Communications for Statistical Applications and Methods
2017, Vol. 24, No. 4, 383–396

https://doi.org/10.5351/CSAM.2017.24.4.383
Print ISSN 2287-7843 / Online ISSN 2383-4757

Model selection algorithm in Gaussian process regression for computer experiments

Youngsaeng Lee[a], Jeong-Soo Park[1,a]

[a]Department of Statistics, Chonnam National University, Korea

Abstract

The model in our approach assumes that computer responses are a realization of a Gaussian processes superimposed on a regression model called a Gaussian process regression model (GPRM). Selecting a subset of variables or building a good reduced model in classical regression is an important process to identify variables influential to responses and for further analysis such as prediction or classification. One reason to select some variables in the prediction aspect is to prevent the over-fitting or under-fitting to data. The same reasoning and approach can be applicable to GPRM. However, only a few works on the variable selection in GPRM were done. In this paper, we propose a new algorithm to build a good prediction model among some GPRMs. It is a post-work of the algorithm that includes the Welch method suggested by previous researchers. The proposed algorithms select some non-zero regression coefficients (β's) using forward and backward methods along with the Lasso guided approach. During this process, the fixed were covariance parameters (θ's) that were pre-selected by the Welch algorithm. We illustrated the superiority of our proposed models over the Welch method and non-selection models using four test functions and one real data example. Future extensions are also discussed.

Keywords: Bayesian information criterion, best linear unbiased prediction, covariance matrix, Kriging, maximum likelihood estimation, metamodel, numerical optimization

1. Introduction

The development of computer technology has enabled researchers to replace a physical experiment using complex computer simulation codes. In addition, computer codes often have high dimensional inputs. In these cases, computer simulation codes can be computationally expensive; therefore, it can be impossible to directly use a computer simulation code for the design and analysis of computer experiment (DACE), because it needs to run many computer simulation codes for the optimization of objective functions. However, one can use a statistical model as a metamodel to approximate a functional relationship between the input variables and response values of a computer simulation instead of the simulation code itself.

Typically, a computer code is deterministic or it has a small measurement error. For this reason, Sacks *et al.* (1989) suggested adopting a Gaussian process regression model (GPRM) as a metamodel for the computer simulation code. The GPRM has often been successfully used in the past for the modeling of computer simulation data. A few examples of recent works that used GPRM for the modeling of computer simulation data are as follows:

- Mechanical engineering: Slonski (2011), Lee and Gard (2014), and Dubourg *et al.* (2013)

[1] Corresponding author: Department of Statistics, Chonnam National University, 77 Yongbong-ro, Buk-gu, Gwangju 61186, Korea. E-mail: jspark@jnu.ac.kr

Published 31 July 2017 / journal homepage: http://csam.or.kr

- Health economics: Rojnik and Naveršnik (2008), and Stevenson *et al.* (2004)
- Computer and system engineering: Kennedy *et al.* (2006), Johnson *et al.* (2011), and Zhang *et al.* (2017)
- Chemical engineering: Gomes *et al.* (2008), Caballero and Grossmann (2008), and Liu *et al.* (2013)
- Other fields: Kapoor *et al.* (2010), Rohmer and Foerster (2011), Deng *et al.* (2012), Silvestrini *et al.* (2013), Tagade *et al.* (2013), and Kumar (2015).

However, many of these researches have not applied a systematic model selection method, which has motivated the current work. In classical regression, when there are many independent variables, we select the subset of independent variables which gives good fit. The reason why is because over-fitting can lead to a significant variance of predicts and under-fitting can lead to a bias of predicts. Therefore, for the building of prediction model, it is important to find best subset of independent variables which gives good fit (Jung and Park, 2015; Lee, 2015). The same reasoning and approach can be applicable to GPRM. However, only a few works about the variable selection in GPRM have been done (Linkletter *et al.*, 2006; Marrel *et al.*, 2008; Welch *et al.*, 1992), and almost all previous researches employed simple GPRM withou a variable and parameter selection. Linkletter *et al.* (2006) used a Bayesian approach while Marrel *et al.* (2008) used a corrected Akaike information criterion, which are both different from our approach.

The classical regression model has only regression coefficients β's; however, the GRRM has β's as well as coefficients θ's in the correlation function that complicates the variable selection task. In this paper, we propose to select some θ's, first by the Welch method and then, by under fixing the θ's, select some β's by forward selection and backward elimination. The proposed algorithms are validated and compared to the simple models and Welch method through four test functions as well as one real data example. From the test functions study, we found that the model obtained from the proposed methods provide better result than the other models in relation to the prediction error.

Following Sacks *et al.* (1989), we consider a Gaussian process model defined on an index set $\mathcal{X} \subseteq R^d$ for DACE:

$$y(\underline{x}) = \sum_{j=1}^{p} \beta_j f_j(\underline{x}) + Z(\underline{x}), \tag{1.1}$$

where f's are a known function and β's are unknown regression coefficient. Here the random process $Z(\cdot)$ is assumed to be a Gaussian process with mean zero and covariance matrix $\sigma^2 V$ for $\sigma^2 > 0$, where σ^2 is the process variance (a scale factor) and V is a correlation matrix. Among many possible covariance functions (see Santner *et al.*, 2003), we consider the "power exponential family" which is given by

$$\operatorname{cov}\left(Z\left(\underline{x}_i\right), Z\left(\underline{x}_j\right)\right) = \sigma^2 \exp\left(-\sum_{k=1}^{d} \theta_k \left|x_{ik} - x_{jk}\right|^{\alpha_k}\right) + \sigma_e^2 \delta_{i,j}, \tag{1.2}$$

where $\theta_k \geq 0$, $0 < \alpha_k \leq 2$ for all k, σ_e^2 is the variance due to nugget effect, and $\delta_{i,j}$ is the Kronecker delta. Then correlation function is

$$v_{i,j} = \operatorname{corr}\left(Z\left(\underline{x}_i\right), Z\left(\underline{x}_j\right)\right) = \exp\left(-\sum_{k=1}^{d} \theta_k \left|x_{ik} - x_{jk}\right|^{\alpha_k}\right) + \gamma \delta_{i,j}, \tag{1.3}$$

where $\gamma = \sigma_e^2/\sigma^2$ and $\gamma \geq 0$. Throughout this study, we set $\alpha = 2$ and $\gamma = 0$.

Once data have been collected at the observation sites $\{\underline{x}_1, \ldots, \underline{x}_n\}$, we use maximum likelihood estimation (MLE) method to estimate parameters in the model (1.1) and (1.2). Since we assume $y(x)$ is a Gaussian process with mean $F\beta$ and covariance matrix $\sigma^2 V$, the likelihood function of y is

$$L\left(y; \theta, \beta, \sigma^2, \gamma, \alpha,\ x\right) = \frac{\left(2\pi\sigma^2\right)^{-\frac{n}{2}}}{\sqrt{|V|}} \exp\left(-\frac{(y - F\beta)^t V^{-1}(y - F\beta)}{2\sigma^2}\right), \tag{1.4}$$

where F is a design matrix. A numerical optimization procedure is required because the likelihood equations do not lead to a closed form solution. We need an efficient MLE searching program to build a good prediction model. Its implementation details are presented in Park and Baek (2001).

The MLE's of the parameters are then plugged into the spatial linear model to predict $y(\underline{x})$ at which $\underline{x}$ is not an observation site (the prediction is called Kriging). The empirical best linear unbiased prediction with the MLE $(\hat{\theta}, \hat{\beta})$ of parameters plugged into is given by

$$\hat{Y}(x_0) = f_0^t \hat{\beta} + r_0^t \hat{V}^{-1} \left(\underline{y} - F\hat{\beta}\right), \tag{1.5}$$

where f_0 is the known linear regression functions vector, r_0 is the correlation vector between x_0 and design S, and $\underline{y}$ is the vector of observation collected at the design sites.

2. Existing model selection algorithm

The GPRM from (1.1) has regression coefficients β's as well as coefficients θ's in the correlation function that complicates the variable selection. Among the many possible combinations of the β's and θ's, we consider the following four models as basic ones (Cox *et al.*, 2001):

- Model 1: β_0 + common θ_c
- Model 2: β_0 + all θ's
- Model 3: first order liner model + common θ_c
- Model 4: first order liner model + all θ's.

Here the "common θ" means that d number of θ's are forced to be a common θ_c such that $\theta_1 = \theta_2 = \cdots = \theta_d := \theta_c$.

One purpose of computer experiments is to establish a cheap metamodel using the above Gaussian process model and Kriging prediction. For this purpose, a good prediction model should be built. Our experience leads to a model with a combination of some β's and θ's that gives a good prediction model. In this paper, we describe an algorithm for building a good prediction model.

Welch *et al.* (1992) described an algorithm to screen important input variables in computer experiments that used a Gaussian process model. They proposed using a dimensional reduction scheme to perform a series of presumably simpler optimization. The idea is to make tractable, the high-dimensional minimization by constraining the number of free θ's; only "important" θ's are allowed to posses their own values. Following Santner *et al.* (2003), we describe the algorithm below because our algorithm is a post-work of it. Hereafter, it is referred to as the Welch6 or W6 algorithm, because six people including the first author Welch wrote the paper (Welch *et al.*, 1992). At each stage of the process, let C denote the indices of the variables having a common θ for that step and let $C_{-j} = C$

$-j$. Notice that [S-0] is an initialization step, while [S-1] and [S-2] are induction steps. Only β_0 is estimated for the linear model term.

• Model 5: W6 algorithm

S-0 Set $C = \{1, 2, \ldots, d\}$, i.e., $\theta_1 = \theta_2 = \cdots = \theta_d := \theta_c$. Maximize (1.4) as a function of θ_c and the resulting log likelihood by l_0.

S-1 For each $j \in C$, maximize (1.4) under the constraint that θ's with in C_{-j} have a common value and θ_j varies freely. Denote the result by l_j.

S-2 Let j^M denote the θ producing the largest increase in $l_j - l_0$ for $j \in C$.

S-3 If $l_{j^M} - l_0$ represents a significant increase in the log likelihood as judged by a stopping criterion, then set $C = C_{-j^M}$, $l_0 = l_{j^M}$, and fix θ_{j^M} at its value estimated in [S-1]. Continue the next iteration at [S-1]. Otherwise, stop the algorithm and take $\hat{\theta}$'s produced by the previous iteration.

For stopping criterion, they used the number 6 which is $\chi^2_{.05}(2)$ based on the 2 times log likelihood ratio.

3. Proposed algorithms

The algorithm of Welch *et al.* (1992) works for screening input variables. But for building prediction model, we attach more steps to select some of β terms in the model. Let us assume that k θ's are selected in the above W6 algorithm; we next then fix it as true. Now we will consider five approaches: forward selection and Backward elimination of β's based on likelihood ratio test (LRT), forward selection and Backward elimination of β's based on Bayesian information criterion (BIC), and least absolute shrinkage and selection operator (Lasso) guided selection. Note that all of these are a kind of "W6 + β selection" algorithm.

• Model 6: forward β selection based on LRT

S-4 Estimate such k θ's under constant linear model (only β_0) by k-dimensional optimization routine. Here the k θ's are allowed to posses values freely. Then fix the MLE of k θ's throughout the iterations.

S-5 Select β's producing the largest increase of the log likelihood as the same as in the forward selection algorithm in ordinary regression model building. The significance of the increase is judged by a stopping criterion.

S-6 When the β selection is done, estimate the k θ's and the selected β's using the MLE program.

• Model 7: backward β elimination based on LRT

S-4 Estimate such k θ's under the first order linear model by k-dimensional optimization routine. Then fix the MLE of k θ's throughout the iterations.

S-5 Eliminate β's producing the smallest decrease of log likelihood as the same as in the backward elimination algorithm in ordinary regression model building. The significance of the decrease is judged by a stopping criterion.

S-6 When the β elimination is done, estimate the k θ's and the remained β's using the MLE program.

When the values of some θ's are fixed, the computation involved in selection or in elimination of β's are relatively fast, no optimization is required; $\hat{\beta} = (F^t V^{-1} F)^{-1} F^t V^{-1} y$. Here $\hat{\beta}$ is the generalized least squares estimator of β.

• Model 8: forward β selection based on BIC (James *et al.*, 2013)

S-4 Let M_0 denote the *null* model, which contains no predictors.

S-5 For $k = 0, \ldots, p-1$:

(a) Consider all $p-k$ models that augment the predictors in M_k with one additional predictor

(b) Choose the best among these $p-k$ models, and call it M_{k+1}. Here the *best* is defined as having smallest negative log likelihood

S-6 Select a single best model from among $M_0, \ldots, M_p$ using the smallest BIC, where

$$\mathrm{BIC}_k = -2\ln(L) + k \cdot \ln(n).$$

• Model 9: backward β elimination based on BIC

S-4 Let M_p denote the $full$ model, which contains all p predictors.

S-5 For $k = p, p-1, \ldots, 1$:

(a) Consider all k models that contain all but one of the predictors in M_k, for a total of $k-1$ predictors

(b) Choose the best among these k models, and call it M_{k-1}. Here the *best* is defined as having smallest negative log likelihood

S-6 Select a single best model from among $M_0, \ldots, M_p$ using BIC.

• Model 10: Lasso guided β selection

The lasso coefficients, $\hat{\beta}^L_\lambda$, minimize the quantity

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{3.1}$$

with respect to β's. The Lasso shrinks the coefficient estimates towards zero. The l_1 penalty on β has the effect of forcing some of the coefficient estimates to be exactly zero when the tuning parameter λ is sufficiently large. Hence, the Lasso performs *variable selection* much like the best subset selection (James *et al.*, 2013). Selecting a good value of λ is usually done by cross-validation.

We first apply the Lasso to our data, and find some non-zero $\hat{\beta}^L_\lambda$'s. Then our selected model is the GPRM with only the non-zero $\hat{\beta}^L_\lambda$'s and with the fixed covariance terms obtained by the Welch algorithm. Note that the Lasso here is applied without the covariance assumption (1.2) and is only used to select non-zero coefficients under the independent error model.

Now we have five models in which some β's are selected by the proposed methods. Table 1 presents a description on the considered 10 GPR models (GPRMs) (M1–M10).

Table 1: Description on the considered Gaussian process regression models (M1–M10)

Model	β	θ	Description
M1	β_0	θ_c	Model 1 in Section 2
M2	β_0	all θ's	Model 2 in Section 2
M3	all β's	θ_c	Model 3 in Section 2
M4	all β's	all θ's	Model 4 in Section 2
M5	β_0	θ selected by W6 algorithm	W6 algorithm in Section 3
M6	β selected	θ selected by W6 algorithm	Forward β selection based on LRT
M7	β selected	θ selected by W6 algorithm	Backward β elimination based on LRT
M8	β selected	θ selected by W6 algorithm	Forward β selection based on BIC
M9	β selected	θ selected by W6 algorithm	Backward β elimination based on BIC
M10	β selected	θ selected by W6 algorithm	Lasso guided β selection

LRT = likelihood ratio test; BIC = Bayesian information criterion.

4. Test function study

We studied simple functions as test toy models in order to check the performance of the proposed algorithm compared to the basic models and Welch's approach. After obtaining sample responses from selected design sites, we followed the above algorithms. The performance is evaluated by comparing the true function values and the predictions of the model and by using the BIC. We tried four test functions and one real data example in this study. For the evaluation of the prediction model, the following root mean squared prediction error (rmspe) is calculated for 500 or 1,000 random points on $(-0.5, 0.5)^d$:

$$\text{rmspe} = \sqrt{\frac{1}{n}\sum_{j=1}^{N}\left(y_j - \hat{y}_j\right)^2}. \tag{4.1}$$

The relative improvement of the best model over M5 (Welch algorithm) is calculated by

$$\frac{\text{rmspe(M5)} - \text{rmspe(best)}}{\text{rmspe(M5)}} \times 100.$$

4.1. Test function 1

$$y = y_1 + y_2 + y_1 \times y_2, \tag{4.2}$$

where

$$y_1 = 4x_1 + x_2 + \frac{x_3}{4} + \frac{x_4}{16}, \tag{4.3}$$

$$y_2 = x_3^2 - x_2^2, \quad -0.5 \le x_i \le 0.5. \tag{4.4}$$

As a design site, 12 runs optimal Latin hypercube design is constructed by Park (1994)'s algorithm (Figure 1).

Table 2 shows the selected prediction models and performance. Based on rmspe, M5 does not work well compared to M1–M4. The models M6 and M8 are similar and work best. The improvement over M5 is 58.0%. Based on BIC, M7, and M9 work best. This test function is a linear model that may be adequately approximated by the model of several included β terms. Figure 2 shows the residual plots of the models, in which the plot of M6 looks better than others. The Lasso guided selection, M10 did not work well. This may be because M10 uses the Lasso without taking the covariance matrix into account, while our GPRM remains the covariance dependent.

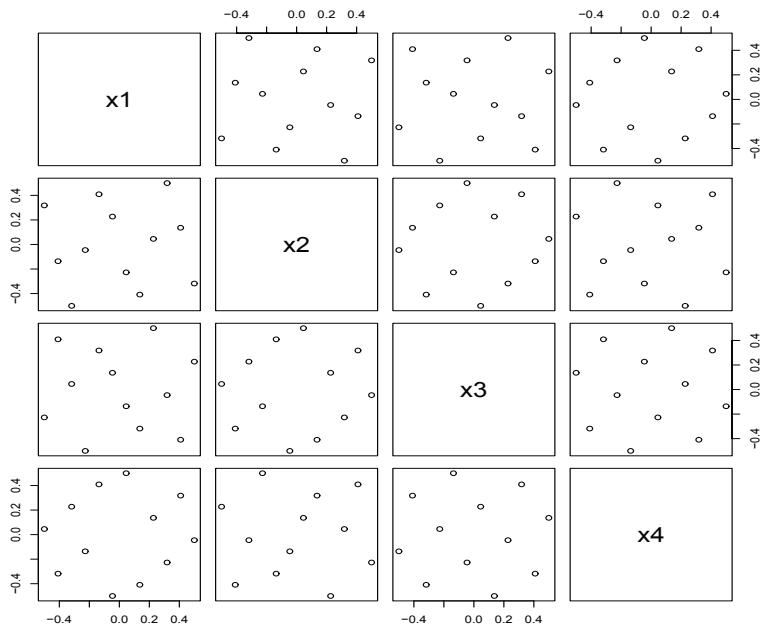

Figure 1: *A 4-dimension 12 runs optimal Latin-hypercube design used for test function 1.*

Table 2: Selected models and their rmspe and BIC value for test function 1

Model	β	θ	rmspe	BIC
M1	β_0	θ_c	0.1783	24.9748
M2	β_0	all θ's	0.1921	19.6810
M3	all β's	θ_c	0.1777	−4.9507
M4	all β's	all θ's	0.1709	−5.6084
M5	β_0	θ_1, θ_2	0.3667	24.2830
M6	$\beta_0, \beta_1, \beta_2, \beta_3$	θ_1, θ_2	0.1540	−7.2326
M7	$\beta_0, \beta_1, \beta_3, \beta_4$	θ_1, θ_2	0.2226	−8.4816
M8	$\beta_0, \beta_1, \beta_2, \beta_3$	θ_1, θ_2	0.1540	−7.2326
M9	$\beta_0, \beta_1, \beta_3, \beta_4$	θ_1, θ_2	0.2226	−8.4816
M10	$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$	θ_1, θ_2	0.1646	−5.4570

A description on models (M1–M10) is provided in Table 1.
rmspe = root mean squared prediction error; BIC = Bayesian information criterion.

4.2. Test function 2

$$y = \frac{2\pi x_3(x_4 - x_6)}{\ln(x_2/x_1)[1 + U(x)]}, \tag{4.5}$$

where

$$U(x) = \frac{2x_7x_3}{\ln(x_2/x_1)x_1^2 \times 9855} + \frac{x_3}{x_5}.$$

This equation from Morris and Mitchell (1995) has a physical interpretation that y represent steady-state flow of water through a borehole between two aquifers. A 7-dimensional 100 runs Latin hyper-

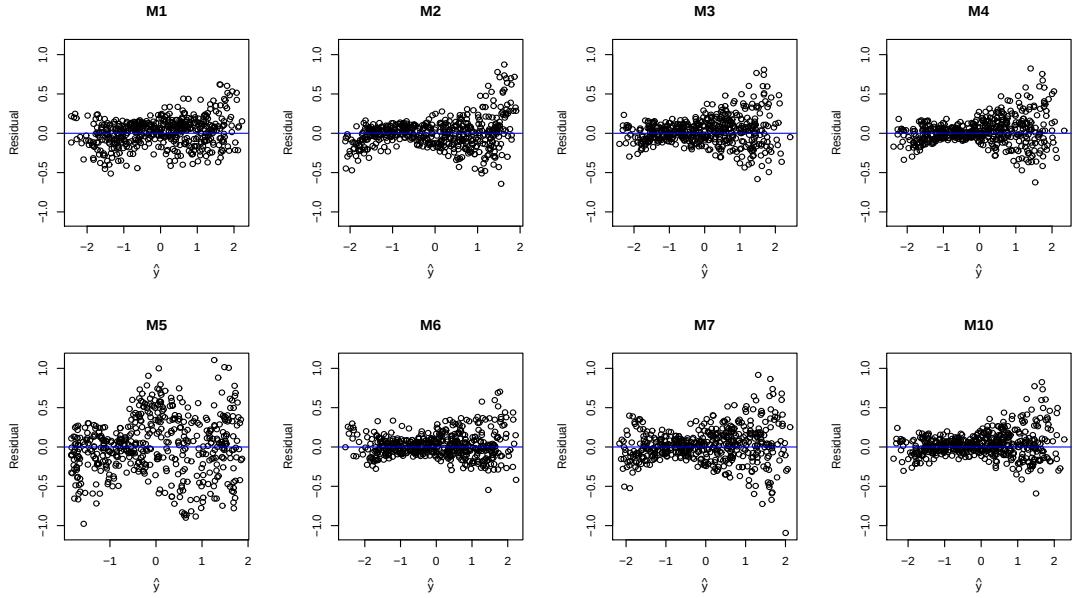

Figure 2: *Residual plots of the prediction models for the test function 1 with 500 random prediction points. Plots for M8 and M9 are omitted because they are the same for M6 and M7, respectively. See Table 1 for a description on models (M1–M10).*

Table 3: Result for test function 2

Model	β	θ	rmspe	BIC
M1	β_0	θ_c	1.6435	562.8069
M2	β_0	all θ's	1.0843	502.4237
M3	all β's	θ_c	1.2010	577.0617
M4	all β's	all θ's	0.9083	441.5618
M5	β_0	$\theta_1, \theta_4, \theta_6, \theta_7$	0.5585	340.9313
M6	$\beta_0, \beta_1, \beta_2, \beta_4, \beta_5, \beta_6, \beta_7$	$\theta_1, \theta_4, \theta_6, \theta_7$	0.3689	264.2891
M7	$\beta_0, \beta_1, \beta_2, \beta_4, \beta_5, \beta_7$	$\theta_1, \theta_4, \theta_6, \theta_7$	0.3731	262.3349
M8	$\beta_0, \beta_1, \beta_2, \beta_4, \beta_5, \beta_6, \beta_7$	$\theta_1, \theta_4, \theta_6, \theta_7$	0.3689	264.2891
M9	$\beta_0, \beta_1, \beta_2, \beta_4, \beta_5, \beta_7$	$\theta_1, \theta_4, \theta_6, \theta_7$	0.3731	262.3349
M10	$\beta_0, \beta_1, \beta_2, \beta_4, \beta_5, \beta_7$	$\theta_1, \theta_4, \theta_6, \theta_7$	0.3689	264.2891

The others are the same as Table 2.
rmspe = root mean squared prediction error; BIC = Bayesian information criterion.

cube design is constructed as a design site. One thousand random points on the domain were used to compute the rmspe.

Table 3 shows the selected prediction models and their performance. Based on rmspe, a M5 model is better than M1–M4, but worse than M6–M10. The best are M6, M8, and M10. The improvement over M5 is 33.9%. Based on BIC, M7, and M9 work best. Figure 3 shows the residual plots of the models in which the plot of M6 looks better than the others.

4.3. Test function 3

$$y = y_1 + y_2, \tag{4.6}$$

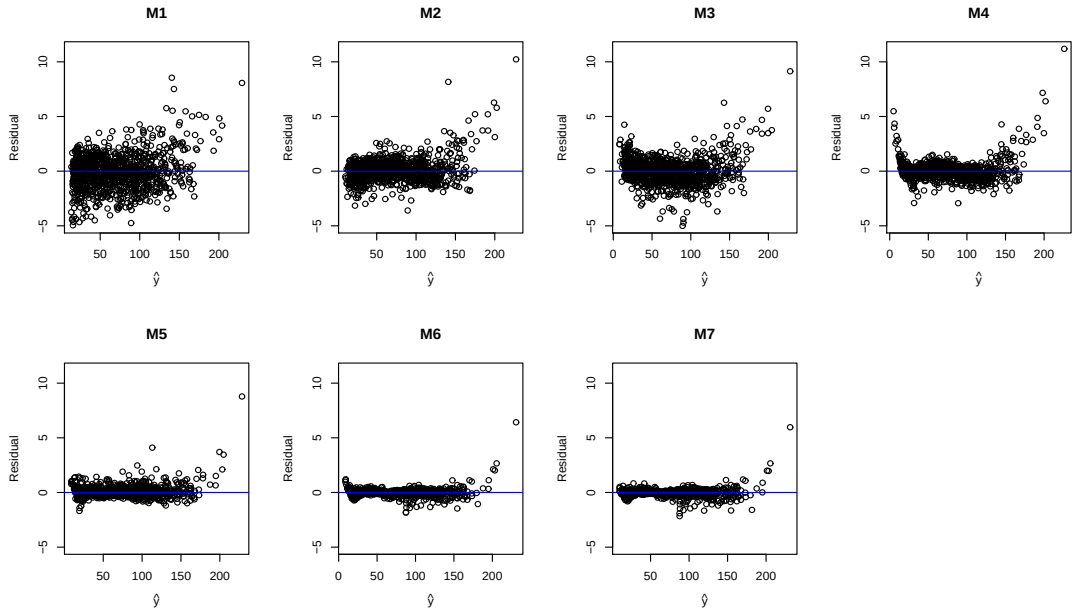

Figure 3: *Residual plots of the prediction models for the test function 2 with 1,000 random prediction points. Plots for M8, M9, and M10 are omitted because they are the same for M6 and M7, respectively. See Table 1 for a description on models (M1–M10).*

Table 4: Result for test function 3

Model	β	θ	rmspe	BIC
M1	β_0	θ_c	3.4471	290.3785
M2	β_0	all θ's	1.2965	215.6780
M3	all β's	θ_c	3.9277	332.6453
M4	all β's	all θ's	1.5580	266.4429
M5	β_0	$\theta_1, \theta_4, \theta_5, \theta_{12}, \theta_{19}, \theta_{20}$	2.4937	254.8991
M6	$\beta_0, \beta_5, \beta_9, \beta_{11}, \beta_{12}$	$\theta_1, \theta_4, \theta_5, \theta_{12}, \theta_{19}, \theta_{20}$	1.0237	188.5468
M7	$\beta_0, \beta_5, \beta_9, \beta_{11}, \beta_{12}$	$\theta_1, \theta_4, \theta_5, \theta_{12}, \theta_{19}, \theta_{20}$	1.0237	188.5468
M8	$\beta_0, \beta_5, \beta_9, \beta_{11}$	$\theta_1, \theta_4, \theta_5, \theta_{12}, \theta_{19}, \theta_{20}$	1.0499	187.6149
M9	$\beta_0, \beta_5, \beta_9, \beta_{11}$	$\theta_1, \theta_4, \theta_5, \theta_{12}, \theta_{19}, \theta_{20}$	1.0499	187.6149
M10	$\beta_0, \beta_2, \beta_7, \beta_9, \beta_{12}, \beta_{18}, \beta_{19}$	$\theta_1, \theta_4, \theta_5, \theta_{12}, \theta_{19}, \theta_{20}$	1.2551	207.4258

The others are the same as Table 2.
rmspe = root mean squared prediction error; BIC = Bayesian information criterion.

where

$$y_1 = \frac{5x_{12}}{1+x_1} + 5(x_4 - x_{20})^2 + x_5 + 40x_{19}^3 - 5x_{19}, \tag{4.7}$$

$$y_2 = 0.05x_2 + 0.08x_3 - 0.03x_6 + 0.03x_7 - 0.09x_9 - 0.01x_{10} - 0.07x_{11} + 0.25x_{13}^2 - 0.04x_{14} + 0.06x_{15} - 0.01x_{17} - 0.03x_{18} \tag{4.8}$$

for $-0.5 \le x_i \le 0.5$. This test function is from Welch *et al.* (1992). As a design site, a 20-dimensional 50 runs Latin hypercube design is used. One thousand random points on the domain were used to compute the rmspe.

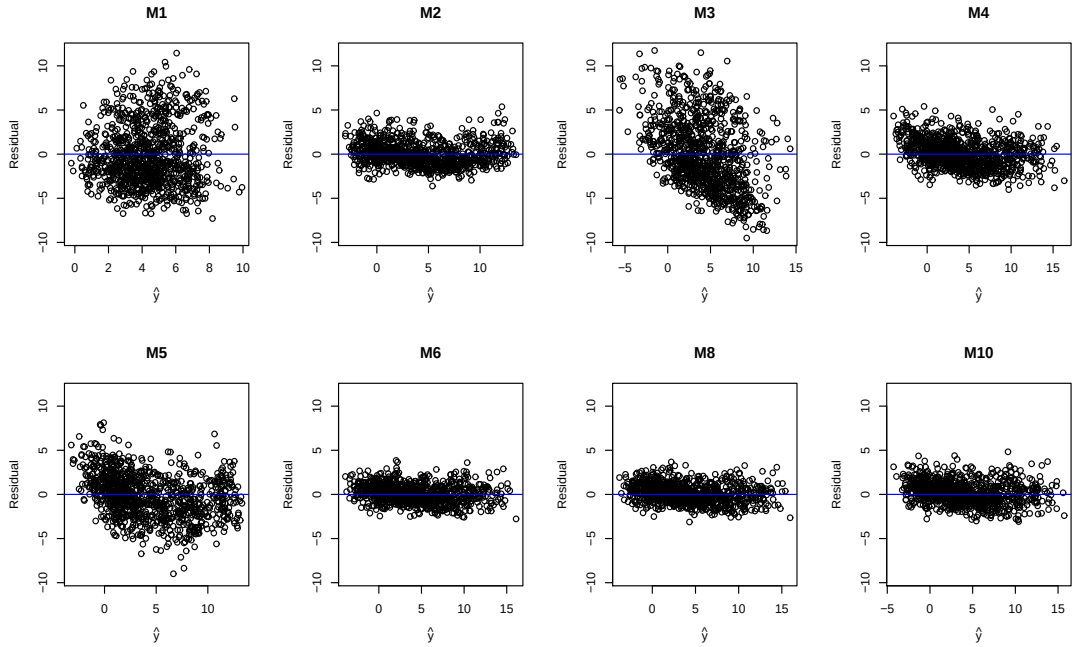

Figure 4: *Residual plots of prediction models for test function 3 with 1,000 random prediction points. Plots for M7 and M9 are omitted because they are the same for M6 and M8, respectively. See Table 1 for a description of the models (M1–M10).*

Table 4 shows the selected prediction models and their performance. Based on rmspe, M6 and M7 are the same and work best. The improvement over M5 is 58.9%. Based on BIC, M8 and M9 are the same and work best. Figure 4 shows the residual plots of the models in which the plot of M6 looks better than others.

4.4. Test function 4

$$y = 2\pi \sqrt{\frac{x_1}{x_4 + x_2^2 \frac{x_5 x_3}{x_7} \frac{x_6}{V^2}}}, \tag{4.9}$$

where

$$V = \frac{x_2}{2x_4}\left(\sqrt{A^2 + 4k\frac{x_5 x_3}{x_7}x_6} - A\right), \tag{4.10}$$

$$A = x_5 x_2 + 19.62x_1 - \frac{x_4 x_7}{x_2}, \tag{4.11}$$

where the response y is the time it takes to complete one cycle (sec); $x_1 \in [30, 60]$ is the piston weight (kg); $x_2 \in [0.005, 0.02]$ is the piston surface area (m^2); $x_3 \in [0.002, 0.01]$ is the initial gas volume (m^3); $x_4 \in [1{,}000, 5{,}000]$ is the spring coefficient (N/m); $x_5 \in [90{,}000, 110{,}000]$ is the atmospheric pressure (N/m^2); $x_6 \in [290, 296]$ is the ambient temperature (K); $x_7 \in [340, 360]$ is the filling gas temperature (K). This piston simulation function is from Moon (2010) and the Virtual Library of

Table 5: Result for test function 4

Model	β	θ	rmspe	BIC
M1	β_0	θ_c	0.01705	−779.3298
M2	β_0	all θ's	0.00296	−1354.8725
M3	all β's	θ_c	0.01999	−823.0619
M4	all β's	all θ's	0.00170	−1581.1772
M5	β_0	$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$	0.00082	−1768.6100
M6	$\beta_0, \beta_1, \beta_4, \beta_6$	$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$	0.00069	−1943.1533
M7	$\beta_0, \beta_1, \beta_4, \beta_6$	$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$	0.00069	−1943.1533
M8	$\beta_0, \beta_1, \beta_4, \beta_6$	$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$	0.00069	−1943.1533
M9	$\beta_0, \beta_1, \beta_4, \beta_6$	$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$	0.00069	−1943.1533
M10	all β's	$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$	0.00070	−1923.4304

The others are the same as Table 2.
rmspe = root mean squared prediction error; BIC = Bayesian information criterion.

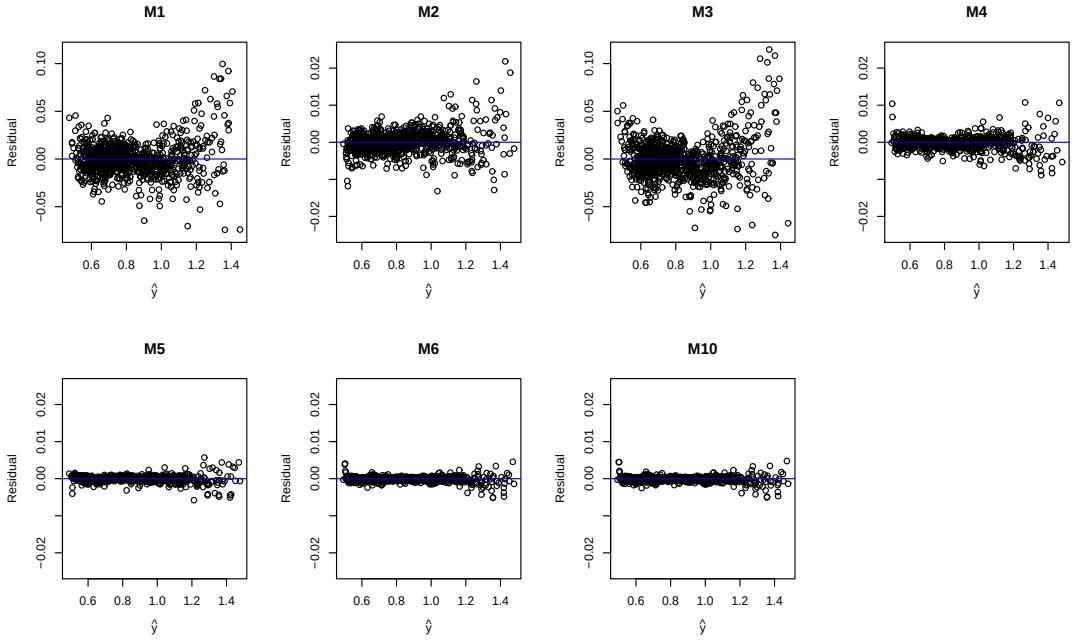

Figure 5: *Residual plots of prediction models for test function 4 with 1,000 random prediction points. Plots for M7, M8, and M9 are omitted because these are the same for M6. See Table 1 for a description of the models (M1–M10).*

Simulation Experiment (Surjanovic and Bingham, 2015) that models the circular motion of a piston. As a design site, a 7-dimensional 200 runs Latin-hypercube design is used. One thousand random points on the domain computed the rmspe.

Table 5 shows the selected prediction models and performance. Based on rmspe and BIC, M6, M7, M8, and M9 are the same and work best. The improvement over M5 is 15.9%. Figure 5 shows the residual plots of the models, in which the plot of M6 looks best.

4.5. Real example: MARTHE dataset

The MARTHE dataset is realization of the MARTHE code, which is about numerical simulation of

Table 6: Result for MARTHE dataset

Model	β	θ	rmspe	BIC
M1	β_0	θ_c	0.3550	95.2770
M2	β_0	all θ's	0.2838	2.8409
M3	all β's	θ_c	0.3339	123.0978
M4	all β's	all θ's	0.3072	52.0115
M5	β_0	$\theta_2, \theta_3, \theta_4, \theta_6, \theta_{14}, \theta_{15}, \theta_{19}, \theta_{20}$	0.2803	−6.3102
M6	$\beta_0, \beta_1, \beta_2, \beta_3, \beta_6, \beta_9, \beta_{15}, \beta_{19}, \beta_{20}$	$\theta_2, \theta_3, \theta_4, \theta_6, \theta_{14}, \theta_{15}, \theta_{19}, \theta_{20}$	0.2817	5.7842
M7	$\beta_0, \beta_1, \beta_6, \beta_{12}, \beta_{15}, \beta_{19}, \beta_{20}$	$\theta_2, \theta_3, \theta_4, \theta_6, \theta_{14}, \theta_{15}, \theta_{19}, \theta_{20}$	0.2822	1.0033
M8	$\beta_0, \beta_2, \beta_{12}, \beta_{15}, \beta_{20}$	$\theta_2, \theta_3, \theta_4, \theta_6, \theta_{14}, \theta_{15}, \theta_{19}, \theta_{20}$	0.2779	−7.7810
M9	$\beta_0, \beta_2, \beta_{15}, \beta_{20}$	$\theta_2, \theta_3, \theta_4, \theta_6, \theta_{14}, \theta_{15}, \theta_{19}, \theta_{20}$	0.2764	−11.9247
M10	$\beta_0, \beta_2, \beta_6, \beta_{13}, \beta_{14}, \beta_{15}, \beta_{20}$	$\theta_2, \theta_3, \theta_4, \theta_6, \theta_{14}, \theta_{15}, \theta_{19}, \theta_{20}$	0.2769	−5.7742

The others are the same as Table 2.
rmspe = root mean squared prediction error; BIC = Bayesian information criterion.

Strontium-90 transport in the upper aquifer of the "Kurchatov Institute" radiation waste disposal site. The computer code is not accessible; therefore, we obtained the dataset from the Virtual Library of Simulation Experiment (see Surjanovic and Bingham, 2015). The data consists of 300 observations with 20 input variables and 10 output dimensions. In this study, we only in 20 input variables and the first output variable 'p102K'. Two hundred observations are used for training data to build models, and the other 100 observations are used for test data to compute rmspe.

Table 6 shows the selected prediction models and performance. M9 is best based on rmspe and BIC. The improvement over M5 is 1.4%.

5. Summary and discussion

This study proposes an algorithm to build a good prediction model. It is a post-work of the algorithm by Welch *et al.* (1992). It selects some β's while the pre-selected θ's are fixed. Using four test functions and one real data example, we illustrated the superiority of the proposed models over other models. Forward selection and backward elimination of β's based on the likelihood ratio test or BIC work well; however, the models built by the algorithm by Welch *et al.* (1992) and by the Lasso guided selection did not work well.

We tried several alternative approaches during the progress of this study as follows. In addition, we should study all of them further despite the problems faced in our brief experience to reach a conclusion on which is the best.

1. β first approach: Selecting β's first (and then selecting θ's) did not work well.

2. θ backward elimination: The strategy that estimating all θ's first, and eliminate some θ's, and then select some β's seems good. Based on our experience, its performance is fair compared to the proposed algorithm; however, it is computationally expensive compared to Welch algorithm and our proposed methods.

3. θ Forward selection: Select θ one-by-one method like as the forward selection algorithm in the ordinary regression model building. This did not work well.

4. Pingpong approach: Select one θ, and select one β, and select another one θ, and select another one β. It seems poor; however, still needs a future study.

5. Backward elimination from the full model: After estimating all β'a and all θ's from the Model 4,

compute Wald's t-statistics ($T = \hat{\theta}/\mathrm{SE}(\hat{\theta})$) to eliminate some β's and θ's, and continue this until no more deletion. The Fisher information matrix may be required to obtain $\mathrm{SE}(\hat{\theta})$.

Sample size is an important setting in the test function study. To reliably evaluate a stable performance, we think a large sample is especially required when the dimension of input variables is high. When there are measurement errors in computer codes or in physical experiments, we set the parameter γ in (1.3) as positive. So our model selection methods in GPRM can also be applicable to physical or computer experiments with the measurement error. For a practical situation, we cannot compute the prediction error as we did in the test function study. Therefore, a leave-one-out or k-fold cross-validation version of prediction error is required. Finally, we may go further to select or eliminate more θ's and β's from the models (M6–M9) considered in this paper that may provide a better model in the sense of less prediction error. However, these remain topics for future study.

Acknowledgements

This study was financially supported by Chonnam National University, 2014.

References

Caballero JA and Grossmann IE (2008). Rigorous flowsheet optimization using process simulators and surrogate models, *Computer Aided Chemical Engineering*, **25**, 551–556.

Cox DD, Park JS, and Singer CE (2001). A statistical method for tuning a computer code to a data base, *Computational Statistics & Data Analysis*, **37**, 77–92.

Deng H, Shao W, Ma Y, and Wei Z (2012). Bayesian metamodeling for computer experiments using the Gaussian Kriging models, *Quality and Reliability Engineering International*, **28**, 455–466.

Dubourg V, Sudret B, and Deheeger F (2013). Metamodel-based importance sampling for structural reliability analysis, *Probabilistic Engineering Mechanics*, **33**, 47–57.

Gomes MVC, Bogle IDL, Biscaia EC, and Odloak D (2008). Using Kriging models for real-time process optimisation, *Computer Aided Chemical Engineering*, **25**, 361–366.

James G, Witten D, Hastie T, and Tibshirani R (2013). *An Introduction to Statistical Learning: with Applications in R*, Springer, New York.

Johnson JS, Gosling JP, and Kennedy MC (2011). Gaussian process emulation for second-order Monte Carlo simulations, *Journal of Statistical Planning and Inference*, **141**, 1838–1848.

Jung SY and Park C (2015). Variable selection with nonconcave penalty function on reduced-rank regression, *Communications for Statistical Applications and Methods*, **22**, 41–54.

Kapoor A, Grauman K, Urtasun R, and Darrell T (2010). Gaussian processes for object categorization, *International Journal of Computer Vision*, **88**, 169–188.

Kennedy MC, Anderson CW, Conti S, and O'Hagan A (2006). Case studies in Gaussian process modelling of computer codes, *Reliability Engineering & System Safety*, **91**, 1301–1309.

Kumar A (2015). Sequential tuning of complex computer models, *Journal of Statistical Computation and Simulation*, **85**, 393–404.

Lee JH and Gard K (2014). Vehicle-soil interaction: testing, modeling, calibration and validation, *Journal of Terramechanics*, **52**, 9–21.

Lee S (2015). An additive sparse penalty for variable selection in high-dimensional linear regression model, *Communications for Statistical Applications and Methods*, **22**, 147–157.

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

Liu YJ, Chen T, and Yao Y (2013). Nonlinear process monitoring by integrating manifold learning with Gaussian process, *Computer Aided Chemical Engineering*, **32**, 1009–1014.

Marrel A, Iooss B, van Dorpe F, and Volkova E (2008). An efficient methodology for modeling complex computer codes with Gaussian processes, *Computational Statistics and Data Analysis*, **52**, 4731–4744.

Moon H (2010). Design and analysis of computer experiments for screening input variables (Doctoral dissertation), Ohio State University, Columbus, OH.

Morris MD and Mitchell TJ (1995). Exploratory designs for computational experiments, *Journal of Statistical Planning and Inference*, **43**, 381–402.

Park JS (1994). Optimal Latin-hypercube designs for computer experiments, *Journal of Statistical Planning and Inference*, **39**, 95–111.

Park JS and Baek J (2001). Efficient computation of maximum likelihood estimators in a spatial linear model with power exponential covariogram, *Computers & Geosciences*, **27**, 1–7.

Rohmer J and Foerster E (2011). Global sensitivity analysis of large-scale numerical landslide models based on Gaussian-Process meta-modeling, *Computers & Geosciences*, **37**, 917–927.

Rojnik K and Naveršnik K (2008). Gaussian process metamodeling in Bayesian value of information analysis: a case of the complex health economic model for breast cancer screening, *Value in Health*, **11**, 240–250.

Sacks J, Welch WJ, Mitchell TJ, and Wynn HP (1989). Design and analysis of computer experiments, *Statistical Science*, **4**, 409–423.

Santner TJ, Williams BJ, and Notz WI (2003). *The Design and Analysis of Computer Experiments*, Springer, New York.

Silvestrini RT, Montgomery DC, and Jones B (2013). Comparing computer experiments for the Gaussian process model using integrated prediction variance, *Quality Engineering*, **25**, 164–174.

Slonski M (2011). Bayesian neural networks and Gaussian processes in identification of concrete properties, *Computer Assisted Mechanics and Engineering Science*, **18**, 291–302.

Stevenson MD, Oakley J, and Chilcott JB (2004). Gaussian process modeling in conjunction with individual patient simulation modeling: a case study describing the calculation of cost-effectiveness ratios for the treatment of established osteoporosis, *Medical Decision Making*, **24**, 89–100.

Surjanovic S and Bingham D (2015). Virtual Library of Simulation Experiments: test functions and datasets: emulation/prediction test problems, Retrieved January 20, 2017, from: https://www.sfu.ca/~ssurjano/emulat.html

Tagade PM, Jeong BM, and Choi HL (2013). A Gaussian process emulator approach for rapid contaminant characterization with an integrated multizone-CFD model, *Building and Environment*, **70**, 232–244.

Welch WJ, Buck RJ, Sacks J, Wynn HP, Mitchell TJ, and Morris MD (1992). Screening, predicting and computer experiments, *Technometrics*, **34**, 15–25.

Zhang J, Taflanidis AA, and Medina JC (2017). Sequential approximate optimization for design under uncertainty problems utilizing Kriging metamodeling in augmented input space, *Computer Methods in Applied Mechanics and Engineering*, **315**, 369–395.

Received February 9, 2017; Revised June 29, 2017; Accepted June 30, 2017