Multicollinearity: A tale of two nonparametric regressions

Richard D. De Veaux
Princeton University

Lyle H. Ungar
University of Pennsylvania

Abstract

The most popular form of artificial neural network, feedforward networks with sigmoidal activation functions, and a new statistical technique, multivariate adaptive regression splines (MARS) can both be classified as nonlinear, nonparametric function estimation techniques, and both show great promise for fitting general nonlinear multivariate functions.

In comparing the two methods on a variety of test problems, we find that MARS is in many cases both more accurate and much faster than neural networks. In addition, MARS is interpretable due to the choice of basic functions which make up the final predictive equation. This suggests that MARS could be used on many of the applications where neural networks are currently being used.

However, MARS exhibits problems in choosing among predictor variables when multicollinearity is present. Due to their redundant architecture, neural networks, however, do not share this problem, and are better able to predict in this situation. To improve the ability of MARS to deal with multicollinearity, we first use principal components to reduce the dimensionality of the input variables before invoking MARS. Using data from a polymer production run, we find that the resulting model retains the interpretability and improves the accuracy of MARS in the multicollinear setting.

1 Introduction

In many estimation problems, the functional form governing the relationship between predictors and response is not known. Examples abound everywhere and include such diverse applications as modeling chemical plants, time series modeling, and relationships between spectrographic data and chemical concentrations. When many predictor variables are present, choosing the best subset of predictors is a formidable task, even if one assumes that the response is a linear function of the predictors. When the linearity assumption fails, as it often does, the problem becomes daunting.

Historically, many different equation forms have been suggested, some based on experience, others by mathematical convenience. Stepwise procedures of systematically adding and removing terms have often led to reasonable, if not optimal models. However, when the predictors exhibit a high degree of multicollinearity, great instability in the selection process can be present. In the linear case, a great deal can be learned by examining the correlation structure of the predictor variables. Solutions such as principal component regression (PCR), ridge regression or partial least squares (PLS) may then be appropriate. Such projection methods take linear combinations of the original variables to create new variables, of which a subset often gives an accurate model.

Neural networks, in many forms, are also being widely used to learn nonlinear mappings (see Rumelhart *et al.*, 1986 or Ripley, 1992 for an introduction to neural networks). These networks have very large numbers of parameters (called weights) estimated by minimizing an error over a set of training patterns. Because they are nonlinear projection methods, and because of their tendency to overparameterize, neural networks tend to be fairly insensitive to problems of multicollinearity. However, precisely because they are overparameterized, they are typically not used for interpretation of the system, but only for prediction. Neural networks offer a nonlinear method which uses projection; the input variables are linearly combined before being passed through their nonlinear transformations.

Multivariate adaptive regression splines (MARS, Friedman 1991), use forward selection to adaptively build a set of basis functions for the function approximation. Unlike projection methods, MARS works in the original coordinate system and finds linear and nonlinear combinations of these coordinates. Recently, De Veaux *et al.* (1992) evaluated and compared neural networks and MARS via their performance on several benchmark problems. In most cases, MARS was able to predict as well or better as the neural network. However, this superiority was not uniform over the test problems.

It is the goal of this paper to better understand when different types of methods work or fail, particularly on nonlinear problems. We will show that whether projection or selection methods perform better is situation dependent. We furthermore show how a nonlinear selection-based method, MARS, can be combined with linear projection methods such as principal components to give models which are as accurate as neural networks, but are simpler and more interpretable.

The paper is structured as follows. The next section recalls certain properties of MARS and neural networks. Section 3 presents the problem of multicollinearity and gives a solution to the collinearity problem which combines principal components and MARS. In section 4 the method is applied on a data set from a polymer production run and compared to a neural network. Section 5 summarizes our results.

2 MARS and neural networks

We assume that most readers are familiar with principal component analysis (PCA) and with neural networks, and so describe them only very briefly. MARS is less widely used, but due to space limitations we must refer the reader to descriptions in Friedman (1991), De Veaux *et al.* (1992) or Sekulic and Kowalski (1992a).

2.1 MARS

The goal of MARS is to produce a simple, understandable model of a response to an arbitrary function of a set of predictor variables. MARS adds basis functions by forward selection. For each predictor variable, x_i, and every possible value, t of x_i, MARS divides the data into two parts, one on either side of the "knot", t. MARS keeps the knot and variable pair which gives the best fit. To each part, it fits the response using a pair of linear functions, each non-zero on one side of the knot.

After one variable has been selected, a split on a subsequent variable can occur in two different ways. The split can either depend on the previous split (splitting the input space only on one side of the previous knot), or it can ignore the previous split, splitting the entire input space on the new knot. In this former case, the associated basis functions of the original knot are said to be parents of the new basis functions.

MARS adds to the set of basis functions using a penalized residual sum of squares, or generalized cross validation criterion. Like any forward selection procedure, when confronted with two highly correlated input variables, it must choose between them. This may be reflected in the final set of basis functions, where only one of the two may be represented. In many applications, this may not be the optimal choice. For example, when multiple sensors are present, a weighted average of the sensors, may be preferable to the single one with the highest correlation with the response.

In general, the set of basis functions selected will be too large, and must be pruned back using backward elimination. The final model is obtained by performing backward selection using the generalized cross validation criterion. The linear basis functions are then replaced by cubic splines to make the approximation smoother.

2.2 Neural networks

The most common form of neural networks, feedforward networks with sigmoidal activation functions or, as they are sometimes called, "backpropagation" networks can, in the simplest case of a single layer of N hidden nodes, be written in the form

$$y_i = \sum_{j=1}^{N} w_{ij}\sigma_j(x) \tag{1}$$

where

$$\sigma_j(x) = 1/(1 + \exp - \sum_k w_{jk}x_k) \tag{2}$$

and the weights w_{ij} and w_{jk} are selected by a nonlinear optimization method to minimize the mean squared error over the training set.

The simplest algorithm to use is gradient descent ("backpropagation") in which each weight is iteratively changed proportionately to its effect on the error, but more advanced methods such as conjugate gradient methods and sequential quadratic programming are being used increasingly.

Neural networks are attractive as automatic model-building tools because they can be proven, given enough nodes, to be able to represent any well-behaved function, with arbitrary nonlinear interactions between the inputs. They are also, besides being suitable for implementation on massively parallel computers, relatively robust to outliers and poor data. In short, they have been seen as a way of doing nonlinear model-building without the pain of learning statistics. They have one major disadvantage: due to the high degree of interaction and collinearity between the variables and basis functions, it is almost impossible to interpret the models.

2.3 Comparisons

MARS and neural networks have some obvious similarities and differences. Both are methods of deriving nonlinear models from data. Both require significant amounts of data to build a reasonable model, and do so by using models with (potentially) large numbers of parameters - sufficiently large numbers of parameters that the models may be considered to be nonparametric. The methods differ in that MARS is based on subset selection, while neural networks offer a nonlinear projection method.

MARS and neural networks also share a subtle but important similarity. Both can be viewed as methods which select basis functions adaptively, based on the data. When there are multiple independent variables, methods which construct basis functions based on the data can be shown to give more accurate models (for a given amount of data) than methods which use a fixed set of basis functions (Barron, 1992). Thus neural networks and MARS are more efficient than Fourier series expansions or higher order NARMA models.

To understand further how the two methods work, it is useful to look at a their performance on an example problem. MARS was compared with neural networks on modeling a nonlinear model of a chemical plant (a non-isothermal continuously stirred tank reactor) which has been used to test different nonlinear control schemes (Psichogios *et al.*, 1992). The reactor temperature, the concentrations of the two chemical species in the reactor, and the temperature of the feedstream to the reactor were used to predict the reactor temperature and concentrations one sample time in the future.

When trained with sufficient data (over 100 data points), both MARS and neural networks were found to be have substantially lower predictive error than linear regression. (Testing must, of course, be done on a separate validation data set since comparisons on the training data are not meaningful.) MARS was found to be much faster than neural networks and, for sufficiently large data sets, it also produced more accurate models.

MARS is, however, not universally superior to neural nets. When not all the state variables are accessible, for example if only the inlet temperature and one of the concentrations on the above reactor problem are available, an autoregressive moving average (ARMA) model must be constructed by using past measurements. For this time series problem, the accuracies for the neural network and for MARS were not significantly different. As will be explained below, we believe that this "failure" of MARS is due to the collinearity of the input data. MARS was also found to be more sensitive to outliers and high leverage points in the predictor space.

3 Multicollinearity

In linear regression,

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon, \tag{3}$$

severe estimation problems arise if the $x_1, \ldots, x_k$ are highly collinear. The problem is manifested through increased variance of the estimated parameters. Prediction, *per se*, of the response is unaffected, however, as the least squares fit will ensure that the residuals are still as small as possible. However, prediction of points not contained in the original data set may be affected due to the instability of the coefficients. The variance inflation factor (VIF) for each predictor is defined as

$$VIF = \frac{1}{1 - R_i^2} \tag{4}$$

where R_i^2 is the multiple correlation between x_i and the other predictors, $x_j (j \neq i)$. The VIF indicates how much the variance of the predicted coefficient β_i of equation 3 is inflated compared to orthogonal predictors. In highly collinear data sets, the VIF may be hundreds of times higher than an orthogonal set, rendering the interpretation of the coefficient meaningless.

A general approach to the problem is to consider the principal components of the predictor variables as predictors rather than the original variables. Usually, the predictor variables are taken to be standardized to have mean 0 and variance 1, so that principal components can be derived from the correlation rather than the covariance matrix. We shall assume that this is the case here. These new variables are orthogonal linear combinations of the original (standardized) predictors in decreasing order of variance. The idea of principal component regression is to use only the first k $(k < p)$ principal components in a regression equation of the form:

$$y = \alpha_0 + \alpha_1 z_1 + \ldots + \alpha_k z_k + \epsilon, \tag{5}$$

where z_i is the i^{th} principal component. It is then hoped that this will still retain good predictive behavior and that the z_i will be interpretable.

For an example of a case where principal components are natural, consider the temperature of a flow measured by six different devices at various places in a production process. Even though the inputs are highly correlated, a better prediction of the response may be gained by using a weighted combination of all six predictors rather than choosing the single best measurement. Variable subset selection attempts to circumvent the problem by choosing a small subset of the original p predictors which are not collinear, but still able to predict y to a reasonable degree of accuracy. However, as in the case of the "redundant" temperature measurements, there are cases when dropping a predictor out of the equation completely is not desired.

Neural networks and MARS deal with the problem of collinearity in very different ways. Due to its overparameterization, the coefficients or weights of a neural network are inherently difficult to interpret. However, it is this very redundancy that makes the individual weights unimportant. That is, at each level of the network, the inputs are linear combinations of the inputs of the previous level. The final output is a functions of very many combinations of sigmoidal functions involving high order interactions of the original predictors. Thus neural networks guard against the "problems" of multicollinearity at the expense of interpretability.

MARS, on the other hand, builds its set of basis functions via forward selection of the predictors, and at the end performs a backward selection to prune the number of selected basis functions. It is the forward selection procedure that makes MARS vulnerable to multicollinearity. If two predictor variables are both are correlated with the response, at some stage of the forward selection procedure MARS may be forced to choose between placing a knot on one of these predictors. The choice may be somewhat arbitrary if both result in roughly the same penalized residual sum of squares or gcv criterion value.

Unfortunately, the choice has potentially profound impact on the choice of all further variable and knot selections and thus on the final model as well. This is a result of the tree structure in MARS and the forward method by which it selects. In an extreme case, it may happen that the choice of one variable may be slightly better at the current step, but that a much better model would result if the other predictor had been chosen. As in any subset selection procedure, the interpretability of the final model when one correlated predictor is chosen over another is degraded as well.

3.1 Principal component MARS

In the case of linear regression, the use of principal components stabilizes the coefficients since the predictor variables are orthogonal. The problems of using a selection method are ameliorated by using orthogonal predictors. Multicollinearity in the predictors may make selection procedures for nonlinear methods such as MARS even worse than in the linear case. Friedman (1991) was aware of this problem and proposed two strategies for dealing with multicollinearity. First, he suggested fitting a series of increasing interaction order models, comparing their gcv scores and then selecting the model with the lowest interaction order with an acceptable fit. His second strategy was to invoke a penalty on introducing new variables into the model, so that the change of entering two highly collinear inputs would be lessened.

Unfortunately, neither of these strategies alleviates the problem of choosing between two inputs in a collinear setting. Merely blocking out one does not ensure optimality or even reasonableness of fit. Moreover, increasing the interaction order says nothing directly about the problem of choosing a sub-optimal predictor at an early stage.

We propose to first transform the predictor variables using principal components before using MARS. This preprocessing of the variables into orthogonal variables will alleviate the problem of selection among highly collinear inputs. While this is certainly not the only method for reducing collinearity, it serves as a generic way to relieve the multicollinearity of tree based subset selection procedures. We will discuss other possible strategies in section 5, but first show how the method can dramatically improve the resulting fit from MARS on data from a chemical processing production run.

4 Example

To illustrate the problem with MARS in dealing with multicollinearity, we will use a data set consisting of measurements taken every 4 hours from a polymer production run. The predictors consists of 18 "control" variables which the engineers have some ability to adjust during the run, and 5 "material" variables which describe properties of the raw material being introduced into the system. The response y is the viscosity of the material as measured by a surrogate of molecular weight. A great deal of multicollinearity is present as is evident in Table 1. The variance inflation factors range from a low of about 9 to a high of over 2000.

If we consider only the 18 control variables as our predictors, restrict the interaction order to 2, and use the default parameters settings of MARS, we find two curves (x_6, x_9), and two surfaces ($x_8 x_{17}$ and $x_{12} x_{17}$) using 11.5 degrees of freedom with a resulting R^2 value of 0.597. In an analogy with linear regression, one might expect that adding the

Control Predictors								
Number	Name	VIF	Number	Name	VIF	Number	Name	VIF
x_1	Upper feed flow	239	x_2	Lower feed flow	139	x_3	Pyro pressure	181
x_4	C02 flow	1119	x_5	Pyro pot temp	11	x_6	Mix flow rate	168
x_7	Brine temp	33	x_8	Hot purge rate	31	x_9	Poly pressure	49
x_{10}	Poly level	16	x_{11}	Spray flow rate	45	x_{12}	Spray temp	258
x_{13}	Spray cat feed	550	x_{14}	Suction leg level	31	x_{15}	MEOH inj rate	80
x_{16}	Swirl temp	166	x_{17}	Swirl pump amps	40	x_{18}	Swirl cat feed	2156
Material Predictors								
Number	Name	VIF	Number	Name	VIF	Number	Name	VIF
x_{19}	H20	31	x_{20}	HCOOH	25	x_{21}	SALT	11
x_{22}	MEOH	23	x_{23}	CH2O	9			

Table 1: Predictors and their variance inflation factors

5 additional material variables will result in a model which may increase the number of degrees of freedom used, with the benefit of an increase in R^2. However, the introduction of these new predictors, results in a model using only one surface ($x_8 x_{17}$) with 4.0 degrees of freedom and an R^2 of 0.359. It seems that not only have the 5 new input variables not appeared in the new model, but their introduction has changed the stepwise procedure significantly.

The output from the MARS model shows what has happened. The first output shows the knot placement with only the control variables:

```
forward stepwise knot placement:
 basfn(s)    gcv      variable      knot   parent
    0       313.3
    1       335.7       17.        92.41     0.
   3 2      363.2        8.        3.149     1.
   5 4      472.7        9.        6.848     0.
   7 6      749.5        6.        17.24     0.
   9 8      1278.       12.        50.84     1.
  11 10     6603.        9.        6.813     7.
```

$R^2 = 0.597$.

The second shows the placement for the full model with all 23 predictors:

```
forward stepwise knot placement:
 basfn(s)    gcv      variable       knot   parent
    0       313.3
    1       335.7       17.         92.41     0.
   3 2      377.6        8.         3.155     1.
   5 4      522.2       10.         42.80     0.
   7 6      767.5       22.        0.1500     0.
    8       1409.       14.         77.26     5.
    9       3365.        9.         5.742     5.
```

$R^2 = 0.359$.

At the third knot placement, MARS has to choose between two relatively close alternatives (in terms of the gcv), placing a knot on variable x_9 or x_{10}. These result in very different future choices. In the second case, this knot generates two daughters which are eventually pruned out, as is the knot itself, resulting in a model with only x_8 and x_{17}. However, if variable x_9 is chosen, the future choices remain after the pruning, and result in a four variable model with a much better fit, as measured by R^2.

To help alleviate the problem of knot placement in the presence of multicollinearity, we use the linear principal components of the predictors rather than the original variables. Rather than choose the first k components, we allow MARS to select among all 23 principal components. The knot placement is shown below:

```
forward stepwise knot placement:
 basfn(s)    gcv      variable        knot      parent
    0       313.3
   2 1      343.3        8.          0.3229       0.
    3       354.2        2.          -3.074       1.
    4       358.9        5.          -3.210       2.
    5       365.6       23.       -0.2505E-01     1.
   7 6      595.2       21.       -0.1507E-01     0.
   9 8      1948.        1.          -1.559       0.
```

$R^2 = 0.784$.

The last two pairs of basis functions are pruned, resulting in model with three surfaces: $(x_2 x_8)$, $(x_5 x_8)$ and $(x_{23} x_8)$. The loading vectors (or weights) for the 4 principal components chosen are shown in Table 2.

Predictor	PC2	PC5	PC8	PC23	Predictor	PC2	PC5	PC8	PC23
Upper feed flow	-0.18	0.34	0.20	-0.04	Lower feed flow	-0.02	0.10	-0.05	0.07
Pyro press	-0.30	0.07	-0.07	0.00	C02 flow	0.05	0.14	-0.03	-0.52
Pyro pot temp	-0.22	-0.08	-0.09	0.01	Mix flow rate	0.16	-0.29	-0.16	-0.04
Brine temp	-0.30	-0.14	-0.35	-0.04	Hot purge rate	-0.18	-0.16	0.11	0.02
Poly press	-0.07	0.08	-0.08	0.02	Poly level	0.07	0.27	-0.20	-0.04
Spray flow rate	-0.13	0.03	0.15	-0.03	Spray temp	-0.15	-0.05	0.07	0.12
Spray cat feed	-0.17	0.13	0.08	-0.36	Suction leg level	-0.17	-0.33	0.35	-0.02
ME0H inj rate	0.24	0.09	-0.08	-0.04	Swirl temp	-0.38	0.15	0.10	-0.10
Swirl pump amps	-0.38	0.11	-0.08	0.02	Swirl cat feed	-0.06	0.17	0.01	0.75
H20	-0.10	-0.09	0.03	0.02	HCOOH	-0.02	-0.59	-0.31	-0.01
SALT	-0.39	0.04	-0.45	0.00	MEOH	-0.13	-0.13	0.09	-0.02
CH2O	-0.19	-0.24	0.50	-0.01					

Table 2: Principal component loadings for chosen directions

The predictor involved in all three surfaces, PC_8 is made up (essentially) of three material variables (HCOOH, SALT and CH_2O), Brine temperature (related to SALT) and the suction leg level. The next most important variable, PC_{23} is essentially three control variables, the swirl catalyst feed rate, the spray catalyst feed rate and the CO_2 flow. PC_2 involves SALT, two variables involving swirl, brine temperature and pyro pressure. Finally, PC_5 is made up of HCOOH, two feed flows and suction leg level.

We have used PCA and MARS to select a small number of linear combinations of variables for non parametric fitting. The technique is not specific to principal components or MARS, but to any dimension reduction technique followed by subset selection and then nonparametric fitting. As an example, we could fit a loess (Chambers and Hastie, 1992) surface to the viscosity using the three combinations (PC_2,PC_8), (PC_5,PC_8) and (PC_8,PC_{23}). This model gave an R^2 value of 0.92. Undoubtedly, MARS could be fine-tuned to produce a better model than the one illustrated above. The point, however, is that MARS automatically chooses the surfaces or curves to fit using subset selection. Trying all single and pairwise combinations in a generalized additive model setting is computationally infeasible. The resulting fitting is a relatively minor matter after the relevant variables have been selected.

5 Discussion and future work

The results presented above show that in cases where there is high collinearity among the predictors, it may be important to linearly project them with a method such as PCA before using a nonlinear model selection method such as MARS. Since nonlinear projection techniques such as neural networks include such linear combination, the same preprocessing will have no effect on their performance.

For linear problems where the original coordinate system is a meaningful one, linear subset selection methods such as stepwise regression work well. If there is a high degree of correlation between the predictor variables, and significant noise in them projection methods such as PCA or PLS may be warranted. If the problem is nonlinear, and a high degree of correlation between predictors exists, neural networks or MARS with a PCA front end will tend to outperform MARS alone.

For nonlinear situations, knowing whether to preprocess need not be obvious. Spectrographic data from IR, NMR and similar methods have very highly correlated values, and PCA and PLS methods have proven fruitful for linear analysis, but a combined PLS/MARS method works less well than MARS on the original variables (Sekulic and Kokalski, 1992b). Why is this? Interactions which occur between individual predictor variables (*e.g.* the strength of given spectrographic peaks) may be masked when interactions between the principal components are chosen, if these are chosen on a linear basis.

The proceeding section of this paper has shown that there are important cases where nonlinear subset selection (*e.g.* MARS) does work well on the principal components. In this case, the results may be more interpretable than those provided by methods such as neural networks which do not involve subset selection. The required computation is also much lower with MARS, and extensive cross validation is not required to avoid overfitting or excessively complex (and hence high variance) models.

In the examples presented in this paper, PCA was used as the projection method to be combined with MARS as the nonlinear model/selection method. It is also possible to use other projection or nonlinear modeling methods. An attractive projection method is PLS, which differs from PCA in that outputs as well as inputs are used in picking the principal directions. (see also Wold *et al.* 1989, Holcomb and Morari, 1992 and Qin and McAvoy, 1992). This complicates interpretation of the model, but has the advantage that one need only keep the dominant "principal components" as (unlike the PCA) they account contain combinations of the inputs which have the largest effect on the outputs. Since PLS uses the output variables and needs an assumed form of relationship between the inputs and outputs (generally, but not necessarily linear), using PLS complicates the algorithm for combining the projection method with the nonlinear model. An optimal method, therefore, requires an iterative scheme in which one first picks a projection for a given form of inner relationship (initially linear) between the projection weights and the outputs and then finds an inner relationship for the projection weights as found in the first step. The new inner relationship requires that the first step be performed again, so the method iterates until convergence.

Many of the methods discussed in this paper are relatively new, and the collective wisdom on their overall attributes and performance in a variety of settings is thin. We have attempted to provide a framework for the possibility of combining some of the features of these tools, and have indicated that some dramatic improvement is possible. Much work remains to be done.

References

[1] Barron, A. R. "Approximation and Estimation Bounds for Artificial Neural Networks.", in press, *Machine Learning* (1992).

[2] Chambers, J. M. and Hastie, T. J., "Statistical Models in S", Wadsworth & Brooks/Cole, Pacific Grove, California, 1992.

[3] De Veaux, R. D., Psichogios, D. C., and Ungar, L. H. "A Comparison of Two Non-Parametric Estimation Schemes: MARS and Neural Networks." *Comp. Chem. Engng.* **17:8** (1993) 819-837.

[4] Friedman, J. H. "Multivariate adaptive regression splines." *The Annals of Statistics* **19:1** (1991) 1-141.

[5] Holcomb, T. R., and Morari, M. "PLS/Neural Networks." *Comp. Chem. Engng.* **16:4**, (1992) 393-411.

[6] Psichogios, D. C., De Veaux, R. D., and Ungar, L. H. "Non-Parametric System Identification: A Comparison of MARS and Neural Networks." *ACC* **TA4** (1992) 1436-1440.

[7] Qin, S. J., and McAvoy, T. J. "Nonlinear PLS Modeling Using Neural Networks." *Comp. Chem. Engng.* **16:4**, (1992) 379-391.

[8] Ripley, B. D. "Statistical Aspects of Neural Networks." Invited lectures for SemStat, Sandbjerg, Denmark, 25-30 April 1992.

[9] Rumelhart, D., Hinton, G., and Williams, R. "Learning Internal Representations by Error Propagation." *Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol 1: Foundations* Cambridge: MIT Press (1986), 318-362.

[10] Sekulic, S., and Kowalski, B. R. "MARS: A Tutorial." *J. Chemometrics* **6**, (1992).

[11] Sekulic, S., and Kowalski, B. R. "Nonlinear Multivariate Calibration Methods Combined with Dimensionality Reduction." submitted to *J. Chemometrics* (1992).

[12] Wold, S., Kettaneh-Wold, N., and Skagerberg, B. "Nonlinear PLS Modeling." *Chemometrics and Intelligent Laboratory Systems*, **7** (1989) 53-65.