 doi:10.20944/preprints201908.0246.v1

Statistical Methods for Computer Experiments with Applications in Neuroscience

Gilad Shapira, Ella Shaposhnik, David M. Steinberg*

Department of Statistics and Operations Research

Tel Aviv University

*Corresponding: dms@tauex.tau.ac.il

Abstract

Many scientific and technological problems are studied with the help of computer codes that simulate the phenomena of interest rather than via traditional laboratory experiments. Such models play an important role in neuroscience where they are used to mimic brain function from the sub-cellular to the macroscopic level. Exploration with computer models carries with it a number of statistical challenges: where to sample the input space for the simulator, how to make sense of the data that is generated, how to estimate unknown parameters in the model, how to validate a model. The simulator setting also has some unique problems and possibilities. This review paper describes statistical research on these issues and how that work might be applied to neural simulations.

Keywords: statistics; in silico experiments; neural simulation; gaussian process regression; Bayesian inference

1. WHAT IS A COMPUTER EXPERIMENT?

Many interesting processes are difficult to study in the laboratory. An increasingly common alternative in science is to explore these processes with the aid of computer simulations that mimic the natural phenomenon via appropriate mathematical equations. The use of simulators is a focal point in modern neuroscience, in general, and of the European Human Brain Project (HBP), in particular. The Human Brain Simulation Platform (BSP) of the HBP states, "the tools provided by the BSP allow researchers to perform *in silico* experiments to validate models and to perform investigations that are not possible in the laboratory".

There are many similarities between *in silico* experiments and traditional laboratory experiments: what factors should be studied, which controlled at constant levels, what combinations of factor levels to simulate, and how to analyze and draw conclusions from the resulting data. There are, though, many important differences. The unique features that arise when the lab bench is

replaced by a simulation model have motivated a substantial body of statistical research over the last 30 years. The purpose of this paper is to give general background and then to review recent developments in the field, highlighting their possible implementation in neuroscience. Our focus will be on methods for analyzing data from a computer simulator. However, it is important to note that much progress has also been made on the design of these experiments (i.e. where to evaluate the simulator). Readers interested in more detail can refer to (Sacks, Welch, Mitchell, & Wynn, 1989), (Levy & Steinberg, 2010) and (Santner, Williams & Notz 2016**)**.

The statistical developments related to computer experiments are useful for a number of related problems, some of them common in neuroscience. In particular, they include efficient tools for Bayesian optimization and for describing complex posterior distributions. We also discuss these methods and their applications.

In Section 2 we present a number of relevant applications with an emphasis on neuroscience. Section 3 gives an overview of methods for analyzing data generated by a computer simulator. Section 4 presents the popular and useful Gaussian process (GP) regression model. Section 5 describes some alternative analysis methods. Section 6 returns to the GP approach, presenting methods for problems with both quantitative and qualitative input factors. In Section 7 we describe methods that are appropriate for the computational challenges posed by large simulator data sets. Section 8 is devoted to problems that can be studied using simulators at varying levels of resolution. In Section 9 we discuss calibration of a computer model using laboratory or field data. Section 10 shows how the ideas for analyzing computer experiments can be used to carry out challenging problems in Bayesian inference. Section 11 describes how the methods can be applied to optimization problems. In Section 12 we consider the case of simulators that generate multivariate output. Section 13 looks at methods for uncertainty assessment of a computer model. Finally, in Section 14, we give a brief summary of some of the tools that are now available to implement these methods.

2. APPLICATIONS

We present here some examples of the use of computer experiments, with an emphasis on applications in neuroscience. One recurring theme is the use of Gaussian processes (GPs) in modeling data. Another is the use of simulator methods to carry out computationally intensive Bayesian statistical analysis.

2.1 Deep Brain Stimulation

Methods from computer experiments can be used to emulate neuron behavior of a single cell or a complete spiking neural network. Deep brain simulations use as building blocks single neuron cell models, which are based on differential equations and therefore require considerable

computational time. (De La Pava et al., 2015) developed a new methodology to reduce the computation time required to estimate the volume of tissue activated during deep brain stimulation. At the heart of their method is an emulator that replaces the system of differential equations with combined multi-compartment axon models coupled to the stimulating electric field with a Gaussian process classifier. The emulator is constructed by fitting a Gaussian process to outcome data. The approach of (De La Pava et al., 2015) reduced by a factor of 10 the average computational runtime of Volume Tissue Activated estimation compared with the gold standard.

2.2 Whole Mouse Brain Dynamics

(Melozzi, Woodman, Jirsa, & Bernard, 2017) describe the benefits of *in silico* experimentation with The Virtual Mouse Brain (TVMB), a platform that facilitates investigation of large-scale mouse brain dynamics. They describe how TVMB might be used to investigate a number of research questions. One example concerns resting state activity in epilepsy. The simulation platform enables the research team to remove connections among different neural regions, reflecting the anatomic reorganization that occurs with neurological diseases. Intermediate settings can be studied by declaring impaired connection strengths between source and target regions, rather than removing them entirely. An interesting feature is that the output may be complex maps of the functional connection dynamics of the brain.

2.3 Inferring Neural Firing Rates from Spike Trains Using Gaussian Processes

The output from a neural simulator is often a spike train. Such data pose challenges to analytical efforts due to their noisy, spiking nature. Many studies of neuroscientific and neural prosthetic importance rely on a smoothed, denoised estimate of the spike train's underlying firing rate. (Cunningham, Byron, Shenoy, & Sahani, 2008) present a new method, based on a Gaussian Process prior, for estimating probabilistically optimal estimates of firing rate functions underlying single or multiple neural spike trains. Whereas current techniques to find time-varying firing rates require *ad hoc* choices of parameters and offer no confidence intervals on their estimates, their method demonstrates improvements over conventional estimators.

2.4 Gaussian Process Behavior in Wide Deep Neural Networks

Neural networks have been extensively used in machine learning applications and have widespread use in neuroscience such as modeling behavior of certain parts of the brain, spike trains, etc. While deep neural networks have shown great empirical success, there is much room for research that will help us understand their theoretical qualities. (Matthews et al., 2018) published an important theoretical work, studying the relationship between feedforward, wide, fully connected, random networks with multiple hidden layers and Gaussian processes with a recursive kernel definition. They based their work on (Neal, 1996), who showed that under some

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

conditions, random, one hidden layer neural networks converge to a Gaussian process as the number of nodes increases. This result provided a connection between flexible Bayesian neural networks and Gaussian processes (Williams, 1998; Rasmussen & Williams, 2006). (Matthews, et al., 2018) provided a strict result on the convergence of certain sequences of finite and fully connected neural networks with multiple hidden layers to Gaussian processes.

The result suggests that a Gaussian process model can effectively emulate the Bayesian neural network. In fact, (Matthews et al., 2018) found that such a model could accurately emulate the network and accelerate computations by orders of magnitude.

2.5 Bayesian Analysis

Some applications in neuroscience apply Bayesian statistical inference as part of their data analysis. For examples, see (Eriksson et al., 2019) and (Papamakarios, Sterratt, & Murray, 2018) . Bayesian inference combines a prior distribution $\pi(\theta)$ on the model parameters with a likelihood function $L(\theta; y)$ that links the observed data to the parameters via a probability model. The analysis is summarized by the posterior distribution for the parameters, $\pi^*(\theta)$, where

$$\pi^*(\theta) \propto \pi(\theta)L(\theta; y).$$

The publications cited here both use models which require running a simulator to evaluate the likelihood. (Eriksson et al., 2019) consider a model related to protein activity in which the simulator is used to compute expected values for the observed data; then one can make standard assumptions for the distribution of the data about the expected values. (Papamakarios et al., 2018) consider a model in which randomness enters inside the simulator, so that the entire shape of the likelihood function, and not just expected values, must be determined from simulator runs.

2.6 Molecular Dynamics

Molecular dynamics uses computationally intensive simulations to study how the motion of atoms and molecules may lead to the formation of proteins or other biomolecules. (Raiteri, Laio, Gervasio, Micheletti, & Parrinello, 2006) modeled a free energy profile of ligand binding, while searching for a local minimum of free energy change. To overcome the computational cost and reach the desired energy trajectory the authors used new sampling techniques. They ran multiple replicas of the system, each with its associated trajectory, in parallel. All the replicas contribute to the estimation of the free energy. By applying and improving computational methods, the researchers were able to build a free energy profile faster than using previous methadymanics method.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

3. MODELING AND DESIGN

In this section we present a broad picture of ideas for modeling the data that is generated by a computer simulator and for designing computer experiments. There are a number of potential benefits from modeling. First, analysis can indicate which input factors have large effects (and which small effects) on the output. In principle, one might think this would be known from the work that went into building the simulator. In practice, many simulators are so complex that it is difficult to determine from first principles which factors are most important. Empirical inquiry may be much simpler. Second, many simulators involve complex mathematical modeling and are too costly, or too slow, to permit many simulation shots. In such cases, statistical modeling can support an emulator, or surrogate model, of the experiment. With the emulator, much more extensive exploration of factor settings is possible. This consideration is especially important in applications where the goal is to optimize a system or to characterize uncertainty (both of which may require many function calls). Third, analysis can be essential for uncertainty quantification, which aims to understand output variation when there is large uncertainty regarding input values.

A variety of flexible regression models could be used to link the output $Y(\underline{x})$ of a computer simulator to the vector x of inputs. We will present a number of options, but with special emphasis on Gaussian Process models, which have proven a popular and effective approach in many applications. These models have roots in geostatistics where they were proposed by (Krige, 1951) as a method for spatial interpolation of two or three-dimensional data observed at locations on the globe. They have a number of attractive features for modeling data from computer experiments. In recent years they have also become an important tool in machine learning (Ghahramani, 2013) (Rasmussen & Williams, 2006) .

Experimental design relates to the collections of input points where the simulator is run and evaluated. A good design enables more informative analysis, with greater potential for generalization. The most popular design for computer experiments has been the Latin hypercube (LHC) (Mckay, Beckman, & Conover, 1979) and its close relatives. The LHC uses settings for each single input that lie on a lattice, usually with different values for each simulation run. A number of methods have been proposed to mate the levels of the different factors. The simplest approach is to mate them at random. More often secondary criteria are adopted, like maximizing the minimum interpoint distance (maximin LHC designs) (Johnson, Moore, & Ylvisaker, 1990) or making the factors first-order orthogonal (Steinberg & Lin, 2006). Other space-filling sequences have also been used.

We will comment on additional ideas in design as they relate to some of the specific data analysis problems that we cover in the ensuing sections.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

4. GAUSSIAN PROCESS MODELS

GP models were suggested for use in the context of computer simulators by (Sacks et al., 1989). They viewed the output $Y(\underline{x})$ as a realization of a random process,

$$Y(\underline{x}) = \sum \beta_i f_i(\underline{x}) + Z(\underline{x})$$

where the $f_i(\underline{x})$ are known regression functions, the β_i are unknown regression parameters and the term $Z(\underline{x})$ represents the deviation of the simulator from the regression model. They assumed that $Z(\underline{x})$ is a GP with $E(Z(\underline{x})) = 0; Var(Z(\underline{x})) = \sigma^2$, and correlation function $R(\underline{x}, \underline{z})$. One could also add a "random error," as in a standard regression model; however, we have in mind problems where $Y(\underline{x})$ is the output of a deterministic simulator, so do not include such a term.

Having observed simulator output Y_i at input settings $\underline{x}_i$, the GP model leads to a natural emulator of the output $Y(\underline{x}^*)$ at the input setting $\underline{x}^*$ – the Best Linear Unbiased Predictor (BLUP). The BLUP depends on the vector $\underline{f}(\underline{x}^*)$ that evaluates the regression functions at $\underline{x}^*$, the vector $\underline{r}(\underline{x}^*)$ that evaluates the correlations $R(\underline{x}^*, \underline{x}_i)$, $i = 1, \ldots, n$, the matrix F whose rows are $\underline{f}^T(\underline{x}_i)$ and the kernel matrix R defined by $R_{i,j} = R(\underline{x}_i, \underline{x}_j)$, , $i, j = 1, \ldots, n$. The BLUP is then given by

$$\hat{y}(\underline{x}^*) = E\left[Y(\underline{x}^*)\middle|\underline{y}\right] = \underline{f}^T(\underline{x}^*)\hat{\underline{\beta}} + \underline{r}^T(\underline{x}^*)R^{-1}(\underline{y} - F\hat{\underline{\beta}})$$

where $\underline{r}(\underline{x}^*) = [R(\underline{x}^*, x_1), \ldots, R(\underline{x}^*, x_n)]^T$ and $\hat{\underline{\beta}} = (F^T R^{-1} F)^{-1} F^T R^{-1} \underline{y}$.

Five useful properties of the GP predictor are:

The predictor is very flexible and can accurately represent a broad spectrum of response functions.

Assumptions about the likely "smoothness" of the response function can be injected into the predictor via appropriate choice of the covariance kernel.

The GP predictor interpolates the observed data: if $\underline{x}^*$is one of the observation sites, then $\hat{y}(\underline{x}^*) = Y(\underline{x}^*)$. This is very appealing with computer simulator data when the response function is observed with no random error.

The method provides not just a predictor, but also a measure of the uncertainty associated with the prediction.

The results can be seen as solutions to a Bayesian inference problem.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

Regarding point (3), it is also straightforward to accommodate a random error term in the model; in that case $\hat{y}(\underline{x}^*)$ smooths the data rather than interpolating it, with the random error variance tuning the degree of smoothing.

As for point (4), the MSE for $\hat{y}(\underline{x}^*)$ is given by

$$MSE\left[Y(\underline{x})\middle|\underline{y}\right] = \sigma^2\left\{1 - [\underline{f}^T(\underline{x}^*)\underline{r}^T(\underline{x}^*)]\begin{bmatrix}0 & F^T\\ F & R\end{bmatrix}^{-1}\begin{bmatrix}\underline{f}(\underline{x}^*)\\ \underline{r}(\underline{x}^*)\end{bmatrix}\right\}$$

4.1 Bayesian view of GP models

In the Bayesian view, the correlation function is interpreted as a Gaussian prior distribution on the response function. Given the data, the posterior distribution of $Y(\underline{x}^*)$ is then normal with expectation $\hat{y}(\underline{x}^*)$ and variance equal to the MSE above. We can compute a credible interval for $Y(\underline{x}^*)$, estimating σ^2 from the data and using a multiplier from the t-distribution. Many works compute these intervals using estimated values of model parameters; however, it is also possible to average over the uncertainty in those parameters to produce fully Bayesian credible intervals.

What functions should be included in the regression model? (Welch et al., 1992) found empirically that it was best to remove all regression terms except a constant. This reduces the model to $Y(\underline{x}) = \beta + Z(\underline{x})$ and reflects all the dependence on x via the correlation function. Researchers in machine learning often drop the constant as well, leading to a slightly simpler formula for the BLUP, as presented in (Rasmussen & Williams, 2006). Support for the use of these models was provided by (Steinberg & Bursztyn, 2004), who showed that commonly used correlation functions effectively model mean value and linear trends, obviating the need to include them explicitly as regression functions.

Use of a sparse regression model can be problematic if the emulator is used to generate predictions outside the range of the observed data. It is easy to show that such extrapolations are dominated by the regression function. For example, in the machine-learning version, extrapolations far from the data will all be 0. (Joseph, Hung, & Sudjianto, 2008) and (Joseph & Melkote, 2009) presented useful databased methods for selecting good regression models.

4.2 Estimation

The GP model includes parameters in the regression function and in the correlation function. (Sacks et al., 1989) showed how these could be estimated by maximum likelihood. The MLE for the regression parameters is the weighted least squares solution shown in the previous

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

section. There is also a closed formula for the MLE of the overall variance of the GP. Any remaining GP parameters must be estimated numerically.

(Welch et al., 1992) proposed an iterative algorithm to deal with estimation with large sets of input factors, where the likelihood is difficult to optimize. The procedure begins by estimating common parameter values for all factors and then, at each stage, identifying one factor whose parameters will be estimated separately and not in the common pool. The chosen factor is the one that generates the largest improvement in the likelihood.

Most works have generated emulators by estimating the correlation parameters and plugging them into the emulator formulas. That ignores the uncertainty associated with estimated values. (Harari & Steinberg, 2013) and (Pronzato & Rendas, 2017) used a fully Bayesian approach, assigning prior distributions to all model parameters and then averaging the emulator predictions with respect to the posterior. Credible intervals also average over these distributions. Both articles found that the Bayesian estimation improved both point estimates and Bayesian credible intervals.

4.3 Alternative kernel options

The quality of prediction using usual Gaussian process (GP) modeling and kriging depends on the choice of kernel function. One common option is the power exponential family, proposed by Sacks et al. (1989),

$$R(\underline{w}, \underline{x}) = \prod_{j=1}^{d} exp\left(-\theta_j \left|w_j - x_j\right|^{p_j}\right), where\ \theta_j \geq 0\ and\ 0 < p_j \leq 2.$$

The powers p_j are related to the smoothness of the underlying simulator function. The scale factors θ_j are indicative of factor importance. The power exponential family has a sharp break between the case when all $p_j = 2$, (Gaussian correlation) which places weight only on analytic response functions, and smaller exponents, for which, with probability 1, the response function has no derivatives. Most statistical articles use different parameters for each input factor, as above. In the machine learning literature, it is more common to adopt a common parameter for all factors. That strategy sacrifices the ability to distinguish more from less important factors in exchange for the simplicity of having far fewer parameters to estimate, an important consideration with a large input space.

The Matèrn family is a parametric collection of intermediate alternatives with a finite number of derivatives, controlled by the value of the parameter. Another option, the cubic correlation, has finite support, which leads to a sparse kernel matrix and improved computational properties. Details on the Matèrn and cubic correlations can be found in **(**Santner, Williams & Notz 2016, Section 2.2**)**.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

See (Chen, Loeppky, & Welch, 2017) for examples of applications with the Matèrn family. (Kaufman, Bingham, Habib, Heitmann, & Frieman, 2011) also exploited the idea of finite support by ``tapering'' covariance functions from other families.

(Harari & Steinberg, 2013) suggested using a convex combination of two simple processes. The approach promotes robustness by considering simultaneously covariance functions from two different families. Alternatively, the method can be used with different scale parameters, creating a combination of a "tame" and a "volatile" process in order to identify both global and local trends.

Their method is applicable in cases where there is uncertainty regarding the family of correlation functions, or when one wishes to characterize both global trends and finer details, all in the same model. In the first case, the method uses correlation functions from different families, thus allowing for robustness in correlation structure choice. In the second case, it combines two GP with the same correlation function, but with Research findings showed that combining Gaussian processes of different scales could improve both the prediction and coverage properties of the ordinary kriging method for moderate training sets, although tend to have no significant advantage for large.

The formulation thus far has assumed that $Z(\underline{x})$ is a mean 0 stationary process, and so can be described by its variance and its correlation function. Predictors derived from stationary models tend to have a homogeneous behavior across the input space and may have large emulation errors when the underlying function changes its behavior across regions. The model and the BLUP can easily be generalized to cover non-stationary processes, in which case the full covariance function is needed to specify the model. A number of useful proposals have been made.

(Ba & Joseph, 2012) also used a convex combination of processes, but with the expanded option that the mixing fraction could be a function of $\underline{x}$, which induces non-stationarity. (Pronzato & Rendas, 2017) also used a convex combination of processes in their Bayesian Local Kriging method. They made their model non-stationary by adding the assumption that the GP variance associated with a design site could be an increasing function of the distance between the design and prediction site. Consequently, the covariance varies as a function of prediction site. A number of test cases showed that Bayesian Local Kriging can substantially improve emulation when the stationarity assumptions are violated with only a small loss in accuracy when stationarity is satisfied.

(Gramacy & Lee, 2008) and (Taddy, Lee, Gray, & Griffin, 2009) merged GP's with regression trees, using the latter to divide the input space into distinct regimes, with separate GP's fitted in each regime. A Bayesian averaging technique smooths the transition across regime borders.

(Plumlee & Apley, 2017) derived a new covariance function form - lifted Brownian covariance – as a limiting form of a Brownian-like covariance model and then extended it to higher-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

dimensional input domain. The lifted Brownian covariance is non-stationary and requires declaring a particular site in the factor space as the origin; the regression model then has the interpretation of being a model that exactly reflects the function value and derivatives at the origin. They found that the lifted Brownian covariance function showed superior predictive performance than did standard covariance functions like the power exponential or Matèrn.

5. OTHER ANALYSIS OPTIONS

There are also useful methods for modeling data from computer simulators that do not follow the Gaussian process framework. We mention a few of those methods here.

(Joseph& Kang, 2011) proposed the use of regression-based inverse distance weighting (IDW). This is a simple data interpolation method that combines an overall linear regression model with inverse distance weighting of the residuals. Including the trend is helpful for improving the fit for most problems. Their distance function involves multipliers for each factor, with the multiplier estimated from the data to reflect the importance of the factor in modeling the residuals about the regression trend. They also developed a method for assessing uncertainty in the fit.

(Storlie, Bondell, Reich, & Zhang, 2011) proposed the adaptive COSSO (ACOSSO) approach to nonparametric regression. Although their framework was to fit data observed with random error, the flexibility of the modeling makes this method appropriate to simulator data, as well. Their model applies ideas from smoothing spline ANOVA (Wahba, Wang, Gu, Klein, & Klein, 1995)) and the Component Selective Shrinkage Operator (COSSO) (Lin & Zhang, 2006). The model adapts the degree of selection and shrinkage to the level of variation shown in the response data.

The idea behind ACOSSO is to exploit an assumption that the response function is relatively smooth (which is realistic for many problems). The simplest model assumes that the response is an additive function of the inputs,

$$Y(\underline{x}_i) = \sum f_j(x_{ij}) + \varepsilon_i$$

where f_j is a smooth function that depends only on the jth input factor and ε_i is an error term. The standard assumption in ACOSSO is that ε_i represents random variation that has no correlation structure, by contrast to the GP models, in which the error process is used to describe the relationship to the inputs. At the next level of complexity, ACOSSO adds two-factor interaction functions, so that

$$Y(\underline{x}_i) = \sum f_j(x_{ij}) + \sum f_{jk}(x_{ij}, x_{ik}) + \varepsilon_i,$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

with appropriate assumptions about the orthogonality of the functions. Clearly we could continue to add higher-order interaction functions. The estimation criterion combines squared error lack-of-fit with a lasso penalty that sums over all the functional components, so that some components are estimated as 0.

(Sung, Wang, Plumlee, & Haaland, 2019) considered problems with many inputs and a large sample, where Gaussian process models become computationally difficult. They suggested a multi-resolution functional ANOVA model similar to that in ACOSSO, but with coefficients estimated by the group lasso (Yuan & Lin, 2006) , which again limits the number of terms included, and emphasizes that some terms should be included/excluded as a group.

6. PROBLEMS WITH QUALITATIVE AND QUANTITATIVE INPUTS

The GP model defined in section 1 was proposed for settings where input factors are all quantitative, with correlations decreasing functions of distance. Many experiments also have some qualitative inputs and adjustments are needed to handle these variables. GP models face two main challenges: constructing a proper covariance structure for the qualitative factors and specifying the relationship between the correlation function for qualitative factors and the correlation function for quantitative factors.

Some of the initial approaches offer a partial solution. One option is to use a correlation function that satisfies some special form (Joseph, & Delaney, 2007) for qualitative factors. This can simplify the computational complexity for model estimation, but lacks the flexibility to provide a general correlation structure for qualitative factors.

(Qian, Wu & Wu, 2008) developed a general framework with a flexible correlation structure expressed as the product of two functions, one for continuous inputs, the other for categorical inputs. The model is capable of accommodating complex interaction effects between qualitative and quantitative factors and proved successful on a number of test cases. (Zhou, Qian, & Zhou, 2011) introduced efficient methods for estimating these models by taking advantage of a hypersphere transformation of the parameters for the categorical terms. The transformation does not sacrifice generality and provides a simple way to guarantee that the covariance matrix for the qualitative effects is positive definite. (Deng, Lin, Liu, & Rowe, 2017) proposed an alternative model with an additive covariance structure for the qualitative factors, whereas that in Qian, Wu and Wu (2008) is multiplicative.

(Zhang & Notz, 2015) gave an excellent summary of ideas for using GP predictors on this problem. They proposed using indicator functions to derive a covariance model for the qualitative factors and compared several alternative ways to parameterize the GP model, with good discussion on the pros and cons of each approach.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

A useful comparison was provided by (Swiler et al., 2014), with extensive discussion of concrete examples. They compared the performance of GP models with special correlation structure, including some with a tree-based structure (Gramacy and Lee, 2008), and ACOSSO (Storlie et al. 2011; see Section 5). Based on their suite of test cases, they concluded that the general model of Qian, Wu and Wu and the ACOSSO analysis are both effective methods that can be widely applied.

Designs that facilitate fitting models with qualitative factors usually involve the following two goals:

1. At each unique setting of the qualitative factors, the quantitative factors should have a good design.
2. Summarizing over all settings of the qualitative factors, the marginal design of the quantitative factors should be efficient.

(Qian,& Wu, 2009) first addressed this problem, proposing the idea of "sliced designs". Here each slice of the design can correspond to a different setting of the qualitative factors. (Qian, Peter Z. G., 2012) showed how to effectively generate LHC designs with good slicing properties. (Ba, Myers, & Brenneman, 2015) extended the method to generate maximin sliced LHC designs. (Yang, Lin, Qian, & Lin, 2013) showed how to generate sliced LHC designs with first-order (or second-order) orthogonality of the factors.

7. MODELING LARGE DATA SETS

It is challenging to model large sets of computer experiment data. In particular, the use of the GP model is problematic because of the need to invert the n×n correlation matrix. When n becomes very large, this matrix will tend to be ill-conditioned, leading to numerical instability in the emulator. Several techniques exist for numerically stabilizing kernel based interpolators, including adding a nugget effect (Linkletter, Bingham, Hengartner, Higdon, & Ye, 2006) **(**Santner, Williams & Notz 2016**)**, using compactly supported kernels (Fasshauer, 2007) (Gneiting, 2002), and covariance tapering (Kaufman, Schervish, & Nychka, 2008) and approximating likelihoods. (Stein, Chi, & Welty, 2004) In addition, some specialized methods have been proposed that circumvent the numerical problems.

(Haaland & Qian, 2011) proposed a multi-step procedure, which is easy to use and can substantially improve the overall accuracy in emulation of large-scale computer experiments. Their idea exploits a sequence of GP fits, at increasingly finer scales of resolution. The procedure begins by forming well-spread nested subsets of the data. The first fit interpolates the lowest resolution subset using a wide kernel and residuals from this fit serve as the output data for the next fit, made on the next subset. At each step, the residuals are interpolated using a

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

narrower kernel, which is the key to preserving computational stability. The final emulator is the sum of the prediction models fitted at each step.

(Gramacy & Apley, 2014) take a different approach to avoid fitting a single GP model to the entire data set. To generate $\hat{y}(\underline{x}^*)$, they use only data from a local sub-design tailored to prediction at $\underline{x}^*$. With those data, they fit a simple GP model with a Gaussian covariance kernel (all factors have $p_j = 2$ and a common value of θ). The complexity of the fit derives from its local nature, in both the sub-design and local estimation of θ. The choice of the local sub-design around $\underline{x}^*$ exploits a greedy criterion that sequentially adds single data points so as to reduce the mean square prediction error of $\hat{y}(\underline{x}^*)$, taking into account the MSE conditional on θ and the uncertainty associated with estimating θ. The former tends to favor adding data sites that are close to $\underline{x}^*$ in Euclidean distance; but the estimation error for θ sometimes leads to adding points that are much further away. They recommend choosing the size of the local designs to meet computational or execution time constraints.

(Park & Apley, 2018) also used local GP models to facilitate fitting GP models to large data sets. They partitioned the factor space into multiple small regions and made local fits in each region, adding a clever scheme that ensures that near the borders between regions, the fits and their standard errors will match one another and thus avoid discontinuity problems that affect some other local methods.

According to (Tzeng & Huang, 2018) the use of GP models in computer experiments has also borrowed ideas developed in the context of spatial statistics, where very large data sets are becoming increasingly common. This includes methods like covariance tapering (Furrer, Genton, & Nychka, 2006) , the predictive process method (Banerjee, Gelfand, Finley, & Sang, 2008; Eidsvik, Finley, Banerjee, & Rue, 2012) fixed rank kriging (Cressie & Johannesson, 2008) and a stochastic partial differential equation approach based on a Markov random field representation (Lindgren, Rue, & Lindström, 2011) .

All these methods involve selection of some tuning parameters (including functions), such as tapering range, basis functions, partitions of spatial regions, number of resolution levels, and boundary treatments. Unfortunately, obtaining an appropriate choice is difficult or computationally impossible, so some heuristic methods are usually applied. (Tzeng & Huang, 2018) proposed a completely automated new method - Fixed Rank Kriging, allowing for a greater number of base functions, so that it can be easily applied in practice without worrying about the allocation of base functions.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

8. MULTIPLE RESOLUTION SIMULATORS

Complex computer codes sometimes require substantial run time to produce a result. In these settings there may be great benefit from running the simulator at several different levels of resolution. The intuition is that one can use results from a fast, but coarse, simulator, to predict what would have been found from running a slow, but high-resolution, simulator. In the context of neural simulations, this might mean replacing a simulated system with hundreds of thousands of neurons by one with just tens of thousands. In molecular dynamics, one could run simulations with different numbers of atoms or molecules.

The idea of exploiting multi-resolution codes was first developed by (Kennedy & O'Hagan, 2001). Given codes at two levels of resolution, they modeled the output from the high-level code as a Gaussian process, with mean and variance functions that depend on the output from the coarse code, the distance between the input points and a collection of hyperparameters that must be estimated. Given s levels of code $z_1(.), \ldots, z_s(.)$, at increasing levels of resolution, the output y from the tth level code is modeled as $y = z_t(\underline{x})$, where $z_t(.)$ is a random function indexed by the input vector $\underline{x}$. Kennedy and O'Hagan proposed using an autoregressive model that predicts the code at each level of resolution conditional on the previous one:

$$z_t(\underline{x}) = \rho_{t-1} z_{t-1}(\underline{x}) + \delta_t(\underline{x}), \quad t = 2, \ldots, s$$

where ρ_{t-1} is a regression parameter, $\delta_t(.)$ is independent of $z_{t-1}(.), \ldots, z_1(.)$ and is modeled as a Gaussian Process with mean $h(.)^T \beta_t$, where $h(.)$ is a vector of q regression functions, and a stationary covariance function $c_t(\underline{x}, \underline{x}') = cov(\delta_t(\underline{x}), \delta_t(\underline{x}'))$. The simplest code $z_1(.)$ is assumed to be a Gaussian Process independent of $\delta_t(.)$ too. Their methods using Bayesian prediction and uncertainty analysis are applicable to a variety of multi-level computer codes and particularly useful when the slowest code is very expensive to run.

Further refinements were introduced by (Qian& Wu, 2008) , who allowed for more flexibility in relating the models across levels of resolution. Their approach replaces the autoregressive model that ties low-fidelity data to high-fidelity data by a pair of Gaussian processes, one for location offset and one for re-scaling. This provides much more flexibility than the model of Kennedy and O'Hagan.

(Tuo, Wu, & Yu, 2014) developed some specialized multi-fidelity models using non-stationary Gaussian processes that are especially appropriate for settings where the resolution is determined by the size of a finite element mesh.

The analysis methods for multi-resolution problems have invariably exploited an assumption that the factor settings evaluated with a high-resolution code are also evaluated at all lower levels of resolution, enabling direct comparison of how the results change with level of resolution.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

Consequently, design for multi-resolution modeling has emphasized the use of sets that have this nesting property. (Qian, Tang, & Wu, 2009) used orthogonal arrays to create a sequence of nested LHC designs. (Qian, Ai, & Wu, 2009) proposed an alternative construction method that uses difference matrices. (Qian, 2009) developed a simple construction scheme in which smaller LHC designs could be augmented to form larger LHC designs. (Haaland & Qian, 2010) derived more general classes of nested space-filling designs.

9. CALIBRATION AND TUNING WITH FIELD OR LAB DATA

Computer models often include parameters or tuning constants whose values are not known. For example, a computer model that describes the mean behavior of a process might be known only up to some set of parameters. Then, much as in other statistical models, it is relevant to obtain data that can be used to calibrate (or estimate) these parameters; for an example, see (Eriksson et al., 2019) . An additional feature in the calibration of computer simulators is the belief that the computer model may not be a perfect description of reality, so that some description of bias is needed.

The term calibration sometimes refers specifically to the idea that some parameter values may be site or application-specific, and so are calibrated to local conditions. Pharmacokinetic models, for example, often include some global parameters related to the substance under study and some individual parameters that describe how a person absorbs and removes the substance from the blood stream. In other cases, the simulator may involve some constants that have no physical meaning and serve the purpose of "tuning" the model so that it more accurately reflects real systems.

The calibration problem was first addressed by (Kennedy & O'Hagan, 2001) . They proposed a Bayesian model in which the simulator, when run at the true parameter values, provides the mean values of data. The goal of the calibration is to estimate the parameters by finding values for which the simulator accurately reproduces the observed data. Kennedy and O'Hagan extended this to a calibration method by adding assumptions about the validity of the computer model, its approximate parametric form, and the extent of error in the data. The method can handle problems where the simulator is run at only a sample of possible settings; there is no requirement to obtain simulator data at all input settings that were observed. The assumptions provide the underpinnings of a Bayesian analysis that leads to estimates of the calibration parameters and to posterior distributions that characterize their uncertainty.

(Han, Santner, & Rawlinson, 2009) proposed a method for simultaneously tuning and calibrating. They also used a hierarchical Bayesian model that combines responses from a computer simulation with those from a physical experiment and a discrepancy function that measures the distance between the computer code output and the expected value of an

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

observation in the physical experiment. They argued that the first step of the process should be to identify, for each calibration setting, the value of the tuning parameter that minimizes the squared discrepancy. Once obtained, this estimated value is used in the calibration process.

(Eriksson et al., 2019) also implemented Bayesian parameter calibration. Their approach was computationally intense. As a first step, they used a broad sample of input settings to identify regions in their parameter space that might give a reasonable fit to the data. They used that region as the starting point for an MCMC analysis in the later phases.

The original calibration formulation of (Kennedy & O'Hagan, 2001) explicitly recognizes the presence of model error (i.e. that the simulator will not be a perfect reflection of the actual process) and includes a bias term to account for it. The error function is general and so can be appended to any value of the calibration parameters, leading some recent authors to question the identifiability of those parameters. (Tuo & Wu, 2016) described the problem and proposed to resolve the identifiability by defining the values of the calibration parameters as the minimizers of the L_2 distance from the true expected response function to the simulator. They then presented a method for consistent estimation of the calibration parameters. (Tuo & Wu, 2015) proved additional results, showing that their calibration method is efficient; by comparison least squares calibration is not efficient for this problem.

(Plumlee, 2017) developed a fully Bayesian method for calibration using a similar framework to that of Tuo and Wu. This calls for careful statement of the prior on the model bias that ensures that the calibration parameter is well-defined; Plumlee solved the problem by adopting a prior in which the bias is orthogonal to the gradient of the computer model.

(Dai & Chien, 2018) extended the ideas of Tuo and Wu by considering more general distance functions and not just the L_2 distance to identify the calibration parameters, by proposing a two-step estimation process, first estimating the calibration parameters and only then estimating the bias function. (Dai & Chien, 2018) derived finite sample properties and showed that their method can provide guarantees on predictive mean squared error.

10. BAYESIAN INFERENCE

Bayesian inference has proved useful for many applications in neuroscience, for example (Obrezanova 2008 et al, Geisler 2011, Wolpert et al. 2011, Park et al. 2013).

As we noted in Section 1.4, Bayesian inference summarizes knowledge about a set of model parameters θ via their posterior distribution,

$$\pi^*(\theta) \propto \pi(\theta)L(\theta; y).$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

where $\pi(\theta)$ is a prior distribution and $L(\theta; y)$ is the likelihood function that links the distribution of the observed data to the parameters. The summary is via a probability distribution, so that methods for handling probability functions can be used to make further summaries. For example, the inference for any single component of θ will be based on its marginal posterior distribution.

Although the recipe looks simple, it hides potentially challenging computational issues. In order to work with $\pi^*(\theta)$ one needs to either normalize the right-hand side by integration, to be able to sample from the distribution (as in Markov Chain Monte Carlo methods) or to be able to approximate the distribution (for example by saddle point or variational Bayes schemes). Moreover, just evaluating the right-hand side for a fixed value of θ may be difficult. This will occur, for example, when a demanding simulation model must be run to evaluate the likelihood.

(Joseph, 2012) proposed a creative marriage of the research on computer experiments with the computational challenges of Bayesian inference. Regarding the product of the prior and the likelihood as a complex function that is difficult to approximate, Joseph's DoIT method made the approximation using a GP model. In particular, he advocated the use of the GP framework with a Gaussian kernel. As a result, the approximation itself is a linear combination of Gaussians and thus is trivial to normalize. Joseph showed that DoIT could be very effective for approximating complex posterior distributions. He also developed useful methods for taking initial samples of the parameter space and for adding further parameter vectors to improve the approximation. See (Steinberg & Jones, 2012) for constructive comments on how to improve the method. (Joseph, 2013) added a simple refinement that guaranteed that the approximation would be non-negative.

11. OPTIMIZATION

The success of computer experiment methods in emulating complex functions has also made them valuable tools for black-box optimization problems, in which the number of function evaluations may be limited by time and resource constraints. (Jones, Schonlau, & Welch, 1998)

(Taddy et al., 2009) developed a new approach, which combines asynchronous parallel pattern search (APPS) and treed Gaussian processes (TGP; see section 4.3), with the EI criterion. APPS was proposed by numerical analysts and is based around continued evaluations at sites near the current best site. The two methods effectively complement one another. By combining these two schemes, their algorithm performs robust local optimization more efficiently than when using either method alone.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

The MCMC algorithm is a very popular choice for optimization problems, yet less suitable for sequential design since it must be restarted and iterated to convergence with the inclusion of each new design point. (Gramacy & Polson, 2011) also extended the application of the EI criterion for optimization in the context of GP models. The major contribution in their work is their method for exploring the GP posterior for the objective function. The most common approach, Markov Chain Monte Carlo, is not a good choice for sequential learning because it requires substantial computing after each additional result is obtained. Gramacy and Polson replace MCMC by a Sequential Monte Carlo algorithm that is much faster than MCMC. This facilitates the generation of sequential evaluation sites and is applicable for online updating of Gaussian process regression and classification models.

Complex optimization problems often involve objective functions that are expensive to compute. An effective strategy to control costs is to exploit a much cheaper surrogate model for part of the optimization process. (Yao, Chen, Huang, & van Tooren, 2014) developed a new method for dealing with two common surrogate-based optimization issues. The first one is the accuracy improvement of the surrogate model. The second relates to the augmentation of the training set with an infill strategy that gradually improves surrogate accuracy and ensures convergence to the real global optimum of the exact model. They propose to use a radial basis function neural network (RBFNN) method for the optimization process. They developed a linear interpolation (LI) based RBFNN modelling method, LI-RBFNN, to enhance the accuracy of RBFNN which is used in combination with a hybrid infill strategy. This strategy uses the lower bound of the surrogate prediction as the optimization objective to locate the promising input region and employs a linear interpolation-based sequential sampling approach to improve the surrogate accuracy globally.

(Bischl et al., 2017) added a number of additional features for optimization via GP models. Their algorithm was designed for both single- and multi-objective optimization with mixed continuous, categorical and conditional inputs. Additional features included multi-point batch proposal, parallelization, visualization, logging and error-handling. The algorithm has a modular implementation that makes it easy to swap single components and permits the user to adapt the method for specific use cases. For example, it is easy to include regression trends as part of the GP.

12. MODELING MULTIVARIATE OUTPUT

Our presentation thus far has focused on models for a univariate output. However, computer simulators often generate multivariate output. In the context of neural simulation, this could refer to a complete spike train, a collection of spike trains, or a set of macroscopic features computed from such a collection.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

One class of methods for emulating multivariate output is directed toward functional output. They are related to statistical models for analyzing functional data (Ramsay, 2005). We will express ideas here in terms of functions of time, but they are equally appropriate for functions of neural location or functions jointly of time and location. One approach has been to express the dependence on time via a collection of p time-trend functions, $g_j(t), j = 1, \ldots, p$. Then the coefficients of these functions depend on the input settings x via a GP model. (Bayarri et al., 2009) developed this model with the trend functions forming a basis expansion; (Bayarri et al., 2007) analyzed highly irregular functional data with the help of a wavelet expansion. (Higdon, Gattiker, Williams, & Rightley, 2008) determined the trend functions via principal component analysis and (Levy, 2008) derived a data-driven method to generate a good set of trend functions. There have been many further works on such models. See, for example, (Goh et al., 2013) , (Hung, Joseph, & Melkote, 2015) , (Plumlee, Joseph & Yang, 2016) , (Chakraborty et al., 2017) , (Salter, Williamson, Scinocca, & Kharin, 2019) .

There has been less attention to the case of multivariate output that is not functional in nature. However, that case seems especially relevant for neural simulations, where attention often focuses on a set of features extracted from complex output, rather than on the functional output itself (e.g. the spike trains for a region of the brain). Even without the functional structure, there will likely be correlations among the outputs that can be exploited in modeling or emulating them. (Rougier, 2008) proposed to model all the outputs in a joint GP model. His approach was made very tractable by an assumption that the overall correlation of the GP could be separated into two distinct, and independent, components, one related to the difference in input factor settings and the other to the outputs. That leads to a Kronecker product form for the correlation matrix which has significant computational advantages.

(Overstall & Woods, 2016) presented a common Bayesian framework for emulation of multivariate simulators using covariance separable Gaussian processes. Their analysis takes advantage of the matrix normal distribution with a covariance that can be expressed as a Kronecker product. They also take a fully Bayesian approach to inference.

The separable structure imposes restrictions that may not always be appropriate, especially when the outputs are related in rather different ways to the inputs. (Fricker, Oakley, & Urban, 2013) developed an emulation scheme for such settings, taking advantage of convolution methods and the theory of coregionalization that has been used in spatial statistics.

13. UNCERTAINTY ASSESSMENT

Uncertainty quantification deals with the quantitative characterization and reduction of uncertainties. Generically, consider the output $Y(\underline{x})$ of a computer simulator, which is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

dependent on the input factors x. Any uncertainty associated with the values of the input factors translates into uncertainty about the value of Y. Challenges that naturally arise are to characterize the nature of that uncertainty and to assess which of the input factors play dominant roles in it. The latter issue is especially important as a guide to future research, indicating what sort of data might be most useful to reduce the degree of uncertainty about Y.

The notion of uncertainty assessment is closely linked to that of sensitivity analysis. It is instructive here to distinguish between so-called local and global sensitivity analysis (GSA). The former is a mainstay of engineering analysis and is termed local because it focuses on the sensitivity of $Y(\underline{x})$ at a particular input setting, say $\underline{x}^*$, to changes in the input settings. Such analyses usually emphasize finding the largest partial derivatives of the function at $\underline{x}^*$. By contrast, GSA looks at the variation of $Y(\underline{x})$ over a large region that is generated by a probability distribution on the inputs. Our interest will be in GSA. See (Saltelli et al. 2008) for a detailed presentation of ideas and methods and (Marino, Hogue, Ray, & Kirschner, 2008) for an account of their application in systems biology.

The study by (Eriksson et al., 2019) that we cited earlier is a good illustration. The study was concerned with dynamical behavior in intracellular models and employed a simulator that depends on a number of unknown parameters, with substantial uncertainty as to their values. Experimental data provided good estimates of some of the parameters, substantially reducing their uncertainty. An important question was to then assess how precisely one could predict the relationship between the active form of a kinase and the active form of a phosphatase and how this relationship depends on certain calcium transients. The authors used GSA to address these questions, using the posterior distribution of the model parameters as the relevant probability distribution on the input space.

Traditional methods for uncertainty quantification and for GSA rely on sampling strategies to probe the input space and analytical summaries of the resulting function values. Thus, there is much common ground with the methods used in computer experiments.

The probabilistic approach offers an advantage of providing the degree of uncertainty associated with the model throughout the whole space. (Salem, Roustant, Gamboa, & Tomaso, 2017) offer a universal method, called Universal Prediction distribution (UP distribution) to define a measure of uncertainty for any surrogate model either deterministic or probabilistic. Their method based on Cross-Validation (CV) sub-models predictions. It also provides a prediction for uncertainty distribution, which is applicable in much more general frames than the Gaussian one, thus allowing a large set of sampling criteria.

Well-known estimation methods of the GP parameters can be classified into two categories based on the Bayesian paradigm. The first is the empirical Bayes approach, using maximum probability estimates (MLE). The second category is a Bayesian posterior as suggested by (Higdon et al., 2008) and the GP method used by (Gramacy & Lee, 2010).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

(Chen et al., 2017) introduced a new approach dealing with uncertainty quantification, which derives from the uncertainty from the statistical emulator of the computer model, including the contribution from parameter estimation. They state that Bayesian methods in principle take into account the parameters' uncertainty and produce better coverage probabilities for reliable intervals. They demonstrated that choosing which category to use is less important for uncertainty quantification than choosing the correlation-function family.

(Iooss & Ribatet, 2009) extended the ideas in GSA to handle problems in which the inputs are themselves functional. Their ideas can be used to assess factor importance when one of the input factors varies over time, such as the time course for providing an electrical stimulus to the brain.

14. TOOLS

In this section, we review tools that address many of the challenges mentioned in this paper, such as modeling the Gaussian processes, the development of emulators for computer simulations, optimization and parameter calibration.

14.1 Gaussian process modeling based on Bayesian approach

laGP is an R package (Gramacy, 2016) that provides approximate GP regression for extensive computer experiments and spatial databases. The approximation method finds small local designs (independently) to predict specific inputs. It can be used for black box constrained optimization via augmented Lagrangians and for computer model calibration via optimization. It implements the following features: ALC, MSPE and NN-based local approximation, EFI-based global heuristics, local MLE/MAP inference for (isotropic and separable) length-scales and nuggets and supplies lower-level (full) GP inference and prediction. It also supports a large number of parallel computation methods and their API's like OpenMP (Open Multi-Processing) for approximation over a large out-of-sample testing set, GPU acceleration for local average length criterion, subroutine evaluations and Simple Network of Workstations (SNOW) parallel-package cluster parallelization.

The **tgp** package (Gramacy, 2007; Gramacy & Lee, 2008) is designed for building surrogates of both stationary and nonstationary noisy simulators. It implements Bayesian techniques like the Metropolis-Hastings algorithm using a GP model for emulating the stationary components of the process. The GP model includes a nugget parameter estimated along with other parameters. The recent version of the tgp package facilitates the emulation of deterministic simulators by removing the nugget parameter from the model, supports designs of experiments and includes one-dimensional and two-dimensional plots, which includes tree drawing functions and higher dimension projection and slice options.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

Plgp - R package developed by Gramacy, implements particle learning (PL) according to principles described in (Carvalho, Johannes, Lopes, & Polson, 2010) (Gramacy & Apley, 2014) (Gramacy & Lee, 2010) . It facilitates Sequential Monte Carlo inference for Bayesian online updating of Gaussian process regression and classification by particle learning (PL). It includes the following features: sequential design for optimization of functions by expected improvement (EI), sequential design for optimization under known and unknown constraints by an integrated expected conditional improvement (IECI) criterion using a hybrid regression-classification GP model, sequential design for exploring classification boundaries by the predictive entropy statistic via a classification GP model. It has a generic PL interface and supplies three types of correlation functions: isotropic, separable, and single-index Gaussian. (Snoek, Larochelle, & Adams, 2012) presented new methods to optimize the hyperparameters used in machine learning algorithms. They have developed a fully Bayesian approach for the expected improvement and algorithms for dealing with variable time regimes and parallelized experiments.

rBayesianOptimization R package by Yachen Yan implements Bayesian Global Optimization with Gaussian Processes algorithm, as was described in (Snoek, Larochelle, & Adams, 2012).

mlrMBO is an additional R package that implements the effective global optimization algorithm and designed to deal with mixed, continuous, categorical, and adaptive optimization parameters, developed by Bischl et al. (2017). The developers used the modular form so that the components replaced or easily adapted by the user.

DiceKriging R package, developed by (Roustant, Ginsbourger, & Deville, 2012), performs estimation, simulation, prediction and validation for various Kriging models. It implements covariance parameter estimation with respect to noise and trend specifications in efficient and fast manner. The package also, has the option to define conditional and unconditional simulations. DiceOptim's R package, also developed by(Roustant et al., 2012), is a complementary package for DiceKriging that performs sequential and parallel Kriging-based optimization, based on the 1-point and multi points Expected Improvement Criteria (EI) .

14.2 Gaussian process modeling using MLE (Maximum Likelihood Estimation)

(Dancik & Dorman, 2008) developed **mlegp** R package, which suitable for modeling Gaussian processes for univariate and multi-dimensional outputs by estimating the maximum likelihood function. The multi-dimensional output can be modeled by fitting independent GPs to each output. The package implements plotting of main effects for functional output. It also supports different correlation structures like product exponential correlation structure, constant or linear regression mean function and allows defining nugget terms like constant nugget term, nugget matrix up to a multiplicative constant or no nugget term at all. The ability to specify the nugget matrix allows some flexibility for using GPs to model heteroscedastic responses. In addition, there is a special version for sensitivity analysis, including functional analysis of Variance

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 August 2019 doi:10.20944/preprints201908.0246.v1

(FANOVA) decomposition, plotting functions to achieve diagnostic plots, main effects, and two-way factor interactions.

GPfit: R package for fitting a Gaussian process model to deterministic simulator outputs (Macdonald, Ranjan, & Chipman, 2015) . Fitting a GP model can be numerically unstable if any pair of design points in the input space are close together. To overcome this problem, (Ranjan, Haynes, & Karsten, 2011) presented the use of a genetic algorithm for maximizing the likelihood, which was robust and numerically stable, but computationally intensive. This package offers an improved method for using a genetic algorithm to maximize the likelihood. It implements a different parameterization of the spatial correlation function and a clustering-based multi-start gradient-based optimization algorithm, which yields robust and faster optimization.

14.3 Alternative surrogate modeling

The SUMO toolbox,(Gorissen, Couckuyt, Demeester, Dhaene, & Crombecq, 2010) written in Matlab and Java , comprises a complete cross-platform, which comes with a large (60+) number of sample problems. It is a flexible framework for accurate global surrogate modeling and adaptive sampling (active learning). It features a rich set of plugins and applies to a wide variety of domains, in an autonomous fashion, a black box, or under full manual control.

DoIt This R-package implements the design of experiments based on the interpolation technique (DoIt, Joseph 2012) for approximate Bayesian estimations of the joint and marginal densities. The method uses estimates of unnormalized density in the space-filling design of parameter values. Normalization is achieved by approximating the target density by a weighted sum of Gaussian kernels centered on the design points. The package contains functions for the optimal selection of additional design points and calculates the optimal kernel width by cross-validation.

Acknowledgements: This work was supported by a grant from the European Commission supporting the European Human Brain Project.

References

Ba, S., & Joseph, V. R. (2012). Composite Gaussian process models for emulating expensive functions. *The Annals of Applied Statistics, 6*(4), 1838-1860. doi:10.1214/12-AOAS570

Ba, S., Myers, W. R., & Brenneman, W. A. (2015). Optimal sliced Latin hypercube designs. *Technometrics, 57*(4), 479-487.

Banerjee, S., Gelfand, A. E., Finley, A. O., & Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70*(4), 825-848. doi:10.1111/j.1467-9868.2008.00663.x

Bayarri, M. J., Berger, J. O., Kennedy, M. C., Kottas, A., Paulo, R., Sacks, J., Tu, J. (2009). Predicting vehicle crashworthiness: Validation of computer models for functional and hierarchical data. *Journal of the American Statistical Association, 104*(487), 929-943.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Tu, J. (2007). A framework for validation of computer models. *Technometrics, 49*(2), 138-154.

Bischl, B., Richter, J., Bossek, J., Horn, D., Thomas, J., & Lang, M. (2017). mlrMBO: A modular framework for model-based optimization of expensive black-box functions. *arXiv preprint arXiv:1703.03373*.

Carvalho, C. M., Johannes, M. S., Lopes, H. F., & Polson, N. G. (2010). Particle learning and smoothing. *Statistical Science, 25*(1), 88-106. doi:10.1214/10-STS325

Chakraborty, A., Bingham, D., Dhavala, S. S., Kuranz, C. C., Drake, R. P., Grosskopf, M. J., . . . McClarren, R. G. (2017). Emulation of numerical models with over-specified basis functions. *Technometrics, 59*(2), 153-164.

Chen, H., Loeppky, J. L., & Welch, W. J. (2017). Flexible correlation structure for accurate prediction and uncertainty quantification in bayesian gaussian process emulation of a computer model. *SIAM/ASA Journal on Uncertainty Quantification, 5*(1), 598-620. doi:10.1137/15M1008774

Cressie, N., & Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society Series B-Statistical Methodology; 70*, 209-226.

Cunningham, J. P., Byron, M. Y., Shenoy, K. V., & Sahani, M. (2008). Inferring neural firing rates from spike trains using gaussian processes. Paper presented at the *Advances in Neural Information Processing Systems*, 329-336.

Dai, X., & Chien, P. (2018). Another look at statistical calibration: A non-asymptotic theory and prediction-oriented optimality. *arXiv Preprint arXiv:1802.00021*,

Dancik, G. M., & Dorman, K. S. (2008). Mlegp: Statistical analysis for computer models of biological systems using R. *Bioinformatics (Oxford, England), 24*(17), 1966. doi:10.1093/bioinformatics/btn329

De La Pava, I., Gómez, V., Álvarez, M. A., Henao, Ó A., Daza-Santacoloma, G., & Orozco, Á A. (2015). A gaussian process emulator for estimating the volume of tissue activated during deep brain stimulation. Paper presented at the *Iberian Conference on Pattern Recognition and Image Analysis*, 691-699.

Deng, X., Lin, C. D., Liu, K., & Rowe, R. K. (2017). Additive gaussian process for computer models with qualitative and quantitative factors. *Technometrics, 59*(3), 283-292. doi:10.1080/00401706.2016.1211554

Eidsvik, J., Finley, A. O., Banerjee, S., & Rue, H. (2012). Approximate bayesian inference for large spatial datasets using predictive process models. *Computational Statistics and Data Analysis, 56*(6), 1362-1380. doi:10.1016/j.csda.2011.10.022

Eriksson, O., Jauhiainen, A., Maad Sasane, S., Kramer, A., Nair, A. G., Sartorius, C., Wren, J. (2019). Uncertainty quantification, propagation and characterization by bayesian analysis combined with global sensitivity analysis applied to dynamical intracellular pathway models. *Bioinformatics, 35*(2), 284-292. doi:10.1093/bioinformatics/bty607

Fasshauer, G. E. (2007). *Meshfree approximation methods with MATLAB*. Singapore; Hackensack, N.J.: World Scientific.

Fricker, T. E., Oakley, J. E., & Urban, N. M. (2013). Multivariate gaussian process emulators with nonseparable covariance structures. *Technometrics, 55*(1), 47-56.

Furrer, R., Genton, M. G., & Nychka, D. (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics, 15*(3), 502-523. doi:10.1198/106186006X132178

Geisler, W. S. (2011). Contributions of ideal observer theory to vision research. *Vision Research, 51*(7), 771-781.

Ghahramani, Z. (2013). Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society A, 371*(1984) doi:10.1098/rsta.2011.0553

Gneiting, T. (2002). Compactly supported correlation functions. *Journal of Multivariate Analysis, 83*(2), 493-508. doi:10.1006/jmva.2001.2056

Goh, J., Bingham, D., Holloway, J. P., Grosskopf, M. J., Kuranz, C. C., & Rutter, E. (2013). Prediction and computer model calibration using outputs from multifidelity simulators. *Technometrics, 55*(4), 501-512.

Gorissen, D., Couckuyt, I., Demeester, P., Dhaene, T., & Crombecq, K. (2010). A surrogate modeling and adaptive sampling toolbox for computer based design. *Journal of Machine Learning Research 2010*

Gramacy, R. B. (2007). Tgp: An R package for bayesian nonstationary, semiparametric nonlinear regression and design by treed gaussian process models. *Journal of Statistical Software, 19*(9) doi:10.18637/jss.v019.i09

Gramacy, R. B. (2016). laGP: Large-scale spatial modeling via local approximate gaussian processes in R. *Journal of Statistical Software, 72*(1), 1-46. doi:10.18637/jss.v072.i01

Gramacy, R. B., & Apley, D. W. (2014). Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics, 24*(2), 1-28. doi:10.1080/10618600.2014.914442

Gramacy, R. B., & Lee, H. K. H. (2008). Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association, 103*(483), 1119-1130. doi:10.1198/016214508000000689

Gramacy, R. B., & Lee, H. K. H. (2010). Optimization under unknown constraints.

Gramacy, R. B., & Polson, N. G. (2011). Particle learning of gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics, 20*(1), 102-118. doi:10.1198/jcgs.2010.09171

Haaland, B., & Qian, P. Z. (2011). Accurate emulators for large-scale computer experiments. *The Annals of Statistics*, *39*(6), 2974-3002. doi:10.1214/11-AOS929

Haaland, B., & Qian, P. Z. (2010). An approach to constructing nested space-filling designs for multi-fidelity computer experiments. *Statistica Sinica, 20*(3), 1063.

Han, G., Santner, T. J., & Rawlinson, J. J. (2009). Simultaneous determination of tuning and calibration parameters for computer experiments. *Technometrics, 51*(4), 464-474. doi:10.1198/TECH.2009.08126

Harari, O., & Steinberg, D. M. (2013). Convex combination of gaussian processes for bayesian analysis of deterministic computer experiments. *Technometrics, 56*(4) doi:10.1080/00401706.2013.861629

Higdon, D., Gattiker, J., Williams, B., & Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association, 103*(482), 570-583.

Hung, Y., Joseph, V. R., & Melkote, S. N. (2015). Analysis of computer experiments with functional response. *Technometrics, 57*(1), 35-44.

Johnson, M. E., Moore, L. M., & Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference, 26*(2), 131-148.

Jones, D., Schonlau, M., & Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization, 13*(4), 455-492. doi:1008306431147

Joseph, V. R. (2012). Bayesian computation using design of experiments-based interpolation technique. *Technometrics, 54*(3), 209-225. doi:10.1080/00401706.2012.680399

Joseph, V. R. (2013). A note on nonnegative DoIt approximation. *Technometrics, 55*(1), 103-107. doi:10.1080/00401706.2012.759154

Joseph, V. R., & Delaney, J. D. (2007). Functionally induced priors for the analysis of experiments. *Technometrics, 49*(1), 1-11. doi:10.1198/004017006000000372

Joseph, V. R., Hung, Y., & Sudjianto, A. (2008). Blind kriging: A new method for developing metamodels. *Journal of Mechanical Design, Transactions of the ASME, 130*(3). doi:10.1115/1.2829873

Joseph, V. R., & Kang, L. (2011). Regression-based inverse distance weighting with applications to computer experiments. *Technometrics, 53*(3), 254-265. doi:10.1198/TECH.2011.09154

Joseph, V., & Melkote, S. (2009). Statistical adjustments to engineering models. *Journal of Quality Technology, 41*(4), 362-375. doi:10.1080/00224065.2009.11917791

Kaufman, C. G., Bingham, D., Habib, S., Heitmann, K., & Frieman, J. A. (2011). Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology. *The Annals of Applied Statistics, 5*(4), 2470-2492.

Kaufman, C. G., Schervish, M. J., & Nychka, D. W. (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association, 103*(484), 1545-1555. doi:10.1198/016214508000000959

Kennedy, M. C., & O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society Series B-Statistical Methodology;*

Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy, 52*(6), 119-139.

Levy, S. (2008). The analysis of time dependent computer experiments. *Unpublished Ph.D. dissertation, Tel-Aviv University*

Levy, S., & Steinberg, D.M. (2010). Computer experiments: A review. *Berlin/Heidelberg: doi:10.1007/s10182-010-0147-9*

Lin, Y., & Zhang, H. (2006). Component selection and smoothing in multivariate nonparametric regression. *Annals of Statistics, 34*(5), 2272-2297. doi:10.1214/009053606000000722

Lindgren, F., Rue, H., & Lindström, J. (2011). An explicit link between gaussian fields and gaussian markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73*(4), 423-498. doi:10.1111/j.1467-9868.2011.00777.x

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., & Ye, K. Q. (2006). Variable selection for gaussian process models in computer experiments. *Technometrics, 48*(4), 478-490. doi:10.1198/004017006000000228

Macdonald, B., Ranjan, P., & Chipman, H. (2015). GPfit: An R package for fitting a gaussian process model to deterministic simulator outputs. *Journal of Statistical Software, 64*(1), 1-23. doi:10.18637/jss.v064.i12

Marino, S., Hogue, I. B., Ray, C. J., & Kirschner, D. E. (2008). A methodology for performing global uncertainty and sensitivity analysis in systems biology. *Journal of Theoretical Biology, 254*(1), 178-196. doi:10.1016/j.jtbi.2008.04.011

Matthews, Alexander G de G, Rowland, M., Hron, J., Turner, R. E., & Ghahramani, Z. (2018). Gaussian process behaviour in wide deep neural networks. *arXiv Preprint arXiv:1804.11271,*

Mckay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics, 21*(2), 239-245. doi:10.1080/00401706.1979.10489755

Melozzi, F., Woodman, M. M., Jirsa, V. K., & Bernard, C. (2017). The virtual mouse brain: A computational neuroinformatics platform to study whole mouse brain dynamics. *eNeuro, 4*(3) doi:10.1523/ENEURO.0111-17.2017

Neal, R. M. (1996). *Bayesian learning for neural networks*. New York: Springer.

Obrezanova, O., Gola, J., Champness, E., & Segall, M. (2008). Automatic QSAR modeling of ADME properties: Blood–brain barrier penetration and aqueous solubility. *Journal of Computer-Aided Molecular Design; Incorporating Perspectives in Drug Discovery and Design, 22*(6), 431-440. doi:10.1007/s10822-008-9193-8

Overstall, A. M., & Woods, D. C. (2016). Multivariate emulation of computer simulators: Model selection and diagnostics with application to a humanitarian relief model. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 65*(4), 483-505. doi:10.1111/rssc.12141

Papamakarios, G., Sterratt, D. C., & Murray, I. (2018). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. *arXiv preprint arXiv:1805.07226*.

Park, C., & Apley, D. (2018). Patchwork kriging for large-scale gaussian process regression. *Journal of Machine Learning Research, 19*, 1-43.

Park, M., Weller, J. P., Horwitz, G. D., & Pillow, J. W. (2014). Bayesian active learning of neural firing rate maps with transformed gaussian process priors. *Neural Computation, 26*(8), 1519-1541.

Plumlee, M. (2017). Bayesian calibration of inexact computer models. *Journal of the American Statistical Association, 112*(519), 1274-1285. doi:10.1080/01621459.2016.1211016

Plumlee, M., & Apley, D. W. (2017). Lifted brownian kriging models. *Technometrics, 59*(2), 165-177. doi:10.1080/00401706.2016.1211555

Plumlee, M., Joseph, V. R., & Yang, H. (2016). Calibrating functional parameters in the ion channel models of cardiac cells. *Journal of the American Statistical Association, 111*(514), 500-509. doi:10.1080/01621459.2015.1119695

Pronzato, L., & Rendas, M. (2017). Bayesian local kriging. *Technometrics, 59*(3), 293-304. doi:10.1080/00401706.2016.1214179

Qian, P. Z. G. (2009). Nested latin hypercube designs. *Biometrika, 96*(4), 957-970. doi:10.1093/biomet/asp045

Qian, P. Z. G. (2012). Sliced latin hypercube designs. *Journal of the American Statistical Association, 107*(497), 393-399. doi:10.1080/01621459.2011.644132

Qian, P. Z. G., & Wu, C. F. J. (2008). Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics, 50*(2), 192-204. doi:10.1198/004017008000000082

Qian, P. Z. G., & Wu, C. F. J. (2009). Sliced space-filling designs. *Biometrika, 96*(4), 945-956. doi:10.1093/biomet/asp044

Qian, P. Z. G., Wu, H., & Wu, C. F. J. (2008). Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics, 50*(3), 383-396. doi:10.1198/004017008000000262

Qian, P., Ai, M., & Wu, C. (2009). Construction of nested space-filling designs. *Annals of Statistics, 37*(6), 3616. doi:10.1214/09-AOS690

Qian, P., Tang, B., & Wu, C. (2009). Nested space-filling designs for computer experiments with two levels of accuracy. *Statistica Sinica, 19*(1), 287-300.

Raiteri, P., Laio, A., Gervasio, F. L., Micheletti, C., & Parrinello, M. (2006). Efficient reconstruction of complex free energy landscapes by multiple walkers metadynamics. *The Journal of Physical Chemistry.B, 110*(8), 3533. doi:10.1021/jp054359r

Ramsay, J.O., Silverman, B.W. (2005). Functional Data Analysis, *2nd edn. Springer, New York*

Ranjan, P., Haynes, R., & Karsten, R. (2011). A computationally stable approach to gaussian process interpolation of deterministic computer simulation data. *Technometrics, 53*(4), 366-378. doi:10.1198/TECH.2011.09141

Rasmussen, C. E., & Williams C. K. I. (2006). Gaussian processes for machine learning *(Vol. 2, No. 3, p. 4). Cambridge, MA: MIT press.*

Rougier, J. (2008). Efficient emulators for multivariate deterministic functions. *Journal of Computational and Graphical Statistics, 17*(4), 827-843. doi:10.1198/106186008X384032

Roustant, O., Ginsbourger, D., & 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) doi:10.18637/jss.v051.i01

Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science, 4*(4), 409-423. doi:10.1214/ss/1177012413

Salem, M. B., Roustant, O., Gamboa, F., & Tomaso, L. (2017). Universal prediction distribution for surrogate models. *SIAM/ASA Journal on Uncertainty Quantification*, *5*(1), 1086-1109. doi:10.1137/15M1053529

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S. (2008). *Global Sensitivity Analysis*. The Primer. Wiley, New York

Salter, J. M., Williamson, D. B., Scinocca, J., & Kharin, V. (2019). Uncertainty quantification for computer models with spatial output using calibration-optimal bases. *Journal of the American Statistical Association*, 1-24. doi:10.1080/01621459.2018.1514306

Santner, T. J., Williams, B. J., & Notz, W. I. (2016). The Design and Analysis of Computer Experiments.

Stein, M., Chi, Z., & Welty, L. (2004). Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society Series B-Statistical Methodology;*

Steinberg, D. M., & Bursztyn, D. (2004). Data analytic tools for understanding random field regression models. *Technometrics, 46*(4), 411-420. doi:10.1198/004017004000000419

Steinberg, D. M., & Jones, B. (2012). Comment: DoIt—Some thoughts on how to do it. *Technometrics, 54*(3), 236-238. doi:10.1080/00401706.2012.697247

Steinberg, D. M., & Lin, D. K. J. (2006). A construction method for orthogonal latin hypercube designs. *Biometrika, 93*(2), 279-288. doi:10.1093/biomet/93.2.279

Storlie, C., Bondell, H. D., Reich, B., & Zhang, H. (2011). Surface estimation, variable selection, and the nonparametric oracle property. *Statistica Sinica*. *21*(2), 679-705. doi:10.5705/ss.2011.030a

Sung, C., Wang, W., Plumlee, M., & Haaland, B. (2019). Multi-resolution functional ANOVA for large-scale, many-input computer experiments. *Journal of the American Statistical Association*, 1-32. doi:10.1080/01621459.2019.1595630

Swiler, L. P., Hough, P. D., Qian, P., Xu, X., Storlie, C., & Lee, H. (2014). Surrogate models for mixed discrete-continuous variables. *Constraint programming and decision making* (pp. 181-202) Springer.

Taddy, M. A., Lee, H. K. H., Gray, G. A., & Griffin, J. D. (2009). Bayesian guided pattern search for robust local optimization. *Technometrics, 51*(4), 389-401. doi:10.1198/TECH.2009.08007

Tuo, R., & Jeff Wu, C. F. (2016). A theoretical framework for calibration in computer models: Parametrization, estimation and convergence properties. *SIAM/ASA Journal on Uncertainty Quantification, 4*(1), 767-795. doi:10.1137/151005841

Tuo, R., & Wu, C. (2015). Efficient calibration for imperfect computer models. *Annals of Statistics, 43*(6), 2331. doi:10.1214/15-AOS1314

Tuo, R., Wu, C. F. J., & Yu, D. (2014). Surrogate modeling of computer experiments with different mesh densities. *Technometrics, 56*(3), 372-380. oi:10.1080/00401706.2013.842935

Tzeng, S., & Huang, H. (2018). Resolution adaptive fixed rank kriging. *Technometrics, 60*(2), 198-208. doi:10.1080/00401706.2017.1345701

Wahba, G., Wang, Y., Gu, C., Klein, R., & Klein, B. (1995). Smoothing spline anova for exponential families, with application to the wisconsin epidemiological study of diabetic retinopathy. *The Annals of Statistics, 23*(6), 1865-1895. doi:10.1214/aos/1034713638

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., & Morris, M. D. (1992). Screening, predicting, and computer experiments. *Technometrics, 34*(1), 15-25. doi:10.1080/00401706.1992.10485229

Wolpert, D. M., Diedrichsen, J., & Flanagan, J. R. (2011). Principles of sensorimotor learning. *Nature Reviews.Neuroscience, 12*(12), 739-751. doi:10.1038/nrn3112 [doi]

Yang, J., Lin, C. D., Qian, P. Z. G., & Lin, D. K. J. (2013). Construction of sliced orthogonal latin hypercube designs. *Statistica Sinica, 23*(3), 1117-1130. Retrieved from http://www.jstor.org/stable/24310788

Yao, W., Chen, X. Q., Huang, Y. Y., & van Tooren, M. (2014). A surrogate-based optimization method with RBF neural network enhanced by linear interpolation and hybrid infill strategy. *Optimization Methods and Software, 29*(2), 406-429. doi:10.1080/10556788.2013.777722

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68*(1), 49-67. doi:10.1111/j.1467-9868.2005.00532.x

Zhang, Y., & Notz, W. I. (2015). Computer experiments with qualitative and quantitative variables: A review and reexamination. *Quality Engineering, 27*(1), 2-13. doi:10.1080/08982112.2015.968039

Zhou, Q., Qian, P. Z. G., & Zhou, S. (2011). A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics, 53*(3), 266-273. doi:10.1198/TECH.2011.10025