Transformation and Additivity in Gaussian Process

Li-Hsiang Lin and V. Roshan Joseph
H. Milton Stewart School of Industrial and Systems Engineering,
Georgia Institute of Technology, Atlanta, GA 30332

Abstract

We discuss the problem of approximating a deterministic function using Gaussian Processes (GP). The role of response transformation in GP modeling is not well understood. We argue that transformations can be used for making the deterministic function approximately additive, which can then be easily estimated using an additive GP. We call such a GP a Transformed Additive Gaussian (TAG) process. To capture possible interactions which are unaccounted for in an additive model, we propose an extension of TAG process called Transformed Approximately Additive Gaussian (TAAG) process. We develop efficient techniques for fitting a TAAG process. In fact, we show that it can be fitted to high-dimensional and big data much more efficiently than the usual GP. Furthermore, we show that the use of TAAG process leads to better estimation, interpretation, visualization, and prediction.

Keywords: Additive models; Computer experiments; Correlation function; High-dimensional data; Kriging.

1 INTRODUCTION

Transformation of response is a common technique used in regression analysis, but not so much in the modeling of deterministic functions. There are many reasons for this. In regression analysis, transformations are used as a way to fix the violations in the statistical modeling assumptions such as constant variance or normality of the errors. Since there are no errors in a deterministic function, there does not seem to be any need for transformations! From a function approximation point of view, there also does not seem to be any advantage in transforming the response and therefore, transformations are rarely studied in numerical analysis literature. To see this, suppose we are trying to approximate a function $y = f(\boldsymbol{x})$, $\boldsymbol{x} \in [0,1]^p$, using the data $\boldsymbol{y} = (y_1, \ldots, y_n)'$ observed over an experimental design $\boldsymbol{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. We can obtain the function approximation $\hat{f}(\boldsymbol{x}|\boldsymbol{D}, \boldsymbol{y})$ directly using this data or $g^{-1}\{\widehat{g \circ f}(\boldsymbol{x}|\boldsymbol{D}, g(\boldsymbol{y}))\}$ using the transformed data, where $g(\cdot)$ denotes the transformation function, $g(\boldsymbol{y}) = (g(y_1), \ldots, g(y_n))'$, and $g \circ f(\cdot) = g\{f(\cdot)\}$. Although these two approximations can be quite different, they are asymptotically equivalent as long as the technique used for function approximation converges (see, for example, Fasshauer (2007) for the conditions on convergence). Since the quality of a function approximation is assessed using its asymptotic convergence properties, the transformation does not seem to play any role in the mathematical analysis and therefore, it is ignored. Yet practitioners have often found it useful to transform the response, but its usage seems to be sporadic with no proper guidelines. For example, a logarithmic transformation is used for making the predictions nonnegative, but many times at the cost of accuracy.

Gaussian process (GP) models, also known as kriging, are widely adopted for modeling deterministic functions (Sacks et al. (1989), Santner et al. (2013)). Because of its probabilistic formulation, a case can be made for transforming the output. This approach is known by the name Trans-Gaussian kriging in spatial statistics (Cressie (1992), De Oliveira et al. (1997)) and warped Gaussian process in machine learning (Snelson et al. (2004), Lázaro-Gredilla (2012)). However, GP is used in modeling mainly due to its mathematical convenience and does not posses a strong justification as in the case of regression analysis. Thus, transforming the response to make its distribution looks more like Gaussian does

look questionable. Stationarity is another common assumption for GP modeling. However, since we observe only a single realization of the stochastic process, assessing the validity of this assumption and achieving constancy of variance is not straightforward.

We propose transformation in GP modeling to improve additivity, that is, to find a transformation so that the deterministic function becomes approximately additive in the variables. An additive function is easier to approximate and therefore the approximation obtained using such a transformation is expected to perform better. To illustrate the idea, consider the functions $f(\boldsymbol{x}) = 1/(x_1^2 + x_2^2)$ and $f(\boldsymbol{x}) = \exp(x_1 + x_2 + .01x_1x_2)$. By setting $g(y) = 1/y$, the first function becomes perfectly additive in the two variables, whereas $g(y) = \log y$ makes the second function approximately additive, both can be well-approximated using fewer data points than what would be needed in the original scale.

Additive models is not a new concept and has a long history in statistics (see, for example, Hastie and Tibshirani (1990)). An obvious disadvantage of additive models is that they cannot entertain higher-order interactions among the variables. Friedman and Stuetzle (1981) extended the additive modeling framework to include linear combinations of the variables, which has the ability to capture interactions. Different from previous works, we employ a GP model as the nonparametric smoother in the additive modeling framework. In this sense our approach is closer to the additive GP models introduced by Duvenaud et al. (2011), but there are major differences. Their objective was to decompose the function into a sum of low-dimensional functions that include interactions, whereas our objective is to identify a transformation so that the function can be represented by a first-order low-dimensional function. The idea of transformation is also not new in additive models. Tibshirani (1988) proposed additivity and variance stabilization (AVAS) algorithm in conjunction with additive models, but variance stabilization is not relevant to our problem because there is no error in deterministic computer experiments. Our approach is similar in spirit to the Alternating Conditional Expectation (ACE) method of Breiman and Friedman (1985), but differs in terms of the smoothing method used for the variables. Moreover, as we demonstrate in this paper, the use of GP models facilitate better uncertainty quantification of deterministic functions.

The article is organized as follows. In Section 2, we develop the main methodology for identifying transformations to make the function as additive as possible. Efficient estimation techniques for the unknown parameters in the model are developed in this section. In general, the additive model can only provide an approximation, whereas interpolation is desired in deterministic computer experiments. In Section 3, we introduce approximately additive GP models which can achieve interpolation. Some examples are provided in Section 4 to illustrate the advantages of the proposed methodology. We conclude with some remarks in Section 5.

2 TRANSFORMED ADDITIVE GAUSSIAN PROCESS

Our aim is to find a transformation for the response $g(y)$ so that the inverse transformed additive model

$$y = g^{-1}\{\mu + z_1(x_1) + \ldots + z_p(x_p)\} \tag{1}$$

is a good approximation to $y = f(\boldsymbol{x})$, where $\boldsymbol{x} = (x_1, \ldots, x_p)'$. We assume each function $z_k(\cdot)$ to follow a stationary GP:

$$z_k(x_k) \sim GP(0, \tau_k^2 R_k(\cdot)),$$

for $k = 1, \ldots, p$, where τ_k^2 is the variance and $R_k(h) = Cor\{f_k(x), f_k(x+h)\}$ is the stationary correlation function. Let $\tau^2 = \sum_{k=1}^p \tau_k^2$ and $\omega_k = \tau_k^2/\tau^2$. Then,

$$g(y) \sim GP(\mu, \tau^2 R(\cdot)), \tag{2}$$

where

$$R(\boldsymbol{h}) = \sum_{k=1}^{p} \omega_k R_k(h_k) \tag{3}$$

with $\sum_{k=1}^p \omega_k = 1$. We call this model as *Transformed Additive Gaussian (TAG) process*. The weights ω_i's can be interpreted as the first-order Sobol indices of $g\{f(\boldsymbol{x})\}$, provided it is an additive function (Sobol', 1990).

Let $z(\boldsymbol{x}) = g\{f(\boldsymbol{x})\}$. Given the data $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)$, we can obtain the posterior mean of $g(y)$ as

$$\widehat{z}(\boldsymbol{x}) = \mu + \boldsymbol{r}(\boldsymbol{x})'(\boldsymbol{R} + \delta \boldsymbol{I})^{-1}(g(\boldsymbol{y}) - \mu \boldsymbol{1}), \tag{4}$$

where $\boldsymbol{r}(\boldsymbol{x})$ is the vector of correlations $(R(\boldsymbol{x} - \boldsymbol{x}_1), \ldots, R(\boldsymbol{x} - \boldsymbol{x}_n))'$, $\boldsymbol{R}$ is the correlation matrix with the ijth element $R(\boldsymbol{x}_i - \boldsymbol{x}_j)$, $\boldsymbol{1}$ is a vector of 1's, and $g(\boldsymbol{y}) = (g(y_1), \ldots, g(y_n))'$. We have intentionally added a nugget term $\delta > 0$ because $\boldsymbol{R}$ is guaranteed to be only semi-positive definite even if we use positive definite correlation functions $R_k(\cdot)$. This is expected because the function we are trying to approximate need not be additive and thus, we cannot interpolate the observed data using the additive correlation function in (3).

Let

$$\hat{\boldsymbol{c}} = (\boldsymbol{R} + \delta \boldsymbol{I})^{-1}(g(\boldsymbol{y}) - \mu \boldsymbol{1}). \tag{5}$$

Then, (4) can also be written as the sum of n basis functions:

$$\widehat{z}(\boldsymbol{x}) = \mu + \sum_{i=1}^{n} \hat{c}_i R(\boldsymbol{x} - \boldsymbol{x}_i). \tag{6}$$

This basis function approximation view point of (4) is crucial for the estimation technique that we devise below. It is easy to see that (5) is the solution to the optimization problem

$$\arg\min_{\boldsymbol{c}} \{g(\boldsymbol{y}) - \mu \boldsymbol{1} - \boldsymbol{R}\boldsymbol{c}\}'\{g(\boldsymbol{y}) - \mu \boldsymbol{1} - \boldsymbol{R}\boldsymbol{c}\} + \delta \boldsymbol{c}'\boldsymbol{R}\boldsymbol{c}.$$

Thus, (6) can be viewed as the posterior mean of fitting the normal linear model

$$g(y) = \mu + \sum_{i=1}^{n} c_i R(\boldsymbol{x} - \boldsymbol{x}_i) + \epsilon,\ \epsilon \sim N(0, \sigma^2) \tag{7}$$

using the prior $\boldsymbol{c} \sim N(\boldsymbol{0}, \tau^2 \boldsymbol{R}^{-1})$, where $\delta = \sigma^2/\tau^2$.

Now substituting (3) into (7), we obtain

$$\begin{aligned} g(y) &= \mu + \sum_{i=1}^{n} c_i \sum_{k=1}^{p} \omega_k R_k(x_k - x_{ik}) + \epsilon \\ &= \mu + \sum_{k=1}^{p} \omega_k \tilde{z}_k(x_k) + \epsilon, \end{aligned} \tag{8}$$

where

$$\tilde{z}_k(x_k) = \sum_{i=1}^{n} c_i R_k(x_k - x_{ik}).$$

Note that $\tilde{z}_k(x_k)$ is just a scaled version of $z_k(x_k)$ in (1). Thus, given $\boldsymbol{c}$, we can estimate $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)'$ by fitting the normal linear model in (8) under the constraints $\boldsymbol{\omega}'\mathbf{1} = 1$ and $\boldsymbol{\omega} \geq \mathbf{0}$. This can be done by solving the quadratic program:

$$\min_{\boldsymbol{\omega} \in \boldsymbol{\Omega}} \{g(\boldsymbol{y}) - \mu\mathbf{1} - \tilde{\boldsymbol{Z}}\boldsymbol{\omega}\}'\{g(\boldsymbol{y}) - \mu\mathbf{1} - \tilde{\boldsymbol{Z}}\boldsymbol{\omega}\} \tag{9}$$

where $\tilde{\boldsymbol{Z}}$ is an $n \times p$ matrix with kth column as $\boldsymbol{R}_k\boldsymbol{c}$, where $\boldsymbol{R}_k$ is the correlation matrix corresponding to the kth variable and $\boldsymbol{\Omega} = \{\boldsymbol{\omega} : \boldsymbol{\omega}'\mathbf{1} = 1,\ \boldsymbol{\omega} \geq \mathbf{0}\}$. This quadratic program can be solved very efficiently and is the beauty of our procedure. The foregoing developments suggest an iterative estimation of $\boldsymbol{\omega}$ and $\boldsymbol{c}$, that is, given $\boldsymbol{\omega}$, estimate $\boldsymbol{c}$ using (5) and given $\boldsymbol{c}$, estimate $\boldsymbol{\omega}$ using (9). The proof of the convergence of such algorithm is given in Appendix A.

There are many unknown parameters to estimate such as μ, τ^2, and δ. Moreover, the correlation function can have unknown parameters. A commonly used correlation function in computer experiments is the Gaussian correlation function given by

$$R_k(h) = \exp(-h^2/s_k^2),$$

where s_k is an unknown length-scale parameter. Thus, we also need to estimate $\boldsymbol{s} = (s_1, \ldots, s_p)'$. Furthermore, we also need to estimate the transformation function $g(\cdot)$. Here we use a parametric approach. A commonly used parametric transformation for nonnegative data ($y > 0$) is the Box-Cox transformation (Box and Cox (1964)) given by

$$g_\lambda(y) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, \text{if } \lambda \neq 0 \\ \log y, \text{if } \lambda = 0 \end{cases}. \tag{10}$$

This transformation contains an unknown parameter λ. A two-parameter Box-Cox model can be used if the data is not restricted to be nonnegative. In this paper we will focus on the foregoing one-parameter transformation, but the methods that we propose below are general and can be applied to more general cases. We now discuss the estimation of all these unknown parameters: $\mu, \tau^2, \sigma^2, \boldsymbol{s}$, and λ.

Since our aim is to find a transformation to make the function as additive as possible, it makes sense to estimate λ from (8). The likelihood is proportional to

$$\frac{1}{\sigma^n} \exp\{-(g_\lambda(\boldsymbol{y}) - \mu\mathbf{1} - \tilde{\boldsymbol{Z}}\hat{\boldsymbol{\omega}}(\lambda))'(g_\lambda(\boldsymbol{y}) - \mu\mathbf{1} - \tilde{\boldsymbol{Z}}\hat{\boldsymbol{\omega}}(\lambda))/(2\sigma^2)\} \prod_{i=1}^{n} y_i^{\lambda-1}, \tag{11}$$

where the last term is due to the Jacobian of transformations. We have used $\hat{\boldsymbol{\omega}}(\lambda)$ to explicitly show its dependence on λ. Maximizing (11) with respect to σ^2 gives

$$\hat{\sigma}^2(\lambda) = \frac{1}{n}\{g_\lambda(\boldsymbol{y}) - \mu\mathbf{1} - \tilde{\boldsymbol{Z}}\hat{\boldsymbol{\omega}}(\lambda)\}'\{g_\lambda(\boldsymbol{y}) - \mu\mathbf{1} - \tilde{\boldsymbol{Z}}\hat{\boldsymbol{\omega}}(\lambda)\}. \tag{12}$$

Substituting this in (11) gives the profile likelihood for λ. By maximizing the profile likelihood, we obtain

$$\hat{\lambda} = \arg\min_{\lambda} \mathrm{PL}(\lambda) \equiv n \log \hat{\sigma}(\lambda) - (\lambda - 1)\sum_{i=1}^{n} \log y_i. \tag{13}$$

Since $g_\lambda(\boldsymbol{y})|\boldsymbol{c} \sim N(\mu\mathbf{1} + \boldsymbol{R}\boldsymbol{c}, \sigma^2\boldsymbol{I})$ and $\boldsymbol{c} \sim N(\mathbf{0}, \tau^2\boldsymbol{R}^{-1})$, we can easily integrate out $\boldsymbol{c}$ to obtain

$$g(\boldsymbol{y}) \sim N(\mu\mathbf{1}, \tau^2\boldsymbol{R} + \sigma^2\boldsymbol{I}).$$

This gives empirical Bayes estimates of μ, τ^2, $\delta = \sigma^2/\tau^2$, and $\boldsymbol{s}$ as

$$\hat{\mu} = \frac{\mathbf{1}'(\boldsymbol{R} + \delta\boldsymbol{I})^{-1} g_\lambda(\boldsymbol{y})}{\mathbf{1}'(\boldsymbol{R} + \delta\boldsymbol{I})^{-1}\mathbf{1}}, \tag{14}$$

$$\hat{\tau}^2 = \frac{1}{n}(g_\lambda(\boldsymbol{y}) - \hat{\mu}\mathbf{1})'(\boldsymbol{R} + \delta\boldsymbol{I})^{-1}(g_\lambda(\boldsymbol{y}) - \hat{\mu}\mathbf{1}),$$

$$(\hat{\delta}, \hat{\boldsymbol{s}}) = \arg\min_{\delta, \boldsymbol{s}} \mathrm{EB}(\delta, \mathbf{s}) \equiv \log|\boldsymbol{R} + \delta\boldsymbol{I}| + n \log \hat{\tau}^2. \tag{15}$$

Although motivated differently, all these estimates agree with the estimates that one would obtain by fitting the GP model in (2).

The whole estimation procedure is shown as Algorithm 1. The procedure may look more complicated than the usual estimation in GP. But note that we are fitting a correlation function involving $2p$ correlation parameters ($\boldsymbol{\omega}$ and $\boldsymbol{s}$), which is twice as that in the usual GP modeling with say, using a product Gaussian correlation function. The extra parameters ($\boldsymbol{\omega}$) in our correlation function makes our modeling more flexible as we discuss later. The most attractive feature of the procedure is that these extra parameters can be obtained without much additional computational cost, thanks to the quadratic program.

Because of the possibility of multiple local optima, it is important to choose a good initialization of the parameters in the algorithm. There are efficient implementations available for fitting an additive model using the backfitting algorithm (Breiman and Friedman, 1985). So we start by fitting an additive model using the *mgcv* package (Wood (2017)) in

R and then estimate all the parameters using the fitted additive model. The details are described in the Appendix B.

Algorithm 1 Estimation of Transformed Additive Gaussian (TAG) process

procedure TAG($\mathbf{y}, \mathbf{D}, \epsilon > 0$) ▷

  Obtain initial estimates $\lambda^{(0)}, \boldsymbol{\omega}^{(0)}, \boldsymbol{s}^{(0)}, \delta^{(0)}$ using Algorithm 3.

  Set $t = 0$ and $\Lambda = \{\lambda^{(0)} - 0.5, \lambda^{(0)}, \lambda^{(0)} + 0.5\}$.

  while $\max |\boldsymbol{\omega}^{(t+1)} - \boldsymbol{\omega}^{(t)}| > \epsilon$ **do**

    $t \leftarrow t + 1$

    For each $\lambda \in \Lambda$, obtain $\hat{\mu}$ from (14), $\hat{\mathbf{c}}$ from (5), and $\hat{\boldsymbol{\omega}}(\lambda)$ from (9).

    $\lambda^{(t)} \leftarrow \arg\min_{\lambda \in \Lambda} \mathrm{PL}(\lambda)$, where $\mathrm{PL}(\lambda)$ is from (13) and $\hat{\sigma}(\lambda)$ is from (12).

    $\boldsymbol{\omega}^{(t)} \leftarrow \hat{\boldsymbol{\omega}}(\lambda^{(t)})$

    Update $(\delta^{(t)}, \mathbf{s}^{(t)})$ using (15).

  end while

  return $(\hat{\boldsymbol{\omega}}, \hat{\mathbf{s}}, \hat{\delta}, \hat{\lambda}) = (\boldsymbol{\omega}^{(t)}, \mathbf{s}^{(t)}, \delta^{(t)}, \lambda^{(t)})$.

end procedure

3 TRANSFORMED APPROXIMATELY ADDITIVE GAUSSIAN PROCESS

Even with the best possible transformation, we may not be able to make the function additive and thus the approximation that we obtain using TAG can be unsatisfactory. In this section, we propose a simple extension of TAG to improve the approximation.

The main limitation of the additive correlation function in (3) is that it is not positive definite and thus cannot be used for interpolating arbitrary functions. But we can make it positive definite by adding a positive definite correlation function $L(\boldsymbol{h})$ to $R(\boldsymbol{h})$. Thus the GP model becomes

$$g(y) \sim GP(\mu, \tau^2\{(1-\eta)R(\cdot) + \eta L(\cdot)\}), \tag{16}$$

where $\eta \in [0, 1]$ and $R(\cdot)$ is as in (3). The resulting predictor is only approximately additive.

Therefore we call this model as *Transformed Approximately Additive Gaussian (TAAG) Process*. Plate (1999) proposed a closely related GP model, but the motivation behind TAAG process and its estimation techniques are completely different. TAAG process is also related to some of the other ideas proposed in the literature such as that of using a convex combination of GPs (Harari and Steinberg (2014)) and composite GPs (Ba and Joseph (2012)).

Since the additive part is expected to capture most of the functional characteristics of the output, we may choose $L(\boldsymbol{h})$ based on the $R_k(\cdot)$, $k = 1, \ldots, p$. So we let

$$L(\boldsymbol{h}) = \prod_{k=1}^{p} R_k\left(h_k\right).$$

A common choice for $R_k(\cdot)$ is the Gaussian correlation function. Then $L(\boldsymbol{h})$ becomes

$$L(\boldsymbol{h}) = \exp\left(-\sum_{k=1}^{p} \frac{h_k^2}{\theta_k^2}\right)$$

with length-scale parameters $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_p)$, where θ_k is for the k-th kernel.

The unknown parameters in the new GP model $(\mu, \tau^2, \lambda, \boldsymbol{\omega}, \boldsymbol{s}, \boldsymbol{\theta}, \eta, \phi)$ can be estimated using maximum likelihood or cross validation methods. But since λ, $\boldsymbol{\omega}$, and $\boldsymbol{s}$ should be chosen so that the function becomes as additive as possible, it makes sense to fix those parameters at the estimates that we obtained earlier using the additive GP. The length scale parameter $\boldsymbol{\theta}$ of $L(\cdot)$ can also be estimated conviniently from a standard GP. Therefore, we only estimate the new parameter η along with μ and τ^2. Their estimates can be obtained as Santner et al. (2013) with a prior on η following a beta distribution with parameters $\hat{\delta} + 1$ and 2.

$$\hat{\eta} = \arg\min_{\eta} \mathrm{M}(\eta) \equiv \log\left|(1-\eta)\boldsymbol{R} + \eta\boldsymbol{L}\right| + n\log\hat{\tau}^2 + 2\log(\eta^{\hat{\delta}}(1-\eta)), \tag{17}$$

where

$$\begin{aligned}
\hat{\mu} &= \frac{\mathbf{1}'\{(1-\eta)\boldsymbol{R} + \eta\boldsymbol{L}\}^{-1} g(\boldsymbol{y})}{\mathbf{1}'\{(1-\eta)\boldsymbol{R} + \eta\boldsymbol{L}\}^{-1}\mathbf{1}}, \\
\hat{\tau}^2 &= \frac{1}{n}(g(\boldsymbol{y}) - \hat{\mu}\mathbf{1})'\{(1-\eta)\boldsymbol{R} + \eta\boldsymbol{L}\}^{-1}(g(\boldsymbol{y}) - \hat{\mu}\mathbf{1}),
\end{aligned}$$

and $\boldsymbol{L}$ is the $n \times n$ matrix with ijth element $L(\boldsymbol{x}_i - \boldsymbol{x}_j)$. The estimation procedure is shown in Algorithm 2.

Algorithm 2 Estimation of Transformed Approximated Additive Gaussian (TAAG) Process

1: **procedure** TAAG($\mathbf{y}, \mathbf{D}$) ▷
2: Obtain $\hat{\delta}, \hat{\lambda}, \hat{\boldsymbol{\omega}}$, and $\hat{\boldsymbol{s}}$ using Algorithm 1 and $\hat{\boldsymbol{\theta}}$ from a standard GP model.
3: $\hat{\eta} \leftarrow \arg\min_{\eta} \mathrm{M}(\eta)$, using an initial value $\eta^{(0)} = \hat{\delta}/(1+\hat{\delta})$, where $\mathrm{M}(\eta)$ is from (17).
4: **return** $\hat{\eta}$
5: **end procedure**

The prediction and uncertainty quantification can be done as follows. The posterior distribution of $z(\boldsymbol{x})$ for given μ and τ^2 is given by

$$z(\boldsymbol{x})|\boldsymbol{y}, \mu, \tau^2 \sim N(\widehat{z}(\boldsymbol{x}), V(\boldsymbol{x})), \tag{18}$$

where

$$\begin{aligned} \widehat{z}(\boldsymbol{x}) &= \mu + \{(1-\eta)\boldsymbol{r}(\boldsymbol{x}) + \eta\boldsymbol{l}(\boldsymbol{x})\}'\{(1-\eta)\boldsymbol{R} + \eta\boldsymbol{L}\}^{-1}(g(\boldsymbol{y}) - \mu\mathbf{1}) \\ V(\boldsymbol{x}) &= \tau^2\left[1 - \{(1-\eta)\boldsymbol{r}(\boldsymbol{x}) + \eta\boldsymbol{l}(\boldsymbol{x})\}'\{(1-\eta)\boldsymbol{R} + \eta\boldsymbol{L}\}^{-1}\{(1-\eta)\boldsymbol{r}(\boldsymbol{x}) + \eta\boldsymbol{l}(\boldsymbol{x})\}\right]. \end{aligned}$$

We can either plug-in the estimates of μ and τ^2 or integrate them out (Santner et al. (2013)), but for simplicity we will use the plug-in approach. From (18), we can obtain the probability density function of $f(\boldsymbol{x})|\boldsymbol{y}, \mu, \tau^2$ as

$$|\dot{g}(f(\boldsymbol{x}))|\frac{1}{\sqrt{2\pi V(\boldsymbol{x})}}\exp\{-(z(\boldsymbol{x}) - \widehat{z}(\boldsymbol{x}))^2/(2V(\boldsymbol{x}))\},$$

where $\dot{g}(\cdot)$ is the derivative of $g(\cdot)$. In general, this can be a nonstandard distribution and computing its mean and variance may require numerical integration. Cressie (1992) derives an approximate expression for the mean using Taylor series expansion. But as Snelson et al. (2004) pointed out, it is much easier to use the median, which is given by

$$\hat{f}(\boldsymbol{x}) = g^{-1}\left\{\widehat{z}(\boldsymbol{x})\right\}.$$

Similarly, we can obtain a 95% credible interval for the prediction as

$$\left[g^{-1}\left\{\widehat{z}(\boldsymbol{x}) - 2\sqrt{V(\boldsymbol{x})}\right\}, g^{-1}\left\{\widehat{z}(\boldsymbol{x}) + 2\sqrt{V(\boldsymbol{x})}\right\}\right].$$

Note that when using a Box-Cox transformation (10), we need constraints $y > 0$ and $g_\lambda(y) > -1/\lambda$ to make sure that $g(\cdot)$ is one-to-one. Therefore we force the lower bound to be 0 if $\widehat{z}(\boldsymbol{x}) - 2\sqrt{V(\boldsymbol{x})} \leq -1/\lambda$.

4 SOME ADVANTAGES OF TAAG

In this section we discuss the many advantages of TAAG using numerical examples.

4.1 New Correlation Function

Although our aim was not to develop a new correlation function, the one that came out of our modeling

$$(1-\eta)\sum_{k=1}^{p}\omega_k R_k(h_k;s_k)+\eta\prod_{k=1}^{p}R_k(h_k;\theta_k)$$

is of independent interest. It has certain properties not possessed by any of the existing correlation functions in the literature. To illustrate its advantages, consider a simple function

$$y=\exp\{2\sin(0.5\pi x_1)+0.5\cos(2.5\pi x_2)\} \tag{19}$$

where $\boldsymbol{x}\in[0,1]^2$. The marginal plots of the function are shown in Figure 1.

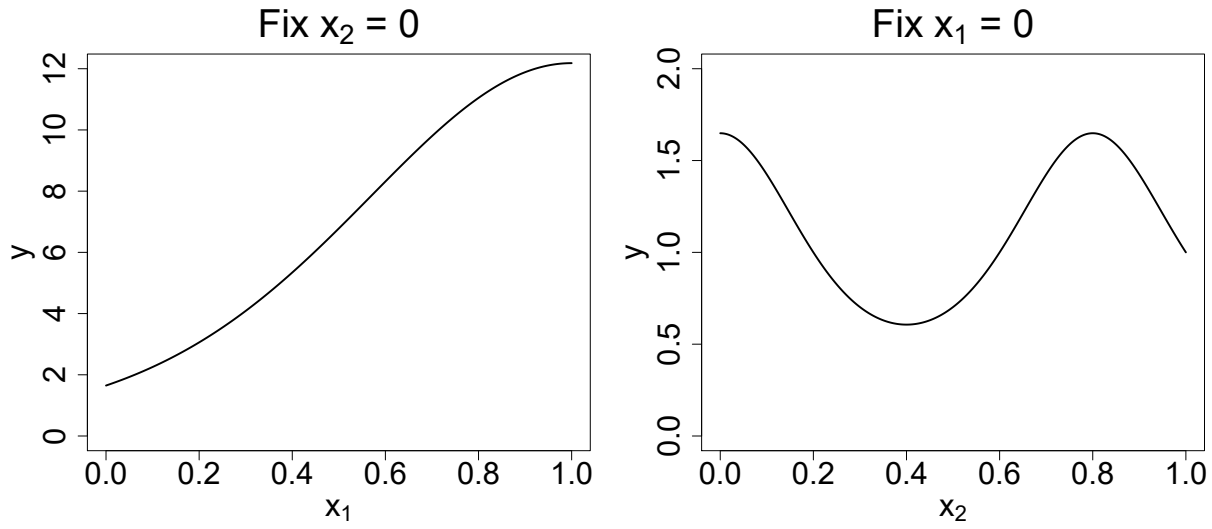

Figure 1: Marginal plots of the function in (19).

Suppose we generate data using a randomized Sobol' sequence (Christophe and Petr (2018)) of $n = 20$ points. First we fit a GP using the commonly used Gaussian product correlation function $R(\boldsymbol{h}) = \exp(-\sum_{k=1}^{2} h_k^2/s_k^2)$. We used a standard R package *mlegp*

(Dancik and Dorman (2008)) for obtaining the maximum likelihood estimates of the parameters. The left panel of Figure 2 shows boxplots of s_1 and s_2 from 100 repetitions of the randomized Sobol' sequence. The length scale parameters have long been used for identifying important variables (Neal (2012), Williams and Rasmussen (1996), Linkletter et al. (2006)). Smaller s_i implies the variable x_i is more important to the output. As s_i increases, the importance of the variable decreases and in the limit $s_i \to \infty$, the variable gets eliminated from the model. Thus, Figure 2 suggests that x_2 is more important than x_1. This is a complete contradiction to the reality as the marginal plot in Figure 1 shows x_1 is at least four times more important than x_2! The function is less wiggly in x_1 than in x_2 and therefore, s_1 is estimated to be larger than s_2. But, clearly, this does not capture the importance of these two variables.

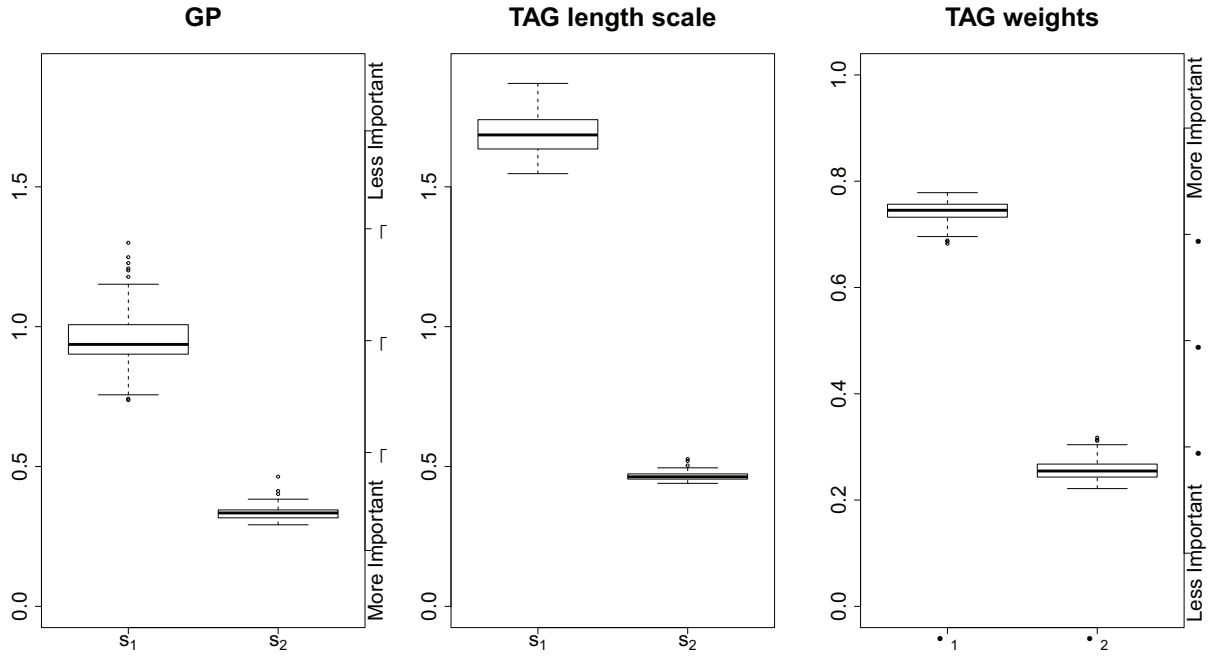

Figure 2: Parameter estimates for GP and TAG for the function in (19).

Now consider the new correlation function in (16). The middle panel shows the boxplots of s_1 and s_2 from the same 100 randomized Sobol' sequences estimated using Algorithm 1. The right panel shows the boxplots of ω_1 and ω_2. In the new correlation function s_i's

can be used to understand how wiggly the function is and ω_i's can be used to understand the importance of the variables. Since s_1 is larger than s_2, the function is expected to be less wiggly in x_1 than x_2, which agrees with Figure 1. Moreover, since ω_1 is more than ω_2, TAG process correctly identifies x_1 to be more important than x_2.

There are other correlation functions proposed in the literature with more parameters such as the power exponential and Matérn correlation functions. But the extra parameters in them only controls the smoothness or roughness of the function. The function considered in this example is very smooth and infinitely differentiable in both the variables and therefore, those correlation functions cannot rectify the confounding issues between scale and importance.

4.2 Prediction Performance

TAAG is expected to perform well in the example function in (19) becuase it becomes perfectly additive under log-transformation. So consider a slightly modified version

$$\exp\{2\sin(0.5\pi x_1) + 0.5\cos(2.5\pi x_2)\} + 0.25\sin(\pi x_1)\cos(0.5\pi x_2), \tag{20}$$

which cannot be made additive through transformation. We will use this function to assess the prediction performance of TAAG process.

As before, we generate data using a randomized Sobol' sequence of $n = 20$ points and fit TAG and TAAG processes. Predictions are made on 1,000 test points in $[0, 1]^2$ and the root mean squared prediction error (RMSPE) is computed. This is repeated 100 times by generating a new randomized Sobol' sequence each time. The resulting RMSPEs are shown as boxplots on the left side of Figure 3. We also fitted the commonly used GP with product Gaussian correlation function on the original data as well as the transformed data. Their RMSPEs are also shown in the same figure denoted as "GP" and "Transformed GP", respectively. As a further check, we also fitted an additive model on the original data and the transformed data ("AM" and "Transformed AM") using the R package *mgcv*. We can see that AM does not perform well, but surprisingly the transformed AM does well, even better than GP. This clearly shows the benefit of transformations. On the other hand,

TAG improves over the Transformed AM and Transformed GP. TAAG performs better than TAG and seems to be the best among the six methods.

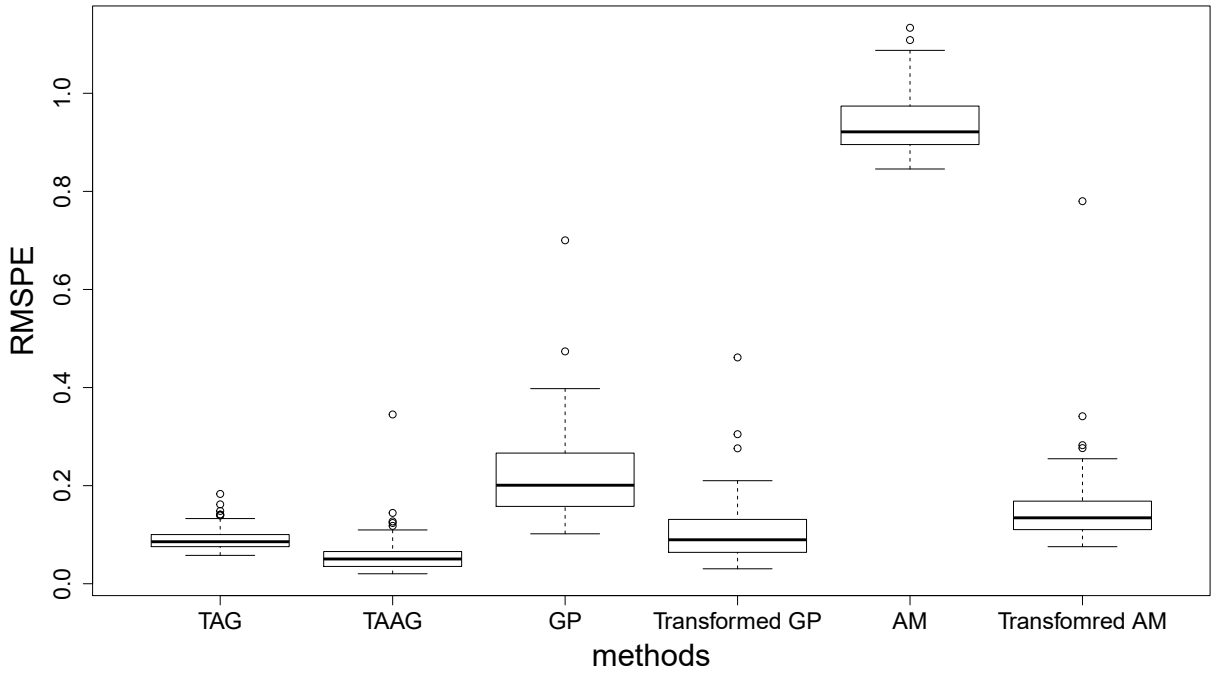

Figure 3: Prediction performance of TAG, TAAG, GP, transformed GP, AM, and transformed AM using the example function in (20), where the simulation is performed by randomizing the design.

As mentioned before, uncertainty quantification is one of the main advantages of the TAG/TAAG processes. To assess their performance, we computed the interval score (Gneiting and Raftery (2007)), which is defined as $(u-\ell)+(2/\alpha)(\ell-x)I\{x<\ell\}+(2/\alpha)(x-u)I\{x>u\}$ with $\alpha=95\%$. A smaller interval score indicates a better prediction interval. The interval scores for the 100 simulation cases are shown as boxplots in Figure 4. Clearly, TAAG is again the best among the six methods.

4.3 Interpretation and Visualization

Another advantage of the TAAG process is that it enables better interpretation and visualization of the effects. The weights, ω_i's, can be used to quickly understand the importance of each variable, which represent the first-order Sobol' indices when the transformed

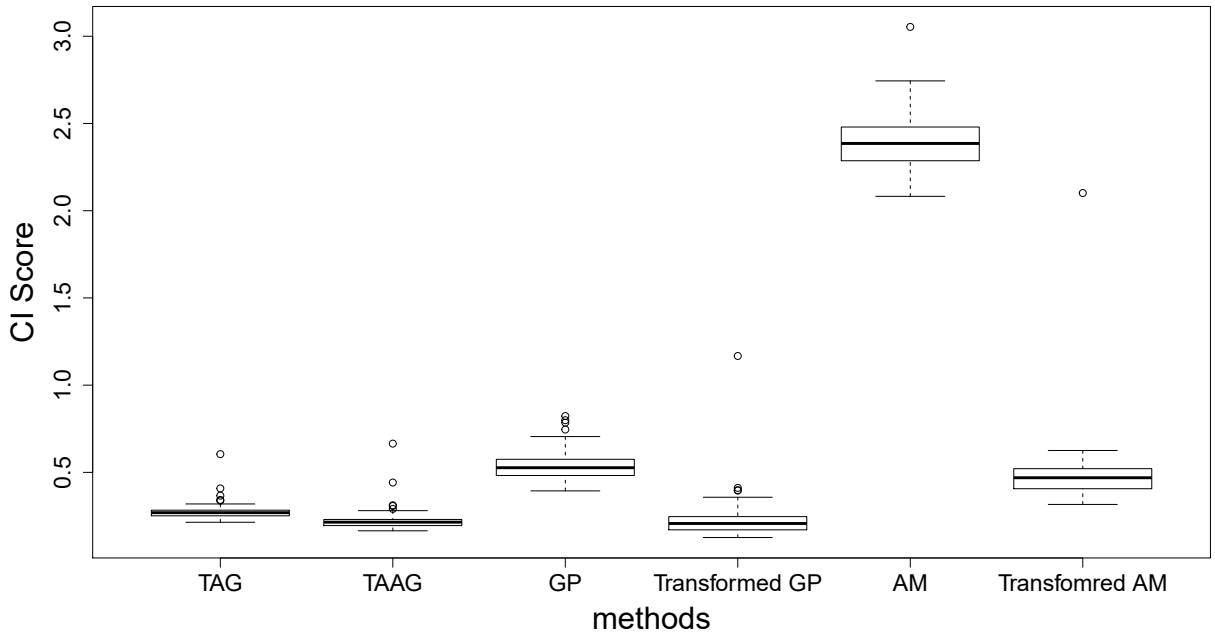

Figure 4: Interval scores (smaller-the-better) of TAG, TAAG, GP, transformed GP, AM, and transformed AM using the example function in (20), where the simulation is performed by randomizing the design.

response is perfectly additive (Sobol', 1990). If $f(\boldsymbol{x})$ is not additive, η will be greater than 0 and its value can be used to understand the overall interaction effect. Moreover, the main effects of the variables in the transformed scale can be quickly visualized using

$$\hat{z}_k(x_k) = \omega_k \sum_{i=1}^{n} \hat{c}_i R_k(x_k - x_{ik}),$$

which does not require any extra computations. On the other hand, one needs to use the computationally intensive functional ANOVA decomposition to get the main effects if we were to fit the usual GP. Of course, the main effects are meaningful only if there are no higher order interactions. Because we use transformation to minimize the interaction effects, the main effects that we obtain using TAAG process are more trustworthy.

We illustrate the foregoing advantages using the borehole function (Morris et al. (1993)):

$$y = \frac{2\pi T_u (H_u - H_l)}{\log(\frac{r}{r_w}) \left[1 + \frac{2LT_u}{\log(\frac{r}{r_w}) r_w^2 K_w} + \frac{T_u}{T_l}\right]},$$

where the ranges for the eight variables are $r_w : (0.05, 0.15)$, $r = (100, 50000)$, $T_u = (63070, 115600)$, $H_u = (990, 1110)$, $T_l = (63.1, 116)$, $H_l = (700, 820)$, $L = (1120, 1680)$, and $K_w = (9855, 12045)$. Suppose we generate $n = 80$ data using the MaxPro design (Joseph et al. (2015)) and fit the TAAG process. It identified a log-transform for the response ($\hat{\lambda} = 0$). The ω_i's and s_i's from the fit are given in Table 1. The first-order Sobol' indices of the log-borehole function is also given in the same table. We can see that ω_i's are very close to the first-order Sobol' indices. This is not a coincidence and happened here because the transformed response is approximately additive.

Input variables	r_w	r	T_u	H_u	T_l	H_l	L	K_w
$\boldsymbol{\omega}$	0.882	0.003	0.002	0.035	0.001	0.037	0.032	0.009
$\boldsymbol{\theta}$	1.502	36.233	1.672	2.878	63.946	2.902	1.795	3.454
First-order Sobol' indices	0.889	0.000	0.000	0.036	0.000	0.035	0.032	0.008

Table 1: The ω_i's and s_i's from the TAAG process for the Borehole function and the first-order Sobol' indices of the log-borehole function

The centered main effects $\hat{\mathbf{z}}_k(x_k) - \bar{z}_k$, where $\bar{z}_k$ is the mean value of $\hat{\mathbf{z}}_k(\cdot)$ are shown in the left panel of Figure 5. The right panel of the figure shows the main effects computed using the borehole function without using any transformation. We can see that the TAAG process approximates the main effects quite well. Moreover, $\hat{\eta} = 0.00158$ is very small showing that the interaction effects are negligibly small in the transformed scale. We can use the difference of Sobol's total index and first-order index to understand the interaction effects. This is shown in Figure 6 for the original and transformed responses. We can see that the log-transformation has greatly helped in reducing the interaction effects. This clearly shows that the main effects plots of $\log y$ are much more meaningful to look at than those of y.

4.4 Big and High-Dimensional Data

Fitting GP models to big and high-dimensional data is always a challenging problem. This is because the likelihood function requires the inversion of the correlation matrix,

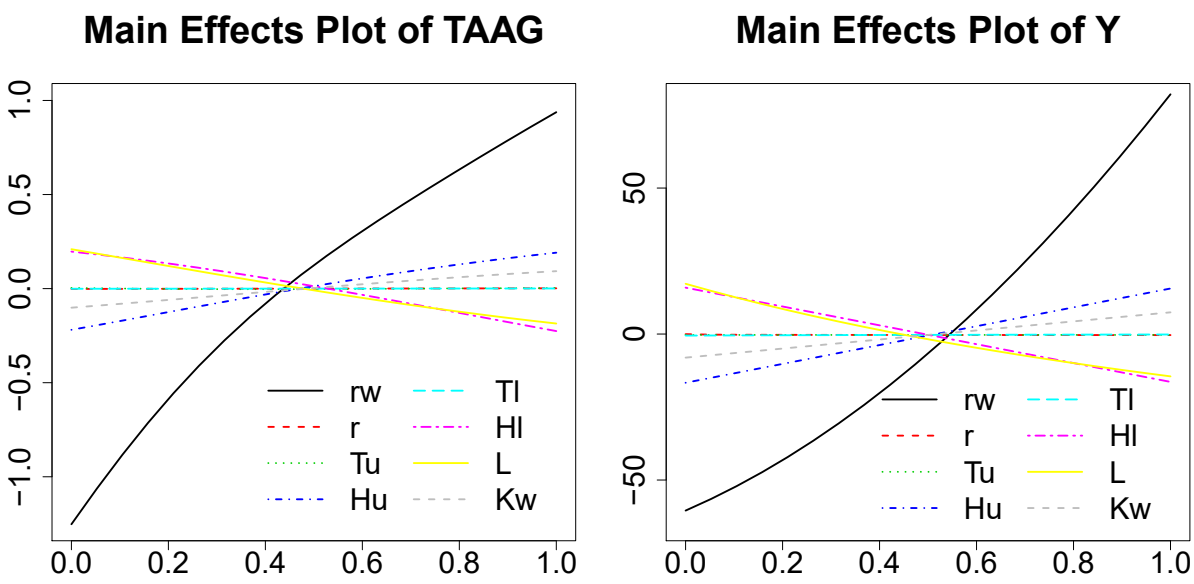

Figure 5: The main effects from TAAG with logarithmic transformation of the response and the true main effects for the original (untransformed) response in the borehole function.

whose computational complexity is $O(n^3)$. Moreover, thousands of evaluations of the likelihood function is needed to optimize it, especially in high dimensions. To understand the computational complexity with respect to the number of dimensions, first note that $O(n^2p)$ computations are needed to construct the correlation matrix. Consider a gradient-based optimization with a fixed number of iterations. Once the correlation matrix is inverted, the gradient of the likelihood can be calculated in $O(n^2p)$ (Williams and Rasmussen (2006), pp.114). So the total computational cost is still $O(n^3 + n^2p)$. However, because the likelihood is likely to be multimodal, the number of initial points for optimization should be at least $O(p)$ to have a fair chance of finding the global optimum (MacDonald et al. (2015)). Thus the computational complexity of optimizing the likelihood is at least $O(n^3p + n^2p^2)$. Since n should be increased at least proportional to p to get a meaningful approximation, the computational complexity with respect to p is at least $O(p^4)$, which can be quite heavy for large p. Much of the recent research in GP modeling has focused on the big n problem, for example, using iterative kriging (Haaland and Qian (2011)) and local GPs (Gramacy and Apley (2015)). But we are not aware of any attempts to extend GP fitting to large

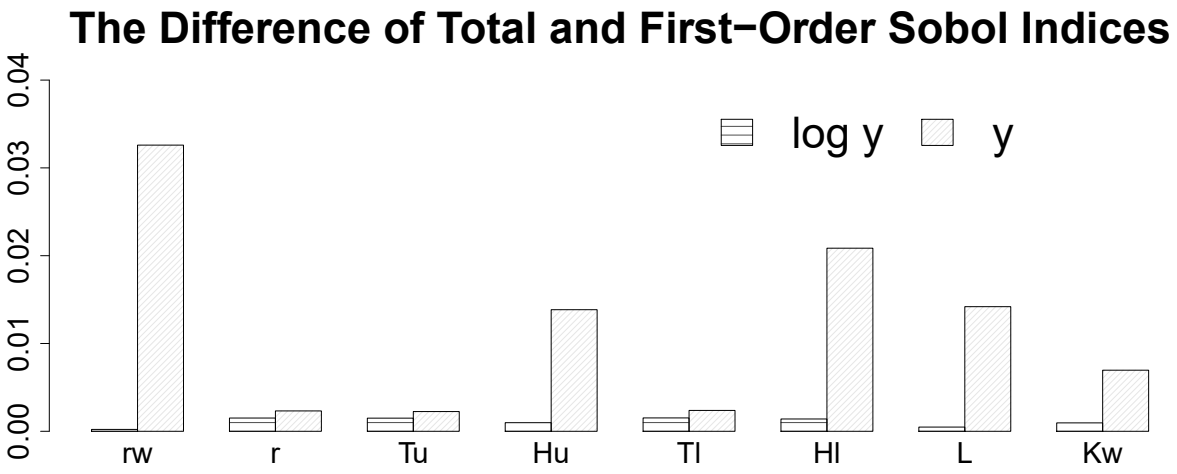

Figure 6: The difference of Sobol's total index and first-order index for the original and log-transformed responses. Large difference shows large interaction effects.

p problems. The additive GP model framework introduced here offers a pathway to fit high-dimensional GP models efficiently.

The key idea is that the additive structure of the model will allow us to fit p one-dimensional GPs instead of the one p-dimensional GP. One-dimensional GPs are easy to fit even with big n. A careful examination of the algorithms described in Sections 2 and 3 reveal that the main time consuming step is the $(p+1)$-dimensional optimization in (15). But as we noted earlier, we have good initial estimates of $\boldsymbol{s}$ obtained by fitting p one-dimensional GPs to the additive functions estimated by the back fitting algorithm. So we let $\boldsymbol{s} = \kappa \boldsymbol{s}^{(0)}$, where $\kappa \in (0, \infty)$ is a unknown parameter and $\boldsymbol{s}^{(0)}$ is obtained from Algorithm 3. Thus, the $(p+1)$-dimensional optimization reduces to a two-dimensional optimization, which is manageable. This considerably simplifies Algorithm 1, which now has only a few one or two dimensional optimizations and a quadratic program, all of which can be done quickly. Similarly, in Algorithm 2, instead of obtaining $\hat{\boldsymbol{\theta}}$ from a standard GP, we can use $\boldsymbol{\theta} = \phi \boldsymbol{s}^{(0)}$, where $\phi \in (0, \infty)$, and then finding the optimizer of (η, ϕ) through optimizing (17). Of course, avoiding the optimization over the full $\boldsymbol{s}$ and $\boldsymbol{\theta}$ can deteriorate the performance, but we found that little is lost by doing this.

To illustrate the idea, consider the function

$$y = \prod_{i=1}^{p} \frac{|4x_i - 2| + a_i}{1 + a_i},$$

where $a_i = i/2$, $i = 1, 2, \cdots, p$, with $p = 10, 20, 30, \cdots, 100$ and $n = 10p$. The designs are generated using Sobol' sequence. Besides the simplifications mentioned in the previous paragraph, we use R function *bam* in package *mgcv* in Algorithm 3, which is similar to *gam* except that the numerical methods are designed for large datasets. The left panel of Figure 7 shows the RMSPEs of GP and TAAG process and the right panel shows the estimation time using a 2.6 GHz laptop. For fitting GP, we use the standard R packages *mlegp* (Dancik and Dorman (2008)) , *GPfit* (MacDonald et al. (2015)), and *DiceKriging* (Roustant et al. (2012)). To make the comparisons fair, we set the number of initial points for optimization in *DiceKriging* to be $2p$ which is the default in *GPfit*. The time taken by GPfit for $p = 30$ is very high (56 hours), so we did not run it for $p > 30$. We can see that the RMSPEs of TAAG process are smaller than those from GP for all $p = 10, 20, \ldots, 100$ and the computational time saving increases with p. For example, it takes about 2.7 hours to fit a 100-dimensional GP with 1000 design points using *mlegp*, which takes only 30 minutes using the TAAG process. Note that unlike *mlegp* and *DiceKriging*, our current implementation of TAAG process is in R. So the actual computational saving in comparable implementations can be even more substantial. We also run the local approximate Gaussian process regression (Gramacy and Apley, 2015) through R package laGP (Gramacy, 2016) in this example: the average of RMSPE is 0.934, which is greater than our results, with standard deviation 0.063, and the running time is from 0.02 to 8.65. Although the running time of laGP is smaller than that of TAAG in this example, we may use their method or other GP based method for large scale, such as Haaland and Qian (2011), to improve TAAG for fitting larger scale data efficiently. Such works would be an interesting future research.

5 More Examples

In this section, we compare TAAG with GP over a broad class of examples of computer simulations. These examples include 5 exemplar functions from Surjanovic and Bingham

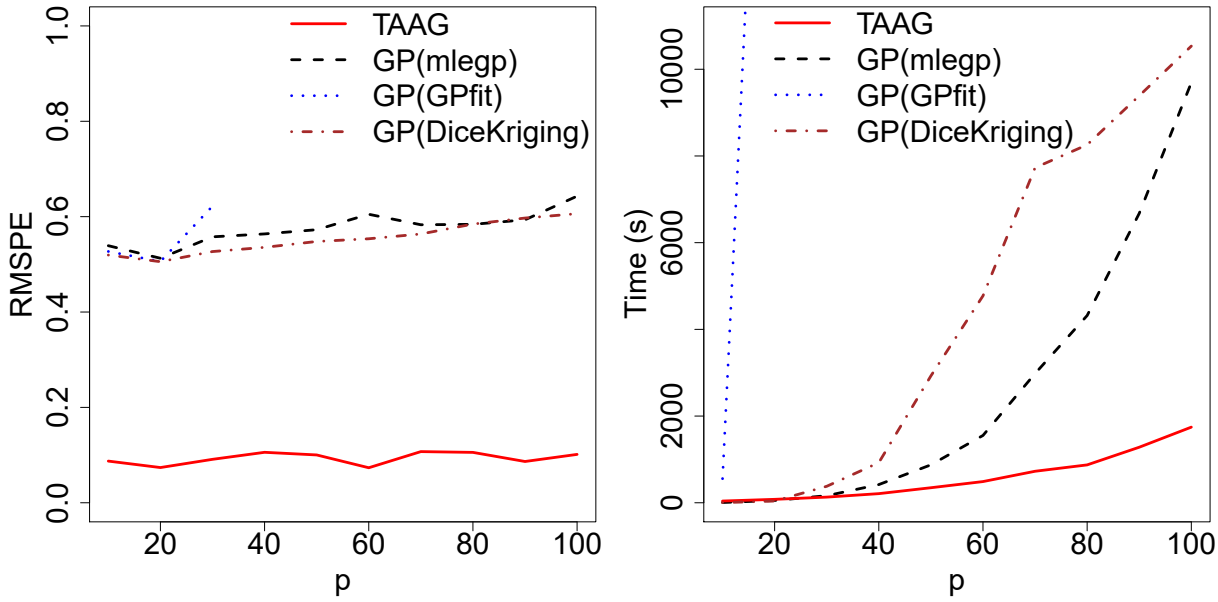

Figure 7: Computational time and root mean squared prediction errors of TAAG process and GP.

(2019): franke function, OTL circuit function, piston simulation function, robot arm function, and wing weight function. The detailed information of all these examples is summarized in the supplemental material. For comparison, each experiment of these functions consists of $10p$ runs of simulations from maximum projection designs (Joseph et al., 2015) for fitting TAAG and GP, and 1000 testing points generated from Sobol sequences with scrambling. Then, over these 1000 test points, the mean square prediction errors (MSPEs) from TAAG and GP are recorded.

The resulting MSPEs are summarized in the first 5 rows of Table 2, with the estimators of λ and η in TAAG. For all the cases, the outcomes of MSPEs from TAAG are better than that from GP. In addition, the small $\hat{\eta}$ values imply, with the estimated transformations ($\hat{\lambda}$), the interaction effects of the 5 transformed functions are small.

We also compare TAAG and GP on two datasets about a heat exchanger (HE) model presented in Qian et al. They also provide a 14-run testing dataset. The results are summarized in the last two rows of Table 2. We observe that the prediction performance of TAAG are better than that of GP, and the $\hat{\eta}$ from TAAG are small. Overall, this simu-

lation study confirms that, with an appropriate transformation, the deterministic function becomes well-approximately additive in its input variables.

		RMSPE		Estimators from TAAG	
Examples	dimension	GP	TAAG	$\hat{\lambda}$	$\hat{\eta}$
Franke	2	0.035	0.031	-0.5	.238
OTL Circuit	6	0.046	0.026	0.5	.029
Piston Simulation	7	0.012	0.00005	0	.000
Robot Arm	8	0.035	0.030	1	.010
Wing Weight	10	2.892	0.217	0	.0001
Approximated HE model	4	4.436	2.028	0.5	.004
Detailed HE model	4	2.217	1.851	1	.001

Table 2: Summary of the results of the simulation examples described in Section 5. The values reported are the dimension of input of each example functions, the MSPEs of GP and TAAG, the estimators of transformed parameters, which is $\hat{\lambda}$, of TAAG, and $\hat{\eta}$ of TAAG.

6 CONCLUSIONS

In this article we have shown that using transformation on the response can be highly beneficial in GP modeling. It can make the deterministic function approximately additive, which can be efficiently approximated using simpler models such as additive models. By exploiting the underlying additive structure, we have developed efficient estimation techniques for fitting the transformed additive GP model. In fact, it can be fitted using a few one or two dimensional optimizations and a quadratic program with initializations provided by the well-known back fitting algorithm. The estimation is so efficient that it can be applied to high-dimensional and big data problems which otherwise would not have been possible with the usual GP models. The development has also led to a new correlation function with much more interpretable parameters than the commonly used correlation

functions such as Gaussian or Matérn. The fitted models can be immediately visualized using main effects plots, which is another advantage of the proposed method. Moreover, the main effects plots are more meaningful in TAG/TAAG processes compared to the usual GP because of the minimization of the interaction effects.

Although we have focused on deterministic functions, the method can be extended to noisy data. Gaussian noise can be addressed by adding a nugget term in the TAAG process, but more work is needed for non-Gaussian data, which we leave as a topic for future research. Another important direction for future research is regarding the transformation. Here we have used the one-parameter Box-Cox transformation, which worked well in the examples we have tried so far. But we anticipate that, in more complex problems, a nonparametric transformation can perform better. The nonparametric transformation needs to be monotonic and easily invertible, which makes this extension nontrivial.

ACKNOWLEDGMENTS

This research is supported by a U.S. National Science Foundation grant DMS-1712642 and a U.S. Army Research Office grant W911NF-17-1-0007.

APPENDIX A: PROOF OF CONVERGENCE OF ALGORITHM 1

In this Appendix, we want to show that the algorithm for fitting TAG presented in section 2 converges to a stationary point. Recall that the algorithm is to iteratively update $\mathbf{c}$ by optimizing (7) with $\boldsymbol{\omega}$ fixed at its current estimator, and update $\boldsymbol{\omega}$ by optimizing (7) with $\mathbf{c}$ fixed at its current estimator. Such algorithm is based on an alternating optimization (AO) process. The convergence of AO was shown in Section 4 of Grippo and Sciandrone (2000). They showed that the sequence generated by an AO converges to a stationary point if the sequence has limit points. Thus, to show the convergence of the algorithm for fitting TAG, we need to show the sequence $\{(\boldsymbol{\omega}^{(k)}, \mathbf{c}^{(k)})\}$ generated by the TAG algorithm has limit points.

To show $\{(\boldsymbol{\omega}^{(k)}, \mathbf{c}^{(k)})\}$ has limit points, our strategy is to verify that the domains of $\boldsymbol{\omega}$ denoted by $\mathcal{W}$ and $\mathbf{c}$ denoted by $\mathcal{C}$ are bounded. If this is true, we know the level set, defined by $\{(\boldsymbol{\omega}, \mathbf{c}) \in \mathcal{W} \times \mathcal{C} : f(\mathbf{c}, \boldsymbol{\omega}) \leq f(\boldsymbol{\omega}_0, \mathbf{c}_0)\}$, for all $(\boldsymbol{\omega}_0, \mathbf{c}_0) \in \mathcal{W} \times \mathcal{C}$ is compact. Since compactness of a set implies every subset of this set has limit points, we show that $\{(\boldsymbol{\omega}^{(k)}, \mathbf{c}^{(k)})\}$ has limit points. Thus, we only need to show $\mathcal{W}$ and $\mathcal{C}$ are bounded.

The domain of $\boldsymbol{\omega}$ is bounded since $\mathcal{W}$ contains all $\boldsymbol{\omega} = (\omega_1, \cdots, \omega_p)$ satisfying $\sum_{i=1}^{p} \omega_i = 1$ and $\omega_i \geq 0$ for $i = 1, \cdots, p$. To show $\mathcal{C}$ is bounded, recall that, given data $\{(\mathbf{x}_1, y_1)\}_{i=1}^{n}$, each $\mathbf{c} \in \mathcal{C}$ can be expressed as $(\mathbf{R} + \delta\mathbf{I})^{-1}(\mathbf{y} - \mu\mathbf{1})$. This means

$$
\begin{aligned}
||\mathbf{c}|| &= ||(\mathbf{R} + \delta\mathbf{I})^{-1}(\mathbf{y} - \mu\mathbf{1})|| \\
&= ||(\sum_{i=1}^{p} \omega_i \mathbf{R}_i + \delta\mathbf{I})^{-1}(\mathbf{y} - \mu\mathbf{1})|| \qquad \text{(by definition of } \mathbf{R}) \\
&\leq ||(\sum_{i=1}^{p} \omega_i \mathbf{R}_i + \delta\mathbf{I})^{-1}||||(\mathbf{y} - \mu\mathbf{1})|| \qquad \text{(the sub-multiplicative property of a norm)} \\
&\leq (\sum_{i=1}^{p} ||(\omega_i \mathbf{R}_i)^{-1}|| + 1/\delta)||(\mathbf{y} - \mu\mathbf{1})|| \quad \text{(the convex property of the inverse operator)}
\end{aligned}
$$

Since each element of R_i is bounded for all $i = 1, \cdots, p$, we can find a bound $M \in (0, \infty)$ such that $||\mathbf{c}|| \leq M$. This implies $\mathcal{C}$ is bounded, and we complete the proof.

APPENDIX B: INITIALIZATION

The details of obtaining initial estimates of the unknown parameters in the TAG process is shown in Algorithm 3. We use the *gam* function in the R package *mgcv* (Wood, 2017) to fit the additive model and use *mlegp* (Dancik and Dorman, 2008) to fit the one-dimensional GPs.

Algorithm 3 Initialization

procedure INITIAL($\mathbf{y}, \mathbf{D}$) ▷

For each $\lambda \in \{-2, -1.5, \ldots, 2\}$, fit an additive model $g_{\lambda^{(0)}}(y) \sim h_1(x_1) + \cdots + h_p(x_p)$ and obtain $\lambda^{(0)}$ that minimizes PL(λ) in (13).

$mod \leftarrow (g_{\lambda^{(0)}}(y) \sim h_1(x_1) + \cdots + h_p(x_p))$.

$\omega_i^{(0)} = \text{var}\{h_i(x_i)\} / \sum_{i=1}^{p} \text{var}\{h_i(x_i)\}$, where $\text{var}\{h_i(x_i)\}$ can be obtained from from *mod*.

$\delta^{(0)} = 1/R^2 - 1$, where R^2 is obtained from *mod*.

Use mod to predict $h_i(x_i)$ at $\mathbf{x}_{test} = \{0, 1/(m-1), \ldots, 1\}$ with $m = 101$. Denote it as $\hat{h}_i(\mathbf{x}_{test})$.

for i from 1 to p **do**

Obtain $s_i^{(0)}$ by fitting a GP on $\{\mathbf{x}_{test}, \hat{h}_i(\mathbf{x}_{test})\}$.

end for

return $\boldsymbol{\omega}^{(0)}, \mathbf{s}^{(0)}, \lambda^{(0)}$, and $\delta^{(0)}$.

end procedure

References

Ba, S., and Joseph, V. R. (2012), "Composite Gaussian process models for emulating expensive functions," *The Annals of Applied Statistics*, 6, 1838–1860.

Box, G. E., and Cox, D. R. (1964), "An analysis of transformations," *Journal of the Royal Statistical Society. Series B (Methodological)*, 26, 211–252.

Breiman, L., and Friedman, J. H. (1985), "Estimating optimal transformations for multiple regression and correlation," *Journal of the American statistical Association*, 80, 580–598.

Christophe, D., and Petr, S. (2018), "randtoolbox: Generating and testing random numbers". R package version 1.17.1.

Cressie, N. (1992), "Statistics for spatial data," *Terra Nova*, 4, 613–617.

Dancik, G. M., and Dorman, K. S. (2008), "mlegp: statistical analysis for computer models of biological systems using R," *Bioinformatics*, 24, 1966–1967.

De Oliveira, V., Kedem, B., and Short, D. A. (1997), "Bayesian prediction of transformed Gaussian random fields," *Journal of the American Statistical Association*, 92, 1422–1433.

Duvenaud, D. K., Nickisch, H., and Rasmussen, C. E. (2011), "Additive gaussian processes," *Advances in Neural Information Processing Systems*, pp. 226–234.

Fasshauer, G. E. (2007), *Meshfree Approximation Methods with MATLAB*, Vol. 6 World Scientific.

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

Gneiting, T., and Raftery, A. E. (2007), "Strictly proper scoring rules, prediction, and estimation," *Journal of the American Statistical Association*, 102, 359–378.

Gramacy, R. B. (2016), "laGP: Large-Scale Spatial Modeling via Local Approximate Gaussian Processes in R," *Journal of Statistical Software*, 72(1), 1–46.

Gramacy, R. B., and Apley, D. W. (2015), "Local Gaussian process approximation for large computer experiments," *Journal of Computational and Graphical Statistics*, 24, 561–578.

Haaland, B., and Qian, P. Z. (2011), "Accurate emulators for large-scale computer experiments," *The Annals of Statistics*, 39, 2974–3002.

Harari, O., and Steinberg, D. M. (2014), "Convex combination of Gaussian processes for Bayesian analysis of deterministic computer experiments," *Technometrics*, 56, 443–454.

Hastie, T. J., and Tibshirani, R. J. (1990), *Generalized Additive Models*, Vol. 43 CRC press.

Joseph, V. R., Gul, E., and Ba, S. (2015), "Maximum projection designs for computer experiments," *Biometrika*, 102, 371–380.

Lázaro-Gredilla, M. (2012), "Bayesian warped Gaussian processes," *Advances in Neural Information Processing Systems*, pp. 1619–1627.

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., and Ye, K. Q. (2006), "Variable selection for Gaussian process models in computer experiments," *Technometrics*, 48, 478–490.

MacDonald, B., Ranjan, P., and Chipman, H. (2015), "GPfit: An R package for fitting a Gaussian process model to deterministic simulator outputs," *Journal of Statistical Software*, 64, 1–23.

Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993), "Bayesian design and analysis of computer experiments: use of derivatives in surface prediction," *Technometrics*, 35, 243–255.

Neal, R. M. (2012), *Bayesian Learning for Neural Networks*, Vol. 118 Springer Science & Business Media.

Plate, T. A. (1999), "Accuracy versus interpretability in flexible modeling: Implementing a tradeoff using Gaussian process models," *Behaviormetrika*, 26, 29–50.

Roustant, O., Ginsbourger, D., and Deville, Y. (2012), "DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization," *Journal of Statistical Software*, 51, 1–55.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and analysis of computer experiments," *Statistical Science*, pp. 409–423.

Santner, T. J., Williams, B. J., and Notz, W. I. (2013), *The Design and Analysis of Computer Experiments,* Springer Science & Business Media.

Snelson, E., Ghahramani, Z., and Rasmussen, C. E. (2004), "Warped gaussian processes," *Advances in Neural Information Processing Systems*, pp. 337–344.

Sobol', I. M. (1990), "On sensitivity estimation for nonlinear mathematical models," *Matematicheskoe Modelirovanie*, 2, 112–118.

Surjanovic, S., and Bingham, D. (2019), *Virtual Library of Simulation Experiments: Test Functions and Datasets* from http://www.sfu.ca/∼ssurjano, Retrieved June 3, 2019.

Tibshirani, R. (1988), "Estimating transformations for regression via additivity and variance stabilization," *Journal of the American Statistical Association*, 83, 394–405.

Williams, C. K., and Rasmussen, C. E. (1996), "Gaussian processes for regression," *Advances in Neural Information Processing Systems*, pp. 226–234.

Williams, C. K., and Rasmussen, C. E. (2006), *Gaussian Processes for Machine Learning,* The MIT Press.

Wood, S. N. (2017), *Generalized Additive Models: an Introduction with R,* Chapman and Hall/CRC.