Inferring Atmospheric Release Characteristics in a Large Computer Experiment using Bayesian Adaptive Splines - Supplementary Material

November 16, 2018

1 BMARS Posterior Sampling

1.1 Likelihood

If we have $(y_i, \mathbf{x}_i)$ pairs for $i = 1, \ldots, N$, we model the relationship as

$$y_i = a_0 + \sum_{m=1}^{M} a_m B_m(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2) \tag{1}$$

$$B_m(\mathbf{x}) = \prod_{j=1}^{J_m} g_{jm}[s_{jm}(x_{v_{jm}} - t_{jm})]_+^{\alpha}, \quad g_{jm} = [(s_{jm}+1)/2 - s_{jm}t_{jm}]^{-\alpha} \tag{2}$$

where g is a scaling factor to make the basis function have maximum of one. Conditional on M and basis parameters, let $\mathbf{B}$ be the matrix of basis functions with a column of ones for the intercept.

1.2 Priors

For the number of basis functions, the weights, and the error variance we use

$$M|\lambda \sim Poisson(\lambda), \ \lambda \sim IG(h_1, h_2) \tag{3}$$

$$\mathbf{a}|\mathbf{B}, \sigma^2, \tau \sim N\left(\mathbf{0}, \frac{\sigma^2}{\tau}(\mathbf{B}'\mathbf{B})^{-1}\right), \ \sigma^2 \sim IG(g_1, g_2), \ \tau \sim Ga(b_1, b_2). \tag{4}$$

The default settings of $h_1 = h_2 = 10$ are fairly robust and suggested in Denison et al. (1998); Nott et al. (2005). The default settings of $g_1 = g_2 = 0$ with $b_1 = 1/2$ and $b_2 = N/2$ result in the Zellner-Siow Cauchy version of the g-prior (Liang et al., 2008). Sometimes it makes sense to give a lower bound to the prior for σ^2 in order to prevent overfitting. Then the lower bound is another parameter, but it is zero by default.

The knots, signs, variables, and degree of interaction (all of which are used to create $\mathbf{B}$) have a discrete uniform prior over the constrained space of possibilities, constrained to have each basis function have at least b non-zero values (to prevent to local fitting at the edges). The discrete uniform constant ends up being a necessary part of the RJMCMC acceptance ratio. Counting the number of possibilities with the constraint (to get the constant) is too difficult, so we use

the unconstrained space constant as a conservative proxy. The unconstrained space has

$$s_{jm} \in \{-1, 1\} \tag{5}$$
$$t_{jm}|v_{jm} \in \{x_{1v_{jm}}, \ldots, x_{nv_{jm}}\} \tag{6}$$
$$v_{jm} \in \{1, \ldots, p\} - \{v_{km}\}_{k \neq j} \tag{7}$$
$$J_m \in \{1, \ldots, J_{\max}\} \tag{8}$$

so that the constant is

$$c_m = \left(\frac{1}{2}\right)^{J_m} \left(\prod_{j=1}^{J_m} \frac{1}{n_{v_{jm}}}\right) \binom{p}{J_{\mathrm{m}}}^{-1} \left(\frac{1}{J_{\max}}\right). \tag{9}$$

This makes

$$\pi\left(\left\{J_m, \{s_{jm}, t_{jm}, v_{jm}\}_{j=1}^{J_m}\right\}_{m=1}^{M}\right) = \prod_{m=1}^{M} c_m 1(b_m > b). \tag{10}$$

1.3 Posterior

Then the posterior up to a constant is

$$\pi(M, \lambda, \mathbf{a}, \sigma^2, \tau, \mathbf{J}, \mathbf{s}, \mathbf{t}, \mathbf{v}|\mathbf{y}) \propto N\left(\mathbf{y}|\mathbf{B}\mathbf{a}, \sigma^2\mathbf{I}\right) N\left(\mathbf{a}|\mathbf{0}, \frac{\sigma^2}{\tau}(\mathbf{B}'\mathbf{B})^{-1}\right) Pois(M|\lambda) Ga(\lambda|h_1, h_2) M!$$
$$Ga(\tau|b_1, b_2) \prod_{m=1}^{M} \left(\frac{1}{2}\right)^{J_m} \left(\prod_{j=1}^{J_m} \frac{1}{n_{v_{jm}}}\right) \binom{p}{J_m}^{-1} \left(\frac{1}{J_{\max}}\right) 1(b_m > b). \tag{11}$$

where the $M!$ accounts for all the ways you can get the M basis functions (ordering). Notice that we can rewrite $N\left(\mathbf{y}|\mathbf{B}\mathbf{a}, \sigma^2\mathbf{I}\right) N\left(\mathbf{a}|\mathbf{0}, \frac{\sigma^2}{\tau}(\mathbf{B}'\mathbf{B})^{-1}\right)$ as

$$(2\pi\sigma^2)^{-N/2}(2\pi\sigma^2/\tau)^{-(M+1)/2}|\mathbf{B}'\mathbf{B}|^{1/2} \exp\left\{\frac{-1}{2\sigma^2} \underbrace{[(\mathbf{y} - \mathbf{B}\mathbf{a})'(\mathbf{y} - \mathbf{B}\mathbf{a}) + \tau\mathbf{a}'\mathbf{B}'\mathbf{B}\mathbf{a}]}_{\mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{B}\mathbf{a} + (1+\tau)\mathbf{a}'\mathbf{B}'\mathbf{B}\mathbf{a}}\right\} \tag{12}$$
$$= (2\pi\sigma^2)^{-N/2}(2\pi\sigma^2/\tau)^{-(M+1)/2}|\mathbf{B}'\mathbf{B}|^{1/2} \exp\left\{\frac{-(1+\tau)}{2\sigma^2}\left(\mathbf{a} - \frac{\hat{\mathbf{a}}}{1+\tau}\right)'\mathbf{B}'\mathbf{B}\left(\mathbf{a} - \frac{\hat{\mathbf{a}}}{1+\tau}\right)\right\}$$
$$\exp\left\{\frac{-1}{2\sigma^2}\left[\mathbf{y}'\mathbf{y} - \frac{1}{1+\tau}\hat{\mathbf{a}}'\mathbf{B}'\mathbf{B}\hat{\mathbf{a}}\right]\right\} \tag{13}$$
$$= (2\pi\sigma^2)^{-N/2}(2\pi\sigma^2/\tau)^{-(M+1)/2}|\mathbf{B}'\mathbf{B}|^{1/2} N\left(\mathbf{a}\middle|\frac{\hat{\mathbf{a}}}{1+\tau}, \frac{\sigma^2}{1+\tau}(\mathbf{B}'\mathbf{B})^{-1}\right)\left(\frac{2\pi\sigma^2}{1+\tau}\right)^{(M+1)/2}|\mathbf{B}'\mathbf{B}|^{-1/2}$$
$$\exp\left\{\frac{-1}{2\sigma^2}\left[\mathbf{y}'\mathbf{y} - \frac{1}{1+\tau}\hat{\mathbf{a}}'\mathbf{B}'\mathbf{B}\hat{\mathbf{a}}\right]\right\} \tag{14}$$
$$= (2\pi\sigma^2)^{-N/2}\left(\frac{\tau}{1+\tau}\right)^{(M+1)/2} N\left(\mathbf{a}\middle|\frac{\hat{\mathbf{a}}}{1+\tau}, \frac{\sigma^2}{1+\tau}(\mathbf{B}'\mathbf{B})^{-1}\right) \exp\left\{\frac{-1}{2\sigma^2}\left[\mathbf{y}'\mathbf{y} - \frac{1}{1+\tau}\underbrace{\hat{\mathbf{a}}'\mathbf{B}'\mathbf{B}\hat{\mathbf{a}}}_{\hat{\mathbf{a}}'\mathbf{B}'\mathbf{y}}\right]\right\} \tag{15}$$

where $\hat{\mathbf{a}} = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{y}$. This shows that we can marginalize over $\mathbf{a}$. We could also marginalize over σ^2.

1.4 Reversible Jump MCMC

Our RJMCMC algorithm allows three types of moves, chosen with equal probability: Birth (add a basis function), death (delete a basis function), and change (move a knot/sign). We use these moves to sample $[M, \mathbf{J}, \mathbf{s}, \mathbf{t}, \mathbf{v}|\mathbf{y}, \sigma^2, \tau, \lambda]$. In these moves $\mathbf{a}$ is marginalized out, which means that we do not need to worry about a transdimensional proposal for $\mathbf{a}$. Then we can sample $[\sigma^2|\mathbf{y}, M, \mathbf{J}, \mathbf{s}, \mathbf{t}, \mathbf{v}, \tau, \lambda]$. If we then sample $[\mathbf{a}|\cdot]$, we can sample $[\tau|\cdot]$.

1.4.1 Birth

Let $b(M^C \to M^P)$ be the proposal probability of going from the current model to the proposed model (which adds one basis function). We sample a proposal by drawing an interaction order $J_{M+1} \sim I(M^C)$, a set of J_{M+1} variables (without replacement) from $z(M^C)$, and corresponding knots and signs from their respective (unconstrained) priors. Here, I is a discrete probability mass function that puts weight on interaction orders according to how often they are used in the current model (with a constant w_1 added, so none are zero). The pmf z is similar but for the variables used in the current model and with constant w_2. Thus

$$b(M^C \to M^P) = P_{\text{birth}} I(J_{M+1}|M^C) z(\mathbf{v}_{M+1}|M^C) \left(\frac{1}{2}\right)^{J_{M+1}} \prod_{j=1}^{J_{M+1}} \frac{1}{n_{v_{jM+1}}} \tag{16}$$

and the acceptance probability is the maximum of one and

$$\alpha_{\text{birth}} = \frac{[M+1, \mathbf{J}^*, \mathbf{s}^*, \mathbf{t}^*, \mathbf{v}^*|\mathbf{y}, \sigma^2, \tau, \lambda] b(M^P \to M^C)}{[M, \mathbf{J}, \mathbf{s}, \mathbf{t}, \mathbf{v}|\mathbf{y}, \sigma^2, \tau, \lambda] b(M^C \to M^P)} \tag{17}$$

and $b(M^P \to M^C) = P_{\text{death}} \frac{1}{M+1}$, as a basis function is chosen at random to kill.

$$\begin{aligned}
[M, \mathbf{J}, \mathbf{s}, \mathbf{t}, \mathbf{v}|\mathbf{y}, \sigma^2, \tau, \lambda] &\propto \left(\frac{\tau}{1+\tau}\right)^{(M+1)/2} \exp\left\{\frac{-1}{2\sigma^2}\left[\mathbf{y}'\mathbf{y} - \frac{1}{1+\tau}\hat{\mathbf{a}}'\mathbf{B}'\mathbf{y}\right]\right\} Pois(M|\lambda) M! \\
&\quad \prod_{m=1}^{M} \left(\frac{1}{2}\right)^{J_m} \left(\prod_{j=1}^{J_m} \frac{1}{n_{v_{jm}}}\right) \binom{p}{J_m}^{-1} \left(\frac{1}{J_{\max}}\right) 1(b_m > b) \qquad (18) \\
&\propto \left(\frac{\tau}{1+\tau}\right)^{(M+1)/2} \exp\left\{\frac{1}{2\sigma^2}\left[\frac{1}{1+\tau}\hat{\mathbf{a}}'\mathbf{B}'\mathbf{y}\right]\right\} \lambda^M \\
&\quad \prod_{m=1}^{M} \left(\frac{1}{2}\right)^{J_m} \left(\prod_{j=1}^{J_m} \frac{1}{n_{v_{jm}}}\right) \binom{p}{J_m}^{-1} \left(\frac{1}{J_{\max}}\right) 1(b_m > b) \qquad (19)
\end{aligned}$$

Now if $\mathbf{B}^*$ and $\hat{\mathbf{a}}^*$ are the candidate set of basis functions and least-squares weights,

$$\frac{[M+1,\mathbf{J}^*,\mathbf{s}^*,\mathbf{t}^*,\mathbf{v}^*|\mathbf{y},\sigma^2,\tau,\lambda]}{[M,\mathbf{J},\mathbf{s},\mathbf{t},\mathbf{v}|\mathbf{y},\sigma^2,\tau,\lambda]} = \left(\frac{\tau}{1+\tau}\right)^{1/2} \exp\left\{\frac{1}{2\sigma^2(1+\tau)}\left[\hat{\mathbf{a}}^*\mathbf{B}^{*\prime}\mathbf{y} - \hat{\mathbf{a}}'\mathbf{B}'\mathbf{y}\right]\right\}\lambda \left(\frac{1}{2}\right)^{J_{M+1}} \left(\prod_{j=1}^{J_{M+1}} \frac{1}{n_{v_{jM+1}}}\right) \binom{p}{J_{M+1}}^{-1} \left(\frac{1}{J_{\max}}\right) 1(b_{M+1} > b) \tag{20}$$

and

$$\alpha_{\text{birth}} = \left(\frac{\tau}{1+\tau}\right)^{1/2} \exp\left\{\frac{1}{2\sigma^2(1+\tau)}\left[\hat{\mathbf{a}}^*\mathbf{B}^{*\prime}\mathbf{y} - \hat{\mathbf{a}}'\mathbf{B}'\mathbf{y}\right]\right\}\lambda \binom{p}{J_{M+1}}^{-1} \left(\frac{1}{J_{\max}}\right) 1(b_{M+1} > b) \frac{P_{\text{death}}/(M+1)}{P_{\text{birth}} I(J_{M+1}|M^C) z(\mathbf{v}_{M+1}|M^C)} \tag{21}$$

1.4.2 Death

The death step is largely the reciprocal of the birth step:

$$\alpha_{\text{death}} = \frac{[M-1,\mathbf{J}^*,\mathbf{s}^*,\mathbf{t}^*,\mathbf{v}^*|\mathbf{y},\sigma^2,\tau,\lambda] b(M^C \to M^P)}{[M,\mathbf{J},\mathbf{s},\mathbf{t},\mathbf{v}|\mathbf{y},\sigma^2,\tau,\lambda] b(M^P \to M^C)} \tag{22}$$

$$= \left(\frac{\tau}{1+\tau}\right)^{-1/2} \exp\left\{\frac{1}{2\sigma^2(1+\tau)}\left[\hat{\mathbf{a}}^*\mathbf{B}^{*\prime}\mathbf{y} - \hat{\mathbf{a}}'\mathbf{B}'\mathbf{y}\right]\right\}(1/\lambda) \binom{p}{J_m} J_{\max} \frac{M P_{\text{birth}} I(J_m|M^P) z(\mathbf{v}_m|M^P)}{P_{\text{death}}}. \tag{23}$$

1.4.3 Change

Since there is no dimension change, the proposals and priors cancel resulting in

$$\alpha_{\text{change}} = \frac{[M,\mathbf{J},\mathbf{s}^*,\mathbf{t}^*,\mathbf{v}|\mathbf{y},\sigma^2,\tau,\lambda]}{[M,\mathbf{J},\mathbf{s},\mathbf{t},\mathbf{v}|\mathbf{y},\sigma^2,\tau,\lambda]} \tag{24}$$

$$= \exp\left\{\frac{1}{2\sigma^2(1+\tau)}\left[\hat{\mathbf{a}}^*\mathbf{B}^{*\prime}\mathbf{y} - \hat{\mathbf{a}}'\mathbf{B}'\mathbf{y}\right]\right\}. \tag{25}$$

1.5 Gibbs Steps

$$[\sigma^2|\mathbf{y},M,\mathbf{J},\mathbf{s},\mathbf{t},\mathbf{v},\tau,\lambda] \propto IG(\sigma^2|g_1,g_2)(2\pi\sigma^2)^{-N/2} \exp\left\{\frac{-1}{2\sigma^2}\left[\mathbf{y}'\mathbf{y} - \frac{1}{1+\tau}\hat{\mathbf{a}}'\mathbf{B}'\mathbf{y}\right]\right\} \tag{26}$$

$$\propto (\sigma^2)^{-N/2-g_1-1} \exp\left\{\frac{-1}{\sigma^2}\left[g_2 + \frac{1}{2}\left(\mathbf{y}'\mathbf{y} - \frac{1}{1+\tau}\hat{\mathbf{a}}'\mathbf{B}'\mathbf{y}\right)\right]\right\} \tag{27}$$

$$\sim IG\left(N/2 + g_1, g_2 + \frac{1}{2}\left[\mathbf{y}'\mathbf{y} - \frac{1}{1+\tau}\hat{\mathbf{a}}'\mathbf{B}'\mathbf{y}\right]\right) \tag{28}$$

$$[\mathbf{a}|\cdot] \sim N\left(\frac{\hat{\mathbf{a}}}{1+\tau}, \frac{\sigma^2}{1+\tau}(\mathbf{B}'\mathbf{B})^{-1}\right) \tag{29}$$

$$[\tau|\cdot] \propto N\left(\mathbf{a}|\mathbf{0}, \frac{\sigma^2}{\tau}(\mathbf{B}'\mathbf{B})^{-1}\right) Ga(\tau|b_1, b_2) \tag{30}$$

$$\propto \tau^{(M+1)/2+b_1-1} \exp\left\{-\tau\left[b_2 + \frac{1}{2\sigma^2}\mathbf{a}'\mathbf{B}'\mathbf{B}\mathbf{a}\right]\right\} \tag{31}$$

$$\sim Ga\left((M+1)/2 + b_1, b_2 + \frac{1}{2\sigma^2}\mathbf{a}'\mathbf{B}'\mathbf{B}\mathbf{a}\right) \tag{32}$$

2 Sobol' Decomposition

If we have a function $f(\mathbf{x})$ where $\mathbf{x} = (x_1, \ldots, x_p)$, we decompose it into

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^{p} f_i(x_i) + \sum_{i=1}^{p}\sum_{j>i} f_{ij}(x_i, x_j) + \cdots + f_{1\ldots p}(x_1, \ldots, x_p) \tag{33}$$

where all terms added above are orthogonal. In addition, all terms except f_0 are centered at zero. This is achieved by building each term such that

$$f_0 = \int f(\mathbf{x}) d\mathbf{x} \tag{34}$$

$$f_i(x_i) = \int f(\mathbf{x}) d\mathbf{x}_{-i} - f_0 \tag{35}$$

$$f_{ij}(x_i, x_j) = \int f(\mathbf{x}) d\mathbf{x}_{-ij} - f_i(x_i) - f_j(x_j) - f_0 \tag{36}$$

as so on, where integrals are over the bounds of each x_i. Without loss of generality, we assume that the bounds are zero and one.

We are interested in partitioning $Var(f(\mathbf{x}))$ into variance from each main effect and interaction. First, note that if we assume that x_i is a random variable with uniform distribution $f_i(x_i) = E(f(\mathbf{x})|x_i) - f_0$ and $f_0 = E(f(\mathbf{x}))$. Also note that $Var(f(\mathbf{x})) = E(f^2(\mathbf{x})) - E(f(\mathbf{x}))^2$. Now, squaring Equation 33 and integrating, we obtain

$$E(f^2(\mathbf{x})) = f_0^2 + \sum_{i=1}^{p}\int f_i^2(x_i)dx_i + \sum_{i=1}^{p}\sum_{j>i}\int f_{ij}^2(x_i, x_j)dx_i dx_j + \cdots + \int f_{1\ldots p}^2(x_1, \ldots, x_p)d\mathbf{x}$$

which lacks any crossproduct terms because all the terms are orthogonal. We could also write each term above as a variance, i.e., $Var(f_i(x_i)) = \int f_i^2(x_i)dx_i - 0$. We subtract 0 because $(\int f_i(x_i)dx_i)^2 = 0$. This is the case for each term. Thus,

$$E(f^2(\mathbf{x})) = f_0^2 + \sum_{i=1}^{p} Var(f(x_i)) + \sum_{i=1}^{p}\sum_{j>i} Var(f(x_i, x_j)) + \cdots + Var(f_{1\ldots p}(x_1, \ldots, x_p))$$

and we have that

$$Var(f(\mathbf{x})) = \sum_{i=1}^{p} Var(f(x_i)) + \sum_{i=1}^{p}\sum_{j>i} Var(f(x_i, x_j)) + \cdots + Var(f_{1\ldots p}(x_1, \ldots, x_p)),$$

thus decomposing the variance of f into variance due to each main effect and interaction. Note that the way we construct the main effects and interactions in Equations 34 through 36 is

sequential, so that a two way interaction functions is the effect after taking into account the two main effects.

We outline some practical considerations for obtaining the variance decomposition above, as discussed in Chen et al. (2005). First, if $\mathbf{u} \subseteq \{1, \ldots, p\}$ of size s and $\mathbf{x_u} = \{x_{u_1}, \ldots, x_{u_s}\}$, then we write the effect $f_u(\mathbf{x_u})$ (interaction if $s > 1$, main effect if $s = 1$) can be written as

$$f_u(\mathbf{x_u}) = \hat{f}_u(\mathbf{x_u}) - \sum_{\mathbf{v} \in \{P(\mathbf{u}) - \mathbf{u} - \emptyset\}} f_v(\mathbf{x_v}) - f_0$$
$$\hat{f}_u(\mathbf{x_u}) = \int f(\mathbf{x}) d\mathbf{x}_{-\mathbf{u}}$$

where $P(\mathbf{u})$ is the power set of $\mathbf{u}$. Note that $\hat{f}_u$ denotes the non-centered version of f_u. This recursive definition gives rise (with some algebra) to a simpler formulation only in terms of the non-centered effects,

$$f_u(\mathbf{x_u}) = \sum_{\mathbf{v} \in P(\mathbf{u})} (-1)^{|\mathbf{u}| - |\mathbf{v}|} \hat{f}_v(\mathbf{x_v})$$

where we define $\hat{f}_v(\mathbf{x_v}) = f_0$ if $\mathbf{v} = \emptyset$. With some further algebra, we can see that

$$Var(f_u(\mathbf{x_u})) = \sum_{\mathbf{v} \in \{P(\mathbf{u}) - \emptyset\}} (-1)^{|\mathbf{u}| - |\mathbf{v}|} Var\left(\hat{f}_v(\mathbf{x_v})\right)$$

and it is easy to show that

$$Var\left(\hat{f}_u(\mathbf{x_u})\right) = \int \hat{f}_u^2(\mathbf{x_u}) d\mathbf{x_u} - f_0^2. \tag{37}$$

Thus, if we can evaluate Equation 37 analytically, we can obtain the variance decomposition analytically.

2.1 BASS Sobol' Decomposition

The Sobol' decomposition for the BASS model can be done analytically. The details for obtaining the Sobol' decomposition and functional Sobol' decomposition when inputs are continuous are in Francom et al. (2018). Here, we will describe the details when we have categorical inputs. We will also describe how to get functional sensitivity indices when we use the EOF approach to emulation described in this paper.

First, we review (Chen et al., 2005) that when we use a tensor product basis function approach such that $f(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_m \prod_{j=1}^{J_m} h_{jm}(x_{v_{jm}})$, the quantities we need to obtain have to do with the integral of h. To simplify the notation, and without loss of generality, rewrite the previously stated function in terms of interactions between all the variables rather than only the variables active in the particular basis function $f(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_m \prod_{v=1}^{p} h_{vm}(x_v)$. Then the integrals required are $C^1_{vm} = \int h_{vm}(x_v) dx$ and $C^2_{vm_1 m_2} = \int h_{vm_1}(x_v) h_{vm_2}(x_v) dx_v$. These quantities are discussed when inputs are continuous in Francom et al. (2018). For categorical

inputs we use sums rather than integrals, so that

$$C^1_{vm} = \frac{1}{|D_v|}\sum_{z\in D_v} 1(z \in D_{vm}) = \frac{|D_{vm}|}{|D_v|}$$
$$C^2_{vm_1m_2} = \frac{1}{|D_v|}\sum_{z\in D_v} 1(z \in D_{vm_1})1(z \in D_{vm_2}) = \frac{|D_{vm_1} \cap D_{vm_2}|}{|D_v|}$$

where D_v is the set of all categories of variable v and D_{vm} is the subset of categories of v used in basis function m. Now if $\mathbf{u}$ is a set of variable indices,

$$Var(\hat{f}_u(\mathbf{x_u})) = \sum_{m_1=1}^{M}\sum_{m_2=1}^{M} a_{m_1}a_{m_2}\left(\prod_{l\notin\mathbf{u}} C^1_{lm_1}C^1_{lm_2}\right)\left(\prod_{l\in\mathbf{u}} C^2_{lm_1m_2}\right) - f_0^2.$$

2.2 Functional Sobol' Decomposition - EOFs

While the functional Sobol' decomposition of a functional BASS model is described in Francom et al. (2018), the model we use is slightly different. We use BASS for modeling weights for EOFs. To get a functional Sobol' decomposition in this case, we need to do a little more work.

First, consider the case where we are interested in getting Sobol' indices for the functional variables like we do the other variables. The function we are decomposing is $f(\mathbf{r},\mathbf{x}) = \sum_{i=1}^{k} K_i(\mathbf{r})w_i(\mathbf{x})$ where $\mathbf{r} = (s,t)$. If we define

$$\hat{w}_i^u(\mathbf{x_u}) = \int w_i(\mathbf{x})d\mathbf{x}_{-\mathbf{u}}$$
$$K_i = \int K_i(\mathbf{r})d\mathbf{r}$$

then we can obtain the overall mean

$$f_0 = \int\int \sum_{i=1}^{k} K_i(\mathbf{r})w_i(\mathbf{x})d\mathbf{r}d\mathbf{x} = \sum_{i=1}^{k}\int K_i(\mathbf{r})d\mathbf{r}\int w_i(\mathbf{x})d\mathbf{x}$$
$$= \sum_{i=1}^{k} K_i w_i^0$$

and the non-centered effects as

$$\hat{f}_r(\mathbf{r}) = \sum_{i=1}^{k} K_i(\mathbf{r})w_i^0$$
$$\hat{f}_u(\mathbf{x_u}) = \sum_{i=1}^{k} K_i\hat{w}_i^u(\mathbf{x_u})$$
$$\hat{f}_{ru}(\mathbf{r},\mathbf{x_u}) = \sum_{i=1}^{k} K_i(\mathbf{r})\hat{w}_i^u(\mathbf{x_u}).$$

Then we need to obtain

$$Var\left(\hat{f}_r(\mathbf{r})\right) = \int \hat{f}_r^2(\mathbf{r})d\mathbf{r} - f_0^2$$
$$= \sum_{i=1}^{k}\sum_{j=1}^{k} w_i^0 w_j^0 \int K_i(\mathbf{r})K_j(\mathbf{r})d\mathbf{r} - f_0^2 \tag{38}$$

$$Var\left(\hat{f}_u(\mathbf{x_u})\right) = \int \hat{f}_u^2(\mathbf{x_u})d\mathbf{x_u} - f_0^2$$
$$= \sum_{i=1}^{k}\sum_{j=1}^{k} K_i K_j \int \hat{w}_i^u(\mathbf{x_u})\hat{w}_j^u(\mathbf{x_u})d\mathbf{x_u} - f_0^2 \tag{39}$$

$$Var\left(\hat{f}_{ru}(\mathbf{r},\mathbf{x_u})\right) = \int\int \hat{f}_{ru}^2(\mathbf{r},\mathbf{x_u})d\mathbf{r}d\mathbf{x_u} - f_0^2$$
$$= \sum_{i=1}^{k}\sum_{j=1}^{k} \int K_i(\mathbf{r})K_j(\mathbf{r})d\mathbf{r} \int \hat{w}_i^u(\mathbf{x_u})\hat{w}_j^u(\mathbf{x_u})d\mathbf{x_u} - f_0^2 \tag{40}$$

which simplifies in Equation 38 and Equation 40 when the basis is orthogonal, but not in Equation 39. Then, the only quantity left to seek an analytical solution for is

$$\int \hat{w}_i^u(\mathbf{x_u})\hat{w}_j^u(\mathbf{x_u})d\mathbf{x_u} = a_{0i}a_{0j} + a_{0i}(w_j^0 - a_{0j}) + a_{0j}(w_i^0 - a_{0i})$$
$$+ \sum_{m_i=1}^{M_i}\sum_{m_j=1}^{M_j} a_{mi}a_{mj}\left(\prod_{l\notin\mathbf{u}} C_{lmi}^1 C_{lmj}^1\right)\left(\prod_{l\in\mathbf{u}} C_{lmij}^2\right). \tag{41}$$

Second, if we get Sobol' indices as functions of r, we have

$$f_0(r) = \sum_{i=1}^{k} K_i(r)w_i^0 \tag{42}$$

$$\hat{f}_u(\mathbf{x_u}) = \sum_{i=1}^{k} K_i(r)\hat{w}_i^u(\mathbf{x_u}) \tag{43}$$

and we need to obtain

$$Var(\hat{f}_u(\mathbf{x_u})) = \int \hat{f}_u^2(\mathbf{x_u})d\mathbf{x_u} - f_0^2(r) \tag{44}$$
$$= \sum_{i=1}^{k}\sum_{j=1}^{k} K_i(r)K_j(r) \int \hat{w}_i^u(\mathbf{x_u})\hat{w}_j^u(\mathbf{x_u})d\mathbf{x_u} - f_0^2(r) \tag{45}$$

which again utilizes Equation 41.

3 Emulator Comparison

At the request of a reviewer, we include some discussion of how to choose an emulator. Choosing an emulator is more than just choosing the nonlinear (or possibly linear) regression model that predicts the best, though good prediction is important. If the emulator cannot be evaluated

quickly, it is not useful. If it does not quantify its uncertainty, it can give unreliable results when used for other uncertainty quantification tasks, like calibration or sensitivity analysis. There are many qualitative characteristics that are important to building a good emulator. Below, we discuss important qualitative and quantitative characteristics to consider, and we evaluate a few possible emulators based on these characteristics for a few datasets. This is not meant to be a comprehensive comparison, but merely shows a limited comparison that indicates that BMARS is a suitable choice for our emulation problem.

We give particular attention to three possible emulators in the following sections: (1) treed Gaussian processes (TGP) (Gramacy and Lee, 2008), (2) BMARS as we have described in this paper, and (3) Bayesian additive regression trees (BART) (Chipman et al., 2010). We choose these three to compare because they can be used for data of the type that we have in our case study (i.e., continuous and categorical inputs, not necessarily data size). Other methods could be compared, as in Swiler et al. (2014), but a full comparison is beyond the scope of this work.

3.1 Qualitative Characteristics

Certain qualitative characteristics make some emulators more useful than others in particular situations. When quantitative accuracy is similar for different emulators, qualitative characteristics can help choose between them. In some cases, qualitative characteristics make some emulators impossible to use for given datasets. Below, we consider a number of qualitative characteristics to consider when choosing an emulator.

3.1.1 Data size

Emulators like the Gaussian process do not scale well. Traditionally this was less of a problem, as large numbers of computer model runs were less common. Today, large numbers of computer model runs are more feasible because of larger computers. In addition, as computers get larger, more complex models can be simulated. Naturally, as a model becomes a more complex function of parameters, an emulator will require more training data (more model runs) to be accurate. This puts users who would emulate with GPs, which have desirable properties but are not scalable, in the difficult position of wanting fewer model runs than what may be necessary to capture the model dynamics and, hence, to build an accurate emulator. The common question of why we need an emulator when we can get a large number of model runs misses the fact that the model runs are usually performed in parallel. Thus, while each individual model run may be expensive in time, they can run simultaneously. All of the calibration approaches that we can imagine would require at least some sequential model runs.

BMARS and BART both scale linearly with the number of model runs. Scalable approximations to GP models are popular, as well. TGP becomes more scalable as the tree part of the model becomes more complex.

3.1.2 Stationarity

Typical GP models are stationary. This is powerful when the computer model is roughly stationary. In those cases, the GP can fit much better than non-stationary approaches, especially with small sample sizes.

When stationarity is a bad assumption, GP models will have difficulty. BMARS and BART have nonstationary mean functions. There are nonstationary GP approaches, but they are likely to be costly in other ways. TGPs can partition the input space into regions of stationarity, a powerful ability, though it comes at the expense of continuity. Another approach would be to

use a GP with a nonstationary mean function, for instance, a GP with a BMARS mean function as in Chakraborty et al. (2013).

An element of the stationarity assumption has to do with global versus local fit. The traditional GP provides a global fit (global correlation lengths), where the BART model is a local fit (trees correspond to a partition of the input space). The BMARS model is something of a mixture, as some basis functions can be linear functions over most of the input space while other basis functions pertain to smaller portions of the input space.

3.1.3 Input/output types

Typical computer modeling scenarios have some continuous inputs and a continuous output. The case of multivariate output is well studied and presented in this paper (other references). Any nonlinear regression model could be applied in EOF space. However, when the inputs include categorical variables, there is additional complexity. For instance, incorporating categorical variables into a GP is not as natural incorporating them into tree models. TGP provides an approach for this, where different GP models using the continuous parameters can be used at the leaves of a tree model, which includes the categorical parameters.

BART and BMARS have more natural ways of incorporating categorical variables.

3.1.4 Degree of interaction

While the GP can model any degree of interaction theoretically, in practice, it will require large amounts of data to accurately model high order interactions. This is the best that can be expected from any emulator.

BMARS and BART both adaptively discover the degree of interaction (and similarly perform variable selection), though that degree is typically capped by the user.

3.1.5 Interpolation

The GP can be an interpolator, though most users opt to include a nugget for numerical stability and sometimes better predictive accuracy (Gramacy and Lee, 2012). The property of the GP that allows for it to interpolate, namely that the correlation of outputs depends of the distance between inputs, also results in its desirable heteroscedasticity. GP predictive uncertainty is larger when predicting farther from training inputs.

BMARS and BART are not interpolators and do not have the desirable heteroscedasticity property. However, the approach of Chakraborty et al. (2013), which is a GP with a BMARS mean function, is an interpolator and has the heteroscedasticity property. The same could be done with a BART mean function.

For our case study, interpolation is less realistic, since we are modeling in EOF space. To interpolate, we would not be able to do dimension reduction.

3.1.6 Continuity

Continuity can be a strong and useful assumption. Many computer models are known to be continuous functions of the inputs. The GP has continuity and also possibly differentiability built into it. The treed GP is discontinuous, but when used to only tree on the categorical variables, continuity can be preserved.

BMARS is continuous, though typically not differentiable. BART, as with other tree-based models, is inherently discontinuous. This can result in poor fits for small sample sizes.

3.1.7 Uses

For the purposes of calibration, accurate prediction with appropriate prediction uncertainty is important. This makes most Bayesian nonlinear regression models possibly suitable for calibration. For the purposes of adaptive design, interpolation and the heteroscedasticity of the GP can be crucial. For the purposes of sensitivity analysis, most approaches would require Monte Carlo approximations of ANOVA decompositions. The BMARS approach has analytical ANOVA decomposition (Francom et al., 2018), and the GP has at least a partial analytical ANOVA decomposition in some cases (Oakley and O'Hagan, 2004).

3.2 Quantitative Characteristics

The question of which emulator will be most accurate certainly depends on the dataset. Here, we compare three emulation methods (TGP, BMARS, and BART) based on prediction performance (including uncertainty) for three datasets, each with both categorical and continuous inputs. The datasets are (1) a synthetic example from Gramacy et al. (2010); (2) the EOF weights described in this case study, for the first EOF; and (3) the weights for the 50th EOF. We examine both the 1st and 50th EOF weights because the earlier EOFs tend to have higher signal to noise ratio than the later ones.

3.2.1 Synthetic Example

The first dataset is generated from a function taken from Gramacy et al. (2010), Section 2. The function has 11 inputs and is of the form

$$f(\mathbf{x}) = \begin{cases} 10\sin(\pi x_1 x_2) & x_{11} = 1 \\ 20(x_3 - 0.5)^2 & x_{11} = 2 \\ 10x_4 + 5x_5 & x_{11} = 3 \\ 10x_1 + 5x_2 + 20(x_3 - 0.5)^2 + 10\sin(\pi x_4 x_5) & x_{11} = 4 \end{cases} \tag{46}$$

so that 10 of the inputs are continuous (though five are noise) and one is categorical. Standard Normal noise is added to the output. We generate three ensembles of "model runs" from this model with different samples sizes (n). We term the ensembles small ($n = 50$), medium ($n = 500$), and large ($n = 5000$). For each ensemble, we fit the emulator and predict the output at 1000 new inputs. We also obtain a 90% prediction interval for each of the 1000 points. We use these to assess empirical coverage and interval length. Finally, we keep track of the time that model fitting requires. We repeat all of this around 40 times, each with a new ensemble of inputs. For reference, we also include the random forest prediction performance (with no attempt at quantifying uncertainty). We use default settings for all but the TGP approach, for which we mimic the best values used in Gramacy et al. (2010) (page 8). We exclude the TGP fit for the large ensemble case, where it takes prohibitively long to run.

Figure 1 shows the results of this simulation. We see that, for the small datasets ($n = 50$), there is not a lot of variation in the RMSE, but that BMARS can have bad coverage. This is likely a result of BMARS overfitting, which could be addressed by changing the parameters. We expected TGP to perform best for the small data case, but it could be that there was not enough data to sufficiently grow the tree model. This is a case when the small number of model runs is too few to capture some important dynamics in the model. With more data ($n = 500$), TGP performs much better, and the three Bayesian methods have fairly good coverage. In the large data case ($n = 5000$), TGP would undoubtedly perform very well, but takes prohibitively long. In that case, BMARS performs best, and both BMARS and BART perform better than

the random forest for only moderately longer training time. It is likely that there exist other implementations of the random forest that are faster.

3.2.2 Diablo Canyon Simulations - EOF 1

In this example, we opt for data that is more informative to this case study. We use the first EOF weight, where EOFs are calculated using the full dataset. The EOF weights are then subsampled to sample sizes of 50, 500, and 5000 as in the previous example. (In a perfect world, we would have taken new LHS samples rather than subsampling a LHS design, but this was not the priority of the case study.) We train the three emulators using the subsampled data, repeating that process for about 40 randomly chosen subsamples of the specified sizes. Prediction accuracy is then assessed based on prediction performance of 1000 held out samples. Results are shown in Figure 2.

For small datasets ($n = 50$) we see little variation in accuracy except for the poor coverage of BMARS, again likely because of overfitting. Here, TGP does not see as substantial an improvement in prediction RMSE as the data size increases, even as we adjusted the tree parameter priors. BMARS and BART see substantial improvement as the data size increases, and have similar predictive accuracy.

We note that deciding the number of model runs necessary for building a suitable emulator is nontrivial. Various rules of thumb exist based on strong assumptions, but often the scalability of the GP plays a role. GP users often prefer not having large numbers of model runs because of the scalability issues associated with the GP. This comes contrary to the notion in statistics that more (good) data is always better. The fact that in all the examples we present here, the BMARS model with large data outperforms the TGP model with medium data reinforces the idea that a non-GP emulator trained with more data is better than a GP emulator trained with less data. Similarly, there is some point where BMARS and BART will not be able to handle a dataset and something like nearest neighbors will perform better with more data than more complex models with less data. It is our opinion that, barring design considerations, the model runs should be used to choose the emulator rather than that the emulator should be used to choose the model runs. There are further practical considerations regarding how accurate an emulator needs to be, but those are more case-specific.

3.2.3 Diablo Canyon Simulations - EOF 50

This example is similar to the previous one but uses the 50th EOF weight rather than the first EOF weight. We expect that the later weights in the decomposition will be noisier than the early weights, as they are more apt to pick up seemingly random variations. Results are shown in Figure 3.

In this example, we see more constant performance from all the methods, again with improvement as more data are used for training.

References

Chakraborty, A., Mallick, B. K., Mcclarren, R. G., Kuranz, C. C., Bingham, D., Grosskopf, M. J., Rutter, E. M., Stripling, H. F., and Drake, R. P. (2013), "Spline-based emulators for radiative shock experiments with measurement error," *Journal of the American Statistical Association*, 108, 411–428.

Chen, W., Jin, R., and Sudjianto, A. (2005), "Analytical variance-based global sensitivity analysis in simulation-based design under uncertainty," *Journal of mechanical design*, 127, 875–886.

Chipman, H. A., George, E. I., McCulloch, R. E., et al. (2010), "BART: Bayesian additive regression trees," *The Annals of Applied Statistics*, 4, 266–298.

Denison, D. G., Mallick, B. K., and Smith, A. F. (1998), "Bayesian MARS," *Statistics and Computing*, 8, 337–346.

Francom, D., Sansó, B., Kupresanin, A., and Johannesson, G. (2018), "Sensitivity Analysis and Emulation for Functional Data using Bayesian Adaptive Splines," *Statistica Sinica*, 28, 791–816.

Gramacy, R. B., and Lee, H. K. (2012), "Cases for the nugget in modeling computer experiments," *Statistics and Computing*, 22, 713–722.

Gramacy, R. B., and Lee, H. K. H. (2008), "Bayesian treed Gaussian process models with an application to computer modeling," *Journal of the American Statistical Association*, 103, 1119–1130.

Gramacy, R. B., Taddy, M., et al. (2010), "Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed Gaussian process models," *Journal of Statistical Software*, 33, 1–48.

Liang, F., Paulo, R., Molina, G., Clyde, M. A., and Berger, J. O. (2008), "Mixtures of g priors for Bayesian variable selection," *Journal of the American Statistical Association*, 103.

Nott, D. J., Kuk, A. Y., and Duc, H. (2005), "Efficient sampling schemes for Bayesian MARS models with many predictors," *Statistics and Computing*, 15, 93–101.

Oakley, J. E., and O'Hagan, A. (2004), "Probabilistic sensitivity analysis of complex models: a Bayesian approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66, 751–769.

Swiler, L. P., Hough, P. D., Qian, P., Xu, X., Storlie, C., and Lee, H. (2014), "Surrogate models for mixed discrete-continuous variables," in *Constraint Programming and Decision Making*, Springer, pp. 181–202.

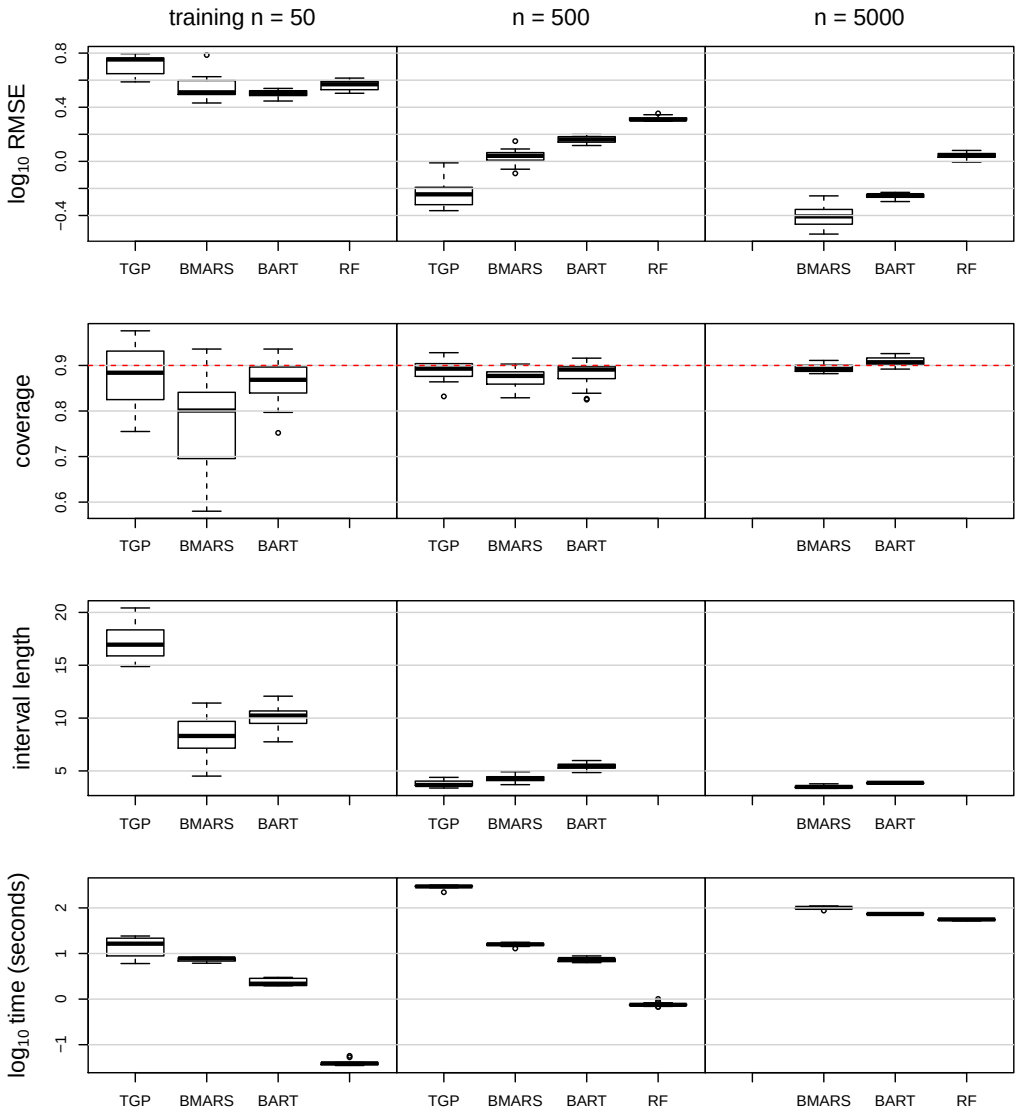

Figure 1: Comparison of emulators for the function given in Equation 46. The top row shows the root mean square error (RMSE) on the $\log_{10}$ scale. The second row shows the empirical coverage of 90% prediction intervals. The third row shows average 90% prediction interval length. The final row shows time that emulator training takes on the $\log_{10}$ scale.

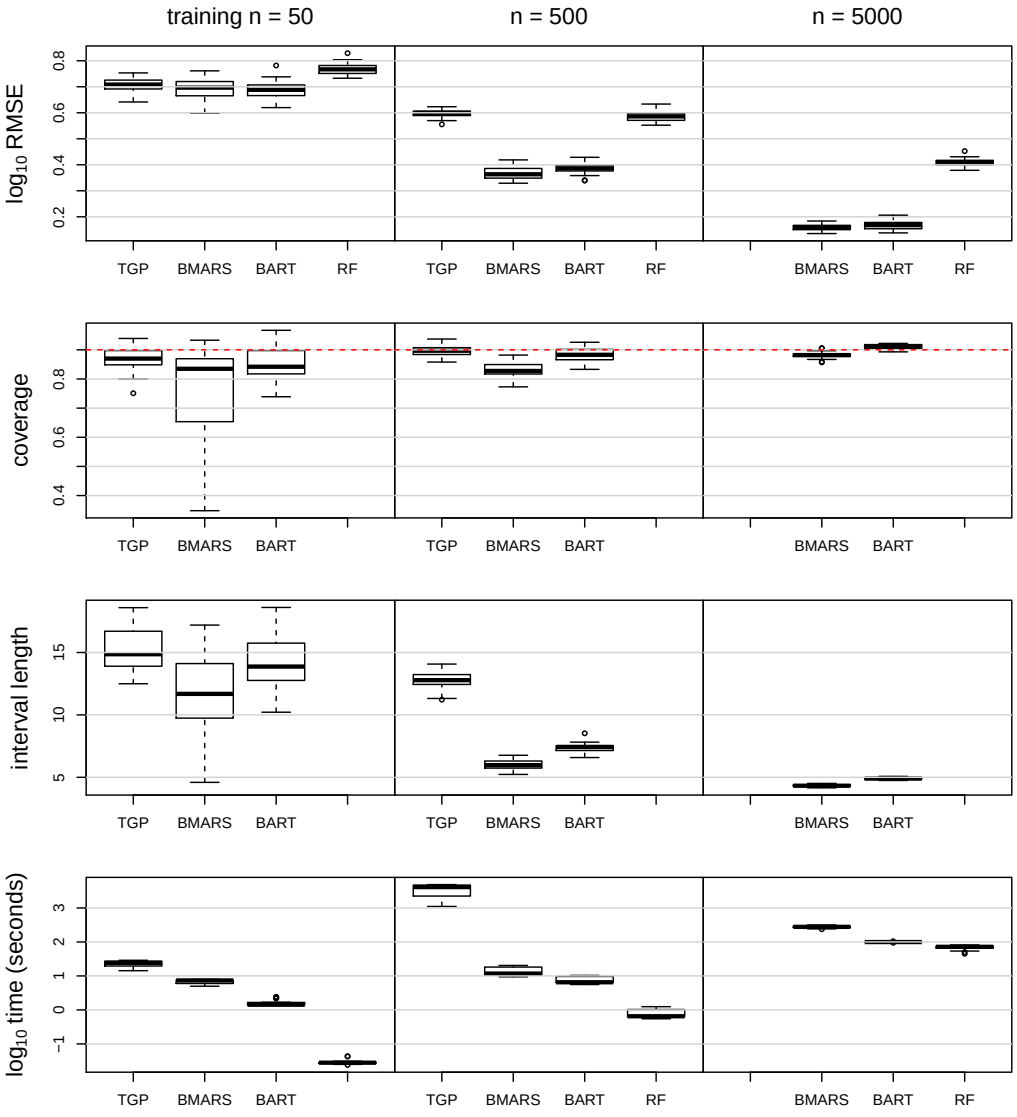

Figure 2: Same as previous figure but for EOF 1.

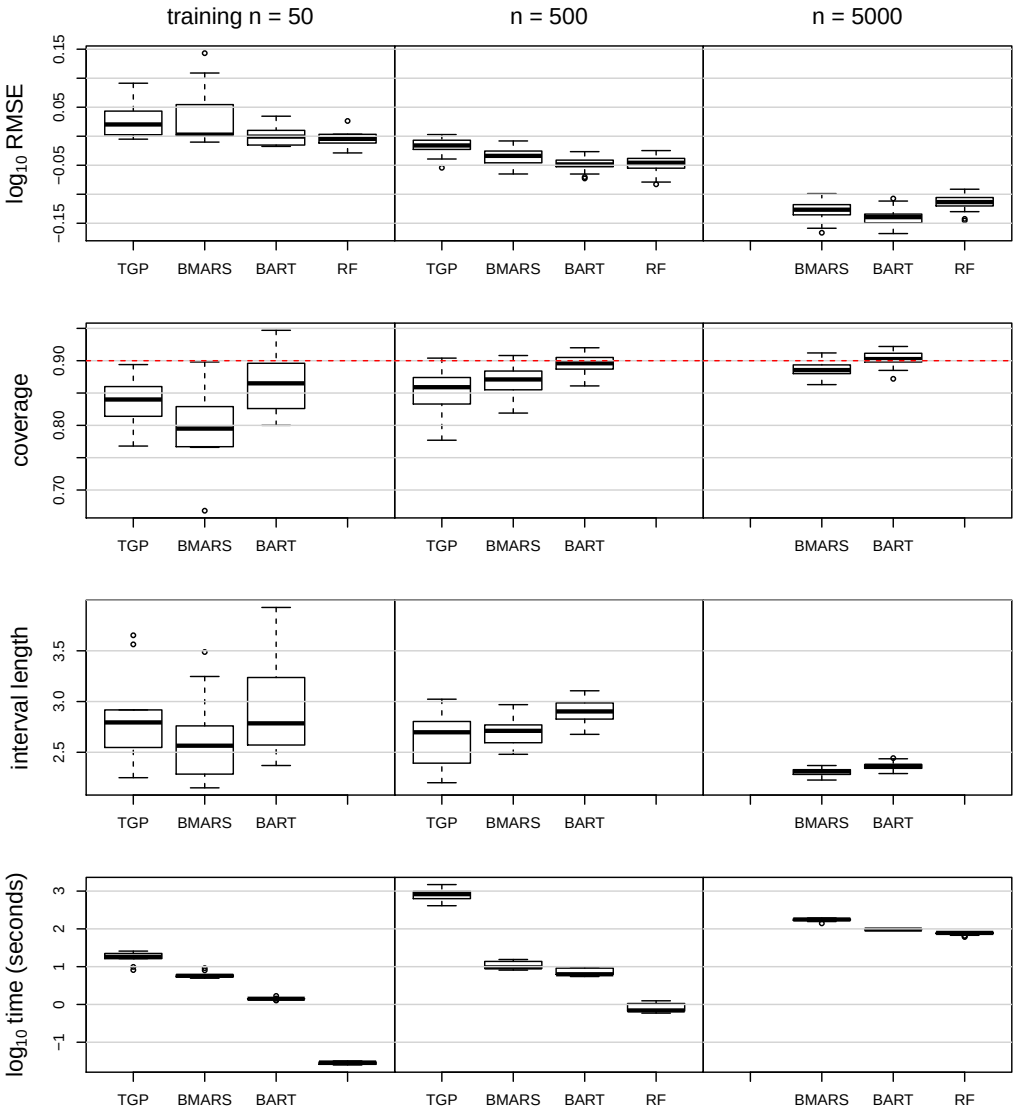

Figure 3: Same as previous figures but for EOF 50.