

Gaussian Models and Correlation Structures

Harshvardhan

June 2018

In many a situations, naturally occurring experiments are either too expensive or it isn't possible for us to experiment with them. For e.g., consider what if we want to see what affects weather of a particular region. In most cases, it isn't possible for us to "experiment" with the weather. Similarly, say we want to study a volcano. We know several parameters that affect the current temperature of a volcano but we can't alter most of them. For such purpose, we need an emulator that can perform just like that volcano or region where we can alter the variables and simulate conditions. Gaussian model is a popular choice for such emulators. These emulators often need to interpolate the simulator and this condition can sufficiently be covered by Gaussian models. Gaussian models make a good emulator and thus are popularly used by various applied statisticians.

1 Forms of Gaussian Models

1.1 Deterministic Model: $y(x) = \mu + Z(x)$

Model Statement Consider the following: x_i is d dimensional i^{th} input vector, $\mathbf{x}_i = (x_{i1}, \dots, x_{id})'$. $y_i = y(x_i)$ is the univariate response variable. The experiment design is $D_0 = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ as set of n input vectors. Output of the simulation trials are stored in n dimensional vector as $\mathbf{Y} = y(D_0) = (y_1, y_2, \dots, y_n)'$.

This model is also known as *Gaussian process model* with fixed mean. The μ is the *overall grand mean*. $Z(x)$ is n dimensional Gaussian process with $E(Z(\mathbf{x}_i)) = 0$, $Var(Z(\mathbf{x}_i)) = \sigma_z^2$ and $cov(Z(x_i), x_j)) = \sigma^2 R_{ij}$, where R is correlation matrix of all \mathbf{x}_i . In most cases, we assume that $y(D_0)$ has a multivariate normal distribution, $N_n(\mathbf{1}_n \mu, \Sigma)$, where $\Sigma = Var(D_0 | y(D_0)) = \sigma_z^2$ and $\mathbf{1}_n$ is a $n \times 1$ vector of all 1's (Sacks et al., 1989). Define a new vector \mathbf{r} , of correlation between various sampled design points, x_i and the unsampled design point, x_i^* , i.e. $\mathbf{r}(x^*) = [corr(x_1, x^*), corr(x_2, x^*), \dots, corr(x_n, x^*)]'$.

Likelihood The negative profile log-likelihood in this GP model is proportional to,

$$-2 \log L_p \propto \log(|R|) + n \log[(Y - \mathbf{1}_n \hat{\mu}(\theta))' R^{-1} (Y - \mathbf{1}_n \hat{\mu}(\theta))].$$

Parameter Estimates The closed form estimates of the parameters, μ and σ^2 are as follows,

$$\begin{aligned} \hat{\mu}(\theta) &= (\mathbf{1}'_n R^{-1} \mathbf{1}_n)^{-1} (\mathbf{1}'_n R^{-1} Y), \\ \hat{\sigma}^2(\theta) &= \frac{(Y - \mathbf{1}_n \hat{\mu}(\theta))' R^{-1} (Y - \mathbf{1}_n \hat{\mu}(\theta))}{n}. \end{aligned}$$

Predictors The predictor $y(x)$ and the responses Y together follow multivariate normal distribution, i.e.

$$\begin{pmatrix} y(x) \\ Y \end{pmatrix} = N_n \left(\begin{pmatrix} \mu \\ \mu \mathbf{1}_n \end{pmatrix}, \begin{pmatrix} \sigma^2 & \sigma^2 \mathbf{r}'(x) \\ \sigma^2 \mathbf{r}(x) & \sigma^2 R \end{pmatrix} \right).$$

So, $E(y(x)|Y) = \mu + \mathbf{r}' R^{-1} (Y - \mathbf{1}_n \mu)$ and $Var(y(x)|Y) = \sigma^2 (1 - \mathbf{r}'(x) R^{-1} \mathbf{r}(x))$. The predicted value at any unsampled point x^* , the predictor is,

$$\hat{y}(x^*) = \hat{\mu} + \mathbf{r}' R^{-1} (Y - \mathbf{1}_n \hat{\mu}),$$

where $\hat{\mu}$ is the estimated mean.

Mean Squared Error $\hat{y}(x^*)$ has a mean squared error of $\sigma^2 (1 - \mathbf{r}'(x) R^{-1} \mathbf{r}(x))$.

1.2 Universal Kriging Model: $y(x) = F\beta + Z(x)$

Model Statement Consider the following: x_i is d dimensional i^{th} input vector; $\mathbf{x}_i = (x_{i1}, \dots, x_{id})'$. $y_i = y(x_i)$ is the univariate response variable. The input data matrix is $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ as set of n input vectors. Output of the simulation trials are stored in n dimensional vector as $\mathbf{Y} = y(X) = (y_1, y_2, \dots, y_n)'$.

The vector β is the Generalised Least Square estimate of true β . $Z(x)$ is n dimensional Gaussian process with $E(Z(x_i)) = 0$, $Var(Z(x_i)) = \sigma^2$ and $cov(Z(x_i), Z(x_j)) = \sigma^2 R_{ij}$, where R is correlation matrix of all \mathbf{x}_i .

Define

$$F = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}.$$

Also define a new vector \mathbf{r} , of correlation between various sampled design points, x_i and the unsampled design point, x_i^* , i.e. $\mathbf{r}(x^*) = [corr(x_1, x^*), corr(x_2, x^*), \dots, corr(x_n, x^*)]'$.

Likelihood The negative profile log-likelihood in this GP model is proportional to,

$$-2 \log L_p \propto \log(|R|) + n \log[(Y - F\hat{\beta})'R^{-1}(Y - F\hat{\beta})].$$

Parameter Estimates The parameters and their values are given by,

$$\hat{\beta} = (F'R^{-1}F)^{-1}F'R^{-1}Y, \text{ and}$$

$$\hat{\sigma}^2 = \frac{(Y - F\hat{\beta})'R^{-1}(Y - F\hat{\beta})}{n}.$$

Predictor The predictor $y(x)$ and the responses Y together follow multivariate normal distribution, i.e.

$$\begin{pmatrix} y(x) \\ Y \end{pmatrix} = N_n \left(\begin{pmatrix} f(x)\beta \\ F\beta \end{pmatrix}, \begin{pmatrix} \sigma^2 & \sigma^2 \mathbf{r}'(x) \\ \sigma^2 \mathbf{r}(x) & \sigma^2 R \end{pmatrix} \right).$$

So, $E(y(x)|Y) = \mu + \mathbf{r}'R^{-1}(Y - F\beta)$ and $Var(y(x)|Y) = \sigma^2(1 - \mathbf{r}'(x)R^{-1}\mathbf{r}(x))$. The predicted value at any unsampled point x^* , the BLUP is,

$$\hat{y}(x^*) = f(x^*)\hat{\beta} + \mathbf{r}'R^{-1}(Y - F\hat{\beta}),$$

where $\hat{\beta} = (F'R^{-1}F)^{-1}F'R^{-1}Y$, i.e. usual generalised least square estimate. The first part in BLUP can be interpreted as least square prediction and the second part is the Gaussian process. So, an statistician can first get a regression model and then interpolate the residuals as if there were no regression model, and that would lead to this model.

1.3 Model: $Y(x) = \mu + Z(x) + \varepsilon$

In this model, consider the following: x_i is d dimensional i^{th} input vector, $\mathbf{x}_i = (x_{i1}, \dots, x_{id})'$. $y_i = y(x_i)$ is the univariate response variable. The experiment design is $D_0 = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ as set of n input vectors. Output of the simulation trials are stored in n dimensional vector as $\mathbf{Y} = y(D_0) = (y_1, y_2, \dots, y_n)'$.

The μ is the overall *grand mean*. $Z(x)$ is n dimensional Gaussian process with $E(\mathbf{Z}(x_i)) = 0$, $Var(\mathbf{Z}(x_i)) = \sigma_z^2$ and $cov(\mathbf{Z}(x_i), \mathbf{Z}(x_j)) = \sigma_z^2 R_{ij}$, where R is correlation matrix of all \mathbf{x}_i . ε is the measurement error in the model as the Gaussian process might not be apt for all kinds of departures from the grand mean. Also, ε follows multivariate normal distribution with $E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma_\varepsilon^2 I$ where I is the identity matrix of size $n \times n$ and thus the errors are

independent of each other. We assume that $y(D_0)$ has a multivariate normal distribution, $N_n(\mathbf{1}_n\mu, \Sigma)$, where $\Sigma = \text{Var}(D_0|y(D_0)) = \sigma_z^2 R + \sigma_\varepsilon^2 I$ and $\mathbf{1}_n$ is a $n \times 1$ vector of all 1's. Also define a new vector \mathbf{r} , of correlation between various sampled design points, x_i and the unsampled design point, x_i^* , i.e. $\mathbf{r}(x^*) = [\text{corr}(x_1, x^*), \text{corr}(x_2, x^*), \dots, \text{corr}(x_n, x^*)]'$.

Likelihood Since new variance is $\Sigma = \sigma_\varepsilon^2 I + \sigma_z^2 R$, so define $\Sigma = \sigma_z^2 \left(R + \frac{\sigma_\varepsilon^2}{\sigma_z^2} I \right) = \sigma_z^2 R_1$. The negative profile log-likelihood in this GP model is proportional to,

$$-2 \log L_p \propto \log(|R_1|) + n \log[(Y - \mathbf{1}_n \hat{\mu}(\theta))' R_1^{-1} (Y - \mathbf{1}_n \hat{\mu}(\theta))].$$

Parameter Estimates The closed form estimates of the parameters, μ and σ_z^2 are as follows,

$$\begin{aligned} \hat{\mu}(\theta) &= (\mathbf{1}'_n R^{-1} \mathbf{1}_n)^{-1} (\mathbf{1}'_n R^{-1} Y), \text{ and} \\ \hat{\sigma}_z^2(\theta) &= \frac{(Y - \mathbf{1}_n \hat{\mu}(\theta))' R_1^{-1} (Y - \mathbf{1}_n \hat{\mu}(\theta))}{n}. \end{aligned}$$

Predictors Following the Best Linear Unbiased Predictor approach at any unsampled point x^* , the predictor is,

$$\hat{y}(x^*) = \hat{\mu} + \mathbf{r}' R_1^{-1} (\mathbf{Y} - \mathbf{1}_n \hat{\mu}).$$

The predictor $y(x)$ and the responses Y together follow multivariate normal distribution, i.e.

$$\begin{pmatrix} y(x) \\ Y \end{pmatrix} = N_n \left(\begin{pmatrix} \mu \\ \mu \mathbf{1}_n \end{pmatrix}, \begin{pmatrix} \sigma_z^2 + \sigma_\varepsilon^2 & \sigma_z^2 \mathbf{r}'(x) \\ \sigma_z^2 \mathbf{r}(x) & \sigma_z^2 R_1 \end{pmatrix} \right).$$

So, $E(y(x)|Y) = \mu + \mathbf{r}' R_1^{-1} (Y - \mathbf{1}_n \mu)$ and $\text{Var}(y(x)|Y) = \sigma_z^2 + \sigma_\varepsilon^2 - \sigma_z^2 \mathbf{r}'(x) R_1^{-1} \mathbf{r}(x)$.

1.4 Model: $Y = F\beta + Z(x) + \varepsilon$

Model Statement Consider the following: x_i is d dimensional i^{th} input vector; $\mathbf{x}_i = (x_{i1}, \dots, x_{id})'$. $y_i = y(x_i)$ is the univariate response variable. The input data matrix is $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ as set of n input vectors. Output of the simulation trials are stored in n dimensional vector as $\mathbf{Y} = y(X) = (y_1, y_2, \dots, y_n)'$.

The vector β is the Ordinary Least Square/Generalised Least Square estimate of true β . $Z(x)$ is n dimensional Gaussian process with $E(\mathbf{Z}(x_i)) = 0$, $\text{Var}(Z(x_i)) = \sigma_z^2$ and $\text{Cov}(Z(x_i), Z(x_j)) = \sigma_z^2 R_{ij}$, where R is correlation matrix of all \mathbf{x}_i .

ε gives a model with an additional measurement error. ε is has multivariate normal distribution with $E(\varepsilon) = 0$ and covariance matrix, $Var(\varepsilon) = \sigma_\varepsilon^2 I$, where I is identity matrix. The covariance after including ε in the model is,

$$Var(Y(x)) = \sigma_\varepsilon^2 I + \sigma_z^2 R.$$

Likelihood Since new variance is $\sigma_\varepsilon^2 I + \sigma_z^2 R$, so define $\Sigma = \sigma_z^2 R_1 = \sigma_z^2 \left(R + \frac{\sigma_\varepsilon^2}{\sigma_z^2} I \right)$. The negative profile log-likelihood in this GP model is proportional to,

$$-2 \log L_p \propto \log(|R_1|) + n \log[(Y - X\hat{\beta})' R_1^{-1} (Y - X\hat{\beta})].$$

Parameter Estimates The only parameter in this model is β . It is estimated using least square techniques. Its value is given by,

$$\hat{\beta} = (F' R^{-1} F)^{-1} F' R^{-1} Y, \text{ and}$$

$$\hat{\sigma}^2 = \frac{(Y - F\hat{\beta})' R_1^{-1} (Y - F\hat{\beta})}{n}.$$

Predictor Define

$$F = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}.$$

Also define a new vector \mathbf{r} , of correlation between various sampled design points, x_i and the unsampled design point, x_i^* , i.e.

$$\mathbf{r}(x^*) = [corr(x_1, x^*), corr(x_2, x^*), \dots, corr(x_n, x^*)]'$$

The predictor $y(x)$ and the responses Y together follow multivariate normal distribution, i.e.

$$\begin{pmatrix} y(x) \\ Y \end{pmatrix} = N_n \left(\begin{pmatrix} f(x)\beta \\ F\beta \end{pmatrix}, \begin{pmatrix} \sigma_z^2 + \sigma_\varepsilon^2 & \sigma_z^2 \mathbf{r}'(x) \\ \sigma_z^2 \mathbf{r}(x) & \sigma_z^2 R_1 \end{pmatrix} \right).$$

So, $E(y(x)|Y) = f(x)\beta + \mathbf{r}' R_1^{-1} (Y - F\beta)$ and $Var(y(x)|Y) = \sigma_z^2 + \sigma_\varepsilon^2 - \sigma_z^2 \mathbf{r}'(x) R_1^{-1} \mathbf{r}(x)$.

When $\varepsilon = 0$, there is no error in prediction, i.e. $MSE(x_i) = 0$ and the model is perfect interpolator at the design points, i.e. $Y(x_i) = y(x_i)$.

The BLUP can be written as

$$\hat{y}(x^*) = f(x^*)\hat{\beta} + \mathbf{r}'(x^*) R_1^{-1} (Y - F\hat{\beta}),$$

where $\hat{\beta} = (F'R_1^{-1}F)^{-1}F'R_1^{-1}Y$, i.e. usual generalised least square estimate. The first part in BLUP can be interpreted as least square prediction and the second part is the Gaussian process. So, an statistician can first get a regression model and then interpolate the residuals as if there were no regression model, and that would lead to this model.

2 Correlation Structures

2.1 Power exponential correlation

Structure of this correlation family is,

$$R_{ij} = \text{corr}(z(x_i), z(x_j)) = \prod_{k=1}^d \exp\{-\theta_k |x_{ij} - x_{jk}|^{p_k}\},$$

where θ is vector of hyper parameters. These are estimated while estimating other parameters of the model. The variable p_k is known as smoothness parameter and its value lies between $(0, 2]$. This model is most common in practice. When $p_k < 2$ the correlation structure is not differentiable at zero and the process is not mean square continuous. When $p_k = 2$, the process is infinitely continuous and differentiable (Kaufman et al., 2011).

Also, taking $p_k = 1$ would make the correlation structure which have first order derivative from one side. They are called Ornstein–Uhlenbeck process. For details see Sacks and Ylvisaker (1966) and Sacks et al. (1989). Integrating this process would give a process that is smoother than this but less smoother than when $p_k = 2$. This can be useful in situations where some differentiability is present and analyticity is in response variable (Sacks et al., 1989).

2.2 Gaussian correlation

Structure of this correlation is,

$$R_{ij} = \text{corr}(z(x_i), z(x_j)) = \prod_{k=1}^d \exp\{-\theta_k (x_{ij} - x_{jk})^2\},$$

where θ is vector of hyper parameters. Gaussian correlation is a specific form of power exponential correlation when $p_k = 2$. Linkletter et al. (2006) say that for most purposes, $p_k = 2$ and the if the responses suggest for $p_k < 2$ in the power exponential correlation, it would be due to numerical “jitters” and not due to model. ε usually takes care of such “jitters” and it isn’t advised to use $p_k < 2$ for that. Gaussian correlation is especially popular

because of its smoothness effect which is enabled by $p_k = 2$. Also, this makes the realisations of its functions infinitely differentiable, another desired effect when working on interpolation. Being differentiable the function gives us the ability to calculate rate of change, growth, etc.

2.3 Matern correlation

This correlation function was given by Santner et al. (2003). The correlation is defined as,

$$R_{ij} = \prod_{k=1}^d \frac{1}{\Gamma(\nu)2^{\nu-1}} (2\sqrt{\nu}|x_{ij} - x_{jk}|\theta_k)^\nu \kappa_\nu(2\sqrt{\nu}|x_{ij} - x_{jk}|\theta_k),$$

where κ_ν is modified Bessel function of order ν . This correlation was originally obtained by letting the parameter in the Gaussian correlation follow Gamma distribution which yielded a positive and spherically symmetric density proportional to R_{ij} and then finding that its Fourier transform was also a density (Guttorp and Gneiting, 2006).

2.4 Compactly supported correlation

Kaufman et al. (2011) use Gaussian process model but with a different model for correlation with compact support. They provide an algorithm which effectively brings in more zeros in the correlation matrix and then it is then efficiently manipulated using sparse matrix algorithms.

Compact support, mathematically, means that for some $\tau_k > 0$, $R_k(|x_k - x'_k|; \tau_k) = 0$ when $|x_k - x'_k| \geq \tau_k$. This has the effect of introducing zeros in the correlation matrix and thus makes it easy for computationally efficient sparse matrix techniques. They use compact support in *product form* of correlation matrix,

$$R(\mathbf{x}, \mathbf{x}'; \theta) = \prod_{k=1}^d R_k(|x_k - x'_k|; \theta_k).$$

For the *power exponential* correlation function, they list two methods.

1. Bohman: $R(t; \tau) = (1 - t/\tau) \cos(\pi t/\tau) + \sin(\pi t/\tau)/\pi$
2. Truncated power: $R(t; \tau, \alpha, \nu) = [1 - (t/\tau)^\alpha]^\nu$, with $0 < \alpha < 2, \nu \geq \nu_d(\alpha)$. The term $\nu_d(\alpha)$ represents a restriction so that the function is a valid correlation function, with $\lim_{\alpha \rightarrow 2} \nu_d(\alpha) = \infty$. (Golubov, 1981)

There can be some parallel drawn between Truncated power's α and power correlation's α . The truncated power function is not differentiable even once at the origin and corresponds

to a process which is not even mean square continuous, if $\alpha < 2$. When $\alpha = 2$, the function is infinitely differentiable but this is an unrealistic scenario with such high level of “smoothness”. Bohman function, however, is twice differentiable and is mean square differentiable.

The range, τ plays an important role in this approach. First, they control the degree of correlation in each dimension (like power correlation’s parameter, θ). Second, unlike θ , τ_k also controls the degree of sparsity in the matrix. For computational purposes and saving time, some additional restriction is needed which they apply through prior distributions.

3 Correlation Parametrisations

3.1 Original Parametrisation $\theta_k = \theta_k$

This is the original parametrisation in Gaussian Process model. This happens to be most popular approach with applied statisticians. The value of θ_k can be between $(0, \infty)$. This parametrisation can lead to wobbly correlation values especially when θ is close to zero. (MacDonald et al., 2015)

Formulation The correlation matrix with this parametrisation is defined as,

$$R_{ij} = \text{corr}(z(x_i), z(x_j)) = \prod_{k=1}^d \exp\{-\theta_k |x_{ij} - x_{jk}|^{p_k}\}.$$

3.2 Parametrisation $\theta_k = 10^{\beta_k}$

We can also use another parameterisation, $\theta_k = 10^{\beta_k}$, i.e. $\beta_k = \log_{10}(\theta_k)$ for all $k = 1, 2, \dots, d$. This benefits in easier likelihood optimisation as MacDonald et al. (2015) suggest. When θ_k is close to zero, the likelihood functions fluctuates rapidly. Taking 10^{β_k} parametrisation addresses this issue. For $\beta_k \ll 0$, there is very high spatial correlation. For $\beta_k \gg 0$, there is very low spatial correlation. The domain of this parameter, β_k is $(-\infty, \infty)$.

Formulation The correlation matrix with this parametrisation is defined as,

$$R_{ij} = \text{corr}(z(x_i), z(x_j)) = \prod_{k=1}^d \exp\{-10^{\beta_k} |x_{ij} - x_{jk}|^{p_k}\}.$$

3.3 Parametrisation $\theta_k = -2^{\alpha_k} \log(\rho_k)$

Linkletter et al. (2006) use parametrisation $\theta_k = -2^{\alpha_k} \log(\rho_k)$. The correlation function they used was,

$$\text{corr}(z(x_i), z(x_j)) = \prod_{k=1}^d \rho_k^{2^{\alpha_k} |x_{ij} - x_{jk}|^{\alpha_k}}.$$

Since $\theta > 0$, hence ρ_k lies between 0 and 1. They preferred this parametrisation of θ because this facilitated posterior exploration through Markov chain Monte Carlo (MCMC). Also, this makes the interpretation easier. If ρ_k is large (i.e. close to 1), the process does not depend on factor k . Therefore, estimation of the ρ_k s help us to determine which input variables are more important in the emulation.

3.4 Comparison

Comparing correlation values obtained in one dimension we get the following contour plots. In figure 1 power exponential property of Gaussian correlation is visible. Figure 2 gives us better correlation at the values with fewer zeros, though in essence they both look the same. Figure 3 has a different looking graph with is mainly because of its different parametrisation. For a given constant correlation and theta parametrisation, the relationship between θ and h is $\theta h^2 = k$, where k is a constant; for beta parametrisation, the relationship is $10^\beta h^2 = k$, where k is a constant. Both the contour plots are typical for such functions.

4 Simulation Study

4.1 Simulation Setup

As discussed in the section 1, we can have four different types of models for developing a suitable emulator. The models differ in their prediction values, likelihoods and mean squared errors. Moreover, there are three different correlation structures that could be used for prediction. Each of these correlation structures can use three different parametrisations for their hyperparameters, θ . So, we will have to choose between $4 \times 3 \times 3 = 36$ different models for any practical application.

Our aim is to train and test each of these models with various known “test functions” and help applied statisticians to choose between them. For the start, we begin with the previously known models as found in various different works in past. The variety of test functions we use would be of different input dimensions to test models better, since the models work for any

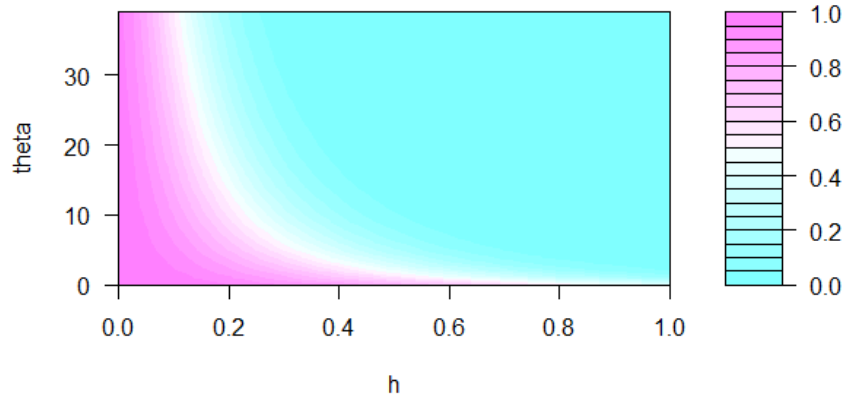


Figure 1: Contour plot grid between parameter, $\theta_k = \theta$ and $x_i - x_j$ in one dimension, $d=1$. Parameter θ varies from 0 to 40. $h = x_i - x_j$ vary from 0 to 1 with 40 values. So, 40×40 correlation values were calculated at all of these combinations.

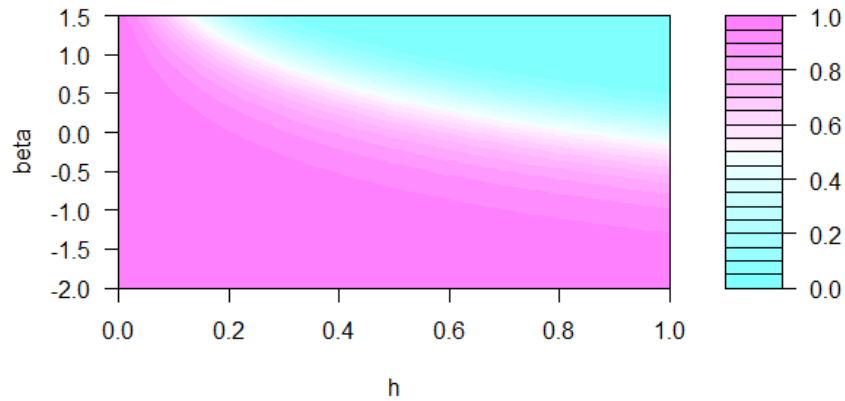


Figure 2: Contour plot grid between parameter, $\theta_k = 10^\beta$ and $x_i - x_j$ in one dimension, $d=1$. Parameter β varies from -1.5 to 2. $h = x_i - x_j$ vary from 0 to 1 with 40 values. So, 40×40 correlation values were calculated at all of these combinations.

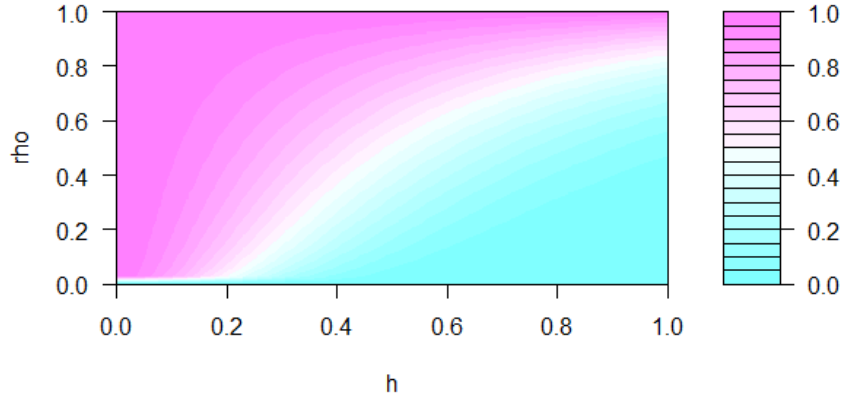


Figure 3: Contour plot grid between parameter, $\theta_k = -2^2 \log(\rho) = -4 \log(\rho)$ and $h = x_i - x_j$ in one dimension, $d=1$. Parameter ρ varies from -1 to 1 and each value at y-coordinate corresponds to the same. $x_i - x_j$ vary from 0 to 1 with 40 values. So, 40×40 correlation values were calculated at all possible combinations.

dimension of input. To compare between the models, we will be comparing Mean Squared Errors for both training and test dataset. The simulation trials would be done several times and then averaged to get better results.

The test test functions are of several applications. They can be together found at the <https://www.sfu.ca/~ssurjano/>. These test functions have been derived and used by their authors in various practical applications and thus make an apt choice for computer designed experiments. The functions found in the literature of emulation and prediction, can be further grouped as physical models, trigonometric models, exponential/logarithmic models, rational models, etc.

4.2 Goodness of fit measures

To test each model's fit, i.e. how well does the model emulates the simulator we will use Mean Squared Prediction Errors (MSPE). MSPE will be calculated for the test data. Then, comparisons will be drawn between the models. Mean Square Prediction Error (MSPE) defined as,

$$MSPE = \frac{(\hat{y}(x^*) - y(x^*))^2}{n - d}.$$

4.3 Correlation Simulation Setup

To find out which parameterisation we should choose for our purpose or for any general purpose application, we compare likelihoods for several values of θ , β and ρ , i.e. the three parameterisations studied earlier, and then, decide the parameter on the basis of least likelihood function. For simulating the results, 500 different equally-spaced values were taken from $\theta \in [0, 60]$, $\beta \in [-1.5, 3]$ and $\rho \in [0, 1]$. Likelihood at each of these points was calculated and then we plotted a likelihood for varying values of parameterisation. Ideally, the curve should be continuous and the value of parameter chosen which gives us the least value of deviance ($-2 \log(L_p)$). However, likelihood at all the values wasn't calculated because of one major reason. At various points the correlation matrix resulting from such parameterisation was near-singular. This was more is cases when θ was small, or β was small, or ρ was large. Although the machine epsilon value is 10^{-16} , which is a common choice for practitioners, in our code we decided ignore calculations for matrices whose determinant was less than 10^{-14} . So, at the remaining points, we had to find the parameter value through these likelihood plots. Finally, we had to choose a parameter that gave us a likelihood function which was easy to minimize over the complete domain of the parameter and gave us the least likelihood value.

In general, the likelihood for θ parameterisation appeared erratic at values of θ close to zero. β parameterisation is an improvement over θ parameterisation to solve this issue MacDonald et al. (2015). The deviance function for β parameterisation definitely was much smoother and looked easier to minimize. The ρ parameterisation looked most difficult to minimize. With increasing values of ρ , the deviance function was increasing. The plot looked very difficult to minimize in most cases.

We tried these parameterisations with two correlation structures – Gaussian correlation structure and Power-exponential correlation structure with $p = 1.95$. Theoretically, Gaussian correlation should look smoother because of it is infinitely differentiable. However, in the plots obtained we found Power-exponential with $p = 1.95$ smoother. This is perhaps because of the fact that Gaussian correlation matrix had much more near-singular conditions than Power-exponential correlation.

Finally, we tested for parameterisations in two different models – $y = \mu + z(x)$ and $y = \mu + z(x) + \epsilon$. The value of $\delta = \frac{\sigma_z^2}{\sigma_\epsilon^2}$ was set as 10^{-2} . In most cases, we found that the model with ϵ had almost the same likelihood curve for both values of p , i.e. for Gaussian correlation as well as Power-exponential correlation.

For this purpose, we started with the sinusoidal function first appeared in Currin (1988)

(hereafter referred to as `test1`). The function is defined as

$$y(x) = \sin(2\pi(x - 0.1)).$$

It takes in univariate input and gives univariate output. The function value of x is found in the range $x \in [0, 1]$. In this case, there were stark differences between Gaussian correlation and Power-exponential correlation with $p = 2$ (unless stated otherwise, consider all Power-exponential correlation values calculated using $p = 1.95$ in this document). When $p = 2$, i.e. Gaussian correlation function and model without ϵ , the optimum value of θ is close to 1; the optimum value of β is close to -1.3; the optimum value of ρ is close to 0.6. These three values are not very consistent with each other as $-4 \log(0.5) = 2.0433$ and $10^{-1.3} = 0.05011$. When we try the same thing for Power-exponential correlation structure without ϵ , we get θ close to 7; β close to 0.85 and ρ close to 0.2. These values are very much consistent with each other as $10^{0.85} = 7.0794$ and $-4 \log(0.2) = 6.437$. When the ϵ was introduced in the model, the optimum value of θ obtained was close to 9; β was close to 0.95; and ρ was close to 0.1. The values didn't change upon changing the power from $p = 2$ to $p = 1.95$. These values were consistent with each other as $10^{0.95} = 8.9125$ and $-4 \log(0.1) = 9.2103$.

We did the same treatment with several other functions. Another one dimensional test function was $y(x) = \frac{\sin(10\pi x)}{2x} + (x - 1)^4$. In this case for Gaussian correlation function, the graph was very smooth but difficult to minimize. When using Gaussian correlation function, optimum value of θ looks close to 60; optimum value of β looks close to 0; optimum values of ρ looks close to 0. Clearly, this doesn't seem to be an ideal fit. For Power-exponential correlation, the graph was much wobbly but easier to minimize. We speculate that this was perhaps because of fewer points for likelihood calculation in former than latter. The optimum value of θ in this case was close to 40, β close to 1.7 and ρ close to 0. Again, these values aren't consistent with each other. Also, the likelihood curves of ρ parametrisation are very bad for this function. Value of ρ close to zero tells us that there is essentially zero correlation which isn't plausible. Introducing ϵ in the model negated the effect of $p = 1.954$ or $p = 2$ and both the plots looked the same. The optimum values were: $\theta = 3$, $\beta = 0.4$ and $\rho = 0.5$. The values are fairly consistent with each other as: $10^{0.4} = 2.5118$ and $-4 * \log(0.5) = 2.7725$.

For third test function, $y(x) = \log(x + 0.1) + \sin(5\pi x)$, we did similar analysis. Again in this case too, we couldn't say for sure which parametrisation was the best. Using Gaussian correlation and no ϵ , we can say that optimum value for θ would be close 50, β would be close to -0.2 and ρ would be close to 0, again. All of this are with likelihood functions that are not easy to minimize. Using Power-exponential correlation, the optimum value of θ would be close to 45, β would be close to 1.7 and ρ would be close 0. Again, other than ρ , all of

these values are somewhat consistent with each other as $10^{1.7} = 50.1187$. When we included ϵ in the model, the effect of using $p = 2$ or $p = 1.95$ was lost. Both the panels of plots looked almost same. Optimum value of θ was 40, β was 1.6 and ρ was 0. Except ρ , they are consistent as $10^{1.6} = 39.810$.

For the fourth test function, $y(x) = \exp(-1.4x) \times \cos(3.5\pi x)$, the likelihood curves were smooth. For model without ϵ and $p = 2$, the optimised values were close to $\theta = 12$, $\beta = -0.7$ and $\rho = 0.7$. These values again, aren't very consistent with each other. When ϵ was included in the model, optimum value of θ was 15, β was 1.2 and ρ was about 0.02. All the three values in this case are consistent as $10^{1.2} = 15.8489$ and $-4 \log(0.02) = 15.64809$. After including ϵ in the model, like previous cases, effect of $p = 2$ or $p = 1.95$ was lost and the plots looked very much the same. Optimum values of θ is close to 19, β is close to 1.3 and ρ is close to 0.01. These values are also consistent with each other as $10^{1.3} = 19.9526$ and $-4 \log(0.01) = 18.42068$.

5 Preliminary Exploration

5.1 Comparing existing packages

Various data scientists have given different statistical packages for fitting data for Gaussian process and the predict the values. To know and compare the correctness of all of them, we do a small simulation exercise. We first train our model using 10 (and 50) design points randomly generated from Latin Hypercube using `maximinLHS` in R. Then, we calculate the true value with various functions. Then we generate another 20 points to test our model for accuracy of prediction. We compare the packages for Mean Square Prediction Error (MSPE) defined as,

$$MSPE = \frac{(\hat{y}(x^*) - y(x^*))^2}{n - d}.$$

We do this for several functions: two 1-d functions, one 2-d function, one 5-d function and one 8-d function.

The MSPE values are put together in a table 1. We can conclude from this that different packages give best results for different dimensions. For 1-d functions with fewer training points ($n = 10$), `mlegp` gives least errors; for more number of training points, `rgasp` gives best results. Conclusion for the 2-d function is also exactly the same. In higher dimensions, `rgasp` is still the winner, but accuracy of `gpfit` seems to be improving atleast for $n = 10$. For 8-d function with $n = 10$ training points `gpfit` is the clear winner.

Function		gfit	mleqp	rgasp
$y(x) = \log(x + 0.1) + \sin(5\pi x)$, $d = 1$	$n = 10$	0.001171601	0.0006340849	0.001732008
	$n = 50$	3.47998e-10	5.55666e-11	8.980093e-12
$y(x) = \exp(-1.4x) \times \cos(3.5\pi x)$, $d = 1$	$n = 10$	4.050959e-06	1.997998e-06	2.199637e-05
	$n = 50$	7.79009e-12	1.53732e-09	3.907107e-12
$y(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$, $d = 2$	$n = 10$	0.06556425	0.01986229	0.06462334
	$n = 50$	9.943076e-05	1.296255e-07	5.240136e-08
$y(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$, $d = 5$	$n = 10$	2.287962	2.420701	1.154225
	$n = 50$	0.009660818	0.0001395483	0.0007247992
$y(x) = \frac{4}{\sum_{i=4}^8} (x_1 - \frac{2}{i} + 8x_2 + 8x_2^2)^2 + (3 - 4x_2)^2 + 16\sqrt{x_3 + 1}(2x_3 - 1)^2$, $d = 8$	$n = 10$	8.010973	11.30924	11.84897
	$n = 50$	0.004879307	0.0011159	0.001621349

Table 1: Mean Squared Prediction Errors (MSPE) for different functions of different dimensions.

Acknowledgements

I sincerely and deeply thank Prof Pritam Ranjan sir for giving me this opportunity to learn. His throughout guidance has immensely enriched my knowledge in the subject. None of this would have been possible without his continuous and unending patience and support.

References

- Currin, C. (1988). A bayesian approach to the design and analysis of computer experiments. Technical report, ORNL Oak Ridge National Laboratory (US).
- Golubov, B. I. (1981). On abel—poisson type and riesz means. *Analysis Mathematica*, 7(3):161–184.
- Guttorp, P. and Gneiting, T. (2006). Studies in the history of probability and statistics xlix on the matérn correlation family. *Biometrika*, 93(4):989–995.
- Kaufman, C. G., Bingham, D., Habib, S., Heitmann, K., and Frieman, J. A. (2011). Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology. *The Annals of Applied Statistics*, pages 2470–2492.
- Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., and Ye, K. Q. (2006). Variable selection for gaussian process models in computer experiments. *Technometrics*, 48(4):478–490.
- MacDonald, B., Ranjan, P., and Chipman, H. (2015). Gpfit : An r package for fitting a gaussian process model to deterministic simulator outputs. 64.
- Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statist. Sci.*, 4(4):409–423.
- Sacks, J. and Ylvisaker, D. (1966). Designs for regression problems with correlated errors. *The Annals of Mathematical Statistics*, 37(1):66–89.
- Santner, T. J., Williams, B. J., and Notz, W. I. (2003). Space-filling designs for computer experiments. In *The Design and Analysis of Computer Experiments*, pages 121–161. Springer.