Supplementary materials for this article are available online.
Please click the JCGS link at http://pubs.amstat.org.

Partial Generalized Additive Models: An Information-Theoretic Approach for Dealing With Concurvity and Selecting Variables

Hong GU, Toby KENNEY, and Mu ZHU

Scientists are often interested in which covariates are important, and how these covariates affect the response variable, rather than just making predictions. This requires inputs from both statistical modeling and background knowledge. Generalized additive models (GAMs) are a class of interpretable, multivariate nonparametric regression models which are very useful data exploration tools for these purposes, but concurvity among covariates (the nonlinear analogue of collinearity for linear regression) can lead GAMs to produce unstable or even wrong estimates of the covariates' functional effects. We develop a new procedure called partial generalized additive models (pGAM), based on mutual information (MI), a measure of nonlinear dependence between variables. Our procedure is similar in spirit to the Gram–Schmidt method for linear least squares. By building a GAM on a selected set of transformed variables, pGAM produces more stable models, selects variables parsimoniously, and provides insight into the nature of concurvity between the covariates by calculating functional dependencies among them. With simulation experiments and real-data examples, we show that pGAM produces much better estimates of the covariates' functional effects, and also incorporates a reasonable and meaningful variable selection method. R code for fitting pGAMs is available online (see Supplemental Materials Section).

Key Words: Concurvity; Generalized additive models; Interpretation; Mutual information; Partial generalized additive models; Variable selection.

1. INTRODUCTION

The generalized additive model (GAM) (Hastie and Tibshirani 1986, 1990) is a popular nonparametric and semiparametric model fitting technique. Let Y be the response variable

Hong Gu is Associate Professor, Department of Mathematics and Statistics, Dalhousie University, Halifax, Nova Scotia, B3H 3J5, Canada (E-mail: *hgu@mathstat.dal.ca*). Toby Kenney is Researcher, Institute for Science and Research, Matej Bel University, Banská Bystrica, Slovakia. Mu Zhu is Associate Professor, Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada.

Journal of Computational and Graphical Statistics, Volume 19, Number 3, Pages 531–551
DOI: 10.1198/jcgs.2010.07139

and $X = (X_1, \ldots, X_p)$ be the covariates. GAM assumes

$$E(Y|X) = h(\eta(X)) = h(f_0 + f_1(X_1) + \cdots + f_p(X_p)), \tag{1.1}$$

where Y has an exponential family distribution; h is a known monotonic link function; and the f_j's ($j > 0$) are unspecified smooth functions. This simple functional form allows for easy and intuitive interpretation of the covariates' functional effects. For this reason, users sometimes prefer GAM even at the expense of predictive accuracy compared with some other available nonparametric methods.

However, the interpretation of GAM is not straightforward when many covariates are involved, since each function $f_j(X_j)$ can only be interpreted as the effect of X_j on the response Y while keeping all other functional effects fixed. When strong functional relationships exist among the covariates themselves, GAM often produces functional estimates that are statistically unstable or even wrong.

The term "concurvity" is used as the analogue of "collinearity" to describe such degeneracies (Hastie and Tibshirani 1990). In a broad sense, concurvity can be defined as a measure of statistical dependency among covariates or the existence of multiple solutions when fitting a GAM. If all covariates are continuous as is the case in this paper, then concurvity can simply be taken to mean the existence of functional relationships among them. Donnell, Buja, and Stuetzle (1994) developed a nonlinear generalization of principal components and used the smallest additive principal component $\sum_j \phi_j(X_j) \approx 0$ as a diagnostic tool for checking concurvity. Using simulation experiments and applying GAM to time series data, Ramsay, Burnett, and Krewski (2003) showed that concurvity can cause us to underestimate standard errors.

When fitting linear models, one way to deal with collinearity is to use penalized regression techniques such as ridge regression. It is also possible, to some extent, to deal with concurvity in GAM by controlling the complexity or smoothness of each fitted function. Green and Silverman (1994), Hastie and Tibshirani (1990), and Wahba (1990) provide nice surveys of penalized likelihood methods. More recently, Simon Wood further advanced the art of GAM fitting in a series of important papers. Wood (2000) applied generalized cross-validation (GCV) to the penalized likelihood function and provided a method to efficiently select multiple smoothing parameters. Wood (2004) solved a difficult numeric rank deficiency problem and showed through simulation that his methods not only provided much more stable functional reconstruction but also gave very competitive mean squared errors when compared with other existing methods for fitting GAMs. Although Wood (2004) simulated all the covariates to be independent in his examples, our experience shows that Wood's method still generates stable and optimally fitted values and properly minimizes mean squared error when the covariates are dependent.

While shrinkage methods can numerically stabilize the model fitting procedure, they are not meant to deal with the inherent difficulty in model simplification, variable selection and model interpretation when there exists concurvity among covariates. In scientific applications, the selection of important covariates and the proper interpretation of their functional effects are sometimes more important than predictive accuracy. To effectively deal with variable selection and concurvity, properly estimate the covariates' functional effects, and

facilitate more precise interpretations of a GAM model, we develop a procedure called partial generalized additive models (pGAM).

Hastie and Tibshirani (1990, p. 125) described a modified backfitting algorithm, which *partially* deals with the concurvity problem. The basic idea is to separate each smoothing operator into a projection part (corresponding to directions with unit eigenvalues) and a shrinking part (corresponding to directions with eigenvalues less than one), combine all the projection parts into one large projection, and only use back-fitting for the shrinking parts. For symmetric smoothers having eigenvalues in [0, 1], exact concurvity occurs only in the projection part, i.e., only if the covariates are perfectly collinear (Hastie and Tibshirani 1990, sections 5.3.5 and 5.4.2), so the modified back-fitting algorithm allows us to deal with concurvity in the projection step alone. However, Hastie and Tibshirani (1990) went on to emphasize that approximate concurvity is still of practical concern, when the covariates are clustered around some lower dimensional manifold. This is precisely the situation that we aim to address with pGAM.

1.1 AN ILLUSTRATIVE EXAMPLE

We first simulate a somewhat exaggerated example to demonstrate how GAM can have difficulty determining functional effects in the case of strong concurvity. A total of $n = 500$ observations are simulated. The covariates X_1, X_2, X_3, X_4 are simulated independently from Unif[0, 1]. The covariate X_5 is generated by

$$X_5 = 2X_1^3 + N(0, \sigma_1^2); \tag{1.2}$$

and the response variable Y is generated by

$$Y = (5e^{-X_1} + 2X_1^3) + X_3 + N(0, \sigma_2^2), \tag{1.3}$$

where $\sigma_1 = 0.01$ and $\sigma_2 = 0.1$. That is, Y is a function of X_1 and X_3, but there is *very strong* concurvity between X_1 and X_5. Of primary interest here is the question: what is the effect of X_1 on Y? This is difficult for GAM to pin down due to the strong concurvity between X_1 and X_5.

The upper left panel of Figure 1 shows the true function $f_1(x_1) = 5e^{-x_1} + 2x_1^3$ and the remaining panels show the result of GAM, using Simon Wood's `mgcv` package in `R` (Wood 2006). Notice the difference between the estimated and the true functional effects of X_1. Notice also the estimated effect of X_5.

Our procedure, pGAM, sequentially maximizes the mutual information (MI) between the response variable and the covariates. Starting with a null model, pGAM first chooses to add the covariate whose mutual information with Y is the largest. It then removes any functional effects of this covariate from all remaining covariates before searching for the next covariate to add. The final result is a model based on a sequence of adjusted predictor variables. The removal of functional dependencies at each step eliminates problems caused by concurvity and gives much more precise and reliable interpretations of the covariates' effects. After the first covariate, all covariates are transformed during the fitting process. We use the notation X_j to denote the original covariates and $X^{(j)}$ to denote the transformed covariates.

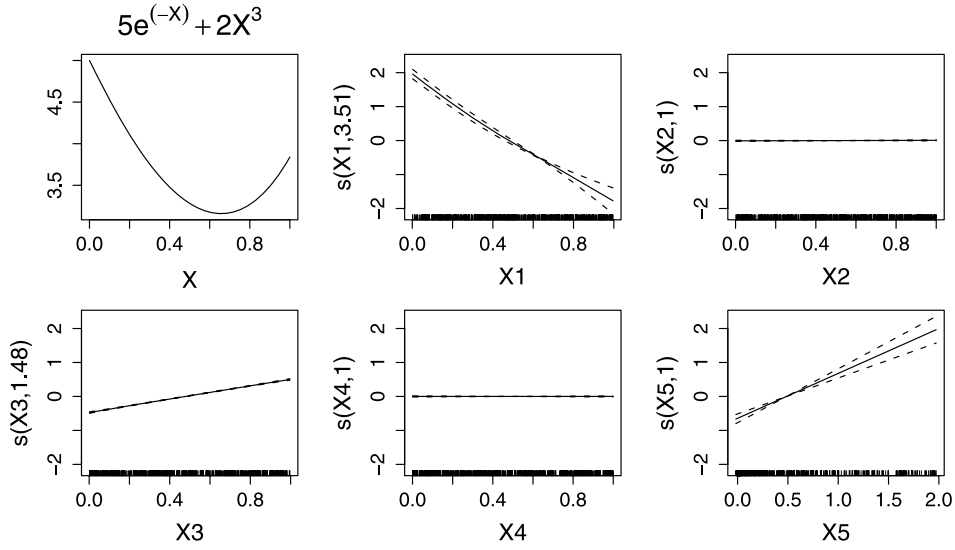

Figure 1. Illustrative example. Upper left: the function $f(x) = 5e^{-x} + 2x^3$ on [0, 1]. Others: effects estimated by GAM.

In this example, the first covariate to enter the model is X_1, so $X^{(1)} = X_1$. We then proceed to remove any effect of $X^{(1)}$ from the remaining covariates by letting

$$X^{(j)} = X_j - g_{j1}\big(X^{(1)}\big), \tag{1.4}$$

where g_{j1} is obtained by smoothing X_j onto $X^{(1)}$ for all $j \neq 1$. The next covariate to enter the model is $X^{(3)}$. Theoretically, since X_3 is independent of $X^{(1)}$, the transformed version $X^{(3)}$ should be identical to the original version X_3 but, in practice, they will not be exactly the same. Notice that, after removing the effect of $X^{(1)}$, $X^{(5)}$ no longer contains any information about Y. In this case, none of the remaining variables is found useful after $X^{(3)}$. The final model thus only includes $X^{(1)}$ and $X^{(3)}$. Figure 2 shows the functional effects of $X^{(1)}$ and $X^{(3)}$ as estimated by pGAM. Notice that the interpretation of each covariate's effect will now depend on previous covariates in the model. Plots of functional dependencies between pairs of covariates thus provide insight into the nature of concurvity, and allow us to better interpret the resulting model.

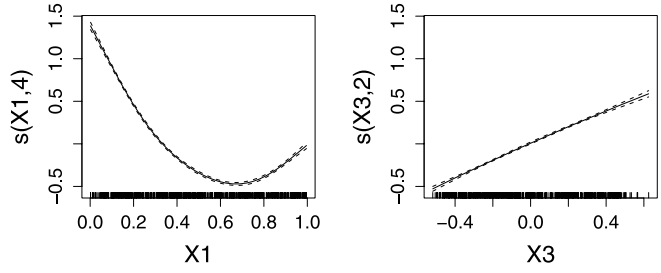

Figure 2. Illustrative example. Effects estimated by pGAM. Notice that only $X^{(1)}$ and $X^{(3)}$ are included in the final model.

1.2 OUTLINE

The remainder of our paper is structured as follows. In Section 2, we develop our main methodology, pGAM. Two simulation studies are presented in Section 3 and two real-data examples are analyzed in Section 4. A number of related issues are discussed in Section 5.

2. MAIN METHODOLOGY

In this section, we first give a brief review of mutual information (Section 2.1). We then look at GAM fitting from the viewpoint of maximizing mutual information (Section 2.2). This leads us to develop the pGAM algorithm in Section 2.3.

2.1 BRIEF REVIEW OF MUTUAL INFORMATION

First introduced by Shannon (1948), mutual information (MI) is a measure of dependence between two random variables. For the bivariate random vector (X, Y), their mutual information is defined as

$$\mathrm{MI}(X; Y) = E\left(\log \frac{f(X, Y)}{f_X(X) f_Y(Y)}\right), \tag{2.1}$$

where f, f_X, and f_Y are their joint and marginal probability distribution functions, respectively. This is closely related to the notion of entropy, $H(X) = -E \log(p(X_1, \ldots, X_p))$, where $X = (X_1, \ldots, X_p)$ is a random vector and $p(x_1, \ldots, x_p)$ is the joint distribution function. Entropy measures the amount of uncertainty in a random variable or a random vector. Mutual information is the relative entropy between the joint distribution and the product distribution. It is easy to show that if $H(Y|X) = -E(\log p(Y|X_1, \ldots, X_p))$ is the conditional entropy, then

$$\mathrm{MI}(Y; X) = H(Y) - H(Y|X) = H(X) - H(X|Y).$$

Thus, MI is the amount of information in X that can be used to reduce the uncertainty of Y. More properties of entropy and mutual information can be found in chapter 2 of Cover and Thomas (1991).

It is well known that Pearson's correlation coefficient ρ measures the linear relationship between two random variables. The following properties of MI, listed in Brillinger (2004), suggest that MI can be used as a measure of nonlinear relationships and thus is to nonparametric regression analysis as Pearson's correlation coefficient is to linear regression analysis:

(1) $\mathrm{MI}(Y; X) = 0$ if and only if X is independent of Y.

(2) For the continuous case, $\mathrm{MI}(Y; X) = \infty$ if $Y = g(X)$ for some function g.

(3) Invariance, i.e., $\mathrm{MI}(Y; X) = \mathrm{MI}(V; U)$ if $U = U(X)$ and $V = V(Y)$ are individually 1–1 measurable transformations.

(4) If (X, Y) has a bivariate normal distribution, then $\mathrm{MI}(Y; X) = -\frac{1}{2}\log(1 - \rho_{XY}^2)$.

2.2 GAM AND MAXIMIZATION OF MUTUAL INFORMATION

Suppose that, given $X = x$, Y has conditional density $h(y, \eta(x))$. Hastie and Tibshirani (1986) showed that GAM chooses $\hat{\eta}(\cdot)$ to maximize the expected log-likelihood, i.e.,

$$E\big(l(\hat{\eta}(X), Y)\big) = \max_{\eta} E\big(l(\eta(X), Y)\big),$$

where $l(\eta(x), y)$ is the log-likelihood of Y given $\eta(X)$, and the expectation is taken over the joint distribution of X and Y. On the other hand, the mutual information between Y and $\eta(X)$ is equal to

$$\mathrm{MI}(Y; \eta(X)) = E\left\{\log \frac{f(\eta(X), Y)}{f_{\eta(X)}(\eta(X)) f_Y(Y)}\right\} = E\big(l(\eta(X), Y)\big) - E \log f_Y(Y). \tag{2.2}$$

Since $E \log f_Y(Y)$ does not depend on $\eta(X)$, choosing $\hat{\eta}(\cdot)$ to maximize the expected log-likelihood $E(l(\eta(X), Y))$ is equivalent to choosing $\hat{\eta}(\cdot)$ to maximize $\mathrm{MI}(Y; \eta(X))$. Therefore, GAM chooses η so as to maximize $\mathrm{MI}(Y, \eta(X))$, subject to the constraint that $\eta(X) = f_0 + f_1(X_1) + \cdots + f_p(X_p)$.

Cover and Thomas (1991, section 2.8) showed that, for any function $\eta(X)$,

$$\mathrm{MI}(Y; \eta(X)) \le \mathrm{MI}(Y; X). \tag{2.3}$$

This is essentially a "no-free-lunch" principle that says we cannot increase the information about Y by transforming the original predictors X. We want to find $\eta(X)$ to maximize $\mathrm{MI}(Y; \eta(X))$, i.e., to make $\mathrm{MI}(Y; \eta(X))$ as close to its upper bound, $\mathrm{MI}(Y; X_1, \ldots, X_p)$, as possible. The chain rule for mutual information gives

$$\mathrm{MI}(Y; X_1, \ldots, X_p) \tag{2.4}$$

$$= \mathrm{MI}(Y; X_1) + \mathrm{MI}(Y; X_2|X_1) + \cdots + \mathrm{MI}(Y; X_p|X_{p-1}, \ldots, X_1) \tag{2.5}$$

$$= \mathrm{MI}(Y; X_1, \ldots, X_{p-1}) + \mathrm{MI}(Y; X_p|X_{p-1}, \ldots, X_1). \tag{2.6}$$

Therefore, one approach to maximize $\mathrm{MI}(Y, \eta(X))$ is to construct $\eta(X)$ term by term, making each term as close as possible to the terms in (2.5). We obtain $f_1(X_1)$ by fitting a (univariate) GAM of Y onto X_1. As we argued above, this makes $\mathrm{MI}(Y, f_1(X_1))$ approach $\mathrm{MI}(Y; X_1)$.

Now suppose $Y = f_1(X_1) + Z$, where Z is independent of X_1 and also conditionally independent of X_1 given X_2. Then,

$$\mathrm{MI}(Y; X_2|X_1) = H(Y|X_1) - H(Y|X_2, X_1) \tag{2.7}$$

$$= H\big((f_1(X_1) + Z)|X_1\big) - H\big((f_1(X_1) + Z)|X_2, X_1\big) \tag{2.8}$$

$$= H(Z|X_1) - H(Z|X_2, X_1) \tag{2.9}$$

$$= H(Z) - H(Z|X_2) \tag{2.10}$$

$$= \mathrm{MI}(Z, X_2). \tag{2.11}$$

Here, (2.7) and (2.11) follow directly from the definition of mutual information; (2.9) is because there is no uncertainty in $f_1(X_1)$ once X_1 is given; and (2.10) is due to the assumption that Z is independent of X_1 and conditionally independent of X_1 given X_2.

These equations suggest that, to construct the next term, $f_2(X_2)$, we can simply take the partial residual $Z = Y - f_1(X_1)$ and then fit a (univariate) GAM of Z onto X_2.

To carry out this argument recursively, suppose $Y = \eta_k(X_1, \ldots, X_k) + Z$, where Z and $(X_1, \ldots, X_k)$ are independent as well as conditionally independent given X_{k+1}. A similar argument leads to

$$\mathrm{MI}(Y; X_{k+1}|X_1, \ldots, X_k) = \mathrm{MI}(Z; X_{k+1}). \tag{2.12}$$

Sequentially approximating the terms in (2.5), using $f_1(X_1) + \cdots + f_k(X_k)$ as an approximation to $\eta_k(X_1, \ldots, X_k)$ when estimating $f_{k+1}(X_{k+1})$, leads to the first pass of the familiar back-fitting algorithm. However, the conditions given immediately before (2.7) and (2.12) suggest that this approximation works only when $X_1, \ldots, X_k$ are independent. Suppose that, for example, there is concurvity between X_k and $X_1, \ldots, X_{k-1}$. Then, even if the additive approximation holds up to $\eta_{k-1}(X_1, \ldots, X_{k-1})$, the partial residual after adding $f_k(X_k)$, namely $Z = Y - \eta_{k-1}(X_1, \ldots, X_{k-1}) - f_k(X_k)$, will still not be independent of $(X_1, \ldots, X_{k-1})$, rendering equation (2.12) false. This is why the back-fitting algorithm requires multiple passes and each function must be iteratively re-fitted. The re-fitting is justified because the chain rules given in (2.5) and (2.6) do not depend on the order of the X_j's.

2.3 PARTIAL GENERALIZED ADDITIVE MODELS

As we have just argued above, when the covariates are not independent, sequentially constructing the terms in $\eta(x)$ to approach the terms in (2.5), as in the first pass of the backfitting algorithm, does *not* lead to the optimal result and iteration is needed. Partial generalized additive models (pGAM) are based on an alternative way to approximate the terms in (2.5). Instead of a recursive application of (2.6), pGAM is based on the recursive application of the following:

$$\mathrm{MI}(Y; X_1, \ldots, X_p) = \mathrm{MI}(Y; X_1) + \mathrm{MI}(Y; X_2, \ldots, X_p|X_1). \tag{2.13}$$

Again, suppose $Y = f_1(X_1) + Z$, where Z is independent of X_1. But now, suppose $X_j = g_{j1}(X_1) + X^{(j)}$ for $j = 2, \ldots, p$, where X_1 and $(X^{(2)}, \ldots, X^{(p)})$ are independent and also conditionally independent given Z. Then, instead of (2.7)–(2.11), we have

$$\begin{aligned}
&\mathrm{MI}(Y; X_2, \ldots, X_p|X_1) \\
&\quad = H(X_2, \ldots, X_p|X_1) - H(X_2, \ldots, X_p|Y, X_1) \\
&\quad = H\big(\big(g_{j1}(X_1) + X^{(j)}\big)_{j=2,\ldots,p}|X_1\big) - H\big(\big(g_{j1}(X_1) + X^{(j)}\big)_{j=2,\ldots,p}|f_1(X_1) + Z, X_1\big) \\
&\quad = H\big(X^{(2)}, \ldots, X^{(p)}|X_1\big) - H\big(X^{(2)}, \ldots, X^{(p)}|Z, X_1\big) \\
&\quad = H\big(X^{(2)}, \ldots, X^{(p)}\big) - H\big(X^{(2)}, \ldots, X^{(p)}|Z\big) \\
&\quad = \mathrm{MI}\big(Z; X^{(2)}, \ldots, X^{(p)}\big).
\end{aligned} \tag{2.14}$$

This suggests a different procedure: first, estimate $f_1(X_1)$ by fitting a (univariate) GAM of Y onto X_1; next, estimate $g_{21}, \ldots, g_{p1}$ by smoothing $X_2, \ldots, X_p$ onto X_1; then, recursively fit Z onto the adjusted variables, $(X^{(2)}, \ldots, X^{(p)})$, which are independent of X_1.

We refer to the functions g_{ji} as "partial effects," which is also where the name "pGAM" comes from.

In practice, we adopt a slightly modified procedure. Since $Y - Z = f_1(X_1)$ is independent of $(X^{(2)}, \ldots, X^{(p)})$, we observe that the same function η will maximize both $\mathrm{MI}(Y; \eta(X^{(2)}, \ldots, X^{(p)}))$ and $\mathrm{MI}(Z; \eta(X^{(2)}, \ldots, X^{(p)}))$. Therefore, instead of using Z, we can keep using the original response, Y, when fitting $X^{(2)}, \ldots, X^{(p)}$. For simple additive models, it is equally convenient whether we use Y or Z but, for *generalized* additive models (GAMs), it is much more convenient to use the original response, Y, due to the presence of a nonlinear link function $h(\cdot)$—see equation (1.1). In this case, the equations above are only an approximation, but we can still fit a GAM on the transformed variables, which are independent, and avoid concurvity. Notice that back-fitting iterations are not necessary for pGAM.

2.3.1 Variable Selection

When fitting a pGAM, the order in which we fit the variables makes a difference to the space of possible models, and thus to the final model chosen. We choose to fit the variables in order of decreasing mutual information with Y, so the variables with the highest mutual information are included first. If, in some applications, a number of covariates are deemed important a priori, we can start by fitting on these variables first. We incorporate a variable selection procedure by only including in the final model those variables which significantly improve the previous model.

A similar variable selection procedure based on the sequential maximization of MI (see Section 2.2) can be derived for classic additive models; we refer to this as "stepwise additive model" or simply *Stepwise AM*. At each step, we enter the covariate having the largest MI with the residual Z, and the procedure stops when the MI between Z and the remaining covariates becomes insignificant. This is different from pGAM in that with pGAM we estimate the MI between the original response variable and the transformed predictor, while in *Stepwise AM*, we estimate the MI of the residual with the original predictor. As mentioned before (Section 2.2), due to the additive approximation of the function $\eta_k(X_1, \ldots, X_k)$ at each step, the residual Z is not necessarily independent of the variables already chosen. Thus *Stepwise AM* is theoretically unsound when there is concurvity among the covariates. It tends to include more covariates which have concurvity with the ones already included, as we shall see later in simulations. We explain the problem with *Stepwise AM* in more detail in the appendix, and give a simple simulation to confirm that this problem can cause *Stepwise AM* to select the wrong variables.

2.3.2 Indirect Estimation of MI

Direct estimation of MI is not a trivial problem. Instead of directly estimating MI, we work with a "proxy" of $\mathrm{MI}(Y; X)$ based on (2.3) and (2.2):

$$\widehat{\mathrm{MI}(Y; X)} = \max_{\eta} \mathrm{MI}(Y; \eta(X)) = \max_{\eta} E\big(l(\eta(X), Y)\big) - E \log f_Y(Y). \tag{2.15}$$

Use of this proxy can be justified by the fact that, if $\eta(X)$ is a sufficient statistic for Y, then $\mathrm{MI}(Y;\eta(X)) = \mathrm{MI}(Y;X)$ (Cover and Thomas 1991, section 2.10). For the purpose of variable selection, we only need to compare the MIs of the covariates with Y. We therefore only need to calculate the maximum value of the conditional log-likelihood. To do this, at each step, we simply fit a univariate GAM of Y onto each remaining covariate and choose the covariate with the largest log-likelihood (or the smallest deviance).

2.3.3 The pGAM Algorithm

The entire pGAM algorithm is laid out in Table 1. The procedure is similar in spirit to the Gram–Schmidt method for linear least squares. At each step, pGAM first chooses the best variable to enter the model. It then tests whether entering this variable into the model provides a significant improvement to the current model. If it does, then pGAM removes any functional effects of this variable from the remaining variables. As a result, we obtain $\mathcal{X}_w$, a group of variables that are approximately independent of this variable, as candidates to enter the model at the next stage. If it does not, then pGAM moves on to the next variable, and so on until all variables have been tried.

Table 1. The pGAM algorithm.

1. Initialization:
   (a) Start with a null model m_0 by fitting a GAM of Y onto a constant; let D_0 be the deviance of m_0.
   (b) Center all X_j's to have mean zero; let
   $$\mathcal{X}_w = \left\{X^{(j)} = X_j;\ j = 1, 2, \ldots, p\right\}$$
   be the initial set of "working variables."
   (c) Set $D = D_0$ and $m = m_0$.
2. Main procedure:
   (a) For each working variable $X^{(j)}$ in $\mathcal{X}_w$, fit a (univariate) GAM of Y onto $X^{(j)}$. Record the deviance, d_j, as well as the degree of freedom for $X^{(j)}$, df_j. Collect d_i into a vector d.
   (b) Choose i such that d_i is the smallest element of d. Remove d_i from d and $X^{(i)}$ from $\mathcal{X}_w$. Form a new model m_{new} by adding $X^{(i)}$ (with df_i degrees of freedom) into m; let D_{new} be the deviance of m_{new}.
   (c) Test whether D_{new} is a significant improvement over D.
   (d) If test 2(c) is not significant:
   - If $\mathcal{X}_w$ is not empty, then go to step 2(b).

   (e) If test 2(c) is significant:
   - For every $X^{(j)} \in \mathcal{X}_w$ $(j \neq i)$, fit $X^{(j)} = g_{ji}(X^{(i)}) + \epsilon_j$ by smoothing $X^{(j)}$ onto $X^{(i)}$; record the fitted functions g_{ji}; and replace each $X^{(j)}$ with $X^{(j)} - g_{ji}(X^{(i)})$ in $\mathcal{X}_w$.
   - Let $D = D_{\text{new}}$; $m = m_{\text{new}}$.
   - If $\mathcal{X}_w$ is not empty, then go to step 2(a).
3. Output: the model m and the g_{ji}'s.

In theory, the set $\mathcal{X}_w$ contains variables that are independent of the ones already in the model, so the variable we select should be the best variable for improving our model. Thus, we should be able to stop after the first rejected variable: that is, if test 2(c) is not significant. However, since step 2(e)—the step that updates the set $\mathcal{X}_w$—is not perfect in practice, we might have selected the first variable to attempt because of some concurvity that was not fully partialled out in step 2(e). Therefore, we might have selected a variable that should not be in the model in preference to a variable that should be in the model. Thus, we continue to check for additional variables to add in step 2(d).

Notice that the $X^{(j)}$'s are generally not the same as the original covariates. Thus the final model is no longer additive in terms of the original covariates. In order to fully interpret the resulting model, pairwise plots of the functions g_{ji} can be used to give us a more complete picture of the relationships among the covariates themselves.

2.3.4 Some Implementation Details

It does not matter which software we use to fit the GAMs. We use Simon Wood's `mgcv` package in `R` because of its efficiency in automatically choosing the smoothing parameters or the degrees of freedom. Also, its ability to deal with numerical rank deficiency (a problem related to concurvity) makes it more suitable for us to compare with our procedure, pGAM.

Section 2.3 suggests that, to add $X^{(i)}$ in step 2(b), we can simply insert into m the term from the corresponding univariate GAM obtained in step 2(a). But since step 2(e)—the "concurvity removal" step—is not perfect in practice, we implement step 2(b) by fitting a multivariate GAM onto the existing covariates in m plus the newly added term, $X^{(i)}$, using fixed degrees of freedom for all terms.

We use thin plate regression splines to fit each function. This ensures that m is nested within m_{new} (Wood 2003) and makes it possible for us to use conventional hypothesis testing in step 2(c), e.g., an F-test if Y is Gaussian and a χ^2-test if Y is binomial or Poisson. When fitting the functions g_{ji} in step 2(e), we use the same degree of freedom for $X^{(i)}$ as when it entered the model m_{new} in step 2(b), because we found this produced better results.

The entire pGAM algorithm involves fitting $O(p^2)$ univariate GAMs and $O(p)$ multivariate GAMs. Univariate GAMs are cheap to fit. The exact complexity depends on the particular method used. The $O(p)$ multivariate GAMs are all of relatively low complexity, since the covariates are made independent, or at least approximately independent, and the degree of freedom for each covariate is pre-specified (see step 2(b)), so these fits are expected to converge quickly. Since pGAM tends to select parsimonious models, there will often not be too many variables in these multivariate GAMs.

For the test in 2(c), we used a fixed significance level α, that we allow the users to specify as a control parameter in the program. Since pGAM involves multiple testing, it may be more appropriate to use a sequence of different significance levels, but we do not expect this to significantly influence the simulation results below. We experimented a little with different values of α in simulation 1, and we use these experiments to suggest reasonable values to set for α in practice.

3. SIMULATION STUDIES

In this section, we present two simulation experiments to illustrate the variable selection effect of pGAM and compare it with *Stepwise AM* (see Section 2.3.1). We also compare the predictive accuracy of pGAM with that of regular GAM (not *Stepwise AM*) using Simon Wood's `mgcv` package in `R`. For both simulations, the sample size is fixed at $n = 500$ for both training and test data sets. The control parameter α for pGAM is set to be 0.001 in both experiments.

3.1 SIMULATION 1

The first simulation is based on the illustrative example in Section 1.1. We use the parameter σ_1 in equation (1.2) to control the degree of concurvity between X_1 and X_5, and consider three levels: strong concurvity ($\sigma_1 = 0.01$), medium concurvity ($\sigma_1 = 0.5$) and weak concurvity ($\sigma_1 = 0.9$). The response variable is generated by equation (1.3). That is, the true model contains only X_1 and X_3, and the parameter σ_2 controls the overall signal-to-noise ratio (SNR). We also consider three levels: high SNR ($\sigma_2 = 0.1$), medium SNR ($\sigma_2 = 0.5$) and low SNR ($\sigma_2 = 1.0$).

Table 2 lists the number of times (out of 500 repetitions) various variable combinations are selected by pGAM and by *Stepwise AM*. It can be seen that pGAM is effective at selecting the correct variables in general. The only case where it had any difficulty was the case of high concurvity and low SNR, where there is a good chance of selecting X_5 instead of X_1. In the high concurvity case, X_5 can be considered as a surrogate of X_1, so this is not a serious mistake.

Perhaps a somewhat counter-intuitive phenomenon is that *Stepwise AM* is more likely to make the mistake of including X_5 in addition to X_1 when SNR is high. We explain this in the Appendix.

Table 3 compares the predictive accuracy of pGAM and regular GAM (using the `mgcv` package in `R`) in terms of their root mean squared errors (RMSE), and root prediction squared errors (RPSE) on an independently generated test set. In general, pGAM performs

Table 2. Simulation study 1. Number of times different variable combinations are selected by pGAM (and by *Stepwise AM* in brackets), out of 500 simulations. The notation "1/5" means either X_1 or X_5. A plus (+) means one of the noise variables from $\{X_2, X_4\}$.

σ_1	σ_2	(1, 3)	(1, 3, 5)	(1/5, 3, +)	(1)	(5)	(5, 3)	(1, 3, 5, +)
0.01	0.1	497 (140)	2 (359)	1 (0)	0 (0)	0 (0)	0 (0)	0 (1)
	0.5	497 (498)	1 (1)	1 (1)	0 (0)	0 (0)	1 (0)	0 (0)
	1.0	425 (426)	0 (0)	2 (1)	1 (1)	0 (0)	72 (72)	0 (0)
0.50	0.1	498 (406)	1 (93)	1 (1)	0 (0)	0 (0)	0 (0)	0 (0)
	0.5	498 (499)	1 (0)	1 (1)	0 (0)	0 (0)	0 (0)	0 (0)
	1.0	496 (498)	1 (0)	2 (1)	1 (1)	0 (0)	0 (0)	0 (0)
0.90	0.1	498 (472)	1 (27)	1 (1)	0 (0)	0 (0)	0 (0)	0 (0)
	0.5	498 (499)	1 (0)	1 (1)	0 (0)	0 (0)	0 (0)	0 (0)
	1.0	496 (498)	1 (0)	2 (1)	1 (1)	0 (0)	0 (0)	0 (0)

Table 3. Simulation study 1. Differences between GAM and pGAM in terms of root mean squared error (RMSE) and root prediction squared error (RPSE) on independent test sets. "DIFF-RMSE" = RMSE(GAM) − RMSE(pGAM) and likewise for "DIFF-RPSE," so positive differences indicate that pGAM performed better than GAM.

		DIFF-RMSE		DIFF-RPSE	
Concurvity	SNR	mean	(stdev)	mean	(stdev)
Strong	High ($\sigma_2 = 0.1$)	0.0033	(0.0036)	0.0005	(0.0010)
($\sigma_1 = 0.01$)	Medium ($\sigma_2 = 0.5$)	0.0072	(0.0189)	0.0008	(0.0040)
	Low ($\sigma_2 = 1.0$)	−0.0018	(0.0486)	−0.0011	(0.0098)
Medium	High ($\sigma_2 = 0.1$)	0.0022	(0.0037)	0.0004	(0.0009)
($\sigma_1 = 0.50$)	Medium ($\sigma_2 = 0.5$)	0.0142	(0.0199)	0.0021	(0.0043)
	Low ($\sigma_2 = 1.0$)	0.0274	(0.0400)	0.0038	(0.0083)
Weak	High ($\sigma_2 = 0.1$)	0.0021	(0.0036)	0.0003	(0.0009)
($\sigma_1 = 0.90$)	Medium ($\sigma_2 = 0.5$)	0.0134	(0.0192)	0.0019	(0.0041)
	Low ($\sigma_2 = 1.0$)	0.0264	(0.0397)	0.0036	(0.0082)

slightly better than GAM, but the differences are not significant. While concurvity greatly affects GAM's ability to estimate the functional effect of each component (see lower panels in Figure 3), we do not in general expect it to significantly affect GAM's ability to make predictions. This is analogous to the effects of collinearity on parameter estimation versus prediction in linear regression. The results here serve mostly as a reassurance that pGAM's

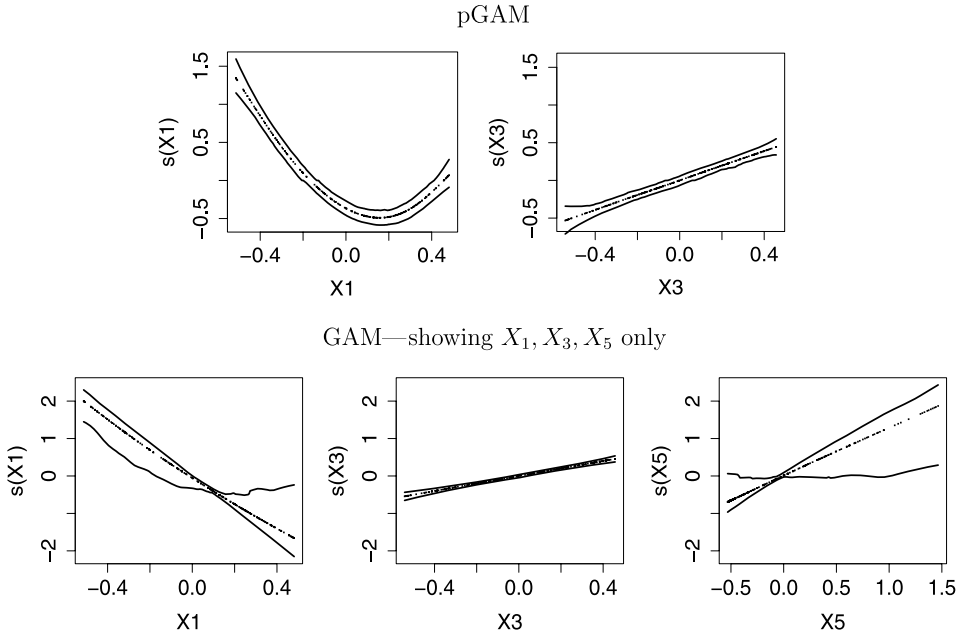

Figure 3. Simulation study 1, strong concurvity ($\sigma_1 = 0.01$) and medium SNR ($\sigma_2 = 0.5$) case. Functional effects as estimated by pGAM (upper panels) and by GAM (lower panels). Pointwise means together with 5th and 95th percentiles (497 out of 500 simulations for pGAM and 500 simulations for GAM).

improved ability at estimating the underlying functional effects does not come at a cost to predictive capability.

Figure 3 provides a graphic illustration of how well pGAM and GAM estimate the underlying functional effects in the "strong concurvity, medium SNR" case ($\sigma_1 = 0.01$, $\sigma_2 = 0.5$). Table 2 shows that, in this case, pGAM picks the right variable combination (X_1, X_3) 497 times (out of 500 simulations). Conditional on pGAM picking the right variable combination, the upper panels in Figure 3 show the functional effects of X_1 and X_3 as estimated by pGAM. The lower panels in Figure 3 show the similar pictures for GAM (from all 500 simulations) on X_1, X_3 and X_5—the estimated effects of X_2 and X_4 are basically 0 and not shown. Here, we can clearly see that GAM is significantly affected by the concurvity between X_1 and X_5.

We also experimented a little with different significance levels. We chose $\alpha = 0.001$ because that worked well—only producing at most $1/500$ false negatives. We also tried setting $\alpha = 0.0001$, and found that (on the same data set) this increased the false negative rate of all three low SNR cases to $17/500$. Therefore, we do not recommend setting α to be smaller than 0.001 unless there is reason to believe that SNR is high.

3.2 SIMULATION 2

The second simulation is designed to study a more complicated situation. First, six iid covariates X_j ($j = 1, \ldots, 6$) are generated uniformly on the interval $[0, 1]$. Then, three more covariates are generated to have various degrees of concurvity with the first six, as follows:

$$X_7 = X_6^3 + N(0, \sigma_1^2),$$
$$X_8 = X_1^2 + N(0, \sigma_1^2),$$
$$X_9 = X_2 X_3 + N(0, \sigma_1^2).$$

The response variable Y is generated by

$$Y = 2X_1^3 + 2\sin X_3 + \exp(X_4) + X_8^2 + N(0, \sigma_2^2).$$

That is, the true model contains only $\{X_1, X_3, X_4, X_8\}$, and there is concurvity among three sets of variables: X_6 and X_7, neither of which is included in the true model; X_1 and X_8, both of which are included in the true model; and finally, X_2, X_3 and X_9, with only one of them (X_3) being included in the true model. The parameters σ_1 and σ_2 play the same roles as in Simulation 1 and the same levels are considered.

Table 4 lists the number of times (out of 500 repetitions) different models are selected by pGAM and by *Stepwise AM*. In cases of medium and weak concurvity, both procedures perform well. In cases of strong concurvity ($\sigma_1 = 0.01$), there is a much higher chance for pGAM to include only one of X_1 and X_8, but this can hardly be called a mistake because, when concurvity is strong, X_1 and X_8 are almost interchangeable, so either one is adequate. On the other hand, in the case of strong concurvity and high SNR ($\sigma_1 = 0.01, \sigma_2 = 0.1$),

Table 4. Simulation study 2. Number of times different variable combinations are selected by pGAM (and by *Stepwise AM* in brackets), out of 500 simulations. The notation "1/8" means either X_1 or X_8. A plus (+) means ≥1 noise variable from $\mathcal{X}_N = \{X_2, X_5, X_6, X_7\}$, and a star (*) means ≥0 noise variables from $\mathcal{X}_N$, e.g., 1348+ means models including X_1, X_3, X_4, X_8 and one or more variables from $\mathcal{X}_N$, while (1/8)349* includes models 1349, 8349, 1349+ and 8349+.

σ_1	σ_2	(1/8)34	(1/8)34+	(1/8)349*	1348	1348+	13489*	489
0.01	0.1	428 (4)	2 (0)	0 (1)	70 (360)	0 (3)	0 (132)	0 (0)
	0.5	498 (457)	2 (10)	0 (1)	0 (32)	0 (0)	0 (0)	0 (0)
	1.0	492 (490)	3 (10)	4 (0)	0 (0)	0 (0)	0 (0)	1 (0)
0.50	0.1	0 (0)	0 (0)	0 (0)	496 (494)	4 (6)	0 (0)	0 (0)
	0.5	0 (0)	0 (0)	0 (0)	498 (496)	2 (4)	0 (0)	0 (0)
	1.0	0 (0)	0 (0)	0 (0)	496 (495)	4 (5)	0 (0)	0 (0)
0.90	0.1	0 (0)	0 (0)	0 (0)	495 (495)	5 (5)	0 (0)	0 (0)
	0.5	0 (0)	0 (0)	0 (0)	500 (493)	0 (7)	0 (0)	0 (0)
	1.0	0 (0)	0 (0)	0 (0)	497 (493)	2 (7)	1 (0)	0 (0)

there is a much inflated chance for *Stepwise AM* to include X_9, which has some concurvity with X_3. This is a far more serious mistake, because X_9 has much weaker concurvity with X_3 than the concurvity between X_1 and X_8, so X_3 and X_9 are not interchangeable.

Finally, Table 5 shows the differences between pGAM and regular GAM in terms of their RMSEs, and RPSEs on an independently generated test set. Most differences are not statistically significant, except in two cases, where pGAM has significantly lower RMSE and RPSE than GAM.

4. REAL DATA EXAMPLES

We now illustrate the use of pGAM with two real-data examples. In the first example, the response variable Y is Gaussian; in the second one, it is Poisson.

Table 5. Simulation study 2. Differences between GAM and pGAM in terms of root mean squared error (RMSE) and root prediction squared error (RPSE) on independent test sets. "DIFF-RMSE" = RMSE(GAM) − RMSE(pGAM) and likewise for "DIFF-RPSE," so positive differences indicate that pGAM performed better than GAM. A star (*) in the last column indicates the differences are statistically significant.

		DIFF-RMSE		DIFF-RPSE		
Concurvity	SNR	mean	(stdev)	mean	(stdev)	SIG
Strong	High ($\sigma_2 = 0.1$)	0.0085	(0.0056)	0.0024	(0.0022)	
($\sigma_1 = 0.01$)	Medium ($\sigma_2 = 0.5$)	−0.0241	(0.0213)	−0.0045	(0.0055)	
	Low ($\sigma_2 = 1.0$)	−0.0435	(0.0434)	−0.0076	(0.0101)	
Medium	High ($\sigma_2 = 0.1$)	0.1584	(0.0391)	0.1101	(0.0360)	*
($\sigma_1 = 0.50$)	Medium ($\sigma_2 = 0.5$)	0.0952	(0.0467)	0.0318	(0.0223)	
	Low ($\sigma_2 = 1.0$)	0.0524	(0.0610)	0.0140	(0.0203)	
Weak	High ($\sigma_2 = 0.1$)	0.1024	(0.0184)	0.0668	(0.0138)	*
($\sigma_1 = 0.90$)	Medium ($\sigma_2 = 0.5$)	0.0438	(0.0250)	0.0130	(0.0103)	
	Low ($\sigma_2 = 1.0$)	0.0049	(0.0440)	0.0014	(0.0151)	

4.1 OZONE DATA

We start by analyzing the ozone data set, a well-known and favorite data set widely used as a benchmark in the nonparametric regression and GAM literature (e.g., Breiman and Friedman 1985; Donnell, Buja, and Stuetzle 1994). The data set contains ozone concentration and eight other meteorological measurements for the Los Angeles area on 330 days in 1976 (Table 6). We treat "ozone" as the response variable and all other variables as covariates. We also standardize all variables to have mean zero and variance one before conducting the analysis.

Regardless of whether the control parameter α is set to 0.1, 0.05, or 0.01, the same set of covariates is selected by pGAM: "temp," "ibh," "humidity," "doy," "vis," and "dpg," in this order. Figure 4 shows the estimated effects of these six variables estimated by pGAM and by regular GAM; the estimated effects of the other three variables by regular GAM are essentially linear and not shown.

To verify that pGAM is an appropriate method for these data, we performed a 10-fold cross-validation, and got a mean squared prediction error (MSPE) of 0.265, with standard deviation (SD) 0.021, for pGAM, compared to a MSPE of 0.206 with SD 0.018 for GAM. It is not surprising, or overly disappointing that GAM is able to achieve better predictive accuracy, since GAM-fitting algorithms were developped to maximize predictive accuracy, while pGAM focusses on model simplification and variable selection. Therefore, the fact that the predictive errors are comparable is satisfactory, and indicates to us that pGAM is a suitable method for studying this data set. Nevertheless, we hope that future improvements to pGAM-fitting algorithms might be able to match, or perhaps even surpass, GAM-fitting methods for predictive accuracy.

The most significant differences between the estimated effects are: (i) the effect of "temp" is much closer to a simple linear effect in pGAM; (ii) for GAM, "humidity" is not a significant covariate (p-value $= 0.06$), whereas, for pGAM, after removing the partial effects of "temp" and "ibh," "humidity" becomes a significant covariate (p-value $=$ 0.0003)—visually, we can also see that the effect of "humidity" is much less flat in pGAM; (iii) the effects of "doy" and "dpg" as estimated by pGAM peak at different locations from those estimated by GAM (more on this below).

Donnell, Buja, and Stuetzle (1994, pp. 1642–1646) analyzed this data set and found the following concurvities: given "ibh," there is a positive relationship between "temp" and "ibt"; given "temp," there is a negative relationship between "ibh" and "ibt"; and the

Table 6. Variables in the ozone data set.

Name	Description	Name	Description
ozone	Logarithm of ozone concentration	humidity	Humidity (%)
temp	Sandburg Air Force Base temperature	wind	Wind speed (mph)
dpg	Daggert pressure gradient	ibh	Inversion base height
ibt	Inversion base temperature	vis	Visibility in miles
vh	Vandenburg 500 millibar pressure height	doy	Day of the year

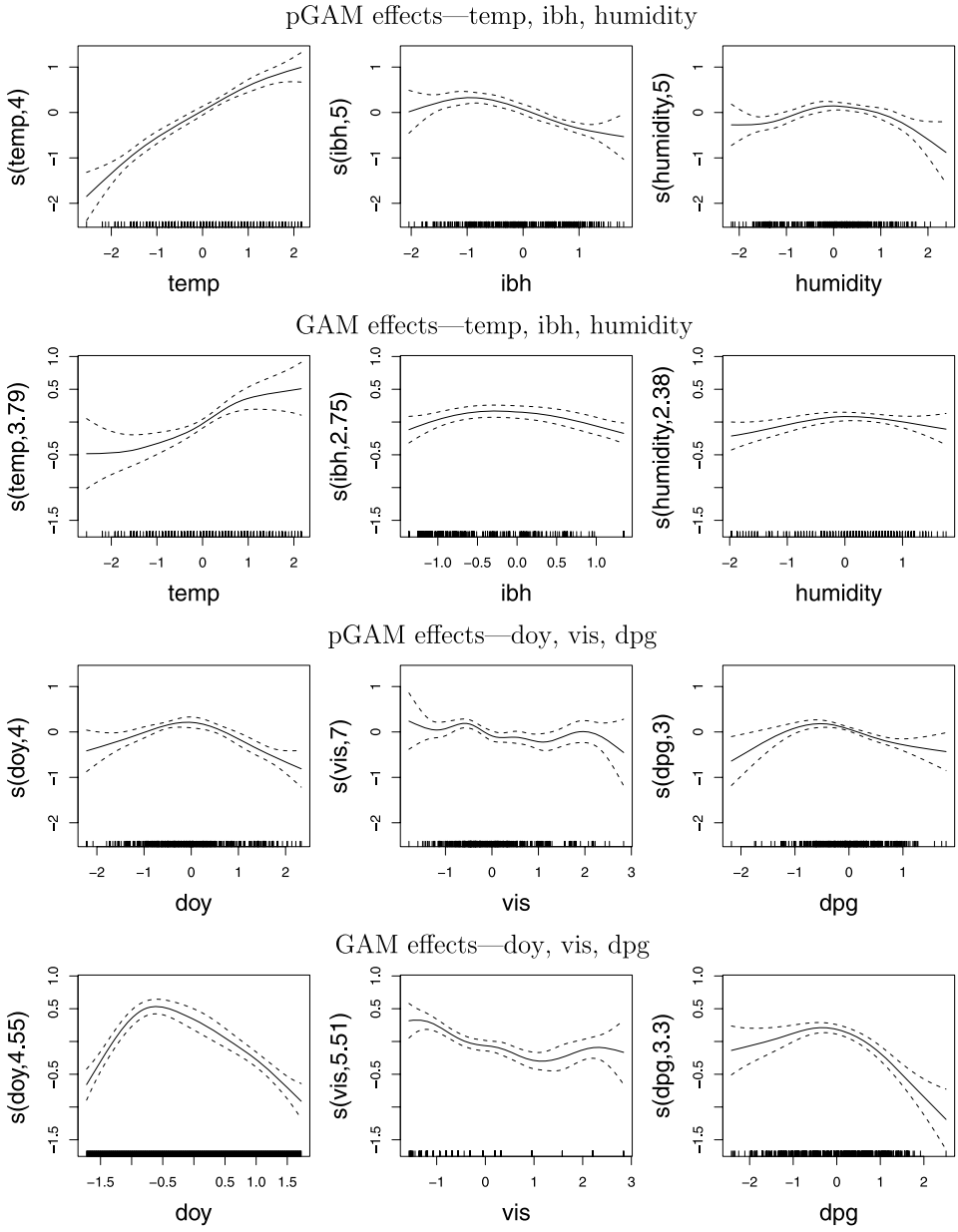

Figure 4. Ozone data set. Effects of covariates estimated by pGAM (1st and 3rd rows) and by regular GAM (2nd and 4th rows). The regular GAM also includes three other variables: "vh," "wind," and "ibt"; their estimated effects are essentially linear and not shown.

covariates "temp" and "vh" tend to increase together. We can see that pGAM has successfully detected and removed these concurvities: the variables "temp" and "ibh" are included whereas the variables "ibt" and "vh" are left out of the model. Donnell, Buja, and Stuetzle (1994) also found that there is a strong and complex (nonlinear) relationship involving

"ozone," "temp," "dpg," and "doy." This suggests that "temp," "dpg," and "doy" are all important covariates when "ozone" is treated as a response variable in a regression model. Indeed, pGAM includes all of them.

Breiman and Friedman (1985) analyzed this data set with their algorithm, ACE. There is good agreement between the variables selected by pGAM and by ACE (see Breiman and Friedman 1985, figure 5), except that ACE excluded the variable "humidity." Breiman and Friedman (1985, p. 587) also reported an extremely interesting finding: if "doy" was used as the single covariate, then its nonlinear effect on ozone actually peaked in late July and early August, whereas, in the ACE model together with other covariates, the peak effect was shifted to the beginning of May (see their figure 5(f) and (g)). This was "puzzling to [them], since the highest pollution days occur from July to September." Our analysis using pGAM suggests that their puzzle was caused by subtle concurvities among the covariates. After removing these concurvities, we see (Figure 4) that the peak effect of "doy" does in fact occur much later than May and somewhere in late July, which agrees with common sense. This is a vivid real-life example showing how concurvity can "mess up" the estimated functional effects in regular GAM and how pGAM can fix this problem.

4.2 AIR POLLUTION AND MORTALITY DATA

GAMs are widely used by environmental epidemiologists to study the relationship between mortality and air pollution (e.g., Ramsay, Burnett, and Krewski 2003). Typically, the response variable, y_t, is the number of deaths or incidents (e.g., pneumonia) in a given day, which is modelled as Poisson(λ_t) with

$$\log(\lambda_t) = f(t) + \sum_{j=1}^{d} g_j(x_{jt}) + h(z_t), \tag{4.1}$$

where x_{jt} $(j = 1, 2, \ldots, d)$ are daily measurements of d covariates, such as temperature and humidity, and the term z_t is the daily concentration of a particular type of air pollutant that is of interest, such as carbon monoxide (CO), nitrogen dioxide (NO_2) or particulate matter. According to Ramsay, Burnett, and Krewski (2003), there often exists strong concurvity between the time component $f(t)$ and the pollution component $h(z_t)$.

The web site *http://www.ihapss.jhsph.edu/data/NMMAPS/R/* contains daily mortality, air pollution, and weather data originally assembled as part of the National Mortality, Morbidity, and Air Pollution Study (NMMAPS). Here, we analyze a small subset of this data to further illustrate our methodology. We use data for the city of Philadelphia between 1995 and 2000 and consider a simple model with four possible covariates. Table 7 lists our variables using the notation of model (4.1).

To simplify matters even further, we also specify the degrees of freedom for all the nonlinear functions a priori: $df(\text{time}) = 20$, $df(\text{temp}) = 3$, $df(\text{dptp}) = 3$, and $df(\text{pollutant}) = 2$. These specifications are suggested to us by the principal author of Ramsay, Burnett, and Krewski (2003) as being typical in the environmental epidemiology literature.

Again, we used 10-fold cross-validation to check the suitability of pGAM for this data set. We found that pGAM had a MSPE of 8.93, with SD 0.19, compared to a MSPE of 8.90 and SD 0.19 for GAM. This is not a significant difference. This is particularly pleasing,

Table 7. Variables in the Philadelphia air pollution and mortality data set.

Variable	Name	Description
y_t	mortality	Number of non-accidental deaths in age group 65–75
t	time	Measured in days, i.e., $1, 2, \ldots, 2191$
z_t	pollutant	Daily NO_2 concentration
x_{1t}	temp	Average daily temperature
x_{2t}	dptp	Daily dewpoint temperature

because the theory behind pGAM is based on additive models (identity link function), so to see it working well even for generalized additive models (nonlinear link function) is very encouraging.

After fitting a regular GAM on this data set, we find that the most significant additive components are $f(t)$ and $h(z_t)$, with p-values $= 0.000$ and 0.020, respectively; the component $g_1(x_{1t})$ is marginally significant (p-value $= 0.030$); and the component $g_2(x_{2t})$ is statistically insignificant (p-value $= 0.745$). Not surprisingly, pGAM selects only two components, $f(t)$ and $h(z_t)$, with p-values $= 0.000$ and 0.002, respectively. Notice that the (nominal) significance level for the pollution component $h(z_t)$ is ten times more dramatic in pGAM, as a result of having adjusted for various concurvity effects among the covariates.

Figure 5 shows all the partial effects. It is perhaps not surprising that we should find a strong nonlinear and periodic relationship between t (time) and x_{1t} (temp) as well as between t (time) and x_{2t} (dptp). This suggests the nonlinear functions $g_1(x_{1t})$ and $g_2(x_{2t})$ are redundant given that the function $f(t)$ is included in the model, which is why pGAM excludes them. Figure 5 also shows the concurvity effect between t (time) and z_t (pollutant).

The final results from pGAM are displayed in Figure 6. Overall, our analysis suggests that mortality was high in winter and low in summer, but it was slowly decreasing during the period of 1995–2000. Most importantly, we find that, after adjusting for the strong seasonal effects, nitrogen dioxide pollution still appears to significantly increase mortality for the elderly population.

5. SUMMARY AND DISCUSSIONS

Using MI as a conceptual framework, we studied fitting GAMs when the covariates exhibit concurvity. First, we explained that fitting a GAM is equivalent to maximizing MI. Next, we explained how the back-fitting algorithm starts by attempting to do it sequentially. Then, we proposed an alternative fitting method, pGAM, which incorporates a variable selection procedure and gives better estimates of the covariates' functional effects when concurvity exists. The advantages of pGAM are illustrated and confirmed by simulation experiments and real-data examples. These advantages are especially important when the main purpose behind using GAMs is to find the most important covariates and

to understand their functional effects. Since pGAM is essentially a forward stepwise algorithm, we expect it to scale well to problems where the number of potential covariates is large. From the experimentation with significance levels, for problems with up to about 100 variables, pGAM can be expected to have reasonably small false positive and false negative rates. For larger problems, we expect pGAM to include a small number of noise variables (a common problem shared by all variable selection procedures that are based on hypothesis tests), but it is still useful as a first-step variable selection tool. Unlike other existing GAM fitting procedures which are mostly based on incorporating proper shrinkage strategies to obtain better numeric stability and predictive accuracy, pGAM explicitly deals with the problem of concurvity by transforming the covariates (to remove concurvity) and selecting informative variables. The main focusses are on model simplification and interpretation. Our simulations and real data examples also show that pGAM has comparable predictive accuracy to the current best GAM fitting procedure.

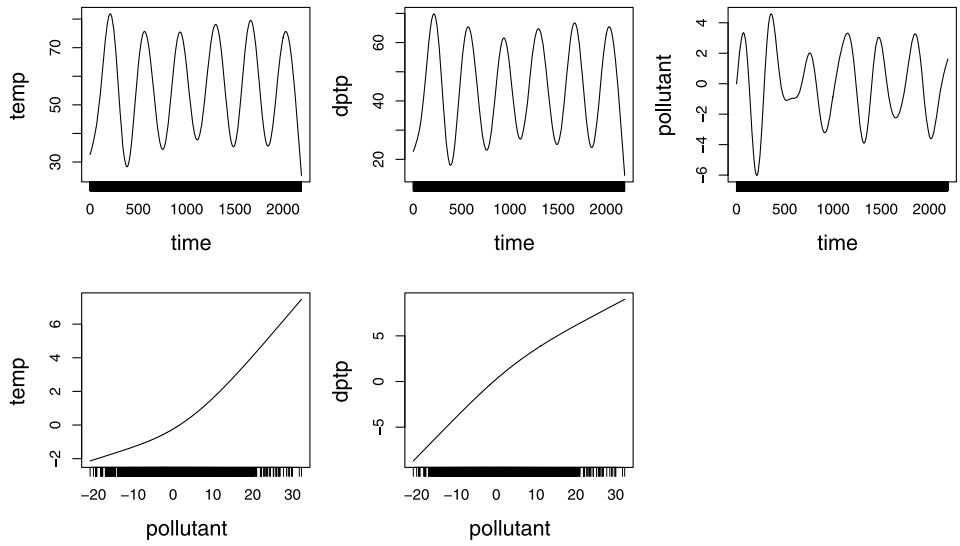

Figure 5. Philadelphia air pollution and mortality data. Partial effects estimated by pGAM.

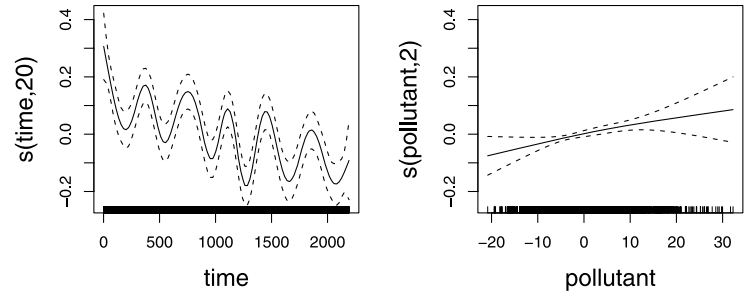

Figure 6. Philadelphia air pollution and mortality data. Effects of covariates estimated by pGAM. Only two covariates, t (time) and z_t (pollutant), are selected by pGAM.

Accurately estimating the mutual information between two random variables X and Y remains a difficult task. In this article, we made a useful observation that, if Y given $X = x$ has an exponential-family distribution with sufficient statistic being a smooth function of x, then $\mathrm{MI}(Y; X)$ can be estimated by first maximizing the conditional log-likelihood of Y given X and then estimating the entropy of Y alone; see equation (2.15). Here, a closely related procedure is Breiman and Friedman's alternating conditional expectations (ACE; Breiman and Friedman 1985). Given two variables X and Y, ACE finds transformations $\theta(Y)$ and $\phi(X)$ to maximize the correlation between $\theta(Y)$ and $\phi(X)$: $\rho_m \equiv \max_{\theta,\phi} \mathrm{corr}(\phi(X), \theta(Y))$. Clearly, one can also use ρ_m as a measure of dependence between X and Y.

We end by briefly discussing some possible further improvements to pGAM. Firstly, the assumption that $X_2 = g_{21}(X_1) + X^{(2)}$ where $X^{(2)}$ is independent of X_1 may not always hold. In this case, it may not be most appropriate to find the transformed variable $X^{(2)}$ simply by fitting a univariate GAM of X_2 onto X_1 and taking the residual; more work can be done on how to better obtain the transformed variable $X^{(2)}$.

Secondly, the approach of ordering candidate variables by their mutual information with Y, while a natural heuristic, can sometimes lead us to include an inferior variable. Therefore, it may be desirable to incorporate a method for deleting variables already in the model, if it turns out that they can be replaced by the ones added later.

SUPPLEMENTAL MATERIALS

Appendix: A description of the theoretical problem with *Stepwise AM* and a simulation demonstrating the problem. (Appendix.pdf)

Computer code: The R file "pGAMnew.R" contains the function for fitting pGAMs. The R file "StepAM.R" contains the function for selecting variables using Stepwise AM. The R files "PGAMsimu1.R" and "PGAMsimu2.R" contain the code for running pGAM, and GAM using Simon Wood's `mgcv` package, for Simulations 1 and 2, respectively. The R files "simu1stepAM.R" and "simu2stepAM.R" contain the code for variable selection using Stepwise AM, for Simulations 1 and 2, respectively. (code.zip)

ACKNOWLEDGMENTS

H. Gu and M. Zhu are supported by NSERC.

[Received November 2007. Revised June 2010.]

REFERENCES

Breiman, L., and Friedman, J. (1985), "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, 80, 580–598. [545,547,550]

Brillinger, D. R. (2004), "Some Data Analyses Using Mutual Information," *Brazilian Journal of Probability and Statistics*, 18, 163–183. [535]

Cover, T. M., and Thomas, J. A. (1991), *Elements of Information Theory*, New York: Wiley. [535,536,539]

Donnell, D. J., Buja, A., and Stuetzle, W. (1994), "Analysis of Additive Dependencies and Concurvities Using Smallest Additive Principal Components," *The Annals of Statistics*, 22, 1635–1668. [532,545,546]

Green, P. J., and Silverman, B. W. (1994), *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach. Monographs on Statistics and Applied Probability*, Vol. 58, London: Chapman & Hall. [532]

Hastie, T., and Tibshirani, R. (1986), "Generalized Additive Models," *Statistical Science*, 1, 297–318. [531,536]

——— (1990), *Generalized Additive Models*, New York: Chapman & Hall. [531-533]

Ramsay, T., Burnett, R., and Krewski, D. (2003), "The Effect of Concurvity in Generalized Additive Models Linking Mortality to Ambient Particulate Matter," *Epidemiology*, 14 (1), 18–23. [532,547]

Shannon, C. E. (1948), "A Mathematical Theory of Communication (Parts I and II)," *Bell System Technical Journal*, 27, 379–423 and 623–656. [535]

Wahba, G. (1990), *Spline Models for Observational Data. CBMS-NSF Regional Conference Series in Applied Mathematics*, Vol. 59, Philadelphia, PA: SIAM. [532]

Wood, S. N. (2000), "Modeling and Smoothing Parameter Estimation With Multiple Quadratic Penalties," *Journal of the Royal Statistical Society, Ser. B*, 62 (2), 413–428. [532]

——— (2003), "Thin Plate Regression Splines," *Journal of the Royal Statistical Society, Ser. B*, 65 (1), 95–114. [540]

——— (2004), "Stable and Efficient Multiple Smoothing Parameter Estimation for Generalized Additive Models," *Journal of the American Statistical Association*, 99, 637–686. [532]

——— (2006), "The `mgcv` Package," Online documentation of `R`. [533]