Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

METAMODELING A SYSTEM DYNAMICS MODEL: A CONTEMPORARY COMPARISON OF METHODS

Rodrigo De la Fuente

Department of Industrial Engineering
University of Concepción
Edmundo Larenas 219
Concepción, CHILE

Raymond Smith III

Department of Engineering
East Carolina University
222 Slay Building
Greenville, NC 27858, USA

ABSTRACT

Advancements in computer technology have resulted in improved computing capability and software functionality. Concurrently, in the simulation community demand to study complex, integrated systems has grown. As a result, it is difficult to perform model exploration or optimization simply due to time and resource limitations. Metamodeling offers an approach to overcome this issue; however, limited study has been made to compare the methods most appropriate for simulation modeling. This paper presents a contemporary comparison of methods useful for creating a metamodel of a simulation model. For comparison we explore the performance of a complex system dynamics model of a community hospital. In our view several characteristics of hospital operations present an interesting challenge to explore and compare the well-known competing methods. We consider three dimensions in our comparison: fit quality, fitting time, and results interpretability. The paper discusses the better performing methods corresponding to these dimensions and considers tradeoffs.

1 INTRODUCTION

Although the computational power available today to run simulations is larger than several years ago, the complexity of simulation models and the systems they represent have also increased. This effect makes it difficult to perform either sensitivity analysis or optimization using end-user computing resources. In order to overcome this obstacle, a metamodel may be used to provide faster execution and/or better comprehension of the simulated system. Specifically, a metamodel could be used for performing (1) model approximation, (2) model exploration, (3) problem formulation, and (4) global or multi-objective optimization (Wang and Shan 2007). Acknowledging the important role that metamodels contribute to the field of simulation, our work seeks to address the following objectives: (1) Provide a methodological procedure to compare metamodels using hyper-parameter optimization and cross-validation; (2) Contribute to the field of simulation with an application of metamodeling for a system dynamics model representing a very complex system; and, (3) Evaluate the usefulness of the most used metamodels to gain insight of the simulated system.

2 LITERATURE REVIEW

We identify in the literature the most applied techniques used to simulate engineering problems and present a comparison of the corresponding metamodeling methods. For discussion, we organize the methods identified as being either a Gaussian process or a non-Gaussian process.

978-1-5386-3428-8/17/$31.00 ©2017 IEEE

2.1 Non-Gaussian Processes Metamodels

Box and Wilson (1951) pioneer work introduced the concept of response surface methodology using a low-order polynomial to approximate the underlying relationship among a set of covariates and a response (Box and Draper 1987). Since then several advancements in metamodeling have appeared. Wei et al. (2015) used metamodels to perform sensitivity analysis. Villa-Vialaneix et al. (2012) compared several parametric and non-parametric models in a agricultural engineering. They concluded that kriging and spline methods were better when data was scarce, but random forest (RF) and support vector regression (SVR) were superior with larger amounts of training data. Can and Heavey (2012) evaluated neural networks (NN) and genetic programming concluding that even though genetic programming performed slightly better in out-of-sample predictions they were more expensive to fit. Wang and Shan (2007) compared SVR, NN, Gaussian processes (GP), and decision trees for optimization. They noticed that more metamodeling comparisons are needed to gain understanding of how the techniques adapt to specific context and to higher dimensions. They emphasized that a good comparison should include more than a goodness-of-fit criteria and the use of cross-validation, as in Hastie et al. (2009) Jin et al. (2001) compared second order polynomial regression, multinomial adaptive regression splines (MARS), radial basis functions (RBF) and Gaussian process with separable Gaussian correlation (GPSGC). After evaluating 14 test problems they concluded that MARS performed better when non-linearity increased. Li et al. (2010) contrasted RBF (with ridge regularization), SVR with Gaussian kernel function, NN, GPSGC with unknown mean, and MARS concluding that MARS was more interpretable, but SVR were more accurate and robust. Ogutu et al. (2011) tested RF, Boosting, and SVR concluding that Boosting and SVR where more accurate, but RF was simpler, faster and more interpretable (Ogutu et al. 2011, p.8). Boutselis and Ringrose (2013) used NN and generalized additive models for location scale and shape (GAMLSS) to metamodel combat simulation, concluding that GAMLSS not only provided a competitive fit but also a better understanding of the covariates and their relationship with the response. Hsieh et al. (2014) proposed the use of a sequential second order polynomial to model cycle time in semiconductor manufacturing. They stated that the sequential procedure improved on the insight of the I/O interactions in the underlying simulation model. Salemi et al. (2016) studied moving least squares regression (MLSR) with anisotropic weight function for high-dimensional stochastic simulation. They compared MLSR, conditional and regression trees (CART), and stochastic kriging (SK) using the M/G/1 queuing model with 5, 25, and 75 dimensions. They found that MLSR did better for large sample sizes and it was competitive in small datasets. Finally, Joseph and Kang (2011) demonstrated that inverse distance weighting (IDW) coupled with linear regression compared favorably to ordinary kriging and required less computational effort.

2.2 Gaussian Process Metamodels

Ginsbourger et al. (2009) tested different configurations of mean response and covariance structure concluding the use of a constant mean is not recommended when the response is highly non-linear. By comparison, they found that Gaussian covariances are preferred. Bachoc et al. (2014) applied Kriging to thermal-hydraulic system testing different separable autocorrelation functions. They conclude that the Matèrn functions performed better. Hengl et al. (2004) compared regression kriging with ordinary kriging (OK). They found that combining regression and Kriging provides both higher accuracy and more interpretable results. Hung (2011) proposed the Iterative Reweighted Least Angle Regression (IRLARS) algorithm for Gaussian Processes. The iterative nature to his method, specifically where the optimization process requires numerous iterations to achieve convergence, are computationally expensive to fit. A detailed explanation of GP is found in Gelfand et al. (2010).

2.2.1 Kriging in Simulation

Current research in the field of simulation has been completed by Kleijnen (2017) who compares how RSM and Kriging relate to different design of experiments (DOE). They associate *fractional fractorial designs* with different resolutions and central composite design with RSM; whereas, latin hypercube designs (LHD) or nearly orthogonal LHD with Kriging methods. Kleijnen and Mehdad (2014) tested multiple response kriging against its univariate counterpart, concluding that independent fittings performed better. Biles et al. (2007) experimented optimizing a small *(s,S) inventory* simulation. Mehdad and Kleijnen (2015) proposed Intrinsic Kriging (IK) which follows the principles of an integrated process as seen in times series. Kleijnen (2009) pointed out that more realistic metamodeling applications were needed since more techniques comparisons focused on the M/M/1 queuing and the (s,S) inventory model. Dellino (2007) used Taguchi methods in combination with kriging to perform robust optimization of production systems. Ankenman et al. (2010) introduced the concept of stochastic kriging (SK) which handles the local variability induced by replications, an effect widely known and applied in geostatistics known as the 'heteroskedastic nugget effect'. Staum (2009) applied SK to the $M/M/1$ queuing model problem emphasizing that misspecification of the regression basis produces poor predictions and a constant mean should be preferred. Chen et al. (2013) showed that incorporating gradient information to SK improves surface prediction.

It is possible to observe that a wide range of techniques have been used to model a variety of different situations. Applications outside of the field of simulation tend to focus on comparing models. Additionally, the lack of metamodel applications for complex simulations has been identified by some authors - a point illuminated by this review. Moreover, based on the reported results it seems that, in general, SVR, NN, and GP are stable and reliable techniques and that random forest (RF) and boosted trees (BT) are not popular in simulation metamodeling. A short description of these techniques is presented in the section that follows.

3 MATERIALS AND METHODS

The variables definitions used in this section are consistent with those defined by Hastie et al. (2009). An input variable is given by X, whose components are $\{X_j\}_1^p$, being p is the total number of features. The signal is given by Y. Capital letters are used to refer the generic aspect of the variable, whereas observed values are written in lowercase, such as sample values x_i, where x_i is the ith observed value of X. Bold uppercase letters represent matrices. The set of samples, e.g. $\{x_i\}_1^N$ would be the $N \times p$ matrix $\boldsymbol{X}$. Vectors with N elements are bold; thus $\boldsymbol{y}$ and $\boldsymbol{x_j}$ represent all observation on the response and the X_j variable respectively. Finally, x_i^T is the transpose of x_i since all vectors are assumed to be column vectors and x_i is the ith row of $\boldsymbol{X}$.

3.1 Modern Machine Learning Techniques

When the relationship among the covariates and the response are non-linear or hard to define, or both, machine learning algorithms provide a better way to approximate the response vector Y at the cost of interpretability (Breiman et al. 2001). In this section, a brief explanation of the machine learning algorithms used in this study is presented. The interested reader can reference Hastie et al. (2009) for a detailed description.

3.1.1 Support Vector Regression (SVR)

Support vector regression (SVR), which was first introduced by Cortes and Vapnik (1995), consists of finding a function $f(X)$ that has at most ε deviation from all training responses Y. Assume that the functional relationship between the covariates and the response can be expressed as $f(X) = \beta_0 + \phi(X)^T\beta$, where $\phi(\cdot)$ is a kernel that maps the feature space to a higher dimension. To locate the optimal β_0 and β while honoring the ε constraint Equation 1 needs to be solved, where C is a tuning parameter that controls the amount of error to be tolerated above ε (Smola and Schölkopf 2004, p.200). To improve prediction accuracy, a λ is introduced in order to constrain the Euclidean norm of the parameters - this is also known as regularization. Finally, this technique ignores errors smaller than ε as seen in Equation 2 which is known as the ε-*insensitive* loss function

$$H(\beta,\beta_0) = C\sum_{i=1}^{N} V(y_i, f(x_i)) + \frac{\lambda}{2}||\beta||^2, \tag{1}$$

where

$$V(y_i, f(x_i)) = \begin{cases} |f(x_i) - y_i| - \varepsilon, & \text{if } |f(x_i) - y_i| \geq \varepsilon; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

This technique is sensitive to the choice of C, ε, and the kernel function $k(\cdot,\cdot)$. The ranges used to tune C and ε are difficult to determine; thus, the procedures proposed by Cherkassky and Ma (2004) are used. According to Hastie et al. (2009), the kernel function is the crucial element used by this technique to induce non-linearity; hence, *linear*, *radial basis function*, and *second and third-order polynomial* are tested. Additionally, the authors reported that SVR performs poorly in *high-dimensional* settings.

3.1.2 Neural Network (NN)

A neural network (NN) consists of a collection of layers, which are composed of nodes, that use transformation functions to induce non-linearity. The training data flows from the input nodes toward the output node, being constantly transformed along the way. At the output node the prediction error is evaluated and the information is passed backward - known as back-propagation - to tune the parameters. This process is repeated until some convergence tolerance is reached. The most used topology to represents a NN is the *feed-forward artificial neural network* (Svozil et al. 1997). In general, nodes are fully connected to their predecessors, and the number of hidden layers (layers that are neither the input nor the output), with their member nodes. All layers need to be tuned in order to avoid either under-fitting or over-fitting. Finally, since the focus of this work is regression, the output node uses a linear transfer function. With regard to the number of hidden layers there are several rules (Hastie et al. 2009); however, in this work the following formula is used $N_h = \frac{N_s}{k(N_i+N_o)}$, where N_h represents the number of hidden nodes, whereas, N_i and N_o are the number of input and output nodes respectively. N_s represents the number of samples and k is a scaling factor between two and ten. Additionally, the most used *transfer functions* are the rectifier linear unit (Relu) defined by $f(z) = \log(1+e^z)$ and the sigmoid function $f(z) = \frac{1}{1+e^z}$, where z is the linear combination of inputs reaching the node. Another important parameter to consider is the *learning rate*, which can be set to be a constant or a variable. The variable considers where the function of the training epochs t though the following function: $f(t) = \text{initial\_learning\_rate}/t^{0.5}$. The *initial learning rate* is set to a number in the interval [0,1]. Finally, L_2 regularization can be used to control over-fitting. In this work we use the Multilayer Perception (MLP) to carry out the analysis.

3.1.3 Random Forest Regression (RF)

The random forest regression (RF) method introduced by Breiman (2001) consists of ensembling several learners (regression trees in this case) to obtain a prediction based on the average vote of all individuals. Although the regression tree is easy to interpret due to its piece-constant structure (Loh 2011), it has several shortcomings important to note: (1) it forces the interaction of variables that could not be related, (2) it is sensitive to small perturbations which has an impact on out-of-sample predictions, and (3) it is not well suited when the response is smooth (Speybroeck 2012). The procedure to fit a RF is simple. Fit a total of M independent trees, but at each iteration get a bootstrapped sample of size N and randomly select k features to obtain more uncorrelated trees; thus, reducing the variance among trees. Following the notation in Hastie et al. (2009), p.589, Θ_b is the set of `if-then-else` rules for the mth tree. Subsequently, the prediction is given by $\hat{f}(X) = \frac{1}{M}\sum_{m=1}^{M} T(X : \Theta_m)$.

3.1.4 Gradient Boosted Regression Tree (GBRT)

Gradient Boosting is another technique based on regression trees. It was introduced by Friedman (2002) who envisioned an additive model constructed by sequential fitting a simple model (base learner) to the residuals of each iteration, such that the prediction is given by $f(X) = \sum_{m=0}^{M} \beta_m b(X;\Theta_m)$, where M are the total number of base learners fitted. In the case of GBRT, the base learner is a tree, then, at each iteration of the algorithm a new tree is fitted to the residuals of the linear combination of previously fitted trees. The author showed that this sequential procedure improved the fitting in regions where the previously fitted trees had poor performance. To induce stochasticity the trees are trained using samples of size $\tilde{N}$ without replacement such that $\tilde{N} \leq N$ where N is the size of the training sample. This procedure is known as *stochastic gradient boosting* and it is helpful to reduce the computational cost of fitting the trees. Moreover, GBRT provides rich information for model exploration and covariate importance. After fitting a GBRT, it bears information about the "relative importance" of each variable which is collected from the number of split associated with the variable weighted by the relative improvement of the split, averaged over all trees. Additionally, it is possible to obtain "partially dependency plots" which are useful to study the marginal effect that one or two variables have on the response.

3.1.5 Gaussian Process

Sacks et al. (1989) presented one of the first works of kriging (in this work, the words kriging and Gaussian process are equivalent) to metamodel an engineering problem. Thereafter, Ankenman et al. (2010), Staum (2009), and Biles et al. (2007) appear as one of the first applications to simulation problems. Gaussian process is a statistical technique that not only can model the trend component of the response vector, but also the stochastic noise associated to it. In general, it can be represented as $Y = F(X)^T\beta + Z(X)$, where X is a p-dimensional vector, $F(X)$ is a mapping function of the input variable X to a k dimensional space (it also adds an intercept), β a vector of k parameters and $Z(\cdot)$ a zero-mean Gaussian process with covariance $C(h) = C[Z(x_i), Z(x_j)] = C(x_i, x_j; \theta)$, where θ is a vector of parameters in $\mathbb{R}^{\mathrm{p}}$. On the one hand, to include the effect of anisotropy in the model the covariance function can be computed as the product of one-dimension $g(\cdot)$ covariances which is defined as *separable covariance*. Knowing that $h^T = (h_1, h_2, \ldots, h_p) = x_i - x_j$. Then the i,jth element of the covariance matrix $C(\cdot)$ is given by $c(h) = \sigma^2 \prod_{j=1}^{p} g(h_j;\theta_j)$ for training samples i and j, where, e.g. $g(h,\theta) = (1 + h/\theta)\exp(-h/\theta)$ for the one-dimensional Matèrn once differentiable. On the other hand, anisotropic behavior can be achieved using a *non-separable covariance* constraining the Euclidean norm of the one dimensional distance between each observation, such as $\|h\|_\theta = \sqrt{h^T \boldsymbol{A}_\theta h}$ where $\boldsymbol{A}_\theta = \mathrm{diag}(1/\theta_1, 1/\theta_2, \cdots, 1/\theta_p)$. For example, the once differentiable Matèrn can be expressed as $g(h,\theta) = (1 + \|h\|_\theta)\exp(-\|h\|_\theta)$ and the i,jth element of the covariance matrix $\boldsymbol{C}(\cdot)$ is given by $c(h) = \sigma^2 g(h,\theta)$.

A detailed derivation can be found in Cressie (2015), Gelfand et al. (2010), Sacks et al. (1989), among others. The important concept to keep in mind is that to obtain the estimated $\hat{\beta}$ it is also necessary to compute the spatial decay parameters θ, thus we can compute $\hat{\beta} = \left(\boldsymbol{F^T}\boldsymbol{\Sigma}(\theta)^{-1}\boldsymbol{F}\right)^{-1} \boldsymbol{F^T}\boldsymbol{\Sigma}(\theta)^{-1}\boldsymbol{y}$. The link between these two set of variables is the *maximum likelihood* equation as presented in Equation 3. To minimize this equation, it is important to notice that β can be computed separately using the aforementioned equation. Subsequently, the problem reduces to the *profile log likelihood*

$$\log \mathscr{L}(\beta,\theta;\boldsymbol{y}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}(\theta)| - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{F}\beta)^T \boldsymbol{\Sigma}^{-1}(\theta)(\boldsymbol{y} - \boldsymbol{F}\beta). \tag{3}$$

Finally, once the β and θ vectors have been computed, the forecasting equation of out-of-sample points is given in Equation 4, where $\boldsymbol{c}(x)^T = [c(x_1,x), c(x_2,x), \ldots, c(x_N,x)]$,

$$\hat{y}(x) = F(x)^T\hat{\beta} + \boldsymbol{c}(x)^T \boldsymbol{\Sigma}(\hat{\theta})^{-1}\left(\boldsymbol{y} - \boldsymbol{F}\hat{\beta}\right). \tag{4}$$

3.2 A Complex Model

This study utilized data generated from a system dynamics model representing a whole hospital model. The purpose of the model was to investigate the effects of patient flow and information on a constrained unit capacity environment for a generalized, medium sized community hospital. The analysis explores the effects of varied unit capacity, along with other uncertain parameters, to explore improvements for overall operating performance and efficiency. The model is an extension of the work presented in Smith III and Roberts (2014a), and Smith III and Roberts (2014b). A detailed description of the covariates and responses can be found in De la Fuente (2016). A latin hypercube design was used to generate 600 combinations of parameters where 13 different output responses were collected. Figure 1 shows a subset of those responses where it is possible to see that responses five and six have extreme behaviors, which is typical when service/production systems are under stress. The only two responses, from the total set that can be thought as "normalizables" are responses seven and eight. Other responses not shown here are multimodal.

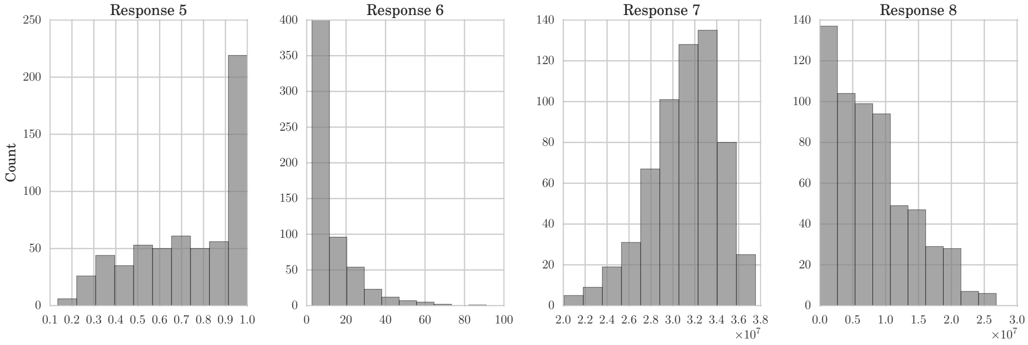

Figure 1: Histograms of some selected responses.

3.3 Experimental Design

The data set was divided in five blocks of 120 observations each to do cross-validation, as recommended in Hastie et al. (2009), p.241. Figure 2 summarizes the experimental design of this work. For hyper-parameters optimization, randomized grid search cross-validation (RGSCV) was used since it performs better than regular grid search cross-validation (Bergstra and Bengio 2012, p.284). For comparability, Python skit-learn implementations of SVR, RF, GBRT, GP, and MLP were tested sequentially and the maximum number of iterations for the randomized search set to 100. For reproducibility, a seed of 100 was set to the RGSCV and 1 for all other techniques with the exception of GP that was 8. With regard to accuracy, $MSPE = [\sum_{i=1}^{n} (y_i - \hat{y}_i)^2]/n$ and $R^2 = 1 - [\sum_{i=1}^{n} (y_i - \hat{y}_i)^2]/[\sum_{i=1}^{n} (y_i - \bar{y})^2]$ will be used. Time and interpretability will be also considered. All covariates are scaled to put them in the same standing. Finally, all computations were performed using an Apple Macbook Air configured with a four core i5 microprocessor and the Python 3.4.3 MKL (math kernel library) libraries installed.

4 RESULTS AND DISCUSSION

In this section, we present and compare a subset of the best tuned models for the techniques yielding the best results. A detailed description of the grid used in identifying the best configuration of tuning parameters per technique can be found in De la Fuente (2016), p.48. Since testing separability of the covariance structures of the GP modes was very important, as shown in Figure 2, these results are presented first. Thereafter, the very best GP model will compete with the top models of the other techniques.

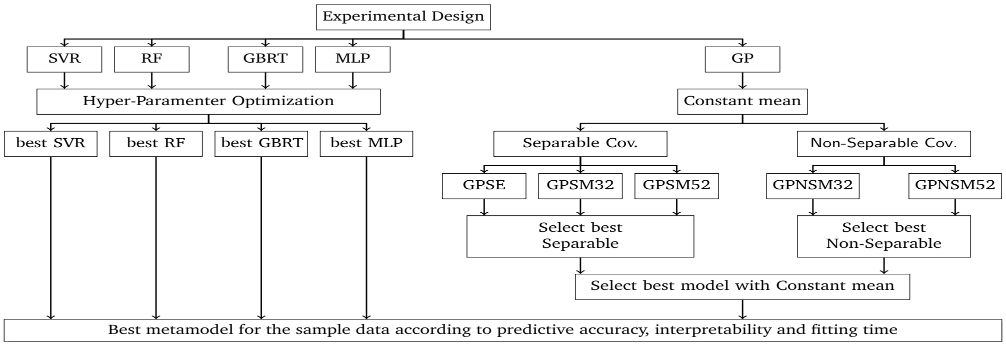
Experimental Design
SVR
RF
GBRT
MLP
GP
Hyper-Paramenter Optimization
Constant mean
best SVR
best RF
best GBRT
best MLP
Separable Cov.
Non-Separable Cov.
GPSE
GPSM32
GPSM52
GPNSM32
GPNSM52
Select best Separable
Select best Non-Separable
Select best model with Constant mean
Best metamodel for the sample data according to predictive accuracy, interpretability and fitting time

Figure 2: Experimental design for metamodel comparison.

4.1 Fit Quality

Figure 3 shows the relative reduction in MSPE for the Gaussian process with non-separable Màtern once differentiable covariance (GPNSM32), contrasted with the Gaussian processes with squared exponential (GPSE), and separable Màtern once differentiable covariance (GPSM32). In both cases, the GPNSM32 outperforms the other models, achieving reductions of more than 50%. Something important to notice is that the GPSM32 is competitive in responses 7 and 8, which are the only responses that look "normalizable". This characteristic of the response may lead to conclusions in some studies suggesting that GPSM32 is more robust than GPSE. The opposite is also true, since the GPSE performed much better than GPSM32 in responses 6 and 7. In relation to the other criteria, GPNSM32 also outperforms its GP counterparts. Finally, there was little difference between the degree of differentiability of the Matérn non-separable covariances, as in the case of GPNSM32 and GPNSM52. For this reason the degree of differentiability is dropped and referenced generally as GPNSM.

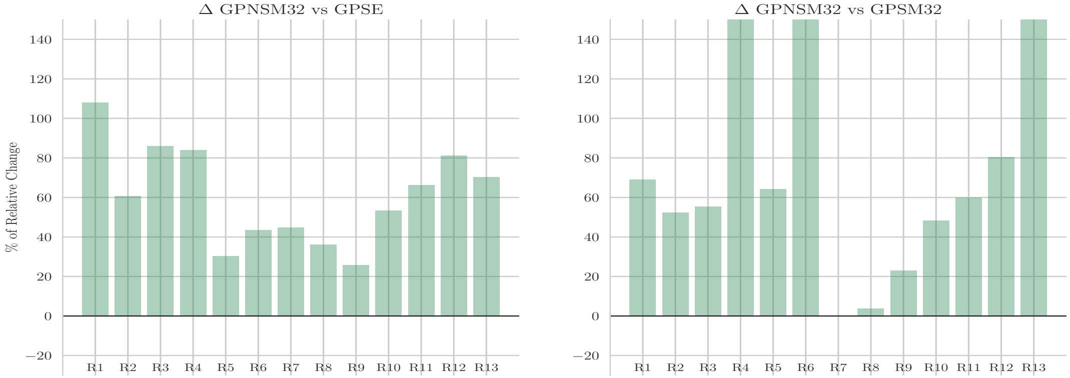

Figure 3: Best Gaussian process metamodel selection, MSPE comparison.

Once the best GP hyper-parameters were chosen, a global comparison was made with the best metamodels of the other techniques. As shown in Figure 4, GPNSM outperforms the MLP and GBRT. An important matter to consider is that the relative improvement of GPNSM, with respect to MLP, is smaller in several responses than what was obtained in Figure 3. This indicates that MLP performed better than GPSM and GPSE, which are the classic configurations used to perform metamodeling in simulation problems. Additionally, GPNSM reduces MSPE by more than 100% in response 7 when compared to MLP. Finally, the third best technique was GBRT which yielded a MSPE more than a 100% higher in the majority of the responses. Both SVR and RF appear further behind, as will be shown next.

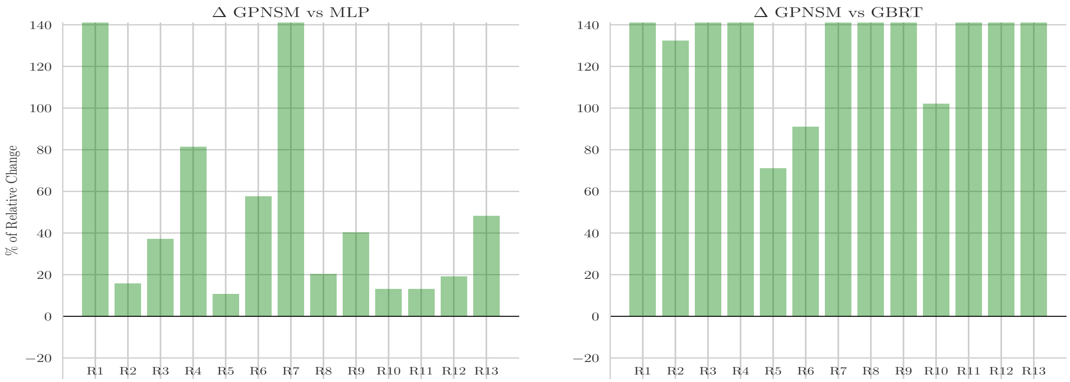

Figure 4: Best metamodel selection, MSPE comparison.

Figure 5 provides a detailed view of the behaviors of the R^2 for each response and each technique. The figure illustrates that GPNSM performs better than all other techniques, achieving in most cases R^2's higher than 90% in the testing (we are reporting the average of the testing folds). Additionally, MLP is a close competitor, but it is observed that a small difference in R^2 can result in a significant reduction in MSPE. Another important finding is that GBRT is better in all responses than RF which has been also pointed out by Hastie (2014). Finally, SVR is the technique that performed the worst in 11 out of 13 responses; however, it was competitive in responses 7 and 8, which are the "normalizable" responses presented in Figure 1. In the literature review several authors reported that SVR where both competitive and fast. This could be attributed to the nature of the responses being studied.

4.2 Fitting Time

With regard to how much time is needed to tune a model, the time collected for the entire RGSCV process for each technique is presented in Figure 6. Although this dimension is subjective, the sklearn module has implemented the algorithms in the most efficient way possible. RF took longer than any other technique to complete mainly due to the number of trees needed to create a forest, where each tree is bushy and complex. Then, GBRT shows the second largest average being noticeable faster than RF. This is because GBRT requires shallow trees (week learners), making the training process faster. The GPSM techniques take longer than their non-separable counterparts. Finally, SVR and MLP are the fastest techniques, which is consistent with the findings stated in the literature review.

4.3 Interpretability

With regard to interpretability, there is not an absolute scale available to rank the models; therefore, a simple classification rule is used. The rule of thumb is that a technique can be considered *excellent* if it is possible to provide information about the gradient and the relationship of each covariate $\{X_j\}_1^p$ with the response. *Good* when it gives only information of dependency but not the gradient. *Regular* and *limited* have fuzzy boundaries but techniques in these ranges only give information after post-processing the results. Finally, *poor* techniques do not render any additional insight beyond that of the predicted value after fitting. In general, complex techniques obtain good fitting performance at the price of interpretability. The entire GP family of methods tested here employ a constant intercept (see Figure 2) as recommended for engineering applications (Kleijnen 2017), but it provides little information to the end-user. However, it can be claimed that the vector of θ values can be used to identify the importance of each covariate. MLP can be interpreted using its arch weights, as proposed in Olden and Jackson (2002) to obtain a relative importance plot. However, neither the gradient nor higher order dependency can be computed. SVR are interpretable only when a linear kernel is used - which is not the case in this work. Finally, tree-based methods provide rich information after fitting. Figure 7 provides an example of the relative importance of

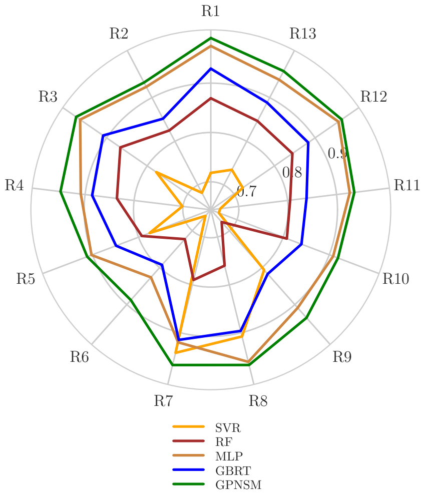

Figure 5: All models - R^2.

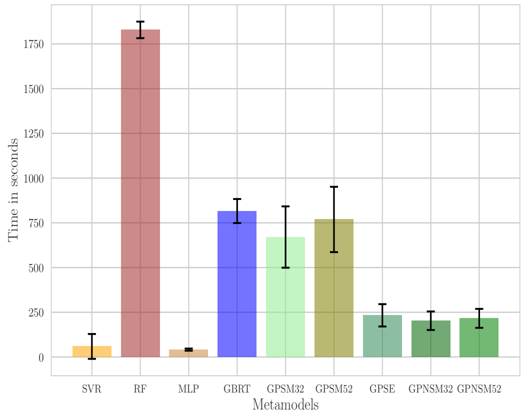

Figure 6: All models - Fitting time.

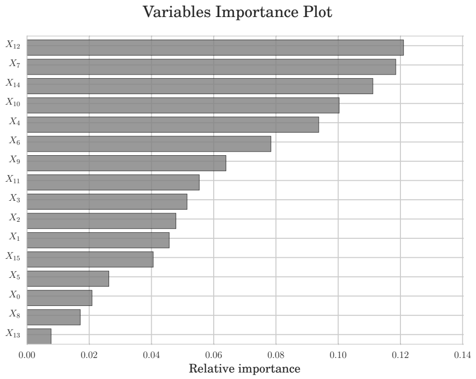

Figure 7: GBRT - Variable importance.

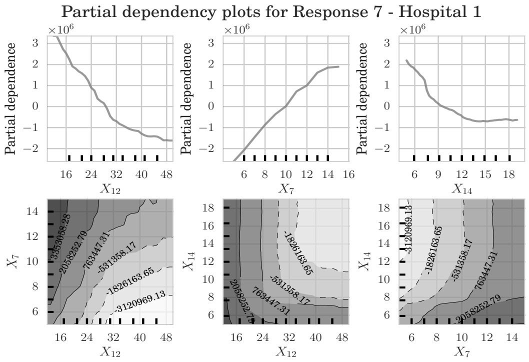

Figure 8: GBRT - Partial dependency.

the variables specifically for response 7. Additionally, the partial dependency of one or two variables with the response can be obtained using tree based methods. Specifically, this relationship may be determined to have a direct, inverse, or concave or convex quadratic effect, as illustrated in Figure 8.

5 CONCLUSIONS

The conclusions regarding this work are illustrated in Figure 9 and summarized in the narrative that follows:

1. When fit quality is the most important criteria the GPNSM models are the most robust technique across responses. They outperform their counterparts in both MSPE and R^2. It seems that the non-separable form of the covariances constrains the range of decay of the θ vector of spatial decay in a more efficient way than the separable form of the same covariances. Additionally, MLP seems unaffected by the dimensionality of the dataset, and SVR showed problems in all highly non-linear responses. These results are presented in the dimension "fit dominance" of the figure which represents the percentage of the number of times a technique dominates the others considering all 52 responses it was competing with (4 techniques times 13 responses). For the case of GPNSM, it dominated all other techniques (4) in all responses (13). The computed fit dominance score is $((4 \times 13)/52) \times 100 = 100$. By comparison, MLP dominated all other techniques, except GPNSM,

resulting in a computed score of $((3 \times 13)/52) \times 100 = 75$. Other techniques values can be inferred directly from Figure 5.

2. With regard to time, MLP was the faster technique and provided the second best performance overall. Non-separable Matèrns where faster than the separable Matèrns. The tree-based methods took longer and their prediction performance was inferior in general. SVR was as fast as MLP but its performance was the worst in the data set. This can be seen in the dimension "$\log_{10}$(time)"
3. The main drawback of GPNSM techniques, MLP, and SVR is their lack of interpretability. This makes it difficult for the end-user to understand the relationship between covariates and the response. In this dimension, tree-based techniques are superior, as shown in the figure's "interpretability" dimension.

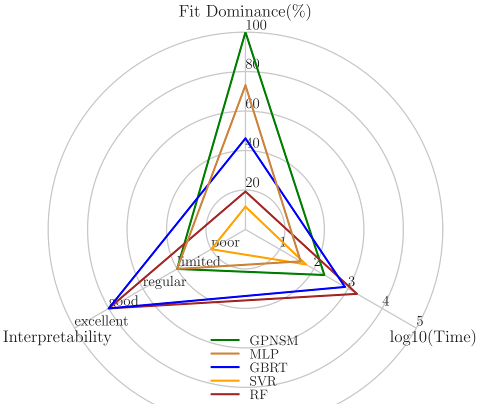

Figure 9: Conclusions per criterion.

The results reveal that what makes a technique stronger in one criterion will make it weaker in another. Future work can be performed to explore the effect in the GPNSM results when a non-stationary trend is included in the model. Additionally, a larger number of covariates can be used to see the effect that increasing the dimensionality has on the results of MLP and GBRT. Finally, while the results presented here are of limited scope it is reasonable to expect similar outcomes when applied to other types of problems with highly unbalanced distributions in the responses. In the case of discrete event simulation, the same logic should apply after adjusting the design matrix due to the intrinsic variability at each design point produced by replications. A future extension of the work may demonstrate this result.

REFERENCES

Ankenman, B., B. L. Nelson, and J. Staum. 2010. "Stochastic Kriging for Simulation Metamodeling". *Operations Research* 58 (2): 371–382.

Bachoc, F., G. Bois, J. Garnier, and J.-M. Martinez. 2014. "Calibration and Improved Prediction of Computer Models by Universal Kriging". *Nuclear Science and Engineering* 176 (1): 81–97.

Bergstra, J., and Y. Bengio. 2012. "Random Search for Hyper-parameter Optimization". *The Journal of Machine Learning Research* 13 (1): 281–305.

Biles, W. E., J. P. Kleijnen, W. Van Beers, and I. Van Nieuwenhuyse. 2007. "Kriging Metamodeling in Constrained Simulation Optimization: An Explorative Study". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 355–362. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Boutselis, P., and T. J. Ringrose. 2013. "GAMLSS and Neural Networks in Combat Simulation Metamodelling: A Case Study". *Expert Systems with Applications* 40 (15): 6087–6093.

Box, G. E., and N. R. Draper. 1987. *Introduction to Response Surface Methodology*. Wiley Online Library.

Box, G. E., and K. Wilson. 1951. "On the Experimental Attainment of Optimum Conditions". *Journal of the Royal Statistical Society. Series B (Methodological)* 13 (1): 1–45.

Breiman, L. 2001. "Random Forests". *Machine Learning* 45 (1): 5–32.

Breiman, L. et al. 2001. "Statistical Modeling: The Two Cultures (with Comments and a Rejoinder by the Author)". *Statistical Science* 16 (3): 199–231.

Can, B., and C. Heavey. 2012. "A Comparison of Genetic Programming and Artificial Neural Networks in Metamodeling of Discrete-event Simulation Models". *Computers & Operations Research* 39 (2): 424–436.

Chen, X., B. E. Ankenman, and B. L. Nelson. 2013. "Enhancing Stochastic Kriging Metamodels with Gradient Estimators". *Operations Research* 61 (2): 512–528.

Cherkassky, V., and Y. Ma. 2004. "Practical Selection of SVM Parameters and Noise Estimation for SVM Regression". *Neural Networks* 17 (1): 113–126.

Cortes, C., and V. Vapnik. 1995. "Support-vector Networks". *Machine Learning* 20 (3): 273–297.

Cressie, N. 2015. *Statistics for Spatial Data*. John Wiley & Sons.

De la Fuente, R. 2016. *Simulation Metamodeling with Gaussian Process: A Numerical Study*. Ph. D. thesis, North Carolina State University.

Dellino, G. 2007. *Robust Simulation-Optimization Methods Using Kriging Metamodels*. Ph. D. thesis, Universita Degli Studi di Bari.

Friedman, J. H. 2002. "Stochastic Gradient Boosting". *Computational Statistics & Data Analysis* 38 (4): 367–378.

Gelfand, A. E., P. Diggle, P. Guttorp, and M. Fuentes. 2010. *Handbook of Spatial Statistics*. CRC press.

Ginsbourger, D., D. Dupuy, A. Badea, L. Carraro, and O. Roustant. 2009. "A Note on the Choice and the Estimation of Kriging Models for the Analysis of Deterministic Computer Experiments". *Applied Stochastic Models in Business and Industry* 25 (2): 115–131.

Hastie, T. 2014. "Gradient Boosting Machine Learning". https://www.youtube.com/watch?v=wPqtzj5VZu. [Online; accessed 20-March-2017].

Hastie, T., R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning*, Volume 2. Springer.

Hengl, T., G. B. Heuvelink, and A. Stein. 2004. "A Generic Framework for Spatial Prediction of Soil Variables Based on Regression-kriging". *Geoderma* 120 (1): 75–93.

Hsieh, L. Y., K.-H. Chang, and C.-F. Chien. 2014. "Efficient Development of Cycle Time Response Surfaces Using Progressive Simulation Metamodeling". *International Journal of Production Research* 52 (10): 3097–3109.

Hung, Y. 2011. "Penalized Blind Kriging in Computer Experiments". *Statistica Sinica* 21 (3): 1171.

Jin, R., W. Chen, and T. W. Simpson. 2001. "Comparative Studies of Metamodelling Techniques Under Multiple Modelling Criteria". *Structural and Multidisciplinary Optimization* 23 (1): 1–13.

Joseph, V. R., and L. Kang. 2011. "Regression-based Inverse Distance Weighting with Applications to Computer Experiments". *Technometrics* 53 (3): 254–265.

Kleijnen, J. P. 2009. "Kriging Metamodeling in Simulation: A Review". *European Journal of Operational Research* 192 (3): 707–716.

Kleijnen, J. P. 2017. "Regression and Kriging Metamodels with their Experimental Designs in Simulation: A Review". *European Journal of Operational Research* 256 (1): 1–16.

Kleijnen, J. P., and E. Mehdad. 2014. "Multivariate Versus Univariate Kriging Metamodels for Multi-response Simulation Models". *European Journal of Operational Research* 236 (2): 573–582.

Li, Y., S. H. Ng, M. Xie, and T. Goh. 2010. "A Systematic Comparison of Metamodeling Techniques for Simulation Optimization in Decision Support Systems". *Applied Soft Computing* 10 (4): 1257–1273.

Loh, W.-Y. 2011. "Classification and Regression Trees". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1 (1): 14–23.

Mehdad, E., and J. P. Kleijnen. 2015. "Stochastic Intrinsic Kriging for Simulation Metamodelling". In *CentER Discussion Paper*, 1–20. Tilburg, Netherlands: Tilburg: CentER, Center for Economic Research.

Ogutu, J. O., H.-P. Piepho, and T. Schulz-Streeck. 2011. “A Comparison of Random Forests, Boosting and Support Vector Machines for Genomic Selection”. In *BMC proceedings*, Volume 5, S11. BioMed Central Ltd.

Olden, J. D., and D. A. Jackson. 2002. “Illuminating the Black Box: A Randomization Approach for Understanding Variable Contributions in Artificial Neural Networks”. *Ecological Modelling* 154 (1): 135–150.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn. 1989. “Design and Analysis of Computer Experiments”. *Statistical Science*:409–423.

Salemi, P., B. L. Nelson, and J. Staum. 2016. “Moving Least Squares Regression for High-Dimensional Stochastic Simulation Metamodeling”. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 26 (3): 16.

Smith III, R. L., and S. D. Roberts. 2014a. “Sensitivity Analysis for a Whole Hospital System Dynamics Model”. In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 1305–1316. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Smith III, R. L., and S. D. Roberts. 2014b. “A Simulation Approach to Exploring Whole Hospital System Operational Performance and Efficiencies”. In *IIE Annual Conference. Proceedings*, 1730. Institute of Industrial Engineers-Publisher.

Smola, A. J., and B. Schölkopf. 2004. “A Tutorial on Support Vector Regression”. *Statistics and Computing* 14 (3): 199–222.

Speybroeck, N. 2012. “Classification and Regression Trees”. *International Journal of Public Health* 57 (1): 243–246.

Staum, J. 2009. “Better Simulation Metamodeling: The Why, What, and How of Stochastic Kriging”. In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 119–133. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Svozil, D., V. Kvasnicka, and J. Pospichal. 1997. “Introduction to Multi-layer Feed-forward Neural Networks”. *Chemometrics and Intelligent Laboratory Systems* 39 (1): 43–62.

Villa-Vialaneix, N., M. Follador, M. Ratto, and A. Leip. 2012. “A Comparison of Eight Metamodeling Techniques for the Simulation of N2O Fluxes and N Leaching from Corn Crops”. *Environmental Modelling & Software* 34:51–66.

Wang, G. G., and S. Shan. 2007. “Review of Metamodeling Techniques in Support of Engineering Design Optimization”. *Journal of Mechanical Design* 129 (4): 370–380.

Wei, P., Z. Lu, and J. Song. 2015. “Variable Importance Analysis: A Comprehensive Review”. *Reliability Engineering & System Safety* 142:399–432.

AUTHOR BIOGRAPHIES

RODRIGO DE LA FUENTE is an Assistant Professor in the Department of Industrial Engineering at University of Concepción (Chile). He holds a Ph.D. in industrial and systems engineering from North Carolina State University. His research interests include discrete-event simulation, system dynamics, agent-based simulation, and machine learning. His e-mail address is rodelafuente@udec.cl.

RAYMOND SMITH III is an Assistant Professor in the Department of Engineering at East Carolina University. He holds a Ph.D. in industrial and systems engineering from North Carolina State University. His research interests include health care systems, simulation modeling, information systems and operations management. His e-mail address is rlsmith5@ncsu.edu.