variable selection with random probes

The relevance of the candidate variables, including the probes, being assessed by their ranks in the ranked list built as described in the previous section, the relevance threshold must be chosen as described on Figure 3 and Figure 4. Instead of estimating the probability distribution function of the rank of the probes, its cumulative distribution function (cdf) is either estimated by enumeration, or computed analytically.

The probability that the rank is smaller than or equal to a given rank r is estimated as the number of realizations of the random probe that have a rank smaller than or equal to r, divided by the total number of realizations of the probe. Alternatively, if the probes are drawn from a normal distribution, the cdf of the rank can be computed exactly as described in (Stoppiglia et al. 2003).

Since the cdf of the relevant variables is unknown, the choice of the relevance threshold can be made as follows: as the number of parameters increases with the number of variables, the latter should be kept as low as possible if data is sparse. Therefore, a small risk of false positive (keeping a candidate variable although it is irrelevant) will be chosen if a small amount of data is available; conversely, if there is a wealth of data, a larger risk may be acceptable. Thus, at each step of the orthogonalization procedure described above, the value of the cumulative distribution function is compared to the predefined risk, and ranking is terminated when that threshold is reached. This has an importance consequence: the method can handle problems where the number of candidate variables is larger than the number of examples; the only limitation is related to the number of *really relevant* variables, which must be smaller than the number of examples.

Figure 2 displays the cumulative distribution function of the rank of the random probe, as estimated from 100 realizations of the random probe, and computed analytically, for a problem with 10 candidate variables. If a 10 % risk is chosen, the first five features should be selected.

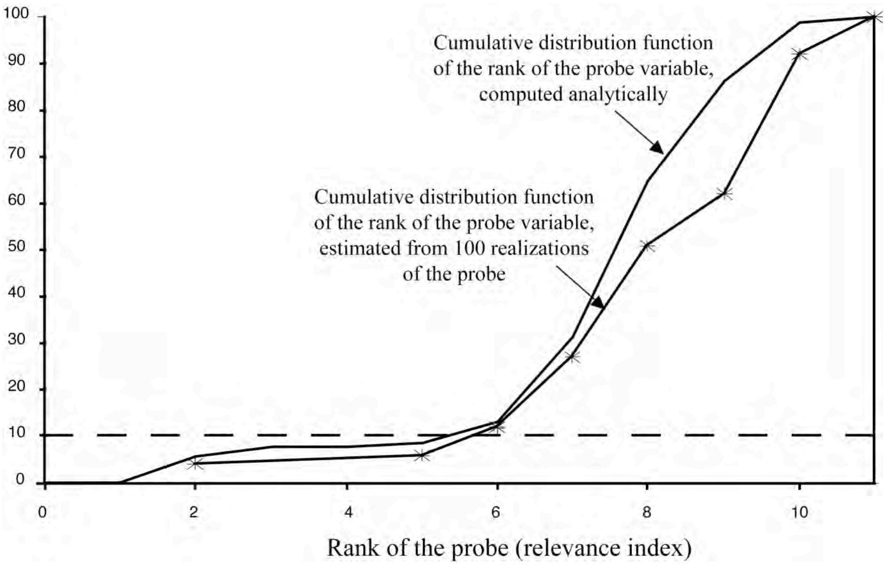

Figure 5
Estimated and computed cumulative distribution function of the rank of a random probe for a problem with ten candidate variables.

It has been shown (Stoppiglia et al. 2003) that the random probe method can be viewed as a generalization of Fisher's test for variable selection, which does not rely on the assumption that the complete model contains the regression, and which does not make any specific assumption about the distribution of the probe.

4 APPLICATIONS

The methods described in the previous sections are generic and completely application-independent. They have been used in a variety of applications.

Both the model selection and the variable selection methods were used in the design of a virtual sensor for car assembly applications. Spot welding is the prevalent technique for metal sheet assembly: the sheets re pressed between two electrodes, and a current of a few kiloamperes is passed through the electrodes during a few tens of milliseconds, resulting in an intense Joule heating that melts the metal locally. After cooling, the melted zone binds the two sheets together, and its diameter is an indicator of the strength of the weld. At present, no fast, non-destructive method exists for measuring the spot diameter, so that there is no way of assessing the quality of the weld immediately after welding. Modeling the dynamics of the welding process from first principles is a difficult task, which cannot be performed in real time. These considerations led to considering black-box model that predicts the diameter of the spot from electrical and mechanical measurements performed during welding for designing a "virtual sensor" of the spot diameter from electrical and mechanical measurements performed during welding. The main concerns for the modeling task were the choice of the model inputs, and the limited amount of examples available initially in the database. Variable selection (Monari 1999) was performed both by the random probe method, and by more classical methods (stepwise backward regression and statistical tests based on performance comparisons), with identical results, but at a lower computational cost. The process experts validated the variable set thus obtained. Model selection was performed by virtual leave-one-out as described above.

Similarly, both the variable selection and the model selection methods were successfully used for the design of a virtual sensor for medical applications: in that area, the purpose is to replace costly chemical tests by a machine learning method that predicts the outcome of that test from results of much cheaper tests. The problem is highly nonlinear; there were a few tens of candidate variables, and model selection was performed as described in sections 2.3 and 2.4.

The above applications are in the area of nonlinear regression. However, the methods described above can be used for classification applications too. It was most successful in natural language processing, for the task of information filtering. The purpose is to find automatically, in a large corpus of texts (e.g. all press releases of Reuter or Agence France Presse), the texts that are relevant to topics that are defined by a few words. That can be formalized as a 2-class classification problem (a text is either relevant or irrelevant). In the so-called "bag of words" approach, each text is described by a vector of numbers that are related to the ration of the frequency of occurrence of each word in the text under consideration to its frequency in the whole corpus; therefore, the size of the vector of variables is equal to the number of words in the dictionary. Therefore, variable selection is a crucial step. It was showed that, on the average, the vocabulary that is specific to a topic (hence discriminant) has 25 words, which is a tremendous complexity reduction. It was a key factor in the success of that application, which won the TREC'9 international competition (Stricker 1999), (Wolinski 2000).

5 DISCUSSION AND CONCLUSION

In the present paper, two all-important problems in nonlinear model design have been addressed: variable selection and model selection. Although they were discussed separately, it is clear that model selection is not independent from variable selection. For instance, instead of choosing a priori a given risk for setting the relevance threshold, one might choose a range of variation for the risk and wrap a variable selection loop around a model design and selection software module. The link between model selection and experimental planning was also emphasized.

There are, however, several important issues that are still under active investigation. First, the probe method using ranking by orthogonalization as a relevance index cannot be safely applied if the number of relevant variables is larger than the number of examples. Although that is an unusual situation, it may arise in biomedical applications related to genomics or proteomics. An efficient solution to that problem is described in (Neal 2005).

For model selection, the estimation of the generalization error by the virtual leave-one-out score is limited in its applicability to the validity of the first-order Taylor expansion of the model output in parameter space. A generalization of the leverages described here, taking into account second derivatives, has been computed and is under investigation.

Despite their limitations, the methods described here have been used very successfully in many applications, in areas ranging from drug design to automatic language processing and virtual sensor design. They are the keys to successful applications given the present state of the art.

6 LITERATURE REFERENCES

Allen DM (1974) The relationship between variable selection and prediction. Technometrics 16: 125-127.

Anders U, Korn O (1999) Model selection in neural networks. Neural Networks 12: 309-323.

Bengio J, Grandvalet Y (2003) No unbiased estimator of the variance of K-fold cross-validation, Journal of Machine Learning Research 5: 1089-1105.

Bi J., Bennett KP, Embrechts M, Breneman CM, Song M (2003) Dimensionality reduction via sparse support vector machines, Journal of Machine Learning Research 3: 1229-1243.

Björck A (1967) Solving linear least squares problems by Gram-Schmidt orthogonalization, Nordisk Tidshrift for Informationsbehadlung 7: 1-21.

Chen S, Billings SA, Luo W (1989) Orthogonal least squares methods and their application to non-linear system identification. International Journal of Control. 50: 1873-1896.

Efron B, Tibshirani RJ (1993) Introduction to the bootstrap. Chapman and Hall, New York.

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection, Journal of Machine

Learning Research 3: 1157-1182

Monari G (1999), Sélection de modèles non linéaires par leave-one-out. Etude théorique et application au procédé de soudage par points. Thèse de Doctorat de l'Université Pierre et Marie Curie (available electronically at `http://www.neurones.espci.fr`).

Monari G, Dreyfus G (2002) Local overfitting control via leverages. Neural Computation, 14: 1481-1506.

Mood AM, Graybill FA, Boes DC (1974) Introduction to the theory of statistics. McGraw-Hill, Singapore.

Neal R, Zhang J (2005) High dimensional classification with Bayesian neural networks and Dirichlet diffusion trees. In: Feature extraction, foundation and applications, I. Guyon, S. Gunn, M. Nikravesh, L. Zadeh, eds. Springer (2005)

Oussar Y, Monari G, Dreyfus G (2004) Reply to the comments on "Local overfitting control via leverages" in "Jacobian conditioning analysis for model validation". Neural Computation 16: 419 - 443.

Quach R, Masson A, Dreyfus G, Gauchi JP, Issanchou S (2004) Plans d'expériences pour réseaux neuronaux, Lambda-mu 14, Bourges.

Rakotomamonjy A (2003) Variable Selection Using SVM based Criteria**.** Journal of Machine Learning Research **3**: 1357-1370

Seber GAF, Wild CJ (1989) Nonlinear regression. John Wiley and Sons, New York.

Stone M (1974) Cross-validatory choice and assessment of statistical predictions. J. Roy. Stat. Soc. B 36: 111-147.

Stoppiglia H, Dreyfus G, Dubois R, Oussar Y (2003) Ranking a random feature for variable and feature selection. Journal of Machine Learning Research 1399-1414.

Stricker M, Vichot F, Dreyfus G, Wolinski F (1999) Two-Step Feature Selection and Neural Network Classification for the TREC-8 Routing. In Proceedings of the Eighth Text Retrieval Conference. Available electronically at:
http://www.neurones.espci.fr/Articles_PS/trec.pdf

Vapnik VN (1982) Estimation of dependencies based on empirical data. Springer, New-York.

F. Wolinski, F. Vichot, M. Stricker, Using Learning-Based Filters to Detect Rule-based Filtering Obsolescence. In *Proceedings RIAO'2000,* 2000. Available electronically at:
http://www.neurones.espci.fr/Articles_PS/riao2000.pdf