Gradient Enhanced Bayesian MARS for Regression and Uncertainty Quantification

Hayes F. Stripling and Ryan G. McClarren

Department of Nuclear Engineering , Texas A&M University, 3133 TAMU, College Station, Tx 77843-3133
h.stripling@tamu.edu rgm@tamu.edu

INTRODUCTION

In this work we describe and implement an extension of the Bayesian Multivariate Adaptive Regression Splines (BMARS) [1] emulator to include gradient information. Emulators (or response surfaces [2–5]) use available pairs of predictor ($\vec{x}_i$) and response (y_i) variables, where $y_i = f(\vec{x}_i)$, to approximate the function f at untried inputs. A "well-tuned" emulator is a valuable tool when large samples of an expensive function f are needed, as in an uncertainty quantification (UQ) study [6–9].

In many applications, the forward model which solves f also provides gradient information using adjoint or automatic differentiation methods, possibly at a small relative cost [10]. Recent work with gradient-informed kriging models [10, 11] found improvement over traditional kriging models in regression and error bound estimates. In this summary, we apply the same techniques for extending the BMARS algorithm. Regression tests on a suite of bivariate test functions as well as on a higher-dimensional particle shielding problem indicate that the inclusion of gradient information does enhance the BMARS emulator, especially for sparsely sampled input spaces.

EXTENSION OF THE BMARS EMULATOR TO INCLUDE GRADIENT INFORMATION

The BMARS algorithm uses Markov Chain Monte Carlo (MCMC) to search for a posterior distribution of MARS basis functions which minimize the error in the approximation

$$B(\vec{x}_i) \approx f(\vec{x}_i), \quad i = 1 \dots I, \quad \vec{x} \in \mathbb{R}^N.$$

For brevity, we will omit many details of the construction of the emulator (see [1, 5]), but a description of the structure of the splines is necessary for an understanding of our work.

The general MARS basis function is a sum of $\mathbf{K}$ multidimensional polynomial splines, and each of these splines (say, of dimension $\mathbf{L}_k$) is the product of $\mathbf{L}$ 1D splines. In equation form, the basis function is

$$B(\vec{x}) = \beta_0 + \sum_{k=1}^{\mathbf{K}} \beta_k \prod_{l=0}^{\mathbf{L}_k} (x_l - t_{k,l})_+^{o_{k,l}}. \tag{1}$$

The function $(y)_+$ evaluates to y if $y > 0$, else it is 0; thus, the contributing 1D splines are zero on part of the domain and have polynomial order $o_{k,l}$ on the remainder. The knot point $t_{k,l}$ is where this definition changes. Finally, the β_ks are regression coefficients which are estimated using a Bayesian least squares approach.

Our contribution is to modify the BMARS algorithm for the case that gradient information is available. We seek to minimize the error in the fit

$$\begin{aligned} B(\vec{x}_i) &\approx f(\vec{x}_i), & i &= 1 \dots I \\ \nabla_x B(\vec{x}_j) &\approx \nabla_x f(\vec{x}_j), & j &= 1 \dots J. \end{aligned}$$

Crucial to our method is the ability to write the derivative of the MARS function with respect to one dimension of its input. For example, the derivative of Eq. (1) w.r.t. dimension $n \in N$:

$$\frac{dB(\vec{x})}{dx_n} = \sum_{k=1}^{\mathbf{K}} o_{k,n} \beta_k \prod_{l=0}^{\mathbf{L}} (x_l - t_{k,l})_+^{o^*_{k,l}}.$$

where

$$o^*_{k,l} = \begin{cases} o_{k,l} - 1 & l = n \\ o_{k,l} & l \neq n \end{cases}.$$

We note that this derivative is exact (analytic) and that it retains the general form of a MARS basis function.

Our modification of the algorithm takes place primarily in the solution for the least squares coefficients. The least squares problem is written as an over-constrained linear system, namely

$$\mathbf{A}\vec{\beta} = \vec{b}, \quad \mathbf{A} \in \mathbb{R}^{P \times K}, \quad P > K.$$

The first I rows of matrix $\mathbf{A}$ contain the $\mathbf{K}$ unscaled splines $\hat{B}\left(\vec{x} \mid o_{k,l}, t_{k,l}\right)$ evaluated at each $\vec{x}_i$. The next N blocks of J rows contain the derivative of the unscaled splines with respect to the dimensions in $\vec{x}$, $\frac{d\hat{B}}{dx_n}(\vec{x})$, evaluated at each $\vec{x}_j$. Here we assume that the derivative with respect to each dimension is available at each $\vec{x}_j$, such that $P = I + N * J$. The vector $\vec{\beta}$ is the vector of regression coefficients which we seek, and the right-hand-side vector $\vec{b}$ contains the response data. In explicit form (for $n = 1 \dots N$):

$$\mathbf{A} = \begin{bmatrix} \hat{B}_1(\vec{x}_1) & \hat{B}_2(\vec{x}_1) & \dots & \hat{B}_K(\vec{x}_1) \\ \hat{B}_1(\vec{x}_2) & \hat{B}_2(\vec{x}_2) & \dots & \hat{B}_K(\vec{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{B}_1(\vec{x}_I) & \hat{B}_2(\vec{x}_I) & \dots & \hat{B}_K(\vec{x}_I) \\ \frac{d\hat{B}_1}{dx_n}(\vec{x}_1) & \frac{d\hat{B}_2}{dx_n}(\vec{x}_1) & \dots & \frac{d\hat{B}_K}{dx_n}(\vec{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{d\hat{B}_1}{dx_n}(\vec{x}_J) & \frac{d\hat{B}_2}{dx_n}(\vec{x}_J) & \dots & \frac{d\hat{B}_K}{dx_n}(\vec{x}_J) \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} f(\vec{x}_1) \\ f(\vec{x}_2) \\ \vdots \\ f(\vec{x}_I) \\ \nabla_{x_n} f(\vec{x}_1) \\ \vdots \\ \nabla_{x_n} f(\vec{x}_J) \end{bmatrix}$$

For the remainder of the algorithm, we maintain the same form of the Bayes factor, acceptance probability, MCMC step probabilities, and Bayesian least squares regression estimator for $\vec{\beta}$ as originally presented by Denison, et. al. [1].

RESULTS USING SYNTHETIC BIVARIATE FUNCTIONS

Following Denison, et. al. [1], we generate training data using five bivariate test functions. The I predictors are uniformly distributed on the unit square, and the response is assumed to be $y_i = f(\vec{x}_i) + N(0, 0.25^2)$. The test functions are written in the appendix of this summary.

To test the regression, we use the emulator to predict the value of the test function at both the original I training points and at 10 000 uniformly-spaced testing points, and compare the predictions to the true test function value. Our error metric is the fraction of variance unexplained, or FVU:

$$\mathrm{FVU} = \frac{\frac{1}{M}\sum_{i=1}^{M}\left(B(\vec{x}_i) - f(\vec{x}_i)\right)^2}{\frac{1}{M}\sum_{i=1}^{M}\left(f(\vec{x}_i) - \bar{f}\right)^2},$$

where $\bar{f}$ is the mean value of the function over the M predictor pairs.

To generate a sample from the posterior MARS distribution, we saved every 20 models from 5 000 iterations after a 50 000 iteration burn-in cycle (this was observed to be sufficient for convergence). Table 1 gives the arithmetic mean of the FVU resulting from five repetitions of the algorithm for each test function and three different values of I. The column "BMARS" includes no gradient information, while the column "gBMARS" includes gradient information w.r.t. both inputs at each of the I samples.

In most cases, the inclusion of gradient information at the predictors increases the accuracy of the fit by an order of magnitude or more. Some of the largest improvements are in the most sparse case, $I = 5$, and only in a few cases did the inclusion of gradient information result in little (or negative) improvement. Finally, we note that the case of I=15 without gradient information has the same amount of information as the case of I=5 with gradient information. In all such pairings, the fit for the former is more accurate, indicating that a spread of function evaluations yields more valuable information than an equal number of function/gradient pairs concentrated on fewer coordinates.

UNCERTAINTY QUANTIFICATION FOR A SHIELDING PROBLEM

We use a shielding example to illustrate the increased return on gradient information for a problem with seven uncertain dimensions. We model a particle flux incident on a three region, purely absorbing shield. The region thicknesses, Δz_q, $q = 1,2,3$, are 1.0, 0.25, and 1.0 cm respectively. We model the angular flux distribution as a nonlinear function of four uncertain predictors drawn from the unit hypercube (this is a common nonlinear regression test function [5]):

$$\psi(\mu|a,b,c,d) = 40\left(2\sin(\pi\mu a) + 4(b-0.5)^2 + \sin(d)(2c+d)\right).$$

Finally we assume the cross-sections $\Sigma_{a,q}$ $q = 1,2,3[cm^{-1}]$ are unknown with probability densities of $\Gamma(5,0.1)$, $\mathrm{N}(5,0.25^2)$, and $\Gamma(5,0.1)$, respectively.

Our model computes the exiting particle flow rate using S_8 Gaussian quadrature

$$J^+(Z) = \sum_{m=1}^{8}\mu_m w_m \psi_m(z=0)\prod_{q=1}^{3}\exp\left(\frac{\Sigma_{a,q}\Delta z_q}{\mu_m}\right),$$

and again our regression metric will be the fraction of variance unexplained.

Our procedure is as follows: we generate two training sets (one of 28 and one of 56 samples) with no derivative information, fit a BMARS surface using each set, and compute the FVU using 100 000 samples of our uncertain inputs (we can afford this high number of samples in this case; other emulator verification methods are more practical for expensive forward models). We then supplement each training set with seven (one

TABLE 1: FVU results for bivariate predictors

Case (I)	Training Data		Testing Data	
Simple	BMARS	gBMARS	BMARS	gBMARS
5	1.359e-03	6.371e-04	7.769e-03	1.110e-03
10	4.606e-04	2.059e-04	6.877e-04	2.769e-04
15	2.786e-04	6.666e-05	2.814e-04	1.313e-04
Radial	BMARS	gBMARS	BMARS	gBMARS
5	3.110e-01	2.055e-03	3.907e-01	4.855e-03
10	1.314e-03	2.964e-04	1.565e-03	2.255e-04
15	8.291e-04	1.171e-04	7.237e-04	8.192e-05
Harmonic	BMARS	gBMARS	BMARS	gBMARS
5	8.761e-01	3.254e-02	9.821e-01	1.176e-01
10	2.464e-03	1.179e-03	8.407e-02	3.221e-03
15	1.926e-03	3.683e-04	3.594e-03	5.553e-04
Additive	BMARS	gBMARS	BMARS	gBMARS
5	1.020e-03	1.112e-03	3.432e-01	4.009e-02
10	6.644e-04	8.399e-04	1.297e-02	3.696e-03
15	7.373e-04	4.269e-04	3.577e-03	8.133e-04
Complicated	BMARS	gBMARS	BMARS	gBMARS
5	8.828e-01	8.756e-01	8.594e-01	6.884e-01
10	1.216e-01	1.209e-01	5.322e-01	2.266e-01
15	2.166e-02	7.844e-03	1.677e-01	8.653e-03

per dimension, each randomly placed at an existing training coordinate), fit a new BMARS model, and recompute the FVU. Table 2 gives the average FVU resulting from 10 repetitions of this procedure.

TABLE 2: Shielding Problem FVU

Gradient	28 Samples	56 Samples
No	0.476	0.373
Yes	0.302	0.240

Again we see that the regression was improved when gradient information was available to the emulator. Each of these data sets are relatively sparse, and we see a fairly large improvement in both cases. Repetitions of the procedure with less than 28 samples saw less success (and in some cases worse fits), likely because the samples were too small to capture the features of the response surface. We note that the errors in these regressions are still large. On average, with these sample sizes, a third and a quarter (respectively) of the output distribution is unexplained by the emulator. Although this is probably too much regression error for meaningful application in a UQ study, we provide the results to illustrate the regression accuracy that may be gained with gradient information.

DISCUSSION AND CONCLUSIONS

In this work we demonstrated an extension of the BMARS emulator to include gradient information and conclude that the derivative information can reduce regression error. This may give reason for a modeler to pay the computational cost for gradient calculations, especially if relatively few samples of the forward model will be available. Further, we note that the addition of gradient information did not result in significant slowdown of the algorithm, which goes roughly as the $O\left(\mathbf{K}^3\right)$ work required to solve the least squares system. We observed that the size of the basis function did not systematically grow in response to the additional training information.

Future work should focus on computing some kind of importance metric for the gradient values, as we do observe that the emulator tends to over-fit gradient information, particularly near the boundary. We also hope to apply the improved emulator to a more complex simulation or UQ study.

ACKNOWLEDGEMENTS

All gradient information produced for this summary was computed using the INTLAB forward automatic differentiation package for Matlab [12]. Funding for the first author was provided by the Department of Energy Computational Science Fellowship under grant number DE-FG02-97ER25308.

APPENDIX: BIVARIATE TEST FUNCTIONS

The explicit form of the bivariate test functions are:

- Simple function

$$f(\vec{x}) = 10.391\left[(x_1 - .4)(x_2 - .6) + 0.36\right]$$

- Radial function

$$f(\vec{x}) = 24.234\left[r^2\left(0.75 - r^2\right)\right], \quad r^2 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$$

- Harmonic function

$$f(\vec{x}) = 42.659\left[0.1 + \hat{x}_1\left(0.05 + \hat{x}_1^4 - 10\hat{x}_1^2\hat{x}_2^2 + 5\hat{x}_2^4\right)\right], \quad \hat{x}_n = x_n - .5$$

- Additive function

$$\begin{aligned} f(\vec{x}) = 1.3356\{&1.5(1 - x_1) \\ &+ \exp(2x_1 - 1)\sin\left(3\pi(x_1 - .6)^2\right) \\ &+ \exp(3(x_2 - .5))\sin\left(4\pi(x_2 - .9)^2\right)\} \end{aligned}$$

- Complicated function

$$\begin{aligned} f(\vec{x}) = 1.9\{1.35 + &\exp(x_1)\sin\left[13(x_1 - .6)^2\right] \\ &\times \exp(-x_2)\sin(7x_2)\} \end{aligned}$$

REFERENCES

1. D. DENISON, B. MALLICK, and A. SMITH, "Bayesian MARS," *Statistics and Computing*, **8**, 4, 337–346 (1998).
2. D. HIGDON, M. KENNEDY, J. C. CAVENDISH, J. A. CAFEO, and R. D. RYNE, "Combining Field Data and Computer Simulations for Calibration and Prediction," *SIAM Journal of Scientific Computing*, **26**, 448–466 (2004).
3. D. DENISON, C. HOLMES, B. MALLICK, and A. SMITH, *Bayesian Methods for Nonlinear Classification and Regression*, Wiley, Chichester, England, chap. 3-4 (2002).
4. A. O'HAGAN, "Curve Fitting and Optimal Design for Prediction," *Journal of the Royal Statistical Society: Series B*, **40**, 1–42 (1978).
5. J. H. FRIEDMAN, "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, **19**, 1, 1–67 (1991).
6. R. MCCLARREN, D. RYU, and E. A. R.P. DRAKE, "A Physics Informed Emulator for Laser-Driven Radiating Shock Simulations," *Reliability Engineering and System Safety* (2010), accepted August 2010.

7. H. F. STRIPLING, R. G. MCCLARREN, C. KURANZ, M. GROSSKOPF, E. RUTTER, and B. TORRALVA, "Calibration of Uncertain Inputs to Computer Models Using Experimentally Measured Quantities and the BMARS Emulator." in "Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering," (May 2011).
8. B. WILLIAMS, D. HIGDON, J. GATTIKER, L. MOORE, M. MCKAY, and S. KELLER-MCNULTY, "Combining Experimental Data and Computer Simulations, With an Application to Flyer Plate Experiments," *Bayesian Analysis*, **1**, 765–792 (2006).
9. B. SANSÓ and C. FOREST, "Statistical Calibration of Climate System Properties," *Journal of the Royal Statistical Society: Series C*, **58**, 485–503 (2009).
10. O. RODERICK, M. ANITESCU, and P. FISCHER, "Polynomial Regression Approaches using Derivative Information for Uncertainty Quantification," *Nuclear Science and Engineering*, **164**, 122–139 (2010).
11. B. LOCKWOOD and M. ANITESCU, "Gradient-Enhanced Universal Kriging for Uncertainty Propagation," in "Proceedings of the American Nuclear Society Summer Meeting," (June 2011), to appear.
12. S. RUMP, "INTLAB - INTerval LABoratory," in T. CSENDES, editor, "Developments in Reliable Computing," Kluwer Academic Publishers, Dordrecht, pp. 77–104 (1999), `http://www.ti3.tu-harburg.de/rump/`.