Variable Selection for Response Surface Modeling Using R-Splines

Sarah W. HARDY
North Carolina State University
Raleigh, NC 27695-8203

Douglas W. NYCHKA
National Center for Atmospheric Research
Boulder, CO 80307

Dennis D. BOOS
North Carolina State University
Raleigh, NC 27695-8203

Perry D. HAALAND
Becton Dickinson Technologies
RTP, NC 27709

Motivated by an application with a moderate number of potential explanatory variables, this paper explores a technique for variable and model selection using R-splines. R-splines are a recently proposed extension to thin plate splines with a modification to the roughness penalty that allows for a reduced polynomial component to be fit. The key model selection idea is a two-stage approach. First, the important explanatory variables are identified using a specific type of R-spline. Then these variables are used to fit different R-spline models from which the most desirable is chosen. This new method is then compared to all subset regressions by leaps and bounds and regression trees. An application of the methodology is also discussed.

KEY WORDS: Thin plate spline, nonparametric, response surface, roughness penalty, leaps and bounds all subset regressions, regression trees.

1. INTRODUCTION

This paper focuses on a nonparametric method for fitting response surfaces known as *R-splines* (Hardy and Nychka, 2000) which are a recently proposed extension to *thin plate splines* (Wahba, 1990, Silverman

© 1999? American Statistical Association
and the American Society for Quality Control
TECHNOMETRICS, Month? 1999, VOL. , NO.

and Green, 1994). This research was motivated by a specific application with twelve potential independent variables. If the functional form(s) of the variables must be designated, for example, x, x^2, x^3, e^x, $ln(x)$, etc., along with interactions between these functional forms, the number of models that could be fit is overwhelming even with relatively small datasets by current standards. Although many model selection techniques have been suggested, there is still no consensus on the superiority of any one specific technique. We will address the issue of variable and model selection using R-splines and make comparisons with some of the existing technologies.

The unique modeling approach proposed in this research is separating the variable selection step from the model selection step. Generally models are selected by optimizing some criterion, and the variables selected are simply those that appear in the selected model. Variable selection and model selection are intimately related, and so the difference is somewhat subtle. It was observed, however, in preliminary simulations that optimizing criterion for judging the model did not satisfactorily select the correct variables. This led to the conclusion that the best results in model fitting with splines could be achieved by first selecting the variables, and then choosing a model from among the different models that could be fit with those variables by varying two model parameters. These model parameters are m and $cost$, where $m-1$ is the degree of the polynomial null model and $cost$ weights the model degrees of freedom in the denominator of the generalized cross-validation function. To address variable and model selection, two sets of simulations were performed. The goal of the first was to examine differences among four strategies of selecting a variable subset using R-splines from among all possible subsets. The goal of the second was to compare three modeling techniques: a two stage R-spline approach using all subset variable selection methodology, standard all subset linear regressions by leaps and bounds, and classification regression trees.

In these simulations, the R-splines methodology using an initial all subsets variable selection method (stage 1), and then optimizing the fit over the model parameters m and $cost$ (stage 2), consistently outperformed the competing regression tree and all subsets regression methods both in terms of variable selection and predictive power. This R-spline methodology was also successful in a larger scale, more complex, practical problem.

2. THIN PLATE SPLINES AND R-SPLINES

A thin plate spline is a smooth surface consisting of a standard polynomial component plus a non-

parametric component that is the sum of radial basis functions. The thin plate smoothing splines are a solution to a minimization problem and can be interpreted as a natural generalization of the traditional polynomial model estimated by minimizing residual sum of squares. Unlike the traditional formulation, however, the thin plate spline minimization problem also involves a roughness penalty. This roughness penalty is a function of m and d where where $m-1$ is the degree of the polynomial component and d is the number of explanatory variables. To guarantee that the roughness penalty is positive and that the function is a true thin plate spline, $2m-d$ must be positive. The number of polynomial terms needed is $\binom{d+m-1}{d}$. Thus, the size of the polynomial component grows very quickly as a function of the number of explanatory variables. For example, if there are $d=5$ explanatory variables, $m \geq 3$; for $m=3$, a full quadratic polynomial component with 21 terms must be fit. If there are $d=6$ explanatory variables, 84 polynomial terms are needed. Many typical industrial experiments do not have enough observations to fit the number of parameters needed for the required polynomial component. Specifically, this becomes problematic when there is a large set of explanatory variables and it is desirable to use an all possible subset approach to variable selection for thin plate spline modeling.

R-splines arose in the context of attempting to modify the thin plate spline to allow for a greater number of explanatory variables in the model. R-splines are splines fit with a polynomial component plus the sum of radial basis functions. Clearly, thin plate splines fall into this category. This broader definition of an R-spline, however, also includes other splines. Specifically, it includes splines in which $2m-d \leq 0$. In other words, the polynomial component may be of a lower order than required by the thin plate spline restraint. When $2m-d \leq 0$, the seamless way in which the polynomial function and roughness penalty fit together (i.e, the space spanned by the polynomial component is the null space of the roughness penalty) no longer holds. Since the relationship is thus "broken," the term *broken* spline will be used to indicate this kind of spline. The term R-splines refers to the broader class of splines including both broken and standard thin plate splines. A more detailed discussion of thin plate and broken splines can be found in Appendix A.

3. SIMULATION OVERVIEW

Two simulation studies were performed. The first examined four strategies of selecting a variable subset using R-splines, i.e., stage 1 of the two-stage approach. The second simulation compared models

using the two-stage R-spline model selection approach to two other modeling methods. For both sets of simulations, three different sets of independent variables, or X matrices, each with 7 columns, i.e., 7 potential explanatory variables, and 100 observations were used. The first two X matrices were created by randomly selecting the points, and the third was a space-filling design created using the FUNFITS (Nychka, et al, 1996) function *cover.design*.

Also for both sets of simulations, the three following test functions were tested:

$$\begin{aligned} f_1(\mathbf{x}) &= s(2x_1) + s(exp(5x_2)) + s(sin(2\pi x_3)) + s(3x_1x_4) \\ f_2(\mathbf{x}) &= s\left[N(x_1;.3,.3)N(x_2;.3,.3) + 1.25N(x_1;.7,.2)N(x_2;.7,.2)\right] + s(3x_3 + x_4) \\ f_3(\mathbf{x}) &= 2x_1 + x_2^2 + 3x_3x_4 \end{aligned}$$

where

$$\begin{aligned} s(x) &= \frac{x - min(x)}{max(x) - min(x)} \\ N(x;\mu,\sigma) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{x-\mu}{\sigma}\right)^2} \end{aligned} \tag{1}$$

All test functions are functions of only x_1, x_2, x_3 and x_4; the last three variables in the X matrices are independent of the test functions. The primary goal of the first set of simulations was to compare the ability of different strategies to select the correct set of variables from a group including spurious variables. The first two functions were selected for their complexity and high degree of non-linearity. These are the types of functions where polynomial models would not be expected to perform well. The notation $s()$ indicates that the term enclosed in the parentheses was scaled between 0 and 1. This was done to equalize the impact of the various terms in the function. The notation $N(x;\mu,\sigma)$ indicates the normal probability density function. The third function is linear in simple polynomial terms. This function was included because it is one in which all subset regressions by leaps and bounds would be expected to perform well.

In both sets of simulations, all nine combinations of X matrices and functions were considered at four different levels of normally distributed random error. These different levels of random error standard deviation, σ, are 1%, 10%, 20%, and 30% of the range of the true function. In the application motivating

this research, within group standard deviations were typically observed to be approximately 22% of the range of the observed responses.

4. VARIABLE SELECTION SIMULATION

4.1 Variable selection simulation criteria

The first stage of the two-stage R-spline modeling approach is to use R-spline models to select a variable subset. In considering different R-spline models there are various criteria that can be examined to judge the success of the model, or in this context the variable subset selected. The purpose of this simulation was to compare three of these criteria: 1) min(GCV), the minimum of the cross-validation function that estimates the average leave-one-out-prediction error (and is used to find λ, the weight given to the roughness penalty); 2) R_p^2, also known as $R^2{}_{prediction}$ (see Appendix D); 3) the root mean squared error; and 4) R^2, the percentage of variation explained by the model. R^2 does not equal the squared correlation between the observed and expected values as it does in traditional linear models. Models that interpolate are generally not useful; so models with degrees of freedom greater than $\frac{2}{3}$ the number of observations (in this case 67) were excluded from consideration. Two other model parameters with an effect on the model fit were also varied. These were 1) $m = 1, 2$, where $m - 1$ is the degree of the polynomial component, and 2) $cost = 1, 2$, which weights the model degrees of freedom in the denominator of the generalized cross-validation function. A *cost* of greater than 1 will tend to discourage over-fitting. In summary, for all combinations of three X matrices, three functions, four levels of random error, $m = 1, 2$, and $cost = 1, 2$, the four criteria were compared in models produced by 100 simulations.

In each one of these sets of simulations, the three criterion were computed for all possible subsets of the seven variables (with the exception of the one variable and constant models), i.e., $2^7 - 7 - 1 = 120$ subsets. For $m = 1$, all the R-spline fits for the subsets are broken splines, as opposed to true thin plate splines, and for m=2 they are broken splines for all subsets containing four or more variables.

4.2 Variable selection results

The variable selection simulation data was tabulated in terms of an all or nothing binary response; the correct set of variables was either chosen or not chosen. The results of the simulations performed at

the 1% level of random error were rather peculiar and not very illustrative, as what most often occurred in the set of 100 simulations was that the *same* variable set was chosen each time, so the correct set was either chosen 100 times or 0 times. Results from simulations at the higher levels of random error showed more discernible differences.

Table 1 contains the mean number of times the correct variable set was chosen for the three criterion with the different functions and m and $cost$ settings. The values are averaged over the three datasets and over the 10%, 20%, and 30% levels of random error. The combination of m and $cost$ which was most successful at choosing the correct variable set is highlighted in each row. The most striking feature of these results, which was common across across all test functions at all levels of random error, was the advantage provided by setting $cost = 2$. Also, when $cost = 2$, setting $m = 1$ led to more successful variable selection. This is an interesting result because with $m = 1$ all variable subsets are fit with broken splines (with $m = 2$ variable subsets containing 2 or 3 variables are fit with true thin plate splines).

Table 1. Comparison of number of correct variable sets chosen in 100 simulations

$Function$	$Criterion$	$cost = 1$		$cost = 2$	
		$m = 1$	$m = 2$	$m = 1$	$m = 2$
1	$min(GCV)$	36.3	39.3	**71.9**	48.9
	R_p^2	33.9	37.9	**70.9**	48.4
	$Rmse$	5.3	6.9	**59.4**	33.7
2	$min(GCV)$	24.7	26.4	33.2	**36.6**
	R_p^2	25.0	27.4	**44.8**	39.0
	$Rmse$	5.0	5.1	**43.2**	42.3
3	$min(GCV)$	46.9	48.4	**87.4**	69.7
	R_p^2	46.2	49.2	**83.3**	70.2
	$Rmse$	0.9	1.9	**76.3**	53.2

Table 2 shows the results above where $cost = 2$ and $m = 1$ are averaged over the three functions. The criterion with the highest mean number of times selecting the correct variable subset when $cost = 2$ and $m = 1$ was R_p^2 with a mean of 66.0. The differences between criteria was not, however, statistically significant.

Table 2. Mean number correct where cost=2 and m=1.

Criterion	*Mean*
$min(GCV)$	64.2
R_p^2	66.0
$Rmse$	59.7

It is somewhat surprising that min(GCV) did not compare more favorably, as it often was the most successful criterion. While the min(GCV) criterion did the best with Functions 1 and 3, it did not do well with Function 2. Function 2 was the most complex function and all the criteria led to choosing smaller variable subsets than for the other functions. The min(GCV) criterion consistently selected smaller variable subsets than the other criteria across all functions. Use of the min(GCV) criterion for Function 2 typically resulted in under-fitting. For Functions 1 and 3, however, the other criteria tended more toward over-fitting.

4.3 Variable selection summary

The all variable subsets approach to variable selection is practical when using R-splines because the functional form of the variables does not have to be specified. Weighting the model degrees of freedom by a factor of two in the denominator of the generalized cross validation function when choosing an optimal smoothing parameter, i.e., setting $cost = 2$, and choosing best subset from amongst all possible subsets is a highly effective way to eliminate extraneous variables from the model. Limiting the polynomial component to a constant term, i.e., $m = 1$, also provides an advantage when choosing a variable subset. The choice of a criterion from among R_p^2, root mean squared error, and min(GCV) was not a significant factor in choosing a variable subset when $cost = 2$ and $m = 1$. A modeler could consider all the subsets indicated as "best" by any of the criteria. Once a subset is selected, the modeler should vary m and $cost$ again to choose the final model as addressed in the following section.

5. MODEL SELECTION SIMULATIONS

5.1 Model selection approaches

The second stage of our model selection approach involves fitting an R-spline model to the variables selected in the first stage. Here we fit R-spline models with different orders of polynomial components ($m = 1$ indicates a constant polynomial component, $m = 2$ indicates a linear polynomial component) and unweighted ($cost = 1$) or weighted ($cost = 2$) model degrees of freedom in the denomimator of the generalized cross validation (GCV) function. This GCV function is used to select the optimal smoothing parameter, i.e, the weight given to the roughness penalty. As this weight increases, the R-spline fit will

become more smooth. Specifically, in this set of simulations, R-spline models were selected from those produced by the four combinations of $m = 1, 2$ and $cost = 1, 2$.

The goal of this set of simulations was to compare models selected by three different modeling techniques: 1) the two-stage R-spline approach, 2) all subset regressions by leaps and bounds (see Appendix B), and 3) regression trees. The quality of the models selected was measured both in terms of their success at selecting the correct model terms and their ability to predict the true function values on a prediction dataset.

In addition to selecting the R-spline model from those produced by the four combinations of $m = 1, 2$ and $cost = 1, 2$, we also considered R-spline models where *cost* was forced to be 2 and only m was varied. These were looked at to see if the less flexible models, which are likely to produce lower criterion values, would be more successful at predicting on the prediction set.

The leaps and bounds approach (Furnival, 1974) uses an efficient algorithm to find the best subset of the given explanatory variables for a traditional linear regression. These simulations used the S-Plus implementation of the algorithm in the function *leaps* to find the best subset for each subset size. Among these subsets, the one that was optimal according the Mallows C_p or the adjusted R^2 criterion was chosen (see Appendix C). For standard regression subset selection the functional form of the variable must be specified. In other words, x_1 is one potential variable, and x_1^2 is another. The S-PLUS leaps function is limited to building subsets from an initial set of 30 variables (or 31 without an intercept). Unfortunately, a full quadratic model in 7 variables has $7 + 7 + \binom{7}{2} = 35$ terms, 5 more than allowed. Thus, the interaction terms x_4x_5, x_4x_6, x_4x_7, x_5x_6, and x_6x_7 were omitted from model selection. All interaction terms containing two of the variables actually in the true functions, and most containing one are included in the model selection. This gave the leaps method some slight advantage over what would be had in practical situation with no *a priori* knowledge of the true underlying function. With Functions 1 and 2, which are highly nonlinear, what is being tested is how well the polynomial function can approximate the true function. Function 3, however, contains only polynomial terms included in the set for model selection.

Both R-splines and regression trees have the advantage that the functional form of the variable does not have to be specified, hence vastly reducing the possible number of models that can be fit. Regression trees actually perform variable selection as a part of their algorithm for fitting the data by searching for

the variable that explains the greatest deviance in the data. The advantage of R-splines, as compared to trees, is that R-splines fit smooth functions to the data, whereas tree models are multidimensional step functions. The advantage of tree models is that both discrete and continuous variables are easily handled, whereas splines are most suitable for continuous variables. These simulations only address the case where all explanatory variables are continuous. These simulations used the S-Plus function *tree* (Chambers and Hastie, 1992) to build the regression trees. Arguments that control the tree size were left at their default settings as follows: 1) growing continues if there are at least 5 observations in a node, and 2) 0.01 is the minimum node deviance before growing stops. Because the number of nodes is intentionally increased until the tree cannot "grow" anymore, some degree of over-fitting can generally be expected. For tree models, there are two approaches to reducing over-fitting. The first is "pruning" where nodes with the least important splits are recursively "snipped." The second approach is "shrinking" in which estimates for observations in lower nodes are "shrunk" to their parent nodes based upon the magnitude of the difference between the fitted values of the lower nodes and the fitted values of their parent nodes. Because nodes are actually removed, pruning leads to a more parsimonious description of the data, while shrinking leads to more accurate prediction. The optimal degree of pruning or shrinking is somewhat arbitrarily chosen by examining plots of the deviance versus the pruning or shrinkage parameters (which can be mapped to the size or effective size of the tree) and determining at which point the reduction in deviance no longer justifies the increased complexity of the tree. These arbitrary decisions, however, cannot be made in a simulation setting. To gain understanding of the general effect of pruning and shrinking, in addition to simulations producing the full tree, as determined by the default values previously described, two additional simulations were also run. These were simulations in which: 1) the shrinkage parameter was set at 70%, and 2) three nodes were pruned.

Specifically, the approaches being compared in these simulations are: 1) R-splines using variables indicated by all subset variable selection with $cost = 2$ and $m = 1$ using the R_p^2 criterion, and then choosing an R-spline model that optimizes R_p^2 over the four combinations of $cost = 1, 2$ and $m = 1, 2$; 2) R-splines using same variable selection method, but fixing *cost* at 2 and then choosing the model that optimizes R_p^2 over $m = 1, 2$; 3) all subset regressions by leaps and bounds using Mallows C_p as a criterion; 4) the same as 3) except using adjusted R^2 as a criterion; 5) a full regression tree; 6) a tree

with 70% shrinkage; and 7) a tree with 3 nodes pruned. The two areas addressed in this section are: 1) a comparison of how successful the methods were at selecting the correct variable set; and 2) a comparison of how well the models selected by the different methods were able to predict true function values on a validation dataset.

5.2 Comparison of variable set selection

Comparisons were made between the R-spline method and the tree methods in terms of the binary response, the correct set of variables was either chosen or it was not chosen. This type of comparison could not be made with the all subset regressions methods because it selects functional forms of the variables. Table 3 shows the number of correct variable sets chosen by each modeling method by function, dataset and random error level. The number of simulations in which the correct variable set was chosen the most times is highlighted in each row. For example, in Table 3, with Function 1, Dataset "a", at the 1% level of random error, the R-spline method did the best choosing the correct variable set all 100 times. The default and pruned regression tree methods chose the correct variable set 66 and 67 times respectively.

For all three functions, at all levels of random error, the R-spline method did spectacularly better than the tree method in terms of choosing the correct variable set. Full regression trees consistently choose more variables than necessary, thus over-fitting the data. Pruning, even this routine fashion, provided an immense improvement over the full tree in terms of choosing the correct variables. Clearly with human intervention, this improvement would be even greater. Note that shrinking a tree does not delete any variables from the full tree, and so is not addressed separately in this section.

5.3 Comparison of predictive power

Figure 1 shows the boxplots of root mean predicted squared error using the models from the simulations to predict values on a prediction set. The prediction set consisted of 400 observations across the 7 variables chosen using a space-filling design. Root mean predicted squared error was based on differences between the predicted values and the true function values at these points. Specifically it was calculated by

$$\frac{1}{400}\sum_{i=1}^{400}(\hat{f}(\mathbf{x}_i) - f(\mathbf{x}_i))^2.$$

Table 3. Comparison of number of correct variable sets chosen in 100 simulations for Functions 1 and 2

Situation			*ModelingMethods*		
Function	*Dataset*	*Error%*	*R − Spline*	*Tree*	*Prune*
1	*a*	1	**100**	66	77
		10	**88**	9	31
		20	**72**	4	8
		30	**42**	0	2
	b	1	**100**	22	72
		10	**95**	4	16
		20	**66**	1	3
		30	**34**	0	1
	c	1	**100**	**100**	**100**
		10	**96**	10	30
		20	**82**	3	6
		30	**55**	0	2
2	*a*	1	**100**	0	0
		10	**67**	0	0
		20	**39**	0	1
		30	**14**	1	2
	b	1	**100**	0	0
		10	**88**	0	3
		20	**59**	0	0
		30	**24**	0	0
	c	1	**97**	0	0
		10	**52**	2	4
		20	**36**	1	2
		30	**24**	0	2
3	*a*	1	**100**	64	63
		10	**96**	10	25
		20	**82**	2	7
		30	**51**	0	0
	b	1	**100**	8	78
		10	**99**	10	21
		20	**91**	0	3
		30	**63**	0	0
	c	1	**100**	**100**	**100**
		10	**100**	15	51
		20	**94**	2	12
		30	**74**	0	2

For each combination of function, dataset, and level of random error, 100 simulations were performed. There were no clear differences between the simulation results for the different datasets; so, results in the boxplots have been combined over the three datasets. Results from each function appear in a column, with the level of random error increasing as the column goes down.

At the lowest level of random error, where the standard deviation of the random normal error is 1% the range of the true function, the all subset regressions methods did approximately as well as the R-splines in prediction on the prediction set in the two non-linear functions and slightly better than R-splines for the polynomial function. They lose this advantage, however, at the 10% level, even for the polynomial function. The difference becomes more distinct as the level of random error increases. This

pattern is observed across all three datasets. Both variants of all subset regressions, i.e., use of C_p and R_a^2 as criterion, perform approximately the same in all cases. Both the R-splines and the all subsets regression consistently outperform the regression tree methods. Shrinking and pruning both consistently improved the predictive power of the regression tree, but only by a small amount. Shrinking improved the predictive power slightly more than pruning. Restricting the R-splines to using $cost = 2$ did not improve the predictive power, and in some cases made it worse.

6. RECOMMENDATIONS

Based on the results of these simulations, the recommendation for model fitting problems involving variable selection with continuous variables, is to use the two-stage R-spline approach. In the first stage a variable set is selected based on comparing the R_p^2 criterion from R-spline models generated with the model parameters $m = 1$ and $cost = 2$ from all possible variable subsets. The criterion is not as important as the settings, and the modeler may also wish to look at variable subsets indicated as best by other criterion. Given this variable set, the second stage is to vary m and $cost$ and select as a final R-spline model that which optimizes R_p^2. Other criterion such as min(GCV) are also candidates for use in the second stage of model selection.

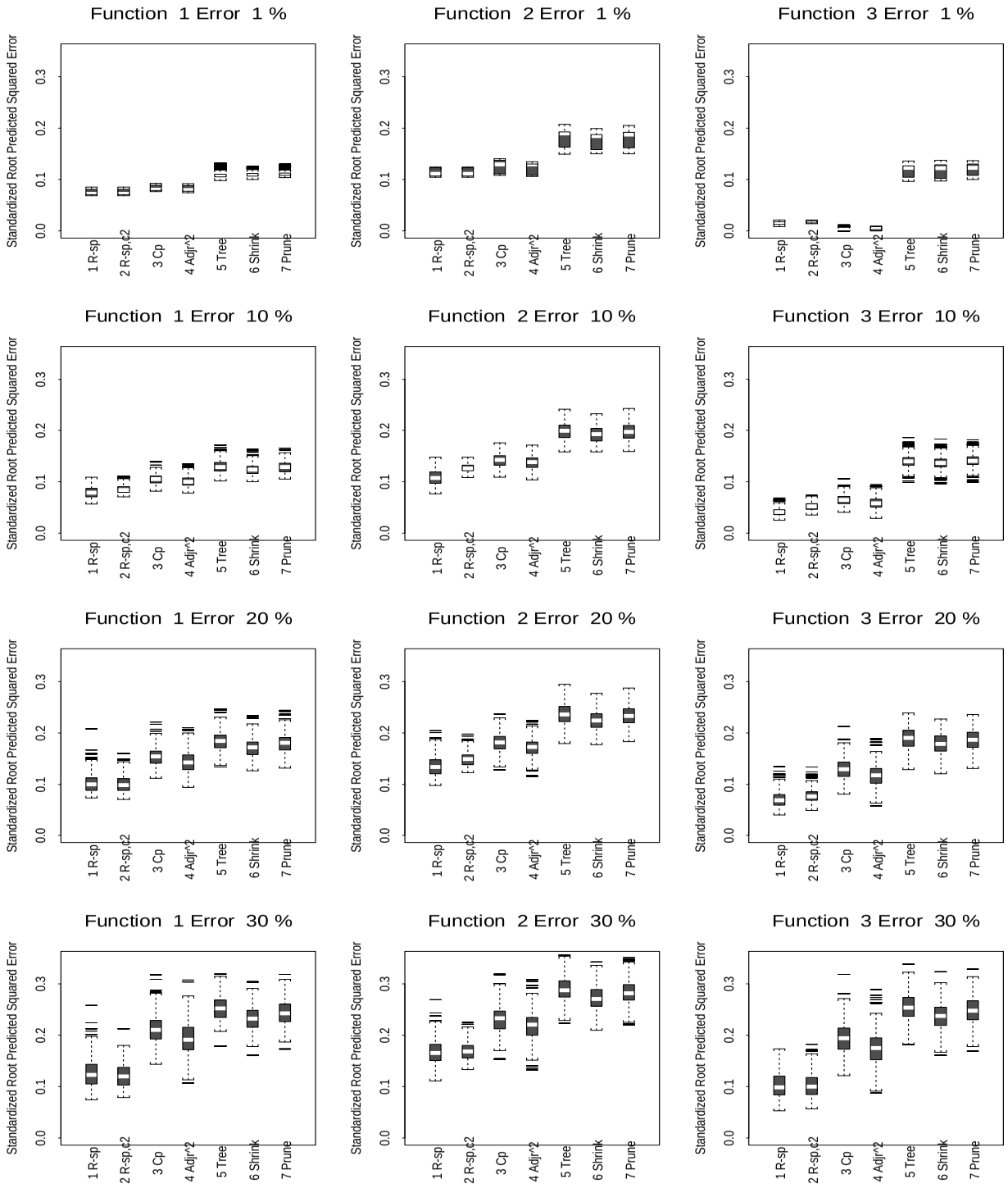

Figure 1. Comparison of distributions of the root mean predicted squared error for the different modeling methods.

7. APPLICATION

The data presented in this section arose from research being conducted at Becton Dickinson Technologies. There is currently a patent pending regarding this research, and unfortunately because of the proprietary nature of the data, the details cannot be discussed. This application involves the modeling of a response for which there are many potential explanatory variables with a high degree of multicollinearity.

A modeling set was selected that consisted of 104 compounds that had been replicated 2 to 15 times each. Of these compounds, 18 had within compound standard deviations of the response that were greater than 2.5. These large standard deviations indicate that at least one of the measurements in each of these compound groups is an outlier. With small numbers of replicates, however, it is not possible to determine which observation is an outlier. For this reason, these 18 compounds were discarded from the modeling group. A set of 12 measurable properties were considered as potential explanatory variables and the goal was to find properties that maximized the response. Two of the 86 compounds were outliers in several of these properties, and so were also omitted because of their potential to be highly influential. Mean responses by compound were aggregated before modeling. Thus, $n = 84$, and the dimension of the variable subsets is at most 12. Since models with only one variable were not considered, $2 \leq d \leq 12$.

An all possible subsets R-spline analysis was conducted using R_p^2 as a criteria for variable selection and fixing $m = 1$ and $cost = 2$. Models with only one variable and the model with only a constant term were omitted from the subset selection analysis. The number of subsets considered was $2^{12} - 12 - 1 = 4083$. This effort, run in batch, took approximately 12 hours on a Sun workstation. The subset selected contained explanatory variables x_4, x_6, and x_{10}. Using these three variables, R-splines with all 6 combinations of $m = 1, 2, 3$ and $cost = 1, 2$ were fit. Note that at this point it is feasible to look at $m = 3$ because with the field of explanatory variables narrowed down to 3, there are only ten terms in the full quadratic polynomial model. Of these 6 trials, the one with the highest R_p^2 was the model using $m = 1$ and $cost = 1$.The generalized cross-prediction function was minimized, indicating that an optimal smoothness was achieved. It appears that this broken spline fit the data better than the true thin plate spline.

Figure 2 contains surface plots that show there are interactions between all three variables. At lower values of x_{10} the optimum can be found with high values of x_4 and low values of x_6. Specifically, the optimimum occurs at $x_4 = 3.1$, $x_6 = -7.5$, and $x_{10} = .6$, and can be seen in the upper left hand plot in Figure 2. As x_{10} increases, the optimum decreases and x_4 has less of an effect.

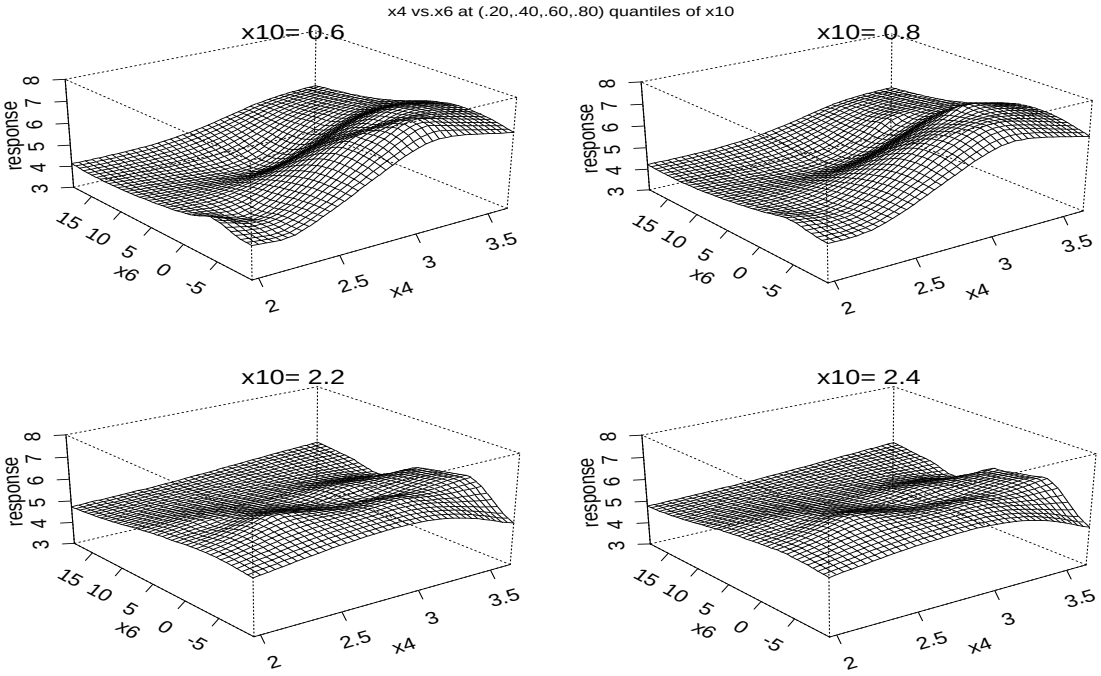

Figure 2. Surface plots of x_4 vs x_6 at 4 levels of x_{10}.

Fitting a leaps model was problematic because the S-Plus function is limited to X matrices containing 30 columns (not including the intercept). With 12 variables a full quadratic model has $12+12+\binom{12}{2} = 80$ terms. To fit a leaps model, the terms were divided into 3 sets, two of 30 and one of 20. Using adjusted R^2 as the criterion, a leaps model was fit to each set. The models selected had 10, 16, and 10 terms. Another model was fit combing the 10 and 16 terms from the first two fits. This resulted in a model with 17 terms, which was then combined with the 10 terms from the third fit. The final model had 18 terms:

$$\hat{y} = x_{11} + x_{12} + x_1x_2 + x_1x_5 + x_1x_6 + x_2x_8 + x_3x_4 + x_3x_7 + x_4x_4 + x_4x_5 + x_4x_7 +$$

$$x_4x_8 + x_4x_{11} + x_4x_{12} + x_5x_{10} + x_5x_{11} + x_5x_{12} + x_6x_{11}.$$

An attempt was made to use the same strategy with the C_p criterion. When the first group of 30 was used, however, the model selected contained all 30 terms and there was no obvious way to proceed in a non-arbitrary fashion.

A full tree model was fit, from which a shrunken tree model and a pruned tree model were constructed. The full tree model contained the variables x_2, x_3, x_5, x_6, and x_{12}. The degree of shrinking and pruning was determined by examining size versus deviance plots. In the pruned tree model, the variable x_5 was eliminated.

Figure 3 shows the diagnostic plots for the R-spline (R^2=.60), full regression tree (R^2=.66), and all subset regressions by leaps and bounds (R^2=.60) models. Here R^2 is calculated by $R^2 = 1 - \sum(y - \hat{y})^2 / \sum(y - \overline{y})^2$. This plots indicate that all the models fit the data fairly well in terms of predicting values close to the observed values. The R-spline model over-estimated low values; but the fit was satisfactory in the higher range of predicted values, where are interest lies. The regression tree models and and the leaps model did not do as well in prediction on a prediction set (as described below) which indicates there is probably some degree of over-fitting.

There were 391 compounds for which there was no replicate data available. These compounds were used as a prediction set for purposes of comparing the R-spline model to those models constructed using leaps and bounds all subets regressions and regression trees. There are some inherent differences between the modeling and prediction dataset: 1) the prediction set contains outliers that cannot be detected, as there are no replicates, 2) all compounds with a high response were re-tested so the prediction dataset

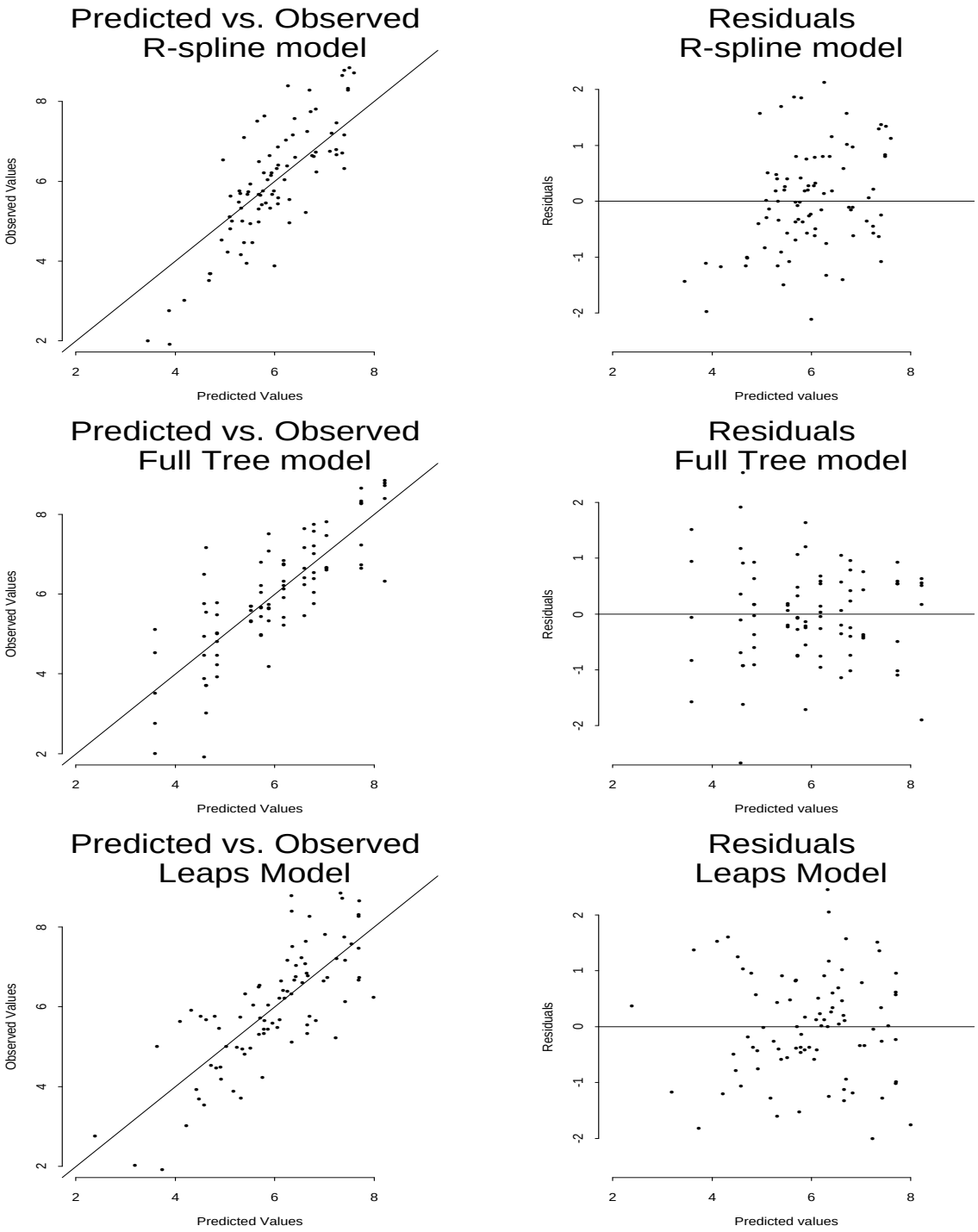

Figure 3. Diagnostic plots for the R-spline, full regression tree, and all subsets regression models.

has overall lower response values, and 3) the values in the prediction set are single measurements, which are less stable than the means that were used in the modeling dataset.

Table 4. Comparison of the model degrees of freedom, root mean predicted squared error, and the median absolute deviation of a prediction of the prediction set using models fit to the experimental dataset with the different methods.

$Method$	$ModelDf$	$Rpse$	Mad
$R-Spline$	23	**2.45**	**1.66**
$FullTree$	13	2.51	1.75
$Shrink$	13	2.53	1.77
$Prune$	8	2.54	1.80
$Prune\&Grow$	13	2.51	1.76
$LeapsR_a^2$	28	3.25	1.73

After all the models were fit, each was used to predict the responses in the prediction set, which were then compared to the observed responses. Table 4 shows the root mean predicted squared error and the median absolute deviation for these predictions from the different modeling methods. The R-spline model was more successful at predicting the responses in the prediction set both in terms of root mean predicted squared error (2.45) and median absolute deviation (1.66) than the other methods. Using a one-sided paired t test to compare the squared residuals, the R-spline predictions of the prediction set are significantly better than the leaps model (p-value=.019). There are not statistically significant differences in the squared residuals between the R-spline and the tree models (p-values .22 to .29).

8. CONCLUSIONS

R-splines using an initial all subsets variable selection method, and then optimizing the fit over the model parameters m and $cost$, consistently outperforms the competing regression tree and all subsets regression methods both in terms of variable selection and predictive power. Starting with a more conservative variable selection method gives the R-spline method a great advantage, as clearly a more satisfactory model can be fit once extraneous variables have been eliminated. Although weighting the model degrees of freedom proved an effective way of choosing variable subsets, unweighted model degrees of freedom led to more optimal models without sacrificing predictive power. A similar approach with regression trees could involve liberally pruning the tree and then re-growing a full tree with the remaining variables. The lower order polynomials fit with the leaps methods yield comparable results to the R-splines at very low levels of random error, even with the nonlinear functions. The R-splines, however, are superior to the leaps models at higher levels of random error even when fitting a polynomial function.

This R-spline methodology was also successful in a larger scale, more complex, practical problem. The all subsets regression method did result in a model that fit the data slightly better, but it did it did

not do as well in terms of prediction on the prediction set. It also had to be adapted to a problem of this size in a heuristic manner, which is not guaranteed to be a workable solution in every case. The regression tree method is the fastest method in terms of computer time, but it also did not do as well in terms of prediction on the prediction set. Thus, the application examined confirms the recommendation indicated by the simulations; the proposed R-spline methodology should be used in datasets containing a large set of continuous explanatory variables with a continuous response, particularly when the model is to be used for predicting responses from unobserved points.

APPENDIX A: THIN PLATE SPLINES AND BROKEN SPLINES

1. THIN PLATE SPLINES

The thin plate spline estimator of f (Wahba, 1990) is the minimizer of the following penalized sums of squares for a d-dimensional explanatory variable $\mathbf{x}$:

$$S_\lambda(f) = \frac{1}{n}\sum_{i=1}^{n} w_i(y_i - f(\boldsymbol{x}_i))^2 + \lambda J_m(f) \text{ for } \lambda > 0, \tag{A.2}$$

where

$$J_m(f) = \int_{\Re^d} \sum \frac{m!}{\alpha_1!...\alpha_d!}\left(\frac{\partial^m f}{\partial u_1^{\alpha_1}...\partial u_d^{\alpha_d}}\right)^2 d\boldsymbol{u}.$$

The sum in the integrand of $J_m(f)$ is taken over all non-negative integer vectors, α, such that $\sum \alpha_1 + \cdots + \alpha_d = m$. Clearly, for $m-1$ order polynomials $J_m(f) = 0$, because all m^{th} derivatives are 0. This will increase in magnitude as a function departs from an $m-1$ order polynomial. Thus, an $\hat{f}$ minimizing (A.2) will result in f with some level of smoothness dictated by λ.

The function that minimizes (A.2) has the form

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{t} \phi_j(\boldsymbol{x}_i)\beta_j + \sum_{k=1}^{N} E(\| \boldsymbol{x}_i - \boldsymbol{x}_k \|; m, d)\delta_k, \tag{A.3}$$

where

$$\sum_{j=1}^{t} \phi_j(\boldsymbol{x}_i)\delta_j = 0,\ 1 \leq j \leq t \text{ and } 2m - d > 0. \tag{A.4}$$

In this formulation, the $\phi_j(\mathbf{x}_i)$ are a set of t polynomial functions (of order $\leq m-1$) and $E(\parallel \mathbf{x}_i - \mathbf{x}_k \parallel ; m, d)$ are a set of N radial basis functions. It is assumed that β_j is estimable.

The radial basis functions are explicitly defined by

$$E(\boldsymbol{r}; m, d) = \begin{cases} a_{md} \parallel \boldsymbol{r} \parallel^{(2m-d)} log(\parallel \boldsymbol{r} \parallel) & d \text{ even} \\ a_{md} \parallel \boldsymbol{r} \parallel^{(2m-d)} & d \text{ odd} \end{cases}$$

where a_{md} depends only on m and d. Here r is the distance between two points.

For linear combinations of basis functions that satisfy (A.4), $J_m(f) = \delta^T M \delta > 0$ where $M_{k,i} = E(\parallel \mathbf{x}_i - \mathbf{x}_k \parallel; m, d)$. In other words, M is guaranteed to be positive definite. Defining the matrices $T_{n \times t}$ such that $T_{k,j} = \phi_j(\mathbf{x}_k)$ along with $W_{n \times n}$, a diagonal matrix proportional to the reciprocal variances of the errors, allows for the following matrix representation of the penalized sums of squares:

$$S_\lambda = \frac{1}{n}(Y - T\beta - M\delta)^T W (Y - T\beta - M\delta) + \lambda \delta^T M \delta. \tag{A.5}$$

After taking partial derivatives of (A.5) with respect to δ and β, a QR decomposition of T, $F^T T = \begin{bmatrix} R \\ 0 \end{bmatrix}$, is used to enforce the constraint $T^T\delta = 0$. $F_{n \times n}$ is an orthogonal matrix that can be partioned $F = [F_1 \mid F_2]$ where F_1 has columns that span the column space of T, i.e $T = F_1 R$, and F_2 is orthogonal to the column space of T. Reparameterizing by letting $\delta = F_2\omega_2$, $\beta \equiv \omega_1$, and $\omega^T = (\omega_1, \omega_2)^T$ allows for the problem to be posed in ridge regression form:

$$X^T W X \omega + \lambda H \omega = X^T W Y,$$

where

$$X = \begin{bmatrix} T & MF_2 \end{bmatrix}^T \text{ and } H = \begin{bmatrix} 0 & 0 \\ 0 & F_2^T M F_2 \end{bmatrix}.$$

Note that in the ridge regression formulation the roughness penalty is represented by the matrix H.

One standard way of determining λ is by generalized cross-validation (GCV) (Bates *et al*, 1987). The GCV function is:

$$V(\lambda) = \frac{Y^T(I - A(\lambda))^T W (I - A(\lambda))Y}{(n - trA(\lambda))^2} = \frac{SSE_{tps}}{(n - trA(\lambda))^2},$$

where

$$A(\lambda) = XG(I + \lambda D)^{-1} G^T X^T W.$$

One variation on this criterion is to the replace the denominator by

$$(1 - (\mathcal{C}(trA(\lambda) - t) + t)/n)^2. \tag{A.6}$$

Here t is the number of polynomial functions that span the null space of J_m and $\mathcal{C}$ is a cost parameter that can give more (or less) weight to the effective number of parameters beyond the base polynomial model.

2. BROKEN SPLINES

The degree of the polynomial component implies a specific roughness penalty in the thin plate spline, and thus if T, the design matrix for the polynomial component does not span P_{m-1} the roughness penalty minimized in the ridge regression formulation will not be the same roughness penalty as the one used to derive the thin plate spline. We will use the term broken spline to describe splines where T does not span P_{m-1}.

For purposes of this comparative discussion the T matrix for a thin plate spline will be denoted T_P and the T matrix for a broken spline will be denoted T_R. Note that the column space of T_R is a subset of that of T_P. Partition F

$$F = [F_a \; F_b \; F_c]$$

where F_a spans the space of T_R, F_b spans the columns of T_P that are not in T_R, and F_c spans the space orthogonal to T_P. Thus in the QR decomposition of T_P, $F_2 = F_c$ and in the QR decomposition of T_R, $F_2 = [F_b \; F_c]$.

For the broken spline H, the roughness penalty matrix, becomes

$$H_R = \begin{bmatrix} \mathbf{0}_{r\times r} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & F_b^T M F_b & F_c^T M F_b \\ \mathbf{0} & F_b^T M F_c & F_c^T M F_c \end{bmatrix},$$

as compared to the H for the thin plate spline which is

$$H_t = \begin{bmatrix} \mathbf{0}_{r\times r} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{t-r\times t-r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & F_c^T M F_c \end{bmatrix}.$$

Hence, H_R for the broken spline is the roughness penalty for the thin plate spline, plus other terms,

$$H_R = H_t + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & F_b^T M F_b & F_c^T M F_b \\ \mathbf{0} & F_b^T M F_c & \mathbf{0} \end{bmatrix}.$$

With thin plate splines, H_t is guaranteed to be non-negative definite. Recall $\delta^T M \delta > 0$ where $T^T \delta = 0$ and $2m - d > 0$, which guarantees that M is positive definite. $F_2^T M F_2$, is a quadratic form of a positive definite matrix and as such is also positive definite. H_t, having $F_2^T M F_2$ in the lower right block and zeros elsewhere, is non-negative definite.

With broken splines, however, H_R is no longer a non-negative definite matrix. Let B denote the inverse symmetric square root of $X^T W X$, i.e., $(B^T B)^{-1} = X^T W X$. Then let UDV^T be the singular value decomposition of BHB so $H_R = B^{-1} U D V^T B^{-1}$. Let ${H_R}^* = B^{-1} U D U^T B^{-1}$, which is a non-negative definite matrix because D is a diagonal matrix of singular values (which are non-negative by definition) and H^* is a quadratic form of D. If H_R is replaced by ${H_R}^*$, a non-negative definite roughness penalty is minimized. Because H_R is a symmetric matrix, U and V will be the same except for sign changes in columns corresponding to the negative eigenvalues. Hence, ${H_R}^*$, is the matrix that results from forcing the negative eigenvalues of H_R to be positive. How "close" the two matrices are depends on the magnitude of these eigenvalues. The negative eigenvalues that occur correspond to the "missing" polynomial terms in the null space. Thus, the roughness penalty is forced into penalizing roughness in the surface that might be modeled by these missing polynomial terms, fitting them instead with the flexible radial basis functions.

APPENDIX B: REGRESSION TREES

Regression tree models (Chambers and Hastie, 1992) use a binary recursive partitioning technique in which the data are successively split along coordinate axes of the explanatory variables. At each node, the split that maximally distinguishes the response variable between the two branches is selected. When

the response variable is numeric the tree is called a regression tree (as opposed to a classification tree when the response is categorical). It is assumed that the response variable has a normal distribution. When x is numeric, as in these simulations, the class of splits consists of the $k-1$ ways to divide the values into two non-overlapping sets where k is the number of distinct values of x. The deviance function for an observation is defined as $D(\mu, y_i) = (y_i - \mu)^2$, where μ is the node mean. The deviance of a node is the sum of the deviances of all observations in the node. The split with the largest difference between the deviances of the two branches is chosen. Splitting continues until nodes are pure (i.e., all observations in the node have the same response) or data are too sparse.

For tree models, there are two approaches to reduce over-fitting. The first is "pruning" where nodes with the least important splits are recursively "snipped." Importance is measured by the cost-complexity measure:

$$D_\alpha(T') = D(T') + \alpha size(T'),$$

where $D(T')$ is the deviance of the subtree T', $size(T')$ is the number of terminal nodes of T', and α is the cost-complexity measure. For a specified α, there is a subtree T', that will minimize $D(T')$ over the set of all possible subtrees. The second approach is "shrinking" in which estimates for observations in lower nodes are "shrunk" to their parent nodes based upon the magnitude of the difference between the fitted values of the lower nodes and the fitted values of their parent nodes. Shrunken fitted values are determined by the recursively applied formula,

$$\hat{y}(node) = \alpha \overline{y}(node) + (1-\alpha)\hat{y}(parent),$$

where the parameter α in this case indexes the degree of shrinkage.

APPENDIX C: LEAPS AND BOUNDS ALL SUBSET REGRESSIONS CRITERIA

The C_p statistic is defined by

$$C_p = \frac{RSS_P}{\hat{\sigma}^2} - n + 2p.$$

RSS_P is the residual sum of squares where P indexes a subset of $(p-1)$ of k potential explanatory values and is an estimate of $(n-p)\hat{\sigma}^2$. Thus, for a satisfactory regression, the statistic C_p should be close to p. In terms of programming for these simulations, the subset with the smallest absolute difference between

p and C_p was chosen. For clarification, p includes an intercept and is 1 more the number of variables in the subset.

Use of R^2 as a criterion is limited because the largest set of explanatory variables will always give the highest R^2. An adjusted R^2 attempts to take into account the number of parameters:

$$R_a^2 = 1 - (1 - R^2)\left(\frac{n}{n-p}\right).$$

Again p is the number of parameters in the model, including the intercept. R_a^2 may become smaller when another variable is added to the model because the increase in R^2 may be more than offset by the loss of a degree of freedom in the denominator. The subset with the maximum adjusted R^2 is chosen as best. Despite this adjustment, it has been noted in the literature, as well as these simulations, that adjusted R^2 still tends to favor large regressions over smaller ones.

APPENDIX D: R_P^2, OR $R^2{}_{PREDICTION}$

$R^2{}_{prediction}$ is a function of the leave-one-out or PRESS (Predicted Error Sum of Squares) residuals, which can be computed using the diagonals of the hat matrix,

$$e_{(i)} = \frac{e_i}{1 - h_{ii}}.$$

PRESS is the sum of the squared PRESS residuals, and can be used to compute an approximate R^2 for prediction (Myers, 1995).

$$R_p^2 = 1 - \frac{PRESS}{SSTO},$$

where SST0 is the total sum of squares. This statistic can be interpreted as the approximate percentage of variability explained in predicting new observations (as opposed to R^2, the variability in the observed responses that can be explained by the model). This statistic, unlike R^2, can be negative when the model fits the data poorly.

1. REFERENCES

Bates, D.M., Lindstrom, M.J., Wahba, G. and Yandell, B.S. (1987). GCVPACK - Routines for generalized cross validation. *Comm. Stat. Sim. Comp.* **16**, 263-297.

Hardy S.W. and Nychka, D.W. (2000) R-splines for Response Surface Modeling. *North Carolina State University Mimeo Series* # 2526. Raleigh, NC.

Furnival, G. M. and Wilson, R. W. Jr. (1974). Regressions by Leaps and Bounds. *Technometrics* **16**, 499-511.

Myers, R. H., Montgomery, D.C. (1995). *Response Surface Methodology*. John Wiley and Sons, Inc., New York, NY.

Nychka, D., B. Bailey, S. Ellner, P. Haaland, and M. O'Connell (1996). *FunFits* data analysis and statistical tools for estimating functions. Software and paper available from *statlib*.

Wahba, G. (1990). *Spline Models for Observational Data*. Society for Industrial Applied Mathematics. Philadelphia PA.