Additive Gaussian Process Regression

Shaan Qamar  
Department of Statistical Science, Duke University, Durham, NC, USA.

Surya T. Tokdar  
Department of Statistical Science, Duke University, Durham, NC, USA.

Summary. Additive-interactive regression has recently been shown to offer attractive minimax error rates over traditional nonparametric multivariate regression in a wide variety of settings, including cases where the predictor count is much larger than the sample size and many of the predictors have important effects on the response, potentially through complex interactions. We present a Bayesian implementation of additive-interactive regression using an additive Gaussian process (AGP) prior and develop an efficient Markov chain sampler that extends stochastic search variable selection in this setting. Careful prior and hyper-parameter specification are developed in light of performance and computational considerations, and key innovations address difficulties in exploring a joint posterior distribution over multiple subsets of high dimensional predictor inclusion vectors. The method offers state-of-the-art support and interaction recovery while improving dramatically over competitors in terms of prediction accuracy on a diverse set of simulated and real data. Results from real data studies provide strong evidence that the additive-interactive framework is an attractive modeling platform for high-dimensional nonparametric regression.

Keywords: Additive model; Dimension reduction; Gaussian process; High dimensional regression; Markov chain Monte Carlo; Multiple-try Metropolis; Nonparametric; Stochastic search; Variable selection; Weak learner.

1. Introduction

Much of the high dimensional regression literature focuses on parametric linear models regularized with sparsity and shrinkage (Tibshirani, 1996; Candes and Tao, 2007; Hastie et al., 2009). Restricted by their linearity and additivity assumptions, these methods often fail to adequately model many naturally occurring predictor-response relations. By requiring the parametric relation to hold globally, these methods remain prone to introducing bias when quantifying intervention effect from nonrandomized studies (Hill, 2011). In addition, these methods also run the risk of over-parametrization when attempting to adjust for non-linearity or to uncover predictor interaction.

Several smoothing based nonparametric regression methods (Lafferty and Wasserman, 2008; Bertin and Lecué, 2008; O’Hagan and Kingman, 1978; Williams and Rasmussen, 1996) accommodate a wider range of predictor-response relations and come with mathematical guarantees of delivering good performance in various settings (Lafferty and Wasserman, 2008; van der Vaart and van Zanten, 2008, 2009; Bhattacharya et al., 2011; Tokdar, 2011b). Unfortunately, the computational demands of these methods scale poorly with predictor dimension due to costly likelihood and score function evaluations. More importantly, with a traditional sparse nonparametric regression model in which \( f \) is assumed to depend on \( d \) of the original \( p \) predictors, statistical estimation greatly suffers from the curse of dimensionality in the high dimensional setting. Under this framework, the minimax \( L_2 \) estimation risk \( r_n \) based on \( n \) observations is of the order
\[
r_n^2 \approx n^{-2\alpha/(2\alpha+d)} + d/n \log(p/d),
\]
where \( \alpha \) denotes the degree of smoothness of \( f \) (Yang and Tokdar, 2014). The second term is the penalty paid for variable selection, and remains small even in large \( p \) small \( n \) situations, i.e., when \( p \) is as large as \( \exp(n^\beta) \), \( \beta \in (0,1) \). For \( p \) of this order, the first term gives the standard fixed-dimension minimax risk for a \( d \)-variate \( \alpha \)-smooth function (Stone, 1982), and is small only when \( d = o(\log n) = o(\log \log p) \), i.e., \( f \) is extremely sparse in the observed predictors.

Popular nonparametric methods based on the theory of ensemble weak learners (Friedman, 1991; Breiman, 2001; Chipman et al., 2010) use simple additive structures and sparsity, and their computational costs scale reasonably well with dimension. However, these methods come with few theoretical guarantees, do not address the large \( p \) small \( n \) curse of dimensionality, may greatly underperform when the actual predictor-response relation is smooth or near parametric, and often behave as black-box forecasting machines offering
little in terms of inference on predictor importance and interaction. Additive structures are also assumed in the widely used generalized additive model (GAM) regression of Hastie et al. (1986) and its high-dimensional extensions (Ravikumar et al., 2007). These approaches consider only univariate components and ignore predictor interaction, however, and the approach by Suzuki (2012) considers interaction only within predetermined subgroups of predictors.

We pursue a novel multivariate regression model and method where the unknown regression function \( f \) is theorized to decompose as \( f = f_1 + \cdots + f_k \) for some \( k \geq 1 \), with each component function \( f_s \) depending nonparametrically on a small number of predictors. Here, predictors are treated exchangeably and learning their interaction patterns is considered a primary objective. Although such an additive-interactive regression resembles the theoretical framework behind ensemble methods (Freund and Schapire, 1997), the emphases are entirely distinct. Ensemble methods add many “weak learners” to boost their overall efficiency while avoiding the risk of overfitting. In contrast, the additive-interactive framework is a modeling assumption which postulates that a high dimensional function \( f \) can be divided into low dimensional pieces added together, enabling a statistical method to estimate each piece with an efficient low-dimensional learner. Seminal works by van der Vaart and van Zanten (2008, 2009) on Gaussian process (GP) regression indicate that a GP prior specification on the component functions could be ideally suited for this task, motivating the additive GP method pursued here.

In a recent work, Yang and Tokdar (2014) establish attractive minimax theory for such additive-interactive regression, with a focus on the high dimensional setting. They show that the additive-interactive regression breaks away from the extreme sparsity assumption, provided the number of predictors included in any single component is bounded. In particular, if every component function is \( \alpha \)-smooth and includes \( d \) predictors, the minimax risk remains small even with \( k = O(n^{1/\alpha}) \), \( \xi \in (0,1) \) components. This corresponds to the inclusion of \( d n^{\xi} \approx (\log p)^{1/\alpha} \) predictors. By restricting the size of each component, which restricts the maximum order of interaction, one can therefore learn the effect of many more predictors by increasing the number of additive components.

We develop a fully Bayesian implementation of additive-interactive regression using an additive Gaussian process (AGP) prior. Yang and Tokdar (2014) show that a theoretical abstraction of our method offers posterior contraction rates that adaptively match the minimax rate across a wide range of regression settings. The significance of our contribution is threefold: First, we present in Section 2.2 details of how to set the global and component specific model hyper-parameters, allowing the method to adapt to varying degrees of sparsity, additivity and complexity of interaction patterns. Second, in high dimensions, computational challenges arise in efficient sampling of the component inclusion vectors while maintaining a reversible Markov chain sampler. Section 3 develops an efficient and reproducible MCMC scheme, adapting concepts from stochastic neighborhood search (Hans et al., 2007) and generalizing the multiple-try Metropolis (Liu et al., 2000) algorithm. Additional strategies are proposed in Sections 3.4.1, 3.4.2 and 3.5 to improve mixing and efficiency. Finally, we demonstrate AGP’s dramatic improvement over competitors in terms of prediction accuracy, as well as its state-of-the-art support and interaction recovery on a diverse set of simulated and real data studies in Sections 4 and 5. Our results provide a strong evidence that the additive-interactive framework is a practically attractive modeling platform for high-dimensional nonparametric regression, and also validate the proposed Markov chain sampler as an effective computational learning tool. Section 6 concludes with a discussion of extensions and future work.

2. Additive-interactive regression with Gaussian processes

2.1. The additive GP model

Recall that a stochastic process \( w = (w(x) : x \in X) \) on a Euclidian domain \( X \) is called a Gaussian process (GP) if every finite collection of elements \( (w(x_1), \ldots, w(x_n)) \), \( n \geq 1 \), \( \{x_1, \ldots, x_n\} \subset X \), has a joint Gaussian distribution. Such a process is completely determined by its mean and covariance functions \( \mu(x) = \mathbb{E}[w(x)], \quad C(x, \tilde{x}) = \mathbb{E}[(w(x) - \mu(x))(w(\tilde{x}) - \mu(\tilde{x}))] \), with the latter being non-negative definite over \( \mathbb{R}^\infty \). For any real function \( \mu(x) \) and non-negative definite function \( C(x, \tilde{x}) \), there exists a GP with these functions as its mean and covariance. We refer to the probability law of such a process as \( \text{GP}(\mu, C) \). When \( \mu \) and \( C \) are continuous, \( w \sim \text{GP}(\mu, C) \) may be viewed as a random element of \( C(X) \), the space of continuous functions over \( X \) equipped with the supremum norm. The smoothness of function valued realizations of such a random element depends on the smoothness of \( C \). For \( C = C_{\text{SE}}(x, \tilde{x}) := \exp(-\|x - \tilde{x}\|^2) \), the squared exponential
covariance function, realizations of \( w \) are infinitely smooth elements of \( \mathcal{C}(X) \).

For paired predictor-response observations \( (x_i, y_i) \in \mathbb{R}^p \times \mathbb{R}, i = 1, \ldots, n \), an additive GP (AGP) model facilitating additive-interactive regression can be written as:

\[
\begin{align*}
    y_i &= f(x_i) + \epsilon_i, \quad \epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2) \\
    f(x_i) &= f_1(x_i) + \cdots + f_k(x_i) \\
    f_l(\rho_l, \lambda_l, \gamma_l, \sigma) &\sim \text{GP}(\mu(\cdot), \sigma^2 \rho_l^2 C_{SE}(\cdot, |\gamma_l, \lambda_l)) \\
    \rho_l &\sim \pi_{\rho_l}, \quad \lambda_l \sim \pi_{\lambda}, \quad \gamma_l \sim \pi_{\gamma}, \quad \sigma^2 \sim \pi_{\sigma}.
\end{align*}
\]

(1)

The proposed model uses a sum of \( k \) GP components to model an unknown regression function \( f_0 \) having \( k_0 \) additive components, namely \( f_0(x) \approx \mathbb{E}(y|x) = \sum_{l=1}^k f_l(x) \). Let \( k_{\text{max}} \) denote a fixed large number that serves as an upper bound on \( k \). Since \( k_0 \) is unknown, components are allowed to be empty to allow the effective number of components be random within this bound. Additional parameters \( \gamma_l \) and \( \lambda_l \) inserted in each GP component’s covariance function facilitate selective predictor inclusion and adaptive local smoothing. In addition, parameter \( \rho_l \) captures the signal-to-noise ratio for each component. Appropriately chosen prior distributions \( \pi_{\rho_l}, \pi_{\lambda}, \pi_{\gamma} \) induce sparsity by imposing regularity on individual GP components, allowing AGP to adapt to the degree of additivity in the data, while ensuring that the number of selected predictors is small relative to \( p \) (see Section 2.2).

2.2. Regularity and prior specification

Response vector \( y = (y_1, \ldots, y_n) \) is assumed centered and scaled having mean 0 and variance 1. AGP model (1) embeds the notion of controlled interaction, with inclusion vector \( \gamma \) modeling a \( d \)-order predictor interaction with common scaling \( \rho, \lambda \). Each GP component is initialized to have mean \( \mu = 0 \) and covariance function \( C_{SE}(x, \tilde{x}) = \rho^2 \exp(-\lambda^2 \|x - \tilde{x}\|_2^2) \). Here, \( x_\gamma = \{x_j \gamma_j, j = 1, \ldots, p\} \) denotes the sub-vector of \( x \) corresponding to the nonzero entries of inclusion vector \( \gamma = (\gamma_1, \ldots, \gamma_p) \in \{0, 1\}^p \). The noise variance is modeled under conjugate prior \( \sigma^2 \sim \text{IG}(a, b) \) with suitably chosen hyper-parameters (see Section 2.2.3).

2.2.1. Sparse priors for inclusion vectors

Sparse priors placed on component inclusion vectors regularize model complexity in terms of the number of active components, in addition to limiting the order of interaction in each. Predictors are treated exchangeably, each having prior inclusion probability \( \tau \in (0, 1) \), with the prior probability for inclusion vector \( \gamma = (\gamma_1, \ldots, \gamma_p) \) given by \( \pi(\gamma|\tau) = \prod_{j=1}^p \tau^{|\gamma_j|}(1 - \tau)^{1-|\gamma_j|} \).

![Figure 1](image-url)  

(a) \( \pi(\tau|d^* = 1) \)  
(b) \( \pi(\gamma|\tau = 1/2) \)  
(c) \( \pi(\gamma|d^* = 1) \) vs. \( \pi(\gamma|\tau = 1/2) \)

**Fig. 1.** Left: prior for predictor inclusion probability \( \tau \) with default hyper-parameters; Middle: prior probability over component inclusion size for varying predictor dimensions \( p \) and fixed inclusion probability \( \tau = 1/2 \); Right: for \( p = 100 \), the latter is compared to \( \pi(\gamma|d^* = 1) \), the induced marginal prior over component inclusion size.
When \( p \) is small, fixing \( \tau = 1/2 \) leads to a distribution over model sizes depicted in Figure 1(b). Having an average model size of \( p/2 \), this prior clearly does not favor sparse configurations as \( p \) grows. Similarly, the “uniform indifference” prior \( \pi(\gamma) = 2^{-\nu}, \; \gamma \in \{0,1\}^p \) fails to adequately penalize large models. The Beta-binomial prior provides the desired degree of regularity, with \( \tau \sim \text{Beta}(\omega, \nu) \) parametrized by its prior \( \omega \) and “sample size.”\(^{†} \) Setting \( \omega = d^*/p \) and \( \nu = p \), the prior expected component size is \( d^* \) independent of the predictor dimension, and \( \text{var}(\tau) \approx 1/p^2 \) when \( p \) is large. In such a case, \( \tau \) concentrates near its prior mean as shown in Figure 1(a). Marginal distribution \( \pi(\gamma|d^*) \) shown in Figure 1(c) places most of its mass on small models, with alternate choices of \( d^* \) allowing for additional flexibility.

2.2.2. Discrete priors for covariance scale parameters
Scale parameters of the GP covariance function play a key role in the ability to learn lower-dimensional interaction effects in the additive-interactive framework. In particular, grouping of predictors within components is encouraged through component-specific scaling. However, coupling between the \( n \)-vector of GP realizations and its hyper-parameters can lead to poor mixing (Murray et al., 2010) even in low dimensions. Variable selection introduces an additional layer of complexity, where component inclusion vector \( \gamma \) and scale parameters \( \rho, \lambda \) are intricately linked within the GP covariance function (see Section 2.2). AGP’s performance depends crucially on good mixing of the component inclusion vectors. Consequently, poor mixing of the scale parameters can significantly hamper MCMC efficiency if updates to \( \gamma \) are rejected because the correct scaling was not proposed.

To address this challenge, we propose a griddy-Gibbs (Ritter and Tanner, 1992) discretization of the support for scale parameters \( \rho, \lambda \). Though simple, this strategy is essential for robust and efficient MCMC over the AGP model space and is highly effective. In particular, this avoids a Metropolis step over \( \{\rho_l, \lambda_l : l = 1, \ldots, k\} \) by allowing posterior weights over the grid to be calculated, and enables marginalization over scale parameters when sampling component inclusion vectors (see Section 3.1). Define \( S_\rho \) as the support grid for signal-to-noise ratio \( \rho \), with prior weight vector \( W_\rho = \{w_i : \rho_i \in S_\rho\} \), and correspondingly \( S_\lambda \) for inverse length-scale \( \lambda \), with prior weight vector \( W_\lambda = \{w_i : \lambda_i \in S_\lambda\} \). Discrete priors for component scale parameters \( \rho, \lambda \) are specified as

\[
\begin{align*}
\pi(\rho) &= \sum_{\rho_i \in S_\rho} w_i \delta(\rho - \rho_i), \quad w_i \propto (1 + \rho_i)^{-\alpha}, \quad \alpha \geq 0 \\
\pi(\lambda) &= \sum_{\lambda_i \in S_\lambda} w_i \delta(\lambda - \lambda_i), \quad w_i \propto \exp(-\beta \lambda_i), \quad \beta \geq 0.
\end{align*}
\]

(2)

Here, if \( X \sim \delta(x) \), then \( X = 0 \) w.p. 1 (i.e., \( \delta(\cdot) \) is the Dirac delta distribution).

2.2.3. Prior hyper-parameter initialization
The response and predictors are standardized to elicit meaningful grid points and default specification for prior hyper-parameters. By default, we set \( a = b = 1 \) for the prior on noise variance \( \sigma^2 \), and \( \alpha = \beta = 0 \) for (2). In experiments, sensitivity to the initialization of these hyper-parameters is negligible, though a careful choice for grid support \( S_\rho \) and \( S_\lambda \) is necessary to allow AGP to adapt to additive structures having varying degrees smoothness and signal strength.

Correlation between nearby points is characterized by inverse length-scale \( \lambda \). Components with larger length-scales capture smoother patterns in the regression surface, while those with smaller length-scales characterize finer shapes and features. Restricting to higher values of correlation offers regularization of GP smoothness. Grid \( S_\lambda \) is chosen so that the GP correlation at a distance of 0.10 given by \( \exp(-0.01 \lambda_i^2) \) ranges over \( \{0.70, 0.80, 0.88, 0.94, 0.99\} \).\(^{‡} \) The covariance scaling for a GP component is parameterized as the product of noise variance \( \sigma^2 \) with signal-to-noise ratio \( \rho^2 \). Components with smaller values of this parameter concentrate increasingly on their prior mean, whereas components with larger values capture important variation in the response. Holding all other components fixed, the ratio of explained variance to

\(^{†}\)The standard Beta parameterization is given by \( \alpha = \omega \nu, \beta = (1 - \omega)\nu \).

\(^{‡}\lambda \approx 1 \) characterizes a near-linear relationship for nearby points, while larger values characterize higher degrees of non-linearity.
total variance for component \( l \) is \( R_l^2 = \rho_l^2/(1 + \rho_l^2) \). Grid values for \( S_p \) are chosen so that \( R^2 \) ranges over \{0.025, 0.50, 0.70, 0.85, 0.99\}.

In addition, we define \( k_\alpha = \sum_{l=1}^{k_{\text{max}}} I(\rho_l > 0 \text{ or } d_l > 0) \) as the number of active components, and \( I_{k_\alpha} \) denotes the active component index set. For \( d^* \ll p \), prior component size \( d_l \) is approximately Poisson\((d^*)\), and \( E[k_\alpha] \approx k_{\text{max}}(1 - w_0 \exp(-d^*)) \). With \( \alpha = 0 \) and \( d^* = 1 \), the prior expected number of active components is approximately 0.95\( k_{\text{max}} \). The minimum and maximum number of active components are set as \( k_{\text{min}} = \lfloor \log(p) \rfloor \) and \( k_{\text{max}} = \lceil p^{1/2} \rceil \), respectively. This allows multiple attempts at identifying important predictors across a larger number of active components initially. Thereafter, the number of active components adapts in a data dependent (see Section 3.4.2).

### 3. Posterior computation for AGP regression

#### 3.1. A basic Bayesian back-fitting algorithm

We propose a fast, efficient, and reproducible Metropolis-within-Gibbs sampler for posterior computation in AGP model (1). Posterior computation is performed over the joint space of component parameters \( \xi = \{ (\rho_l, \lambda_l, \gamma_l) : l = 1, \ldots, k \} \), and sampling proceeds by sequentially updating component-specific parameters via the “back-fitting” algorithm (Friedman and Stuetzle, 1981).

In the regression setting, the \( n \)-vector of GP realizations \((f_1(x_1), \ldots, f_1(x_n))^\top, 1 \leq l \leq k \), over the observed predictors is conjugate and may be analytically integrated out. In addition, the covariance function for AGP priors have scaling which allows marginalization over the variance of the error process. Doing so, one obtains a marginal likelihood for the data which depends only on \( \xi = \{ \mu, \nu \} \), namely \( y(\xi) \sim t_n(0, (b/a)\Sigma) \). Here, \( \Sigma = I_n + \mathbf{K}, \mathbf{K} = \sum_{l=1}^{k} \rho_l^2 \mathbf{C}_l \), and \( \mathbf{C}_l, 1 \leq l \leq k \), denote component covariance matrices defined in Section 2.2. In addition, \( t_n(\mu, \Sigma) \) denotes a multivariate-\( t \) distribution with \( \nu \) degrees of freedom, mean \( \mu \) and variance \( \Sigma \). For \( l = 1, \ldots, k \), back-fitting for AGP proceeds to:

1. Sample component inclusion vector \( \gamma_l \sim \pi(\cdot | \tau, \xi_{(l)}, y) \) according to

   \[
   \pi(\gamma_l | \tau, \xi_{(l)}, y) \propto \pi(\gamma_l | \tau) \sum_{(\rho_l, \lambda_l) \in \mathcal{G}} \pi(\rho_l, \lambda_l) p(y | (\rho_l, \lambda_l, \gamma_l), \xi_{(l)}),
   \]

   where \( \mathcal{G} = S_p \times S_\lambda \) is the discrete grid for covariance scale parameters discussed in Section 2.2.2. Enumeration of \( \gamma_l \in \{0, 1\}^p \) is intractable when \( p \) is large, so sampling here proceeds via a Metropolis step (see Section 3.2).

2. Sample scale parameters \( (\rho_l, \lambda_l) \in \mathcal{G} \) as a block according to

   \[
   \pi(\rho_l, \lambda_l | \gamma_l, \xi_{(l)}, y) = \frac{\pi(\rho_l, \lambda_l) p(y | (\rho_l, \lambda_l, \gamma_l), \xi_{(l)})}{\sum_{(\hat{\rho_l}, \hat{\lambda}_l) \in \mathcal{G}} \pi(\hat{\rho}_l, \hat{\lambda}_l) p(y | (\hat{\rho}_l, \hat{\lambda}_l, \gamma_l), \xi_{(l)})}.
   \]

3. (Optional) Auxiliary posterior draws for \( \sigma^2 \) and \( f_l = (f_l(x_1), \ldots, f_l(x_n))^\top \) are available in closed-form as

   \[
   \pi(\sigma^2 | \xi, y) = \text{IG}(a + n/2, b + y^\top \Sigma^{-1} y/2)
   \]

   \[
   \pi(f_l | \sigma^2, (\rho_l, \lambda_l, \gamma_l), \xi_{(l)}, y) = N(\Sigma_l^{-1} \Sigma_{(l)}^{-1} y, \sigma^2 \Sigma_l),
   \]

   where \( \Sigma_{(l)} = I_n + K_{(l)} \), \( K_{(l)} = \sum_{k \neq l} \rho_k^2 \mathbf{C}_k \) and \( \Sigma_{l} = (\Sigma_{(l)}^{-1} + (\rho_l^2 \mathbf{C}_l)^{-1})^{-1} \). Posterior draws for variance parameter \( \sigma^2 \) may serve as a model fit diagnostic and a measure of MCMC mixing and stability.

The complete AGP sampling scheme is given in Algorithm 2. There, a posterior draw for the predictor inclusion probability precedes component updates, i.e., \( \tau \sim \text{Beta}(\mu', \nu') \), \( \nu' = (1 + k_a)p \) and \( \mu' = (d^* + \sum_{l \in I_{k_\alpha}} d_l)/\nu' \).
3.2. Updating variable inclusion vectors

3.2.1. Reversible neighborhood sampling

On \{0, 1\}^p, the space of \(p\) dimensional inclusion vectors, a random walk that chooses a variable index at random and proposes to flip its inclusion status, quickly becomes inefficient as \(p\) grows as it takes on an average \(p\) moves before any index gets a “look-at.” When the assumed prior concentrates on sparse inclusion vectors, such random walks are heavily biased toward adding additional predictors rather than removing unwanted ones. To overcome these issues, “Shotgun Stochastic Search” (SSS) (Hans et al., 2007) uses a neighborhood search procedure in high dimensional spaces to quickly identify inclusions vectors with large posterior mass, and is demonstrated to perform well in linear regression and graphical modeling. Additional MCMC schemes have been developed in the context of exploring high dimensional discrete spaces in applications to statistical physics and genomics (Hamze et al., 2013; Bottolo et al., 2010; Strens, 2003; Mansinghka et al., 2009).

It is relatively straightforward to adapt shotgun stochastic search to an MCMC setting for the additive-interactive AGP model. Consider the following single-predictor changes to component inclusion vector \(\gamma\), namely

1. **Add neighbors:** \(N_A(\gamma) = \{\gamma' : \gamma = \gamma + 1_j, \ j \in [\gamma]^c\}\)
2. **Remove neighbors:** \(N_R(\gamma) = \{\gamma' : \gamma = \gamma - 1_j, \ j \in \gamma\}\)
3. **Swap neighbors:** \(N_S(\gamma) = \{\gamma' : \gamma - 1_j + 1_k, (j, k) \in \gamma \times [\gamma]^c\}\).

Swap moves are equivalent to adding and then removing a (different) predictor, but does so in a single step. A proposal distribution

\[
p(\gamma' | \gamma) = \frac{\pi(\gamma' | \gamma)}{\sum_{\gamma \in N(\gamma)} \pi(\gamma | \gamma)}, \quad \gamma' \in N(\gamma) = N_A(\gamma) \cup N_R(\gamma) \cup N_S(\gamma), \tag{6}
\]

defined over these “one-away neighbors” may be used in a Markov chain sampler for component inclusion vectors in the AGP model. We call the resulting Markov chain sampler a reversible neighborhood sampler (RNS). RNS allows every predictor to be considered in the context of the current inclusion vector \(\gamma\), and is biased towards moves with larger posterior probability. A Metropolis step with acceptance probability

\[
\alpha(\gamma, \gamma') = \min \left\{1, \frac{\sum_{\gamma \in N(\gamma)} \pi(\gamma' | \gamma)}{\sum_{\gamma \in N(\gamma)} \pi(\gamma | \gamma)} \right\} \tag{7}
\]

ensures that updates satisfy detailed balance and the Markov chain maintains the desired stationary distribution in (3).

The space of inclusion vectors grows exponentially in the predictor dimension, posing potentially serious scalability and mixing issues for RNS when \(p\) is large. For an inclusion vector containing \(d\) predictors, proposal distribution (6) requires evaluating the likelihood score for \((p - d)\) add, \(d\) remove, and \(d(p - d)\) swap neighbors. Hence, the per-iteration complexity for sampling component inclusion vectors via back-fitting (see Section 3) scales linearly in the predictor dimension. Although likelihood scores may be evaluated in parallel across separate processors, scalable Markov chain sampling of inclusion vectors within the additive-interactive framework requires strategies that address the following: (Problem 1) RNS often identifies good configurations, but is unable to quickly transition to them because of the reversibility constraint. This occurs when good inclusion vectors have even better neighborhoods; and (Problem 2) many neighbors offer little improvement to model fit in high dimensions, and a trade-off exists between the number of additive components used and the neighborhood search size (see Section 3.4.2).

The first of these issues is addressed by leveraging the disjoint structure of the defined one-away neighborhood \(N(\gamma)\) in terms of its constituent add, remove and swap configurations. In particular, if \(\gamma' \in N_A(\gamma)\) then \(\gamma \in N_R(\gamma')\); if \(\gamma' \in N_R(\gamma)\) then \(\gamma \in N_A(\gamma')\); and otherwise \(\gamma' \in N_S(\gamma)\) and \(\gamma \in N_S(\gamma')\). Using this fact, we

\[\text{Consider modeling } f(x) = g_1(x_1, x_2) + g_2(x_3, x_4, x_5) \text{ with a two component AGP model, and assume } \gamma_1 = \{1, 2, 5, 6\} \text{ and } \gamma_2 = \{3\}. \text{ The neighborhoods for a remove proposal } \gamma_1' = \{1, 2, 5\} \text{ or an add proposal } \gamma_2' = \{3, 4\} \text{ contains a better inclusion vector than any contained in the current neighborhood, causing the RNS acceptance probability to be small.}\]
propose a paired-moved reversible neighborhood sampler (PRNS) with “Add-remove”, “Remove-add”, and “Swap-swap” forward-reverse paired neighborhoods. By allowing different moves to be proposed separately and in quick succession, PRNS can dramatically improve mixing in the space of single predictor changes to $\gamma$ (see Section 3.2.2).

3.2.2. A paired-move neighborhood sampler

The PRNS proposal distribution is given by

$$p(\gamma' | \gamma) = w_A p_A(\gamma' | \gamma) + w_R p_R(\gamma' | \gamma) + w_S p_S(\gamma' | \gamma),$$

with $w_A, w_R, w_S \geq 0$ and $w_A + w_R + w_S = 1$. These probabilities are allowed to vary with component size to encourage additions to smaller components, removal from larger components, and swaps for intermediately sized ones. An empty component has size $|\gamma| = 0$, so set $w_A = 1$. Otherwise, set $w_A \propto h_A(|\gamma|; \lambda_A)$ and $w_S \propto h_S(|\gamma|; \lambda_S)$, where $h_A$ is monotonically decreasing in $|\gamma|$, and $h_S$ is unimodal at interaction size $\bar{d} = \text{argmin}_d P(|\gamma| > \delta |d^* = 1) \leq 0.01$.

![Fig. 2. Probability of Add/Remove/Swap moves as a function of component-size $|\gamma| = d$. By default, $h_S$ is the Poisson density with $\lambda_S = 4$, $h_A$ is the exponential density with $\lambda_A = d^* = 1$, and $\bar{d} = 4$.](image)

Paired-moves reduce the number of likelihood evaluations per iteration, though $O(p)$ likelihood evaluations in a GP regression setting remains computationally prohibitive for single computer implementations when $p$ is large. The PRNS transition rule for sampling component inclusion vectors proceeds as follows:

1. Select move $m \in \{A, R, S\}$ with probability $w_A, w_R$, and $w_S$, respectively
2. For move $m \in \{A, R, S\}$, construct forward one-away neighborhood $N_m(\gamma)$
3. Propose $\gamma' \sim N_m(\gamma)$ sampled according to PRNS proposal distribution (8) restricted to this set
4. For the corresponding reverse move $m' \in \{R, A, S\}$, construct reverse neighborhood $N_{m'}(\gamma')$. By construction $\gamma \in N_{m'}(\gamma')$
5. Accept proposal $\gamma'$ with probability

$$\alpha_m(\gamma, \gamma') = \min \left\{ 1, \frac{w_{m'}(|\gamma'|) \sum_{\tilde{\gamma} \in N_m(\gamma)} \pi(\tilde{\gamma}|\cdot)}{w_m(|\gamma|) \sum_{\tilde{\gamma} \in N_{m'}(\gamma')} \pi(\tilde{\gamma}|\cdot)} \right\}.$$  

**Lemma 3.1.** The paired-move neighborhood sampler (PRNS) satisfies detailed balance and MCMC samples converge to the desired target posterior $\pi(\gamma|\cdot)$. 
A proof of Lemma 3.1 is provided in Section C of the Appendix.

To further reduce the complexity for sampling component inclusion vectors, one might propose random thinning of paired neighborhoods, where a fixed “budget” of size \( M \) defines the maximum allowable neighborhood size (likelihood evaluations) per component per iteration. Intuitively, \( M \approx o(p) \) seems necessary to be able to explore sufficiently large regions of the inclusion space while being computationally tractable. In particular, consider a randomly thinned paired-move Add transition \( \gamma \to \gamma' = \gamma + 1_j \) having budget \( M \). Here, one selects \( M \) of \( p - |\gamma| \) possible add configurations in the forward set and all \( |\gamma| + 1 < M \) remove neighbors for the paired reverse set. In the opposite direction direction, transitioning from \( \gamma' \to \gamma = \gamma' - 1_j \) does not require selection of predictor \( j \) beforehand as all \( |\gamma| + 1 \) remove neighbors are considered in the forward set, hence the paired reverse (Add) set need only select \( M - 1 \) of \( p - |\gamma| - 1 \) predictors. This introduces a factor of \( M/(p - |\gamma|) \) in the expression for detailed balance which is non-negligible in desired \( M \ll p \) limit. Hence, although random thinning of one-away neighborhoods appears to have good empirical performance, it compromises reversibility of the paired-move Markov chain sampler.

We resolve the issue of reversibility for random neighborhoods by considering a probabilistic formulation for generating Add/Remove/Swap neighbors of a predictor inclusion vector. In particular, we adapt the multiple-try Metropolis (MTM) algorithm (Liu et al., 2000) to the discrete setting where transitions are confined to the set of one-away neighbors of component inclusion vector \( \gamma \). The approach is general, allowing probabilistic control over neighborhood sizes and adaptive importance scores for predictors comprising these configurations. This addresses problem \#2 raised in Section 3.2.1.

3.3. Generalized multiple-try Metropolis on discrete spaces
For a real valued random variable \( x \in \mathcal{X} \subseteq \mathbb{R} \), let \( T : \mathcal{X} \times \mathcal{X} \to [0, 1] \) be a transition kernel such that \( T(x, x') > 0 \iff T(x', x) > 0 \). In addition, let \( \lambda(x, x') \) be some symmetric function and define weight function \( \omega(x, x') = \pi(x)T(x, x')\lambda(x, x') \). The MTM algorithm produces draws from target distribution \( \pi(x) \) by (1) sampling reference set \( x_1^*, \ldots, x_M^* \sim T(x, \cdot) \); (2) calculating weights \( \omega(x_i^*, x) \), \( i = 1, \ldots, M \); (3) drawing \( x' \in \{x_1^*, \ldots, x_M^*\} \) with probability proportional to \( \omega(x_i^*, x) \); (4) sampling reverse set \( \tilde{x}_1, \ldots, \tilde{x}_{M-1} \sim T(x', \cdot) \) having weights \( w(\tilde{x}_i, x') \), \( i = 1, \ldots, M - 1 \) and \( \tilde{x}_M = x \); and (5) accepting proposal \( x' \) w.p. \( \alpha(x, x') = \min \left\{ 1, \frac{\sum_{i=1}^M \omega(x_i^*, x)}{\sum_{i=1}^M w(\tilde{x}_i, x')} \right\} \).

3.3.1. Random neighborhoods via “toggled” predictor inclusion
To adapt the MTM algorithm for sampling predictor inclusion vector \( \gamma \in \{0, 1\}^p \), define \( \eta_j \sim \text{Bern}(\omega_j) \) with \( \omega : \{0, 1\} \times \mathbb{R}^+ \to [0, 1] \) being a weight function taking input \( \gamma_j = I(j \in \gamma) \) and nonnegative predictor importance score \( v_j \), \( j = 1, \ldots, p \). Here, predictor \( j \) is “toggled” on/off corresponding to an Add(Remove) whenever \( \gamma_j = 0(\gamma_j = 1) \) and \( \eta_j = 1 \). The unique set of selected predictor “toggles” ensures no duplicates among the constructed Add/Remove/Swap neighbors. To initialize \( \gamma = 0 \) and predictors are treated exchangeably a priori by setting importance \( v_j = 1 \), \( j = 1, \ldots, p \), for example. This framework lends itself naturally to adaptive schemes which are useful in sampling high dimensional inclusion vectors. In particular, effective strategies will enhance the chance of selecting predictors with large \( v_j \) when \( j \notin \gamma \) and suppress them when \( j \in \gamma \).

Let \( \pi(\gamma) \) stand for the conditional posterior distribution in (3) and define operator \( \text{tog}(j, \gamma) : \gamma_j \leftarrow 1 - \gamma_j \). Below, we restrict ourselves to Add/Remove one-away neighbors for simplicity but note that Swaps are trivially incorporated by considering proposals \( \gamma' = \text{tog}(r, a), \gamma \) for predictor pairs \( r, a \in \{(j, k) : \gamma_j = 1, \gamma_k = 0\} \) subject to \( \eta_r = \eta_a = 1 \). This case is handled in detail in the section that follows. An algorithm for the proposed generalized MTM over a \( p \)-variate discrete space (DMTM) is:

(1) For \( j = 1, \ldots, p \), draw \( \eta_j \sim \text{Bern}(\omega(\gamma_j, v_j)) \) independently
(2) For \( k \in \{j : \eta_j = 1\} \), \( \tilde{\gamma} = \text{tog}(k, \gamma) \) defines the forward set of mixed Add and Remove neighbors
(3) Among the constructed forward neighbors, select \( \gamma' = \text{tog}(k^*, \gamma) \) with probability \( \pi(\gamma')/\sum_j : \eta_j = 1 \pi(\text{tog}(j, \gamma)) \)
(4) For \( j \neq k^* \), draw \( \eta_j' \sim \text{Bern}(\omega(\gamma'_j, v_j)) \) independently and set \( \eta_k' = 1 \). For \( k \in \{j : \eta_j' = 1\} \), \( \tilde{\gamma} = \text{tog}(k, \gamma) \) defines the reverse set of mixed Add and Remove neighbors
(5) Accept proposal $\gamma'$ with probability
\[
\alpha(\gamma, \gamma') = \min \left\{ 1, \frac{\omega(\gamma_j, v_j; m)}{\omega(\gamma_j', v_j; m)} \right\} \cdot \frac{\sum_{j:t_j=1} \pi(tog(j, \gamma))}{\sum_{j:t_j=1} \pi(tog(j, \gamma'))}.
\]

**Lemma 3.2.** The DMTM algorithm satisfies the detailed balance condition, and MCMC samples converge to the desired target posterior $\pi(\gamma)$.

The proof of Lemma 3.2 follows from the proof of Theorem 3.3. See Section C of the Appendix for details.

### 3.3.2. A paired-move reversible MTM sampler

A probabilistic version of the paired-move neighborhood sampler (PRNS) introduced in Section 3.2.2 is obtained as a special case of the DMTM algorithm for a specific choice of weight function $\omega$, namely
\[
\omega(\gamma_j, v_j; m) = (1 - \gamma_j)f(v_j)I(m = A) + \gamma_jI(m = R), \quad \gamma_j = I(j \in \gamma).
\]

The DMTM algorithm is modified to incorporate paired-moves as follows:

1. Select move $m \in \{A, R, S\}$ with probabilities $w_A(\gamma_j), w_R(\gamma_j)$, and $w_S(\gamma_j)$, respectively.
2. (a) If move $m \in \{A, R\}$: for $j = 1, \ldots, p$, draw $\eta_j \sim \text{Bern}(\omega(\gamma_j, v_j; m))$. For $k \in \{j : \eta_j = 1\}$, $\tilde{\gamma} = \text{tog}(k, \gamma)$ defines the set of Add/Remove neighbors
   (b) If move $m = S$ : define $(\eta_r, \eta_a) \in \{0, 1\}^2$ and consider predictor pairs $(r, a) \in \{(j, k) : \gamma_j = 1, \gamma_k = 0\}$.
   For $r \in \gamma$, draw $\eta_r \sim \text{Bern}(\omega(\gamma_r, \eta_r, m = R))$, and independently for $a \in \gamma^c$ draw $\eta_a \sim \text{Bern}(\omega(\gamma_a, \eta_a, m = A))$. For $(r, a) : \eta_r = \eta_a = 1$, $\tilde{\gamma} = \text{tog}((r, a), \gamma)$ defines the set of Swap neighbors
3. Among the constructed forward neighbors, select $\gamma' \in N_m(\gamma)$ with probability $\pi(\gamma') / \sum_{\tilde{\gamma} \in N_m(\gamma)} \pi(\tilde{\gamma})$.
   If $m \in \{A, R\}$, denote $\gamma' = \text{tog}(k^*, \gamma)$; otherwise denote $\gamma' = \text{tog}(r^*, \gamma)$ for $m = S$
4. (a) If move $m = A$ : for $j \neq k^*$, draw $\eta'_j \sim \text{Bern}(\omega(\gamma'_j, v_j; m = R))$ and set $\eta'_k = 1$. Using the defined weight function, note that for $j \notin \gamma'$, $\omega_j = 0 \quad \Rightarrow \eta'_j = 0$. For $k \in \{j : \eta'_j = 1\}$, $\tilde{\gamma} = \text{tog}(j, \gamma')$ defines the set of reverse (remove) neighbors
   (b) If move $m = R$ : for $j \neq k^*$, draw $\eta'_j \sim \text{Bern}(\omega(\gamma'_j, v_j, m = A))$ and set $\eta'_k = 1$. Using the defined weight function, note that for $j \notin \gamma'$, $\omega_j = 0 \quad \Rightarrow \eta'_j = 0$. For $k \in \{j : \eta'_j = 1\}$, $\tilde{\gamma} = \text{tog}(j, \gamma')$ defines the set of reverse (add) neighbors
   (c) If move $m = S$ : define $(\eta''_r, \eta''_a) \in \{0, 1\}^2$ as before, with predictor pairs $(r, a) \in S(\gamma') = \{(j, k) : \gamma'_j = 1, \gamma'_k = 0\}$.
   For $(r, a) \in S(\gamma')$ : $(r, a) \neq (a^*, r^*)$, draw $\eta''_r \sim \text{Bern}(\omega(\gamma'_r, \eta_r, m = R))$ and independently draw $\eta''_a \sim \text{Bern}(\omega(\gamma'_a, \eta_a, m' = A))$, and set $(\eta''_r, \eta''_a) = (1, 1)$. For $(r, a) : \eta'_r = \eta''_a = 1$, $\tilde{\gamma} = \text{tog}(r, a, \gamma')$ defines the set of reverse (swap) neighbors
5. For $m \in \{A, R, S\}$, the corresponding reverse paired neighborhood is $m' \in \{R, A, S\}$. Let $s^* = k^*$ if $m \in \{A, R\}$; otherwise $s^* = (r^*, a^*)$ for $m = S$. With forward and reverse paired neighborhoods $N_m$ and $N_{m'}$ as defined above, accept proposal $\gamma' = \text{tog}(s^*, \gamma)$ with probability
\[
\alpha_m(\gamma, \gamma') = \min \left\{ 1, \frac{\omega(\gamma_j, v_j; m)}{\omega(\gamma_j', v_j; m')} \frac{\sum_{\tilde{\gamma} \in N_m(\gamma)} \pi(\tilde{\gamma})}{\sum_{\tilde{\gamma} \in N_{m'}(\gamma)} \pi(\tilde{\gamma})} \right\}.
\]

**Theorem 3.3.** The paired-move discrete MTM algorithm satisfies the detailed balance condition, and samples therefore converge to the desired target posterior $\pi(\gamma)$.

A proof of Theorem 3.3 is provided in Section C of the Appendix.

This is a non-trivial generalization of the MTM algorithm which extends MCMC for sampling high dimensional inclusion vectors and enables scalable variable selection outside of the linear regression setting. The framework is general and versatile, allowing sparsity and other model assumptions (e.g., additivity, smoothness etc.) to be incorporated in the sampling of inclusion vectors as is done for the AGP model. Adaptive MCMC strategies within this framework are discussed in Section 3.5.
3.4. Additional ensemble-wide strategies

3.4.1. Additive inter-component moves

To enable predictors to be rearranged or exchanged between components rapidly, we propose inter-component moves (ICM). These moves are complementary to the sequential updating of component inclusion vectors using paired-moves in the back-fitting MCMC scheme of Section 3.1. While paired Add/Remove/Swap employ a “divide-and-learn” approach to exploring the space of predictor inclusion, ICM moves propose updates to inclusion state \( \xi = (\gamma, \theta) = \{(\gamma_l, \theta_l) : l = 1, \ldots, k\} \) via transitions \(((\gamma_m, \theta_m), (\gamma_n, \theta_n)) \to ((\gamma'_m, \theta'_m), (\gamma'_n, \theta'_n))\) between active components \((m, n) \in \mathcal{I}_A\). Here, \( \theta_l = (\rho_l, \lambda_l) \in \mathcal{G} \) denotes the scaling for GP component \( l \). The following transitions are considered:

(1) Cross-component donate: choose an active component at random, say \( \gamma_n \). For predictors \( j \in \gamma_n \), define \( \tilde{\gamma}_{jm} = \{\ldots, \gamma_m + 1, \gamma_n - 1, \ldots\} \) for every other active component \( m \neq n \). Set \( N_{CD}(\gamma; n) = \{\tilde{\gamma}_{jm} : j \in \gamma_n, m \neq n\} \)

(2) Paired donate: choose indices of two active components \((m, n)\) at random, with \( \gamma_n \) taken to be the non-empty donor component by convention. For predictors \( j \in \gamma_n \), define \( \tilde{\gamma}_{jm} = \{\ldots, \gamma_m + 1, \gamma_n - 1, \ldots\} \) and set \( N_{PD}(\gamma; (m, n)) = \{\tilde{\gamma}_{jm} : j \in \gamma_n\} \)

(3) Paired swap: choose indices of two active non-empty components \((m, n)\) at random. For predictor pair \((j, k) \in \mathcal{S}(\gamma_n, \gamma_n) = \{(p, q) \in \gamma_m \times \gamma_n : p \neq q\} \), define \( \tilde{\gamma}_{jk} = \{\ldots, \gamma_m - 1 + j, \gamma_n - 1 - k + j, \ldots\} \) and set \( N_{PS}(\gamma; (m, n)) = \{\tilde{\gamma}_{jk} : (j, k) \in \mathcal{S}(\gamma_n, \gamma_n)\} \).

Cross-component moves in (1) facilitate a rapid exchange of predictors from a selected active component to any other active component holding fixed all component scalings (i.e. transitions for \( \theta \) occur via the identity map). In contrast, the proposal neighborhood for (2) and (3) is defined over \( N_{C}(\gamma; (m, n)) \times \mathcal{G}^2 \), where the acceptance probability for paired Donate/Swap proposals between an active component pair is maximized via optimal scaling. Sampling for ICM proceeds as follows:

(1) Select move \( m \in \{CD, PD, PS\} \) with probability \( w_{CD}, w_{PD}, \) and \( w_{PS}, \) respectively

(2) (a) If \( m = CD \) : let \( k \) be the index for the randomly selected non-empty active component. Then select \( (\gamma', \theta' = \theta) \in A_m(\gamma) = N_m(\gamma; k) \) with probability \( \pi(\gamma'|-) / \sum_{\gamma \in A_m} \pi(\gamma|-) \)

(b) If \( m \in \{PD, PS\} \) : let \( (k, l) \) denote indices of the two randomly selected active components. Then select \( (\gamma', \theta') \in A_m(\gamma) = N_m(\gamma; (k, l)) \times \mathcal{G}^2 \) with probability \( \pi(\gamma', \theta'|-) / \sum_{(\gamma, \theta) \in A_m} \pi(\gamma, \theta|-) \)

(3) Accept transition to state \( \xi' = (\gamma', \theta') \) under the defined proposal distribution for move \( m \) with probability

\[
\alpha_m(\xi, \xi') = \min \left\{ 1, \frac{\sum_{\xi \in A_m(\xi)} \pi(\xi|\cdot)}{\sum_{\xi \in A_m(\xi')} \pi(\xi|\cdot)} \right\}.
\]

(4) If \( m = CD \) : update scale parameters \( \theta = (\rho_l, \lambda_l : l \in \mathcal{I}_A) \) via Gibbs transition kernel

\[
T(\theta, \theta'|\gamma, y) = \prod_{l \in \mathcal{I}_A} \pi(\theta_l|\theta'_{<l}, \theta_{>l}, \gamma, y)
\]

ICM moves are particularly useful in low signal and high dimensional predictor settings. Across random test/train replicate runs on real data illustrations in Section 5, variance of the predictive mean-squared-error is consistently lower when MCMC uses ICM moves (see Table 5).

LEMMA 3.4. For state \( \xi = (\gamma, \theta), \xi_l = (\gamma_l, \rho_l, \lambda_l) \) and \( l \in \mathcal{I}_A, \) the ICM transition kernel preserves stationary distribution \( \pi(\xi|y) \).

A proof of Lemma 3.4 provided in Section C of the Appendix.

REMARK 3.5. Transition kernel \( T(\theta, \theta'|\gamma, y) = \prod_{l \in \mathcal{I}_A} \pi(\theta_l|\theta'_{<l}, \theta_{>l}, \gamma, y) \) has stationary distribution \( \pi(\theta|\gamma, y) \). This follows from the fact that \( T \) is a Gibbs transition kernel. ICM moves used in conjunction with such a transition kernel for component scaling parameters also preserves stationarity.
3.4.2. Updating empty components

From Section 2.2.3, recall that \( k_a = \sum_{l=1}^{k_{\text{max}}} I(\rho_l > 0 \text{ or } d_l > 0) \) defines the number of active components in the AGP model. Algorithm 2 is initialized with fixed lower and upper bounds on the number of components, while allowing empty components with no predictors included (i.e., \( d_l = 0 \)). Non-empty components explaining little variation in the response will often have signal-to-noise ratio \( \rho = 0 \), but remain active to allow AGP to discover additional additive structure in subsequent iterations.

Let \( B \) denote a fixed computational budget (likelihood evaluations) per iteration. A trade-off exists between a per-component budget \( M \) and the total number of active components \( k_a \), namely \( B = Mk_a \). This suggest the possibility of including only as many components as needed, and having more detailed neighborhood scans when updating individual inclusion vectors. Component inclusion vectors are updated with probability \( \theta(\rho_l, \gamma_l; \theta_0) \). Active set \( I_A \) is updated by Algorithm 1 in each MCMC iteration, with baseline probability \( \theta_0 \) initialized so that, on average, one inactive component is “switched on.”

Algorithm 1 Updating active component index set

Input: (i) Active index set \( I_A \); (ii) State vector \( \xi_l = (\gamma_l, \rho_l, \lambda_l) \), \( l \in I_A \); (iii) Fixed target budget \( B \) (likelihood evaluations per iteration)

Output: (i) Updated active index set \( I_A \); (ii) Target neighborhood budget \( M \)

1: function UPDATE.ACTIVESET(\( \xi, I_A, B \))
2: Initialize \( \theta_0 = 0 \)
3: Compute \( k_a = \sum_{l=1}^{k_{\text{max}}} I(\rho_l > 0 \text{ or } d_l > 0) \)
4: if \( k_a < k_{\text{max}} \) then
5: \( \text{set } \theta_0 = (k_{\text{max}} - k_a)^{-1} \)
6: end if
7: for \( l = 1 : k_{\text{max}} \) do
8: \( \theta_l = \theta_0 + (1 - \theta_0)I(l \in \mathcal{I}_A) \)
9: if \( (l > k - k_{\text{min}}, | \mathcal{I}_A | < k_{\text{min}}) \) or Unif(0,1) < \( \theta_l \) then
10: \( \mathcal{I}_A \leftarrow \mathcal{I}_A \cup \{l\} \)
11: end if
12: \( \mathcal{I}_A \leftarrow \mathcal{I}_A \setminus \{l\} \)
13: end for
14: Set \( M = B/|\mathcal{I}_A| \)
15: end function

3.5. Adaptive predictor importance

We propose an adaptive version of the Markov chain sampler where a predictor’s chance of being included in an Add move, for example, is proportional to its importance score. Let \( T \) denote the length of the MCMC chain, and define \( b_0 = \max\{100, \lfloor T/10 \rfloor \} \) as a burn-in period. The importance score of a predictor is initialized as \( v_j = 1, j = 1, \ldots, p \), and subsequently updated as

\[
v_j(t + 1) = v_j(t) + \frac{\sum_{l \in I_A} I(\rho_l > 0, j \in \gamma_l)}{k_a^*} \left( \frac{I(t \leq b_0) t}{b_0} + \frac{I(t > b_0)}{(t - b_0)\zeta} \right),
\]

where \( \zeta \in (1/2, 1] \). Following convention for “optimal learning rates” in stochastic gradient descent algorithms, we fix \( \zeta = 2/3 \). Here, importance scores are increased for predictors frequently included in the past, and adaptations designed to be modest, typically converging to an equilibrium where \( \max(v_j) \approx 10 \) (see Figure 9). Stationarity is maintained subject to diminishing adaptation of predictor importance scores used in the paired-move DMTM sampler (Roberts and Rosenthal, 2007).

The “toggle” mechanism introduced in Section 3.3.1 for selecting a subset of predictors from which to form one-away neighborhoods of an inclusion vector is extremely flexible. The weight function (11) defined in the paired-move DMTM algorithm introduced in Section 3.3.2 lends itself naturally to adaptive MCMC.
functions, a non-optimized AGP implementation in R completes in 121.18 seconds for AGP using Algorithm 2 with default initializations. Averaging across independent replicates and test functions, the expected size of an Add neighborhood is\
\[ \sum_{v} f(v_j) = M \left( \frac{p-|I|}{b+M} \right) \approx M. \]
Under the sparsity assumption, the vast majority of predictors retain scores \( v_j \approx 1 \), and so the probabilistic control on neighborhood sizes is maintained. For paired Swap moves, letting \( f(v) = f(v)/|\gamma| \) maintains the same limit. As importance scores are updated, predictors with large \( v_j \) are promoted within proposal neighborhoods (note: by our choice of weight function, all remove configurations are considered in paired forward/reverse neighborhoods). Parameter \( \alpha \) controls the degree of inclusion bias, and we recommend \( \alpha \in [1, 1.5] \). As an example: assume \( \alpha = 1.5 \), \( p = 1000 \), \( k_{\text{max}} = \lceil \sqrt{p} \rceil \), \( B = 10k_{\text{max}} \) and assume an active set of size \( |I_A| = 10 \). Then, predictors with \( v_j = 5(10) \) have inclusion probability \( 0.26(0.50) \).

### 4. Numerical experiments

The AGP method is compared to state-of-the-art competitors in terms of predictive root-mean-squared error (RMSE) and support recovery across a variety of simulated truths. A 1000 iteration MCMC chain is run for AGP using Algorithm 2 with default initializations. Averaging across independent replicates and test functions, a non-optimized AGP implementation in R completes in 121.18 minutes for \( p = 1000 \). Results discard the first 200 samples and thin subsequent samples by selecting every fourth draw.

Synthetic data are generated by drawing \( x_{ij} \sim \text{Unif}(0,1) \) and \( y_i \sim N(f(x_i), 1) \), \( 1 \leq i \leq n = 100, 1 \leq j \leq p \). When the predictor dimension is small, a single GP prior with an ARD squared exponential kernel can identify important predictors but offers no characterization of lower dimensional interaction. Tree models can accomplish both tasks, but often suffer in terms predictive accuracy for nonlinear functions. We select two popular ensemble methods, BART (Chipman et al., 2010) and Random Forests (RF) (Breiman, 2001). The Lasso (Tibshirani, 1996) is selected as a final competitor. The NULL model reports prediction using a naïve average over training outcomes.

#### 4.1. Variable selection and interaction recovery

Simulated experiments in this section concern test functions in Table 1. The degree of nonlinearity and additivity varies substantially across these functions, but all are sparse considering the assumed data generating process and predictor dimension. The number of active AGP components is adaptively tuned using Algorithm 1, and Figure 3 displays a histograms for the number of active (and non-empty) components post burn-in. Figure 4 plots the cumulative variance explained as a function of the number of active components (sorted by increasing importance). The marginal variance explained by an active AGP component is given by \( \rho_{I}^2/(1 + \sum_{k \in I_A} \rho_k^2) \), \( l \in I_A \). Here, one observes that learning is most evenly distributed across active components in the linear regression example, whereas most of the learning for the “single component” example is captured by a single AGP component.

Table 1. Simulated test functions. All experiments fix \( n = 100 \) and \( p = 1000 \).

| Function            | Formula                                                                 |
|---------------------|-------------------------------------------------------------------------|
| Friedman*           | \( f = 10 \sin(\pi x_1 x_2) + 10 \cos(\pi (x_1 x_4 + x_5)) + 20(x_6 - 0.5)^2 + 10 x_7 \) |
| Confounded          | \( f = 10 \cos(\pi (x_1 + x_2 + x_3)) + 10 \sin(\pi (x_2 + x_4)) + 10 x_5 (x_1 + x_2) \) |
| Linear regression   | \( f = 5(x_1 + \ldots + x_5) + 2(x_6 + \ldots + x_{10}) \)              |
| Single component    | \( f = 10 \cos(\pi (x_1 + 5x_2)) \)                                     |

Runtimes are averaged over 5 independent replicates for each test function considered in Section 4.1. Simulations were run on an x86-64 Intel(R) Core(TM) i7-3770.
An interaction graph summarizes co-appearances of a predictor pairs across any of AGP’s active components. Here, marginal inclusion probabilities are thresholded at the $(1 - q/p)$th quantile, focusing on relationships between predictors deemed to be most important (by default, we set $q = 10$). A measure for the marginal importance of the $j$-th predictor is calculated as: (1) LASSO: $|\hat{\beta}_j|$; (2) BART: $\frac{1}{T} \sum_{t=1}^{T} (n_{jt} - \hat{n}_t) / (\hat{n}_t - \bar{n}_j)$, with $\hat{n}_t = \max_j (n_{jt}), \bar{n}_t = \min_j (n_{jt})$, where $n_{jt}$ denotes the number of unique occurrences of the $j$-th predictor across the tree ensemble at MCMC iteration $t$; (3) RF: using ‘importance’ in the randomForest R package; and (4) AGP: $\frac{1}{T} \sum_{t>T-S} \max \{I(j \in \gamma(t)), l \in I_A\}$, namely the fraction of post burn-in iterations containing the $j$-th predictor in any component.

In terms of support recovery, AGP is the clear winner for all test functions, identifying the most important predictors with the least number of false positives. In contrast, support recovery using inclusion probabilities from BART and RF can be challenging even when $p$ is small (comparing Figures 3 and 8). In addition, AGP improves dramatically over all competitors in terms of predictive mean-squared-error (see Section 4.2).

**Friedman** Additive effects are well separated for the modified Friedman function (Friedman, 1991). As shown in Figure 3, the median number of non-empty components used by AGP to recover this structure is 5, two of which are used to isolate the two- and three-way interaction effects, while another two components are used to capture the univariate main effect involving predictors 6 and 7, respectively.

**Confounded effects** Nonlinear additive effects in the true function share predictor 2, while predictors 1 and 2 also appear as bilinear main effects with predictor 5. The latter terms are weak in comparison to the other additive effects present in the regression function. The interaction graph in Figure 3 shows that AGP successfully recovers the $(1,5)$ main effect, in addition to isolating predictor interaction $(2,4)$ from the three-way $(1,2,3)$ predictor interaction. Weaker edge weights between predictor pairs $(2,5), (3,5)$ and $(4,5)$ indicate such configurations were explored during MCMC, but were less persistent. Consistent with this predictor interaction recovery, AGP has 5 or fewer non-empty components 65% of the time. AGP settles on such a configuration in 6 / 10 replicate experiments. The remaining runs discover a local configuration mode which is notoriously difficult to escape from. Here, a single AGP component models the bilinear $(1,5)$ main effect, while another models predictor interaction $(1,2,3,4)$. Local neighborhood moves do not subsequently isolate the latter into $(1,2,3)$ and $(2,4)$ constituent interactions (refer to the discussion in Section 6).

**Linear regression** AGP and the Lasso both correctly identify predictors 1-5 and 9. AGP also identifies predictor 6 (with inclusion probability 0.33 over a single MCMC chain). The Lasso has a number of false positives (i.e., non-zero coefficients for irrelevant predictors), whereas AGP produces but a single false positive (see Figures 3 and 8). Compared to other test functions, AGP uses a larger number of components to model this fully additive function. The histogram in Figure 3 shows that AGP has 6 or fewer non-empty active components roughly 50% of the time. Since AGP identifies 7 of 10 predictors, components model univariate linear terms in at least half of the MCMC iterations. Edge weights in the interaction graph are relatively consistent with this view.

**Single component** This function is difficult to model due to its extreme sparsity, nonlinearity and lack of additive structure. With sparsity inducing priors over the model space, along with an ability to adapt to the underlying smoothness of the regression function (via component adaptive scaling), AGP recovers the true support and predictor interaction. All other competitors fail to do so. AGP contains 3 or fewer non-empty components 90% of the time as depicted in Figure 3. A single component models the $(1,2)$ interaction, while other active components enable occasional exploration of additional joint configurations $\gamma = \{(1,2), 1, -\}$, and $\gamma = \{(1,2), 2, -\}$.

### 4.2. Predictive performance

RMSE is reported on a set of 200 test observations generated for each function described in Section 4. Results are averaged over several independent replications and reported along with standard errors. AGP’s improved support and interaction recovery leads to a dramatic improvement in predictive performance and uncertainty quantification over state-of-the-art methods in low and high dimensional settings. This validates...
Fig. 3. Simulated test functions 1-4 for Section 4.1 with $p = 1000$. Left: number of active vs. non-empty (utilized) components, with median sizes appearing as vertical dashed lines; Middle: marginal inclusion probabilities for the AGP model (index on log-scale); Right: an interaction graph among the most important predictors. Vertex sizes are proportional to predictor importance, and edges between predictor pairs are drawn in proportion to their co-appearance across GP components.

### Components Marginal inclusion Interaction graph

| Friedman $n = 100$ $p = 1000$ |
|-------------------------------|
| Friedman $n = 100$ $p = 1000$ |
| Linear $n = 100$ $p = 1000$ |
| Single $n = 100$ $p = 1000$ |

the additive-interactive modeling framework for GP regression, and confirms that our adaptive MCMC sampler using paired and inter-component moves provides reliable predictor and interaction recovery in high
Additive Gaussian Process Regression

Fig. 4. A plot of the cumulative variance explained as a function of the median number of active components (sorted by increasing importance). The fraction of the marginal variance explained by an active GP component is \( \frac{\rho_l^2}{1 + \sum_{s \in I_A} \rho_s^2} \), \( l \in I_A \). Median active sizes appear as vertical dashed lines in the histograms shown in Figure 3.

Table 2. Predictive RMSE for test functions defined in Section 4.1 with \( n = 100 \) and \( p = 1000 \). Standard errors using 10 replicates appear as subscripts on averaged RMSE values.

|               | AGP       | BART     | RF        | LASSO     | NULL     |
|---------------|-----------|----------|-----------|-----------|----------|
| Friedman      | 1.42_{0.08} | 6.81_{0.50} | 6.76_{0.52} | 6.18_{0.43} | 7.82_{0.39} |
| Confounded    | 2.61_{0.53} | 9.27_{0.41} | 9.00_{0.46} | 7.87_{0.04} | 9.68_{0.28} |
| Linear regr.  | 1.52_{0.12} | 3.38_{0.09} | 3.43_{0.10} | 1.75_{0.16} | 3.70_{0.14} |
| Single comp.  | 1.97_{0.27} | 7.07_{0.27} | 7.08_{0.27} | 7.23_{0.53} | 7.07_{0.28} |

4.2.1. Predictive comparisons for lower dimensional experiments

A single GP prior with an ARD squared exponential covariance function \( \parallel \) placed on the unknown regression function can identify important predictors and has been the primary workhorse for variable selection in the GP regression setting (Neal, 1997; Williams and Rasmussen, 1996). However, it is well known that MCMC and likelihood based methods for estimation of GP-ARD parameters suffer as the predictor dimension grows. This fact, together with recent theoretical developments for GP regression using an additive-interactive framework, are compelling reasons to pursue the proposed APG model (1). It is in instructive, therefore, to compare performance between the two methods.

Data are generated with \( n = 100 \) and \( p = 50 \) for each test function, and predictive performance is summarized in Table 3. The ARD competitor is fit by iteratively computing MAP estimates for discretized covariance parameters until convergence is reached. AGP’s improvement over ARD-MAP indicates how modeling of additive structure enables superior estimation of the regression function even when \( p \) is small. In addition, as predictor dimension \( p \) grows, BART and RF degrade considerably in terms of predictive performance, whereas AGP remains remarkably robust to the 20-fold increase in predictor dimension (compare Tables 2 and 3).

\( \parallel \) The ARD covariance function is \( C^{\text{ARD}}(x, \tilde{x}) = \rho^2 \exp \left( - \sum_{j=1}^{p} \lambda_j^2 (x_j - \tilde{x}_j)^2 \right) \).
In addition, predictor count varies from moderate to very large across these examples. This allows us to offer a wide range of predictor effect types, from mostly additive to including high order interaction.

### 5. Real data illustrations

We analyze four real datasets which are commonly used in the statistics and machine learning literature, and offer a wide range of predictor effect types, from mostly additive to including high order interaction. In addition, predictor count varies from moderate to very large across these examples. This allows us to validate the AGP model (1) on data that may satisfy only some of the additive-interaction assumptions. The number of AGP components (and neighborhood budget for PRNS moves) is adaptively tuned using Algorithm 1, with BART, RF, and the Lasso taken to be our competitors as before.

Competing methods vary dramatically in terms of their relative performance (RMSE) across the four datasets as shown in Table 4. For Riboflavin microarray and biscuit-dough spectroscopy (Cookie) data, regularized linear regression (i.e., the Lasso) clearly gives better averaged test prediction than ensemble nonparametric methods, whereas the opposite holds for Crime and Boston housing data (i.e., BART and RF are good competitors). Note that AGP remains competitive (best or second best predictive performance) across the board, owing to its ability to accommodate varying degrees of sparsity, interaction, and additivity. Moreover, this adaptability appears robust across predictor dimension, accommodating even settings with high predictor correlation (e.g., Cookie data).

To understand how learning is distributed across AGP’s active components, Figure 6 plots the cumulative variance explained (components sorted by increasing importance) for the median-sized model over all MCMC...
Table 4. Performance of AGP against the same by BART, RF, Lasso, and the mean prediction rule (NULL) for well known real datasets. Left: the fraction of explained variance computed as $R^2 = 1 - (\text{RMSE}/\bar{y})^2$ for AGP; Right: predictive (hold-out) RMSE averaged over random test/train splits with standard errors appearing as subscripts.

| Dataset    | AGP $R^2$ | AGP RMSE | BART | RF | Lasso | NULL |
|------------|-----------|----------|------|----|-------|------|
| Boston     | 0.890.04  | 3.000.54 | 3.480.49 | 3.470.63 | 5.020.54 | 9.150.47 |
| Crime      | 0.640.15  | 421.920.86 | 415.540.81 | 430.4078.50 | 458.7371.66 | 741.920.63 |
| Riboflavin | 0.740.13  | 0.470.12 | 0.540.08 | 0.560.07 | 0.430.13 | 0.800.10 |
| Cookie     | 0.950.42  | 1.551.44 | 1.440.25 | 1.980.13 | 0.800.10 | 1.980.13 |

Iterations. These sizes appear as vertical dashed lines in Figure 7. For the Riboflavin and Crime data, several active components contribute to AGP’s learning, whereas learning in the Boston housing and Cookie data is primarily captured in one or two GP components. The right panel in Figure 7 plots posterior predictor inclusion probabilities and interactions for the AGP model. As shown, AGP identifies important predictors (only a handful in some, very many in others) and their interactions (strong interaction effects in some; mostly additive, non-interactive effects in others). Table 5 compares the mean and variance for predictive RMSE (on held out test data) calculated on random train/test partitions of the real datasets. Here, predictive RMSE for MCMC with and without ICM moves are comparable, but estimates from runs using ICM moves generally have smaller variance and are less sensitive to initialization.

For each real dataset, Figure 9 in Section D of the Appendix displays (a) bar-plots for component smoothness parameters $\rho_l, \lambda_l, l \in I_A$; (b) a trace plot of posterior draws for variance parameter $\sigma^2 \sim \pi(\cdot | \xi, y)$ in (5); and (c) predictor importance scores $v_j, j = 1, \ldots, p$ learned using Algorithm 1. The trace plot of the posterior variance serves as a model fit diagnostic and a measure of MCMC mixing and stability.

Fig. 6. A plot of the cumulative variance explained as a function of the median number of active components (sorted by increasing importance). The fraction of the marginal variance explained by an active GP component is $\rho^2_l/(1 + \sum_{s \in I_A} \rho^2_s), l \in I_A$. Median active sizes appear as vertical dashed lines in the histograms shown in Figure 7.

6. Discussion

Extensions of AGP to accommodate binary outcomes via the probit model (Albert and Chib, 1993) are underway, and future work will extend AGP to survival outcome data as well. An optimized R package for AGP is being written to accommodate larger sample sizes and predictor dimensions (i.e., $n \approx 10^4$, $p \approx 10^6$). Here, developments in GPU and distributed computing, together with low-rank matrix approximation, can enable learning of an association graph between predictors over parallelized MCMC chains. Robust estimates
for the predictor correlation matrix (Schäfer and Strimmer, 2005; Bickel and Levina, 2008) may also be used to enhance the efficiency of learning predictor inclusion scores. Finally, additional MCMC strategies for moves in the joint space of component inclusion are required to tackle local modality issues which can arise when additive effects share predictors (e.g., the “confounded” test function in Section 4.1). There, in some MCMC runs where an AGP component models predictor interaction (1,2,3,4), local neighborhood moves do not subsequently isolate the latter into constituent (1,2,3) and (2,4) interactions. The “neighbor of a good neighbor is a better neighbor” effect was a driving force that lead to the paired-move neighborhood sampler (PRNS), which is effective at overcoming this obstacle when additive components are fairly well separated (e.g., the modified Friedman function in Section 4.1). For the “confounded” example, this is far from true – here, a donate ICM move can only overcome this effect in the (low probability) event that predictor 2 is proposed and accepted in another component via an Add move, followed by a donate ICM move from the component in question in a subsequent iteration. In addition, consider the case where (1,2,3) and (4,5) are modeled in separate components. A paired move enables predictor 2 to be added to the second component, and subsequent donate ICM (paired remove) moves allow (2,4,5) to be separated into (2,4) and 5. On the other hand, ICM also enables one to move toward the undesired local mode, i.e., by considering donate ICM moves from the component with (4,5).

Acknowledgement

SQ is partially supported by grants R01-ES-017436 and R01-ES-017240 from the National Institute of Environmental Health Sciences (NIEHS) of the National Institutes of Health (NIH). ST is partially supported by grant R01-ES-017436 from NIEHS of NIH. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NIEHS or NIH.

Table 5. Averaged RMSE (on held out test data) and RMSE variance for real data illustrations for AGP using Algorithm 2 for MCMC with and without the use of ICM moves.

|                  | No ICM |         | With ICM |         |
|------------------|--------|---------|----------|---------|
|                  | Mean   | Var     | Mean     | Var     |
| Boston           | 3.16   | 0.55    | 3.00     | 0.54    |
| Crime            | 424.19 | 74.32   | 421.92   | 70.86   |
| Riboflavin       | 0.46   | 0.12    | 0.47     | 0.12    |
| Cookie           | 0.42   | 0.02    | 0.42     | 0.02    |
Fig. 7. Comparison across well known datasets. **Left:** number of active vs. non-empty (utilized) components, with median sizes appearing as vertical dashed lines; **Middle:** marginal inclusion probabilities for the AGP model (index on log-scale); **Right:** an interaction graph among the most important predictors. Vertex sizes are proportional to predictor importance, and edges between predictor pairs are drawn in proportion to their co-appearance across GP components.

| Dataset   | # predictors | Sample size | Splits: |          |          |          |
|-----------|--------------|-------------|---------|----------|----------|----------|
| Boston    | 13           | 507         | 6, 70 - 30% |          |          |          |
| Crime     | 102          | 416         | 5, 75 - 25% |          |          |          |
| Riboflavin| 4088         | 71          | 5, 80 - 20% |          |          |          |
| Cookie    | 700          |             |          |          |          |          |

Train size: 40
Test size: 34
A. Additive Gaussian process sampling

A.1. Markov chain sampling for AGP regression

Section 3.3.2 introduces a reversible paired-move neighborhood MTM sampler to efficiently explore the space of one-away inclusion vectors. Section 3.4.1 introduces inter-component moves (ICM) to explore local neighborhoods of joint configuration state $\gamma = \{\gamma_1, \ldots, \gamma_k\}$ that preserves the stationarity distribution of the paired-move sampler. Algorithm 2 uses these moves together with predictor importance adaptation (see Section 3.5) to draw approximate posterior samples for AGP parameters $\gamma, \rho, \lambda$.

Algorithm 2 AGP Markov chain sampling

**Input:** (Defaults) $k_{\text{min}} = \lceil \log(p) \rceil$, $k_{\text{max}} = \lceil p^{1/2} \rceil$, $b_0 = 100$, $\Delta = 0.20$.

**Output:** Approximate posterior draws for AGP parameters $\{\rho, \lambda, \gamma\}$

1: function AGP.SAMPLE \(\triangleright\) Sampling for AGP model (1)
2: Call UPDATE.ACTIVESET($\xi, \mathcal{A}, B$) \(\triangleright\) see Algorithm 1
3: Sample $\tau \sim \text{Beta}\left(\mu' = \frac{d' + \sum_{l \in \mathcal{A}} d_l}{\rho'}, \nu' = (1 + k_n)p\right)$
4: if $\text{Unif}(0, 1) < \Delta$ then \(\triangleright\) Do ICM move
5: Select ICM move $m \in \{\text{CD}, \text{PD}, \text{PS}\}$
6: Propose $\xi' = (\gamma', \theta') \sim p(\cdot | A_m(\gamma), \theta, y)$
7: if $\text{Unif}(0, 1) < \alpha_{m}(\xi, \xi')$ then \(\triangleright\) see Section 3.4.1 (13)
8: Set $\gamma, \rho, \theta, \gamma', \rho', \theta'$
9: if $m = \text{CD}$ then
10: go to line 13
11: end if
12: else
13: for $l \in \mathcal{A}$ do
14: Sample $\rho_l, \lambda_l \sim \pi(\cdot | \gamma_l, \xi(l), y)$ \(\triangleright\) see Section 3 (4)
15: end for
16: end if
17: else \(\triangleright\) Do paired-move DMTM
18: for $l \in \mathcal{A}$ do
19: Select pared-move $m \in \{A, R, S\}$
20: Propose $\gamma_l' \sim p(\cdot | N_m(\gamma_l), \xi(l), y)$ \(\triangleright\) see Section 3.3.2 (12)
21: if $\text{Unif}(0, 1) < \alpha_{m}(\gamma_l, \gamma_l')$ then
22: Set $\gamma_l \leftarrow \gamma_l'$
23: end if
24: Sample $\rho_l, \lambda_l \sim \pi(\cdot | \gamma_l, \xi(l), y)$ \(\triangleright\) see Section 3 (4)
25: if $\rho_l > 0, d_l > 0$ then
26: Update $v_j$, $j \in \gamma_l$
27: end if
28: end for
29: end if
30: end if
31: end function

A.2. Predictive inference for AGP regression

Let $\Lambda = \sum_{l=1}^{k} \rho_l^2 C_l$ represent the scaled covariance of the aggregated GP $f(x) = \sum_{l=1}^{k} f_l(x)$, and $C_l$, $1 \leq l \leq k$ are the scaled component covariances defined in Section 2. Let $s_f = (f(x_1), \ldots, f(x_n))^t$ stand for the $n$-vector of aggregated GP realizations, $f(x_i) = \sum_{l \in \mathcal{A}} f_l(x_i)$. For a new observation $(x^*, y^*)$, the joint distribution of $f(x^*)$ with aggregated GP realizations over the training data follows from Kolmogorov
consistency for Gaussian processes, namely

\[ (f(x^*), f(x_1), \ldots, f(x_n))' \sim N(0, \sigma^2 A), \quad A = \begin{pmatrix} \Lambda^+ & \Lambda_{cr} \\ \Lambda_{cr}' & \Lambda_{1:n} \end{pmatrix}. \quad (16) \]

This implies \( f(x^*)|s_f, \sigma^2 \sim N(\Lambda_{cr}A_{1:n}^{-1}s_f, \sigma^2(\Lambda^+-\Lambda_{cr}A_{1:n}^{-1}\Lambda_{cr}')) \), and the posterior predictive distribution for mean response \( f(x^*) \) is given by

\[ \pi(f(x^*)|y) = \int \pi(f(x^*)|s_f, \sigma^2) \pi(s_f, \sigma^2|x, \gamma) \pi(\gamma|x) d(\xi, s_f, \sigma^2). \quad (17) \]

Approximate posterior draws for AGP parameters \( \xi = \{\gamma, \rho, \lambda\} \) are obtained via Algorithm 2. Sampling from (17) proceeds by (1) drawing \( (s_f, \sigma^2)^{(s)} \sim \pi(\cdot, \cdot|\xi^{(s)}, y) \), available in closed-form; and (2) drawing \( f^{(s)}(x^*) \sim \pi(\cdot|s_f, \sigma^2) \) as given. Using the final \( S \) draws, the posterior predictive mean is approximated as

\[ \mathbb{E}(y^*|x^*, y) \approx \frac{1}{S} \sum_{i>T-S} f^{(i)}(x^*). \quad (18) \]

Here, quantiles of predictive estimates \( f^{(1)}(x^*), \ldots, f^{(S)}(x^*) \) provide point-wise posterior credible intervals (see Figure 5). In addition, promising inclusion configurations may be identified by thresholding \( \pi(\gamma^{(t)}, \rho^{(t)}, \lambda^{(t)}|y) \). For a subset of inclusion configurations \( \mathcal{M} \), marginal posterior probabilities may be obtained by running an MCMC chain over AGP parameters \( \rho, \lambda \) (in parallel for every configuration state \( \gamma \in \mathcal{M} \)), and computing

\[ \hat{\pi}(\gamma|y) = \frac{1}{T} \sum_{t=1}^{T} \pi(\gamma|\rho^{(t)}, \lambda^{(t)}, y), \quad \hat{P}(m_\gamma|y) = \frac{\hat{\pi}(\gamma|y)}{\sum_{\gamma \in \mathcal{M}} \hat{\pi}(\gamma|y)}. \quad (19) \]

\section{B. Efficient matrix inversion for AGP sampling}

\subsection{B.1. Low-rank approximation for AGP}

Evaluating the likelihood function under a GP regression model requires inverting the \( n \times n \) (kernel) covariance matrix. The neighborhood sampler for updating component inclusion vectors using back-fitting (see Sections 3.1 and 3.2.1) requires likelihood evaluations for every \( \gamma \in N(\gamma) \) at each MCMC iteration which grows as \( O(n) \). A paired-move neighborhood sampler was introduced in Section 3.3.2 to probabilistically control the number of neighbors that must be scored.

When \( n \) exceeds several hundred observations the \( O(n^3) \) matrix inversion adds significantly to the per-iteration complexity. This arises from Cholesky factorizations for aggregated AGP covariance \( K = \sum_{l=1}^{d} \eta^2 C_l = \tilde{R}' \tilde{R} \), however we employ a pivoted low-rank matrix approximation which enables matrix inversion at a reduced \( O(n^2) \) cost, \( r \ll n^{**} \). Here, \( \tilde{K} \approx \tilde{R}' \tilde{R} + D \), for an upper triangular matrix \( \tilde{R} \times n \) and diagonal matrix \( D \). The marginal likelihood calculation in (3) involving \( \Sigma^{-1} = (I_n + K)^{-1} \approx (A + \tilde{R}' \tilde{R})^{-1}, \quad A = I_n + D \), can then be computed efficiently via an application of the Sherman-Morrison Woodbury formula. Harbrecht et al. (2012) provide an overview of pivoted inversion. Other procedures in the predictive-process literature choose a set of “knot points” to tackle the cubic-order complexity, where points are adaptively selected in response to changes in the canonical metric induced by uncertainty in the parameters controlling the GP covariance function (Banerjee et al., 2008; Tokdar, 2011a; Banerjee et al., 2012).

\section{C. Validity of inclusion vector sampling for AGP}

Below, stationary distribution \( \pi(\cdot|-) \) is abbreviated as \( \pi(\cdot) \).

\[ ** \text{By default, Algorithm 2 sets pivoted rank } r = \lceil n, \frac{1}{2}(\log n)^2 \rceil \].
C.1. Reversibility for paired-move neighborhood sampling

Consider the vanilla paired-move neighborhood sampler having proposal distribution (8) and acceptance probability (9). Below, detailed balance is verified for a paired “Add-remove” (“Remove-add” follows from symmetry, which is precisely the reversibility condition being checked) and “Swap-swap” proposals. Given component inclusion vector \( \gamma \), one first chooses between Add, Remove and Swap with probabilities \( w_A(|\gamma|) \), \( w_R(|\gamma|) \) and \( w_S(|\gamma|) \), respectively.

If \( m = A \): construct one-add neighborhood \( N_A(\gamma) = \{ \tilde{\gamma} = \gamma + 1_j : j \notin \gamma \} \). Select \( \gamma' = \gamma + 1_k \) with probability \( \pi(\gamma')/\sum_{\tilde{\gamma} \in N_A(\gamma)} \pi(\tilde{\gamma}) \), namely the RNS proposal distribution (8) restricted this set. The paired reverse one-remove neighborhood \( N_R(\gamma') \) always contains \( \gamma \). Accept with probability \( \alpha_m(\gamma, \gamma') \) given by (9). Then,

\[
\pi(\gamma) T(\gamma, \gamma') = \pi(\gamma) w_A(|\gamma|) \frac{\pi(\gamma')}{\sum_{\tilde{\gamma} \in N_A(\gamma)} \pi(\tilde{\gamma})} \min \left\{ 1, \frac{w_R(|\gamma'|)}{w_A(|\gamma|)} \frac{\sum_{\tilde{\gamma} \in N_R(\gamma') \pi(\tilde{\gamma})}{\sum_{\tilde{\gamma} \in N_A(\gamma)} \pi(\tilde{\gamma})} \right\}
\]

If \( m = S \): construct one-swap neighborhood \( N_S(\gamma) = \{ \tilde{\gamma} = \gamma - 1_m + 1_n : (m, n) \in \gamma \times [\gamma]^c \} \). Select \( \gamma' = \gamma - 1_j + 1_k \) with probability \( \pi(\gamma')/\sum_{\tilde{\gamma} \in N_S(\gamma)} \pi(\tilde{\gamma}) \), namely the RNS proposal distribution (8) restricted this set. The paired reverse swap neighborhood \( N_S(\gamma') \) always contains \( \gamma \). Accept with probability \( \alpha_m(\gamma, \gamma') \) given by (9). Then,

\[
\pi(\gamma) T(\gamma, \gamma') = \pi(\gamma) w_S(|\gamma|) \frac{\pi(\gamma')}{\sum_{\tilde{\gamma} \in N_S(\gamma)} \pi(\tilde{\gamma})} \min \left\{ 1, \frac{w_R(|\gamma'|)}{w_S(|\gamma|)} \frac{\sum_{\tilde{\gamma} \in N_R(\gamma') \pi(\tilde{\gamma})}{\sum_{\tilde{\gamma} \in N_S(\gamma)} \pi(\tilde{\gamma})} \right\}
\]

The expression \( \pi(\gamma) T(\gamma, \gamma') \) is symmetric in \( \gamma \) and \( \gamma' \) in both cases. This completes the proof.

C.2. Reversibility for paired-move discrete multiple-try Metropolis sampler

The proof below makes use of weight function \( \omega(\gamma, v_j, m) \) defined in (11), and acceptance probability \( \alpha_m(\gamma, \gamma') \) given by (12). Given current inclusion vector \( \gamma \), choose between Add, Remove and Swap with probabilities \( \pi(\gamma) w_A(|\gamma|) \), \( \pi(\gamma) w_R(|\gamma|) \) and \( \pi(\gamma) w_S(|\gamma|) \), respectively. Let \( m \in \{ A, R, S \} \) denote this choice, and \( m' \in \{ R, A, S \} \) is the corresponding paired reverse neighborhood. In each case, the transition kernel is described in Section 3.3.2.

If \( m \in \{ A, R \} \): See steps (2a) and (4a/b) in Section 3.3.2 for construction of transition kernel \( T(\gamma, \gamma') \). Let \( \omega_j = \omega(\gamma, v_j, m) \) denote the forward bernoulli inclusion probability for predictor \( j \), and \( \bar{\omega}_j = \omega(\gamma', v_j, m') \) denotes its inclusion probability in the reverse paired neighborhood. Then for \( \gamma' = \text{tog}(k, \gamma) \),

\[
\pi(\gamma) T(\gamma, \gamma') = \pi(\gamma) w_m(|\gamma|) \sum_{\eta, \eta' \in \{0,1\}^p \eta_k = \eta'_k = 1} \left\{ \prod_{j \neq k} \omega_j^{\eta_j} (1 - \omega_j)^{1-\eta_j} \bar{\omega}_j^{\eta'_j} (1 - \bar{\omega}_j)^{1-\eta'_j} \right\}
\]

\[
\times \frac{\pi(\gamma')}{\sum_{j \neq \gamma_j} \pi(\text{tog}(j, \gamma))} \min \left\{ 1, \frac{w_m'(|\gamma'|)}{w_m(|\gamma|)} \frac{\bar{\omega}_k \sum_{j \neq \gamma_j} \pi(\text{tog}(j, \gamma))}{\omega_k \sum_{j \neq \gamma_j} \pi(\text{tog}(j, \gamma'))} \right\}
\]

\[
= \pi(\gamma) \pi(\gamma') \sum_{\eta, \eta' \in \{0,1\}^p \eta_k = \eta'_k = 1} \left\{ \prod_{j \neq k} \omega_j^{\eta_j} (1 - \omega_j)^{1-\eta_j} \bar{\omega}_j^{\eta'_j} (1 - \bar{\omega}_j)^{1-\eta'_j} \right\}
\]

\[
\times \min \left\{ \frac{w_m(|\gamma|) \omega_k}{\sum_{j \neq \gamma_j} \pi(\text{tog}(j, \gamma))}, \frac{w_m(|\gamma'|) \bar{\omega}_k}{\sum_{j \neq \gamma_j} \pi(\text{tog}(j, \gamma'))} \right\},
\]
If \( m = S \): See steps (2b) and (4c) in Section 3.3.2 for construction of transition kernel \( T(\gamma, \gamma') \). Denote forward bernoulli inclusion probabilities for a pair of predictors \((r, a) \in \gamma \times [\gamma]^c\) as \( \omega_r = \omega(\gamma_r, \nu_r, m = R) \), \( \omega_a = \omega(\gamma_a, \nu_a, m = A) \). Likewise for reverse inclusion probabilities, let \( \omega'_r = \omega(\gamma'_r, \nu_r, m = R) \), \( \omega'_\gamma = \omega(\gamma'_\gamma, \nu_\gamma, m = A) \) and \((r, a) \in \gamma' \times [\gamma]^c \). For \( \gamma' = \text{tog}(r^*, a^*) \), the following facts are noted: (i) \( \omega_r = \omega'_r \), \( r \in \gamma \setminus \{r^*\} \); (ii) \( \omega_a = \omega'_a \), \( a \in \gamma' \setminus \{a^*\} \); (iii) \( \omega_{r=a} = \omega'_{r=a} = 1 \); (iv) \( \omega_{a=r} = f(v_r) \) and \( \omega'_{a=r} = f(v'_r) \), see (13); and (v) \( w_S(\gamma) = w_S(\gamma') \). Then,

\[
\pi(\gamma)T(\gamma, \gamma') = \pi(\gamma)w_S(\gamma) \times \sum_{(r,a) \in \gamma \times [\gamma]^c} \left[ \prod_{(r,a) \neq (r^*, a^*)} \omega_r^{\eta_r} \omega_a^{\eta_a} \omega_{\gamma_r}^{\eta_{\gamma_r}} \omega_{\gamma_a}^{\eta_{\gamma_a}} \right] \\
\times \sum_{(r,a) : \eta_r = \eta_a = 1} \pi(\text{tog}((r,a), \gamma)) \\
\times \min \left\{ \frac{w_S(\gamma)}{\pi(\text{tog}((r,a), \gamma))}, \frac{w_S(\gamma')}{\pi(\text{tog}((r,a), \gamma'))} \right\}.
\]

\[
\pi(\gamma)T(\gamma, \gamma') = \pi(\gamma)w_S(\gamma) \times \sum_{(r,a) \in \gamma \times [\gamma]^c} \left[ \prod_{(r,a) \neq (r^*, a^*)} \omega_r^{\eta_r} \omega_a^{\eta_a} \omega_{\gamma_r}^{\eta_{\gamma_r}} \omega_{\gamma_a}^{\eta_{\gamma_a}} \right] \\
\times \sum_{(r,a) : \eta_r = \eta_a = 1} \pi(\text{tog}((r,a), \gamma)) \\
\times \min \left\{ \frac{w_S(\gamma)}{\pi(\text{tog}((r,a), \gamma))}, \frac{w_S(\gamma')}{\pi(\text{tog}((r,a), \gamma'))} \right\}.
\]

In all cases, i.e. for paired-moves \( m \in \{A, R, S\} \), the expression \( \pi(\gamma)T(\gamma, \gamma') \) is symmetric in \( \gamma \) and \( \gamma' \). This completes the proof.

C.3. Proof of Lemma 3.4

Given state vector \( \xi = (\gamma, \theta) \), \( \xi_f = (\gamma_f, \rho_f, \lambda_f) \) and \( l \in I_A \), an ICM move proceeds by choosing between “cross donate”, “paired donate”, or “paired swap” moves with probabilities \( w_{\text{CD}}, w_{\text{PD}} \) and \( w_{\text{PS}} \), respectively. For move \( m \in \{ \text{CD, PD, PS} \} \), the corresponding ICM neighborhood \( A_m(\gamma) \) is constructed (see Section 3.4.1). Select \( \xi' \) with probability \( \pi(\xi'|-) \sum_{\xi \in A_m(\gamma)} \pi(\xi|-) \) and accept it with probability \( \alpha_m(\xi, \xi') \) given by (13). Below, \( k_{\text{ne}} \leq k_a \) denotes the number of non-empty active components.

If \( m = \text{CD} \): For \( \gamma' = \{\ldots, \gamma_u + 1_k, \gamma_v - 1_k, \ldots\} \), where component \( \gamma_u \) donates predictor \( k \) to component \( \gamma_u \), and

\[
\pi(\gamma|\theta, -)T(\gamma, \gamma') = \pi(\gamma|\theta, -) \frac{w_{\text{CD}}}{k_{\text{ne}}} \sum_{\gamma \in N_{\text{CD}}(\gamma; v)} \pi(\gamma|\theta, -) \min \left\{ \frac{1}{\sum_{\gamma \in N_{\text{CD}}(\gamma; v)} \pi(\gamma|\theta, -)}, \frac{1}{\sum_{\gamma \in N_{\text{CD}}(\gamma'; u)} \pi(\gamma'|\theta, -)} \right\}
\]

If \( m \in \{ \text{PD, PS} \} \): Choose component indices \((u, v) \in I_A \) with \( d_u, d_v > 0 \). If \( m = \text{PD} \), then \( \gamma' = \{\ldots, \gamma_u + 1_k, \gamma_v - 1_k, \ldots\} \), where component \( \gamma_v \) donates predictor \( k \) to component \( \gamma_u \); otherwise \( m = \text{PS} \), and for \( j \in \gamma_u \setminus \gamma_v \) and \( k \in \gamma_v \setminus \gamma_u \), \( \gamma' = \{\ldots, \gamma_u - 1_j + 1_k, \gamma_v - 1_k + 1_j, \ldots\} \). In addition, here \( A_m(\gamma; (u, v)) \) =
In all cases, i.e. $m \in \{CD, PD, PS\}$, the expression $\pi(\xi)T(\xi, \xi')$ is symmetric in its arguments. Hence the mixture ICM proposal distribution comprising “cross donate”, “paired donate”, and “paired swap” moves preserves stationarity of the paired-move DMTM sampler.
D. Additional figures

Fig. 8. Marginal predictor importance (inclusion probability) for BART, RF, and Lasso on each simulated test function considered in Section 4.1. Plots generated below use the same data which used to generate Figure 3 ($n = 100, p = 1000$).
Fig. 9. Posterior plots and comparisons for the real datasets considered in Section 5. Left: plot of predictor importance scores, $v_j$, $j = 1, \ldots, p$. Middle: bar-plots of GP scale parameters $\rho, \lambda$ for contributing components (i.e., having 75% quantile over posterior draws for $\rho_l > 0$). Right: trace plot of posterior draws for variance parameter, $\sigma^2$.

| Dataset   | Importance scores $\{v_j, j = 1, \ldots, p\}$ | GP parameters $\{\rho_l, \lambda_l, l \in I_A\}$ | Trace plot $(\sigma^2)$ |
|-----------|-----------------------------------------------|-------------------------------------------------|------------------------|
| Boston    | ![Importance scores](image1.png)              | ![GP parameters](image2.png)                    | ![Trace plot](image3.png) |
| Crime     | ![Importance scores](image4.png)              | ![GP parameters](image5.png)                    | ![Trace plot](image6.png) |
| Riboflavin| ![Importance scores](image7.png)               | ![GP parameters](image8.png)                    | ![Trace plot](image9.png) |
| Cookie    | ![Importance scores](image10.png)             | ![GP parameters](image11.png)                   | ![Trace plot](image12.png) |
Additive Gaussian Process Regression

References

Albert, J. H. and S. Chib (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association* 88(422), 669–679.

Banerjee, A., D. B. Dunson, and S. T. Tokdar (2012). Efficient gaussian process regression for large datasets. *Biometrika*, ass068.

Banerjee, S., A. E. Gelfand, A. O. Finley, and H. Sang (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70(4), 825–848.

Bertin, K. and G. Lecué (2008). Selection of variables and dimension reduction in high-dimensional non-parametric regression. *Electronic Journal of Statistics* 2, 1224–1241.

Bhattacharya, A., D. Pati, and D. B. Dunson (2011). Adaptive dimension reduction with a gaussian process prior. *arXiv preprint arXiv:1111.1044*.

Bickel, P. J. and E. Levina (2008). Regularized estimation of large covariance matrices. *The Annals of Statistics*, 199–227.

Bottolo, L., S. Richardson, et al. (2010). Evolutionary stochastic search for bayesian model exploration. *Bayesian Analysis* 5(3), 583–618.

Breiman, L. (2001). Random forests. *Machine Learning* 45, 5–32.

Candes, E. and T. Tao (2007). The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics* 35, 2313–2351.

Chipman, H. A., E. I. George, and R. E. McCulloch (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics* 4(1), 266–298.

Freund, Y. and R. E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1), 119 – 139.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The annals of statistics*, 1–67.

Friedman, J. H. and W. Stuetzle (1981). Projection pursuit regression. *Journal of the American statistical Association* 76(376), 817–823.

Hamze, F., Z. Wang, and N. de Freitas (2013). Self-avoiding random dynamics on integer complex systems. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 23(1), 9.

Hans, C., A. Dobra, and M. West (2007). Shotgun stochastic search for large p regression. *Journal of the American Statistical Association* 102(478), 507–516.

Harbrecht, H., M. Peters, and R. Schneider (2012). On the low-rank approximation by the pivoted cholesky decomposition. *Applied numerical mathematics* 62(4), 428–440.

Hastie, T., R. Tibshirani, et al. (1986). Generalized additive models. *Statistical science* 1(3), 297–310.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics* 20(1).

Lafferty, J. and L. Wasserman (2008). Rodeo: sparse, greedy nonparametric regression. *The Annals of Statistics*, 28–63.

Liu, J. S., F. Liang, and W. H. Wong (2000). The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association* 95(449), 121–134.
Mansinghka, V., D. Roy, E. Jonas, and J. Tenenbaum (2009). Exact and approximate sampling by systematic stochastic search.

Murray, I., R. Prescott Adams, and D. J. C. MacKay (2010, December). Elliptical slice sampling. ArXiv e-prints.

Neal, R. M. (1997). Monte carlo implementation of gaussian process models for bayesian regression and classification. arXiv preprint physics/9701026.

O’Hagan, A. and J. Kingman (1978). Curve fitting and optimal design for prediction. Journal of the Royal Statistical Society. Series B (Methodological), 1–42.

Ravikumar, P. D., H. Liu, J. D. Lafferty, and L. A. Wasserman (2007). Spam: Sparse additive models. In NIPS.

Ritter, C. and M. A. Tanner (1992). Facilitating the gibbs sampler: the gibbs stopper and the griddy-gibbs sampler. Journal of the American Statistical Association 87(419), 861–868.

Roberts, G. O. and J. S. Rosenthal (2007). Coupling and ergodicity of adaptive markov chain monte carlo algorithms. Journal of applied probability, 458–475.

Schäfer, J. and K. Strimmer (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statistical applications in genetics and molecular biology 4(1).

Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. The Annals of Statistics 10, 1040–1053.

Strens, M. J. (2003). Evolutionary mcmc sampling and optimization in discrete spaces. In ICML, pp. 736–743.

Suzuki, T. (2012). Pac-bayesian bound for gaussian process regression and multiple kernel additive model. In JMLR Workshop and Conference Proceedings, Volume 23, pp. 8–1.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), 267–288.

Tokdar, S. T. (2011a). Adaptive gaussian predictive process approximation. arXiv preprint arXiv:1108.0445.

Tokdar, S. T. (2011b). Dimension adaptability of gaussian process models with variable selection and projection. arXiv preprint arXiv:1112.0716.

van der Vaart, A. W. and J. H. van Zanten (2008, June). Rates of contraction of posterior distributions based on Gaussian process priors. The Annals of Statistics 36(3), 1435–1463.

van der Vaart, A. W. and J. H. van Zanten (2009). Adaptive bayesian estimation using a Gaussian random field with inverse gamma bandwidth. The Annal of Statistics 37, 2655–2675.

Williams, C. K. I. and C. E. Rasmussen (1996). Gaussian processes for regression. In Advances in Neural Information Processing Systems 8, Cambridge, MA, pp. 514–520. The MIT Press.

Yang, Y. and S. T. Tokdar (2014, January). Minimax risks for high dimensional nonparametric regression. ArXiv e-prints.