Supplement for:

Bayesian Generalized Additive Model Selection Including a Fast Variational Option

BY VIRGINIA X. HE AND MATT P. WAND

University of Technology Sydney

S.1 The Canonical Demmler-Reinsch Spline Basis

Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be a continuous univariate data set. In the context of this article, the x_is correspond to values of a continuous candidate predictor. Let $[a, b]$ be an interval containing the x_is. For an integer $K \leq n - 2$, let $\boldsymbol{\kappa}_{\text{inter.}} \equiv (\kappa_1, \ldots, \kappa_{K-2})$ be a set of so-called interior knots such that

$$a < \kappa_1 < \cdots < \kappa_{K-2} < b.$$

A reasonable default value for K is around 30, or smaller values if the number of unique x_is is lower. It is common to place the interior knots at sample quantiles of the x_is.

We now list steps for construction of the matrix $\boldsymbol{Z}$ containing canonical Demmler-Reinsch basis functions of the entries of $\boldsymbol{x}$. The justification for Steps (3)–(6) is given in Section 9.1.1 of Ngo & Wand (2004).

(1) Use the steps described in Section 4 of Wand & Ormerod (2008) to obtain the matrix denoted by $\boldsymbol{Z}$ in that section's equation (6), which contains canonical O'Sullivan spline basis functions. Denote this matrix by $\boldsymbol{Z}_{\text{OS}}$ and note that it has dimension $n \times K$.

(2) Form the matrix $\boldsymbol{C}_{\text{OS}} = [\mathbf{1}_n \; \boldsymbol{x} \; \boldsymbol{Z}_{\text{OS}}]$ and set $\boldsymbol{D} = \text{diag}(0, 0, \mathbf{1}_K)$.

(3) Obtain the singular value decomposition of $\boldsymbol{C}_{\text{OS}}$:

$$\boldsymbol{C}_{\text{OS}} = \boldsymbol{U}_C \text{diag}(\boldsymbol{d}_C) \boldsymbol{V}_C^T \text{ where } \boldsymbol{U}_C \text{ is } n \times (K+2) \text{ and } \boldsymbol{V}_C \text{ is } (K+2) \times (K+2)$$
$$\text{such that } \boldsymbol{U}_C^T \boldsymbol{U}_C = \boldsymbol{V}_C^T \boldsymbol{V}_C = \boldsymbol{I}_{K+2}.$$

(4) Form the symmetric matrix $\text{diag}(\mathbf{1}/\boldsymbol{d}_C) \boldsymbol{V}_C^T \boldsymbol{D} \boldsymbol{V}_C \text{diag}(\mathbf{1}/\boldsymbol{d}_C)$ and obtain its singular value decomposition:

$$\text{diag}(\mathbf{1}/\boldsymbol{d}_C) \boldsymbol{V}_C^T \boldsymbol{D} \boldsymbol{V}_C \text{diag}(\mathbf{1}/\boldsymbol{d}_C) = \boldsymbol{U}_D \text{diag}(\boldsymbol{d}_D) \boldsymbol{V}_D^T \text{ where } \boldsymbol{U}_D \text{ is } (K+2) \times (K+2)$$
$$\text{and } \boldsymbol{V}_D \text{ is } (K+2) \times (K+2) \text{ such that } \boldsymbol{U}_D^T \boldsymbol{U}_D = \boldsymbol{V}_D^T \boldsymbol{V}_D = \boldsymbol{I}_{K+2}.$$

(5) Set the full (non-canonical) Demmler-Reinsch matrix as follows: $\boldsymbol{C}_{\text{DR}} \longleftarrow \boldsymbol{U}_C \boldsymbol{U}_D$.

(6) The next steps assume that the singular value decompositions follow the convention that $\boldsymbol{d}_D$ is a $(K+2) \times 1$ vector with its entries in non-increasing order. Adjustments to the singular value decompositions are needed if this convention is not used.

(7) Set the $(K+2) \times 1$ vector $\boldsymbol{s}_D$ as follows:

$$\omega_{21} \longleftarrow \sqrt{K\text{th entry of } \boldsymbol{d}_D}, \quad ; \quad \boldsymbol{s}_D \longleftarrow \omega_{21} \mathbf{1}_{K+2} \Big/ \sqrt{\boldsymbol{d}_D}$$

and then set the last two entries of $\boldsymbol{s}_D$ to equal 1.

(8) Set the full canonical Demmler-Reinsch design matrix as follows:

$$\boldsymbol{C}_{\text{cDR}} \longleftarrow \boldsymbol{C}_{\text{DR}} \text{diag}(\boldsymbol{s}_D).$$

(9) Set the O'Sullivan to canonical Demmler-Reinsch transformation matrix as follows:

$$\boldsymbol{L}_{\text{OS.to.cDR}} \longleftarrow \boldsymbol{V}_C \text{diag}(\mathbf{1}/\boldsymbol{d}_C)\boldsymbol{U}_D \text{diag}(\boldsymbol{s}_D).$$

This $(K+2)\times(K+2)$ matrix has the following property:

$$\boldsymbol{C}_{\text{OS}}\boldsymbol{L}_{\text{OS.to.cDR}} = \boldsymbol{C}_{\text{cDR}}$$

and is useful for prediction and plotting purposes. This is because grid-wise analogues of $\boldsymbol{C}_{\text{OS}}$ are readily computed using the structures described in Wand & Ormerod (2008) involving cubic B-spline basis functions.

(10) Reverse the order of the columns of $\boldsymbol{C}_{\text{cDR}}$. Reverse the order of the columns of $\boldsymbol{L}_{\text{OS.to.cDR}}$.

(11) The matrix containing canonical spline basis functions of the inputs $\boldsymbol{x}$ and $\boldsymbol{\kappa}_{\text{inter.}}$ is

$$\boldsymbol{Z} \longleftarrow \text{the } n\times K \text{ matrix consisting of columns 3 to } K+2 \text{ of } \boldsymbol{C}_{\text{cDR}}.$$

A function in the R language for computing $\boldsymbol{Z}$ and $\boldsymbol{L}_{\text{OS.to.cDR}}$ for given $\boldsymbol{x}$ and $\boldsymbol{\kappa}_{\text{inter.}}$ can be accessed by downloading the accompanying gamselBayes package. Assuming that the gamselBayes package is installed, the relevant function is `gamselBayes:::ZcDR()`.

S.2 Approximate Marginal Log-Likelihood Expressions

The approximate marginal log-likelihood is

$$\log \underline{p}(\boldsymbol{y};\mathfrak{q}) = \begin{cases} \log \underline{p}(\boldsymbol{y};\mathfrak{q},\mathrm{C}) + E_{\mathfrak{q}}[\log\{\mathfrak{p}(\boldsymbol{y}|\beta_0,\gamma_\beta,\widetilde{\boldsymbol{\beta}},\gamma_u,\widetilde{\boldsymbol{u}},\sigma_\varepsilon^2)\}] \\ \quad +E_{\mathfrak{q}}[\log\{\mathfrak{p}(\sigma_\varepsilon^2|a_\varepsilon)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\sigma_\varepsilon^2)\}] \\ \quad +E_{\mathfrak{q}}[\log\{\mathfrak{p}(a_\varepsilon)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(a_\varepsilon)\}] & \text{Gaussian response case,} \\ \log \underline{p}(\boldsymbol{y};\mathfrak{q},\mathrm{C}) + E_{\mathfrak{q}}[\log\{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{c})\}] \\ \quad +E_{\mathfrak{q}}[\log\{\mathfrak{p}(\boldsymbol{c}|\beta_0,\gamma_\beta,\widetilde{\boldsymbol{\beta}},\gamma_u,\widetilde{\boldsymbol{u}})\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\boldsymbol{c})\}] & \text{Bernoulli response case,} \end{cases}$$

where

$$\begin{aligned} \log \underline{p}(\boldsymbol{y};\mathfrak{q},\mathrm{C}) &= E_{\mathfrak{q}}[\log\{\mathfrak{p}(\beta_0)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\beta_0)\}] + E_{\mathfrak{q}}[\log\{\mathfrak{p}(\gamma_\beta)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\gamma_\beta)\}] \\ &\quad +E_{\mathfrak{q}}[\log\{\mathfrak{p}(\widetilde{\boldsymbol{\beta}}|\boldsymbol{b}_\beta,\sigma_\beta^2)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\widetilde{\boldsymbol{\beta}})\}] + E_{\mathfrak{q}}[\log\{\mathfrak{p}(\boldsymbol{b}_\beta)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\boldsymbol{b}_\beta)\}] \\ &\quad +E_{\mathfrak{q}}[\log\{\mathfrak{p}(\sigma_\beta^2|a_\beta)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\sigma_\beta^2)\}] + E_{\mathfrak{q}}[\log\{\mathfrak{p}(a_\beta)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(a_\beta)\}] \\ &\quad +E_{\mathfrak{q}}[\log\{\mathfrak{p}(\gamma_u)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\gamma_u)\}] + E_{\mathfrak{q}}[\log\{\mathfrak{p}(\widetilde{\boldsymbol{u}}|\boldsymbol{b}_u,\boldsymbol{\sigma}_u^2)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\widetilde{\boldsymbol{u}})\}] \\ &\quad +E_{\mathfrak{q}}[\log\{\mathfrak{p}(\boldsymbol{b}_u)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\boldsymbol{b}_u)\}] + E_{\mathfrak{q}}[\log\{\mathfrak{p}(\boldsymbol{\sigma}_u^2|\boldsymbol{a}_u)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\boldsymbol{\sigma}_u^2)\}] \\ &\quad +E_{\mathfrak{q}}[\log\{\mathfrak{p}(\boldsymbol{a}_u)\}] - E_{\mathfrak{q}}[\log\{\mathfrak{q}(\boldsymbol{a}_u)\}]. \end{aligned} \tag{S.1}$$

Here "C" signifies the fact that (S.1) is common to both $\log \underline{p}(\boldsymbol{y};\mathfrak{q})$ expressions.

Explicit expressions for $\log \underline{p}(\boldsymbol{y};\mathfrak{q})$ can be obtained by simplifying each of the $\mathfrak{q}$-density moment expressions. For example, the first term of (S.1) is

$$\begin{aligned} E_{\mathfrak{q}}[\log\{\mathfrak{p}(\beta_0)\}] &= -\tfrac{1}{2}\log(2\pi) - \tfrac{1}{2}\log(\sigma_{\beta_0}^2) - \tfrac{1}{2}E_{\mathfrak{q}}(\beta_0^2)/\sigma_{\beta_0}^2 \\ &= -\tfrac{1}{2}\log(2\pi) - \tfrac{1}{2}\log(\sigma_{\beta_0}^2) - \tfrac{1}{2}\{\mu_{\mathfrak{q}(\beta_0)}^2 + \sigma_{\mathfrak{q}(\beta_0)}^2\}/\sigma_{\beta_0}^2. \end{aligned}$$

Also, since $\mathfrak{q}(\beta_0)$ is the $N\big(\mu_{\mathfrak{q}(\beta_0)},\sigma_{\mathfrak{q}(\beta_0)}^2\big)$ density function, the second term of (S.1) is

$$\begin{aligned} -E_{\mathfrak{q}}[\log\{\mathfrak{q}(\beta_0)\}] &= \tfrac{1}{2}\log(2\pi) + \tfrac{1}{2}\log\big(\sigma_{\mathfrak{q}(\beta_0)}^2\big) + \tfrac{1}{2}E_{\mathfrak{q}}\big\{\big(\beta_0 - \mu_{\mathfrak{q}(\beta_0)}\big)^2\big\}/\sigma_{\mathfrak{q}(\beta_0)}^2 \\ &= \tfrac{1}{2}\{\log(2\pi)+1\} + \tfrac{1}{2}\log\big(\sigma_{\mathfrak{q}(\beta_0)}^2\big). \end{aligned}$$

Continuing in this fashion, and accounting for some cancellations, we obtain

$$\begin{aligned}
\log \underline{p}(\boldsymbol{y};\mathfrak{q},\mathrm{C}) = \text{const}_1 &- \tfrac{1}{2}\{\mu^2_{\mathfrak{q}(\beta_0)} + \sigma^2_{\mathfrak{q}(\beta_0)}\}/\sigma^2_{\beta_0} + \tfrac{1}{2}\log\left(\sigma^2_{\mathfrak{q}(\beta_0)}\right) + \text{logit}(\rho_\beta)\sum_{j=1}^{d_\circ+d_\bullet}\mu_{\mathfrak{q}(\gamma_{\beta j})} \\
&- \sum_{j=1}^{d_\circ+d_\bullet}\left[\mu_{\mathfrak{q}(\gamma_{\beta j})}\log\left(\mu_{\mathfrak{q}(\gamma_{\beta j})}\right) + \{1-\mu_{\mathfrak{q}(\gamma_{\beta j})}\}\log\left(1-\mu_{\mathfrak{q}(\gamma_{\beta j})}\right)\right] \\
&- \tfrac{1}{2}\mu_{\mathfrak{q}(1/\sigma^2_\beta)}\sum_{j=1}^{d_\circ+d_\bullet}\mu_{\mathfrak{q}(b_{\beta j})}\left(\mu^2_{\mathfrak{q}(\widetilde{\beta}_j)} + \sigma^2_{\mathfrak{q}(\widetilde{\beta}_j)}\right) + \tfrac{1}{2}\log\left|\boldsymbol{\Sigma}_{\mathfrak{q}(\widetilde{\boldsymbol{\beta}})}\right| \\
&- \tfrac{1}{2}\sum_{j=1}^{d_\circ+d_\bullet}\{1/\mu_{\mathfrak{q}(b_{\beta j})}\} - \mu_{\mathfrak{q}(1/a_\beta)}\mu_{\mathfrak{q}(1/\sigma^2_\beta)} - \tfrac{1}{2}(d_\circ+d_\bullet+1)\log\left(\lambda_{\mathfrak{q}(\sigma^2_\beta)}\right) \\
&+ \mu_{\mathfrak{q}(1/\sigma^2_\beta)}\lambda_{\mathfrak{q}(\sigma^2_\beta)} - \mu_{\mathfrak{q}(1/a_\beta)}/s^2_\beta + \lambda_{\mathfrak{q}(a_\beta)}\mu_{\mathfrak{q}(1/a_\beta)} - \log\left(\lambda_{\mathfrak{q}(a_\beta)}\right) \\
&- \sum_{j=1}^{d_\bullet}\left[\mu_{\mathfrak{q}(\gamma_{uj})}\log\left(\mu_{\mathfrak{q}(\gamma_{uj})}\right) + \{1-\mu_{\mathfrak{q}(\gamma_{uj})}\}\log\left(1-\mu_{\mathfrak{q}(\gamma_{uj})}\right)\right] \\
&+ \text{logit}(\rho_u)\sum_{j=1}^{d_\bullet}\mu_{\mathfrak{q}(\gamma_{uj})} - \tfrac{1}{2}\sum_{j=1}^{d_\bullet}\mu_{\mathfrak{q}(1/\sigma^2_{uj})}\mu_{\mathfrak{q}(b_{uj})}\left(\|\boldsymbol{\mu}_{\mathfrak{q}(\widetilde{\boldsymbol{u}}_j)}\|^2 + \mathbf{1}^T_{K_j}\boldsymbol{\sigma}^2_{\mathfrak{q}(\widetilde{\boldsymbol{u}}_j)}\right) \\
&+ \tfrac{1}{2}\sum_{j=1}^{d_\bullet}\sum_{k=1}^{K_j}\log\left(\sigma^2_{\mathfrak{q}(\widetilde{\boldsymbol{u}}_{jk})}\right) - \tfrac{1}{2}\sum_{j=1}^{d_\bullet}\{1/\mu_{\mathfrak{q}(b_{uj})}\} - \sum_{j=1}^{d_\bullet}\mu_{\mathfrak{q}(1/a_{uj})}\mu_{\mathfrak{q}(1/\sigma^2_{uj})} \\
&- \tfrac{1}{2}\sum_{j=1}^{d_\bullet}(K_j+1)\log\left(\lambda_{\mathfrak{q}(\sigma^2_{uj})}\right) + \sum_{j=1}^{d_\bullet}\mu_{\mathfrak{q}(1/\sigma^2_{uj})}\lambda_{\mathfrak{q}(\sigma^2_{uj})} - (1/s^2_u)\sum_{j=1}^{d_\bullet}\mu_{\mathfrak{q}(1/a_{uj})} \\
&+ \sum_{j=1}^{d_\bullet}\left\{\lambda_{\mathfrak{q}(a_{uj})}\mu_{\mathfrak{q}(1/a_{uj})} - \log\left(\lambda_{\mathfrak{q}(a_{uj})}\right)\right\}
\end{aligned}$$

where const_1 is a constant that does not depend on any $\mathfrak{q}$-density parameters.

In the Gaussian response case, we have

$$\begin{aligned}
\log \underline{p}(\boldsymbol{y};\mathfrak{q}) = \log \underline{p}(\boldsymbol{y};\mathfrak{q},\mathrm{C}) &- \tfrac{1}{2}(n+1)\log(\lambda_{\mathfrak{q}(\sigma^2_\varepsilon)}) - \mu_{\mathfrak{q}(1/a_\varepsilon)}/s^2_\varepsilon - \log\left(\lambda_{\mathfrak{q}(a_\varepsilon)}\right) + \lambda_{\mathfrak{q}(a_\varepsilon)}\mu_{\mathfrak{q}(1/a_\varepsilon)} \\
&+ \text{const}_2,
\end{aligned}$$

where const_2 is a constant that does not depend on any $\mathfrak{q}$-density parameters. In the Bernoulli response case

$$\begin{aligned}
\log \underline{p}(\boldsymbol{y};\mathfrak{q}) = \log \underline{p}(\boldsymbol{y};\mathfrak{q},\mathrm{C}) + \sum_{i=1}^{n}\log\Big\{\Phi\Big((2y_i-1)\Big(\mathbf{1}_n\mu_{\mathfrak{q}(\beta_0)} + \boldsymbol{X}\left(\boldsymbol{\mu}_{\mathfrak{q}(\gamma_\beta)}\odot\boldsymbol{\mu}_{\mathfrak{q}(\widetilde{\boldsymbol{\beta}})}\right) \\
+\sum_{j=1}^{d_\bullet}\boldsymbol{Z}_j\left(\mu_{\mathfrak{q}(\gamma_{uj})}\boldsymbol{\mu}_{\mathfrak{q}(\widetilde{\boldsymbol{u}}_j)}\right)\Big)_i\Big)\Big\}.
\end{aligned}$$

S.3 Additional Simulation Results

We have conducted thorough simulation testing of Algorithms 2 and 3 and the model selection strategies given in Section 3.5. Space considerations are such that Sections 3 and 4 contain only our primary simulation results. Additional simulation results are conveyed in this section.

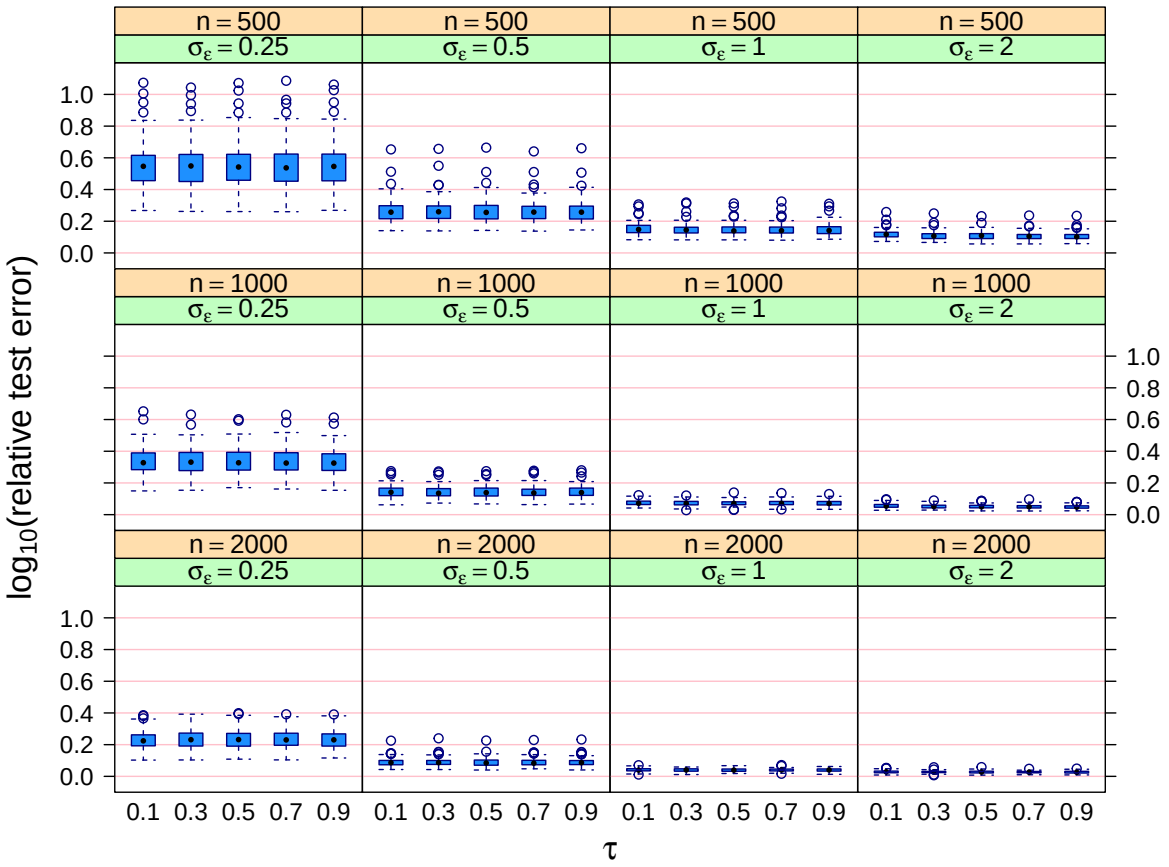

Figure S.1: *Side-by-side boxplots of the logarithms, to base* 10*, of relative test error for the Markov chain Monte Carlo Algorithm 2 for the simulation study described in the text. Each panel corresponds to a different combination of sample size and error standard deviation. Within each panel, the side-by-side boxplots compare relative test error as a function of the threshold parameter* τ*.*

S.3.1 Alternative Evaluation Metrics

The model selection recommendations of Section 3.5 are guided by the effect type misclassification rate since we believe this particular evaluation metric to be best aligned with the practical goal of achieving interpretable and parsimonious models. However effect type misclassification rate is just one of many possible evaluation metrics that could be used in simulation assessment, comparison and the guiding of tuning parameter choice. For example, the simulation studies of Hastie, Tibshirani & Tibshirani (2020), for a different regression-type setting, consider five evaluation metrics.

To see if and how our threshold parameter recommendations change if a different evaluation metric is used, we re-ran the Gaussian response simulation studies of Section 3.5 with effect type misclassification rate replaced by *relative test error*. For the situation where $d_\circ = 0$ and $d_\bullet \in \mathbb{N}$, suppose that the selected model based on the data set $\mathscr{D}$ corresponds to $\widehat{f}$ for some additive function $\widehat{f} : \mathbb{R}^{d_\bullet} \to \mathbb{R}$. If the true model corresponds to $f_{\text{true}} : \mathbb{R}^{d_\bullet} \to \mathbb{R}$ and the predictor $\boldsymbol{x} \in \mathbb{R}^{d_\bullet}$ is a random vector with density function $\mathfrak{p}(\boldsymbol{x})$ then the relative test error is

$$E\big[\{y - \widehat{f}(\boldsymbol{x})\}^2 | \mathscr{D}\big] / \sigma_\varepsilon^2 \quad \text{where} \quad y \sim N\big(f_{\text{true}}(\boldsymbol{x}), \sigma_\varepsilon^2\big). \tag{S.2}$$

Note that the expectation in (S.2) is over the predictor distribution corresponding to $\mathfrak{p}(\boldsymbol{x})$. The denominator in (S.2) is the *Bayes error*, corresponding to the situation where $\widehat{f} = f_{\text{true}}$. Therefore, (S.2) is the test error relative to the Bayes error and is an evaluation metric with a lower bound of 1, and equals 1 when f_{true} is estimated perfectly.

Figures S.1 and S.2 are the analogues of Figures 3 and 4 with effect type misclassification rate replaced by relative test error. Monte Carlo approximations of the (S.2) numerator quantity based on 100,000 draws from the predictor distribution were used. To aid visualization the $\log_{10}$ transformation is applied to the relative test error values.

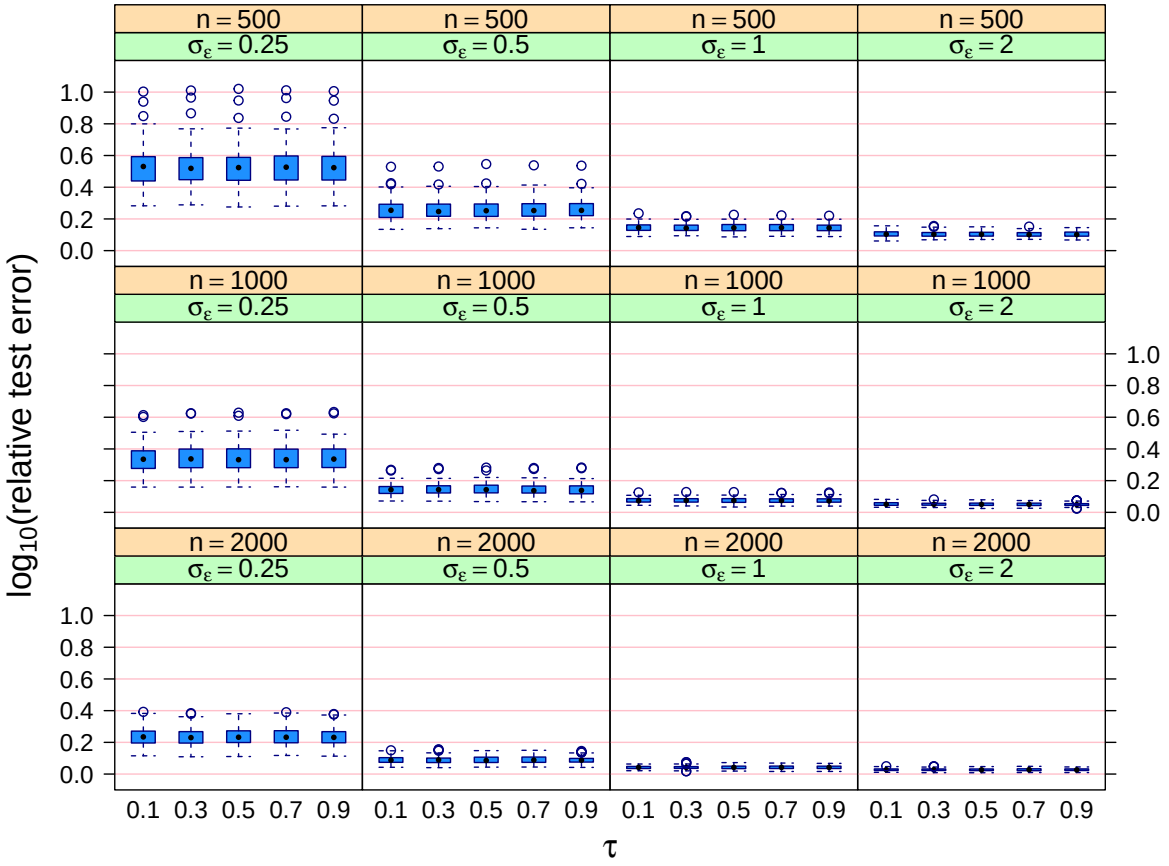

Figure S.2: *Side-by-side boxplots of the logarithms, to base* 10*, of relative test error for the mean field variational Bayes Algorithm 3 for the simulation study described in the text. Each panel corresponds to a different combination of sample size and error standard deviation. Within each panel, the side-by-side boxplots compare relative test error as a function of the threshold parameter* τ*.*

From Figures S.1 and S.2 we see that the relative test errors are lower for higher sample sizes, as expected. Somewhat counter-intuitively the relative test errors are lower for higher noise levels. However, comparisons of relative test error across different values of σ_ε are not clear-cut when the estimators are subject to bias. In addition, relative test errors are barely affected by the choice of the thresholding parameter τ. Lastly, the relative test errors based on mean field variational Bayes approximate inference are similar to those based on Markov chain Monte Carlo. It is interesting that this particular evaluation metric is not affected very much by the choices between Algorithms 2 and 3 and the value of the threshold parameter τ.

S.3.2 Detailed Computing Time Results

We also conducted some more detailed involving computing times. One simulation study looked into the effect of sample size, whilst another one investigated how the number of candidate predictors impacts computing times. The results are presented in this section.

S.3.2.1 Assessment of the Effect of Sample Size

Our first detailed computing time simulation study was concerned with the effect of sample size. We fixed the candidate predictor dimensions to be $(d_\circ, d_\bullet) = (0, 10)$ and let the sample size n to range over the set

$$\{10^k : k = 2, 3, 4, 5, 6\}.$$

The data were generated in a manner similar to that for the simulation studies described in Sections 3 and 4, with 100 replications.

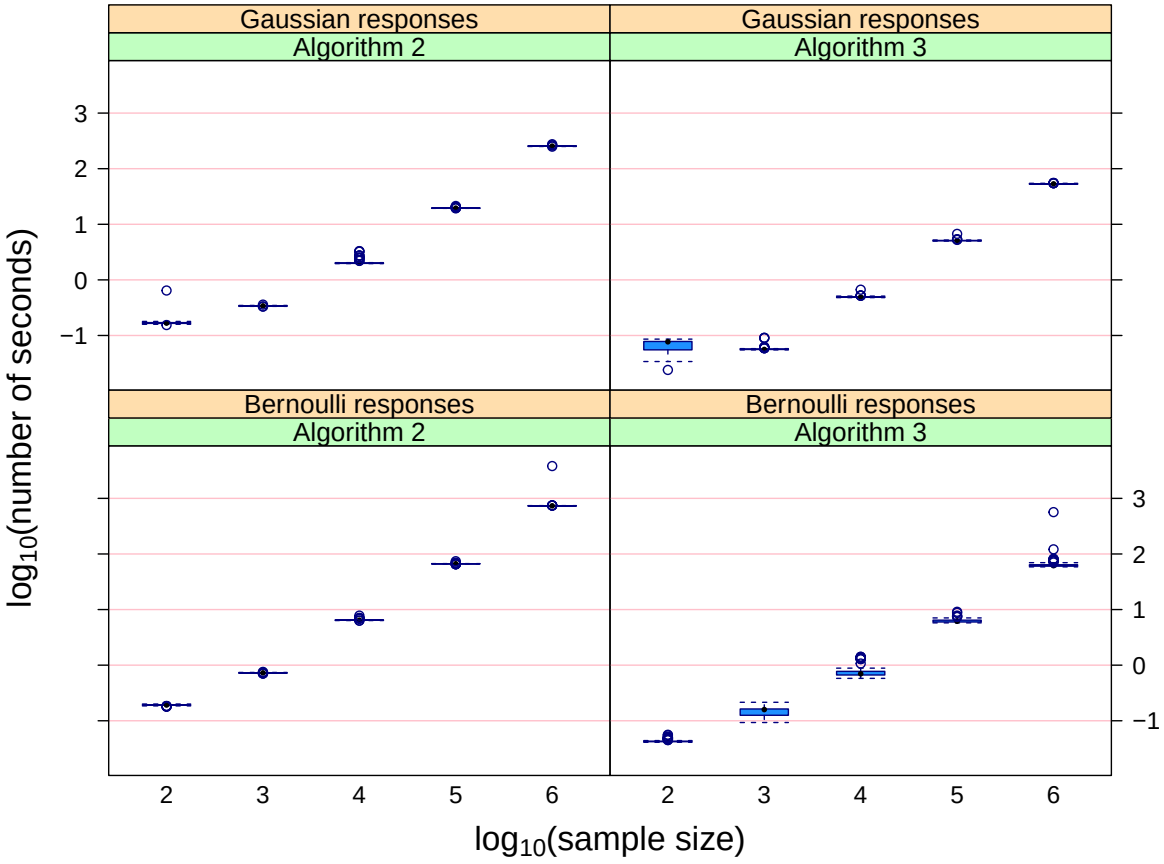

Figure S.3: *Side-by-side box plots of computing time in seconds versus sample size for generalized additive model selection via Algorithms 2 and 3, for the first simulation study described in Section S.3.2. Each axis uses a* $\log_{10}$ *scale.*

Figure S.3 summarizes the results using side-by-side boxplots of the logarithmically transformed computing times, broken down according to sample size, response type and whether or not Algorithm 2 or Algorithm 3 was used. The relationships between the mean logarithmic number of seconds and logarithmic sample size are approximately linear, which suggests a simple power law relationship between computing time and sample size. Simple linear regression analyses suggest that the power is close to 1 and, hence, mean computing time is roughly proportional to sample size.

Figure S.3 also shows that use of Algorithm 3 leads to an approximately ten-fold reduction in computing time compared with Algorithm 2. For example, when $n = 100,000$ the mean computing time of Algorithm 2 for Bernoulli responses is about 100 seconds. For Algorithm 3 it is only about 10 seconds.

S.3.2.2 Assessment of the Effect of the Number of Candidate Predictors

We also ran a simulation study concerned with the effect of the number of candidate predictors on computing time. The sample size was fixed at $5,000$ and $d_\bullet$, the number of candidate predictors that could enter the model non-linearly, varied over the set

$$\{2^k : k = 1, 2, 3, 4, 5, 6\}.$$

We generated the data in a manner similar to that for the simulation studies described in Sections 3 and 4 and, again, obtained 100 replications.

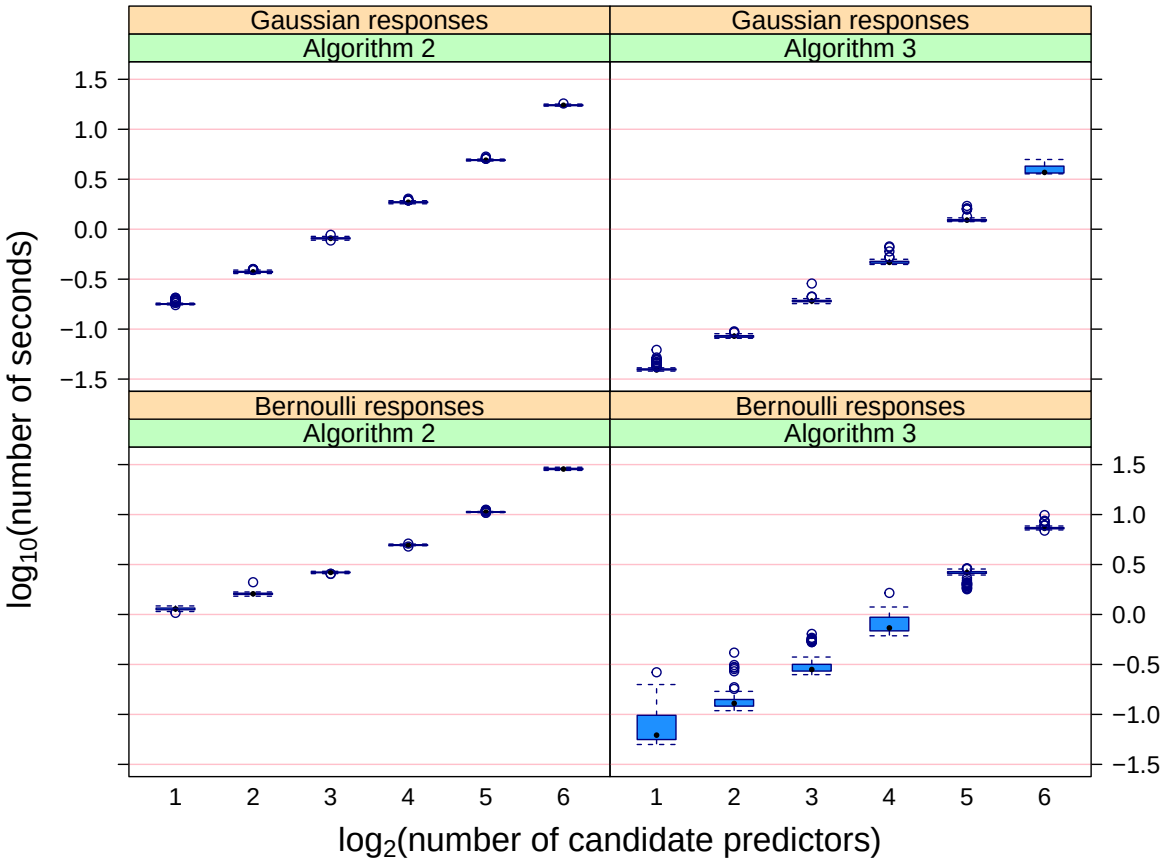

Figure S.4: *Side-by-side box plots of computing time in seconds versus number of candidate predictors for generalized additive model selection via Algorithms 2 and 3, for the second simulation study described in Section S.3.2. The horizontal axis uses a* $\log_2$ *scale and the vertical axis uses a* $\log_{10}$ *scale.*

Figure S.4 summarises the results in similar way to Figure S.3. Once again, there is approximate linearity within each panel with logarithmic scales. Simple linear regression analyses of the data within each panel of Figure S.4 suggest that the mean computing time is approximately proportional to $d_\bullet^\kappa$, with κ dependent on the response distribution and fitting algorithm combination but within the interval $(1.2, 1.5)$.

S.3.3 Hyperparameter Sensitivity Checks

Figure S.5 conveys the effect of the Half Cauchy distribution scale hyperparameters, denoted by s_β, s_ε and s_u in model (9), on the effect type misclassification rate. It is based on the simulation study set-up of Section 3.5 with the Markov chain Monte Carlo approach of Algorithm 2 and the threshold parameter τ set to our recommended default value of 0.5. The scale hyperparameters

ranged over the set

$$\{10^k : k = 1, 2, 3, 4\}.$$

Figure S.5 indicates that our default version of Algorithm 2 is not sensitive to the Half Cauchy distribution scale hyperparameter values.

Figure S.5: *Side-by-side boxplots of misclassification rate for varying values of the Half Cauchy distribution scale hyperparameter for the Gaussian response version of Algorithm 2, for the first simulation study described in Section S.3.3.*

Figure S.6 is similar to Figure S.5, but is for the mean field variational Bayes approach used by Algorithm 3 with τ set to the default value of 0.1. Once again, low sensitivity to the Half Cauchy distribution scale hyperparameter values is exhibited.

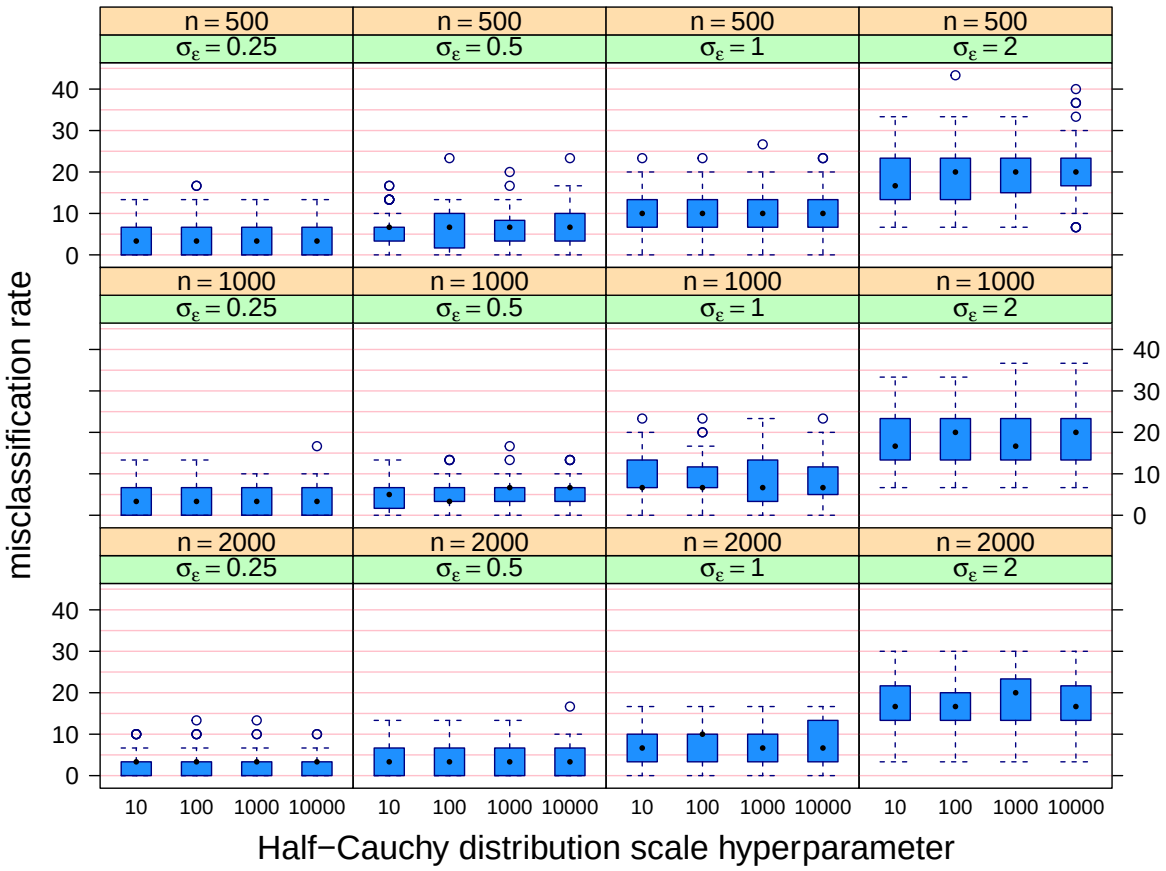

Figure S.6: *Side-by-side boxplots of misclassification rate for varying values of the Half Cauchy distribution scale hyperparameter for the Gaussian response version of Algorithm 3, for the first simulation study described in Section S.3.3.*

Reference

Hastie, T., Tibshirani, R. & Tibshirani, R. (2020). Best subset, forward stepwise of lasso? Analysis and recommendations based on extensive comparisons. *Statistical Science*,**35**, 579–592.