Uncertainty analysis: the variance of the variance

Tony O'Hagan

August 27, 2011

Abstract

When computing uncertainty analysis using a Gaussian process emulator, we often focus on the mean and variance of the uncertainty distribution. Using an emulator, we make inference about these quantities, and again it is usual to derive the emulator mean of each as an estimate. The natural expression for the accuracy of these estimates would be the corresponding emulator variances. However, hitherto we have had formulae for the emulator variance of the uncertainty mean but not for the emulator variance of the uncertainty variance. No valid expression for this quantity appears to have been published to date, and the purpose of this note is to fill that gap.

1 Background

1.1 Uncertainty analysis

Suppose that we have a simulator expressed as a function $f(x)$, where the simulator input x is in p dimensions but we consider only a single output, so that f is a scalar function. The idea of uncertainty analysis (UA) is that the inputs are random (or uncertain), so that we represent them as a random vector X with some distribution $g(x)$. Usually this distribution will be absolutely continuous with density $\omega(x)$, but in the most general case we can express expectations with respect to this distribution using the Stieltjes integral $\int \ldots dg(x)$, which becomes $\int \ldots \omega(x)dx$ in the absolutely continuous case.

Because of uncertainty about X, the simulator output is uncertain. The role of UA is to characterise the distribution of $f(X)$ that is induced by the distribution of X. Using $\mathrm{E}[\cdot]$ and $\mathrm{Var}[\cdot]$ to denote expectation and variance with respect to X, two summaries of the uncertainty distribution that are often used are the uncertainty mean and variance, respectively

$$M = \mathrm{E}[f(X)] \quad \text{and} \quad V = \mathrm{Var}[f(X)] \; .$$

We cannot evaluate these quantities analytically using the simulator, so we make inference about them using an emulator. We will use $\mathrm{E}^*[\cdot]$, $\mathrm{Var}^*[\cdot]$ and

$\text{Cov}^*[\cdot,\cdot]$ to denote the operations of expectation, variance and covariance with respect to the emulator. In this notation, the natural estimates of M and V are $\text{E}^*[M]$ and $\text{E}^*[V]$, while code uncertainty in these estimates is expressed in the corresponding variances $\text{Var}^*[M]$ and $\text{Var}^*[V]$. In release 7 and preceding releases of the MUCM toolkit, expressions have been given for $\text{E}^*[M]$, $\text{E}^*[V]$ and $\text{Var}^*[M]$, but there has as yet been no expression for $\text{Var}^*[V]$ in the toolkit. This is for the very simple reason that no formula has hitherto been published. (An early attempt is known to have been wrong.)

1.2 Outline of this paper

In this note, I will derive formulae for computing $\text{Var}^*[V]$. I will establish what I believe to be improved notation compared to that in the toolkit release 7 and will also present revised formulae for $\text{E}^*[M]$, $\text{E}^*[V]$ and $\text{Var}^*[M]$ using this new notation. The results will go into release 8 of the toolkit and this note will also be added to the toolkit as a supporting document to show their derivation.

In due course, I hope to extend the results to handle code uncertainty about the elements of a variance-based sensitivity analysis, but I want to expose the computations here to wider scrutiny first.

The formulae are developed in several stages. After discussing the most general computation method by simulated draws from the emulator, Section 2 assumes a Gaussian process (GP) emulator of $f(x)$, such as is obtained using the standard toolkit approach set out in ThreadCoreGP. In particular, the emulator is actually a t process, but in Section 2 it is allowed to have quite general mean and covariance functions. We then introduce a series of restrictions that correspond to the usual assumptions in ThreadCoreGP, which make the computations steadily much more efficient (although the formulae become more complex). In Section 3 we assume a linear mean and vague prior distributions. Section 4 then assumes a normal distribution for X and the Gaussian form of correlation function. Finally, Section 5 considers further specialisation to the case where both of these are associated with diagonal variance matrices, and also to a specific form of prior mean function.

2 Computation using a general GP or t process emulator

The most general way to compute $\text{E}^*[M]$, $\text{E}^*[V]$, $\text{Var}^*[M]$ and $\text{Var}^*[V]$ is to simulate realisations of the function $f(x)$, as described for instance in the toolkit page ProcSimulationBasedInference. For each such realisation, we can evaluate M and V, for example by taking a large Monte Carlo sample of values of x from $g(x)$. The sample of realisations of M and V then allow $\text{E}^*[M]$, $\text{E}^*[V]$, $\text{Var}^*[M]$ and $\text{Var}^*[V]$ to be evaluated. This is highly computationally intensive, requiring a large sample of realisations of $f(\cdot)$ and then for each realisation the realised values of $f(x)$ computed for a large sample of x values. This method works for just about any emulator, and may still be more efficient than running the simulator itself for a large sample of x values. However, for any GP-based emulator much more efficient computations are available.

In this section we suppose simply that the emulator mean function $m^*(x) = \mathrm{E}^*[f(x)]$ and covariance function $v^*(x, x') = \mathrm{Cov}^*[f(x), f(x')]$ can be readily evaluated for any x and x'. For $\mathrm{E}^*[M]$, $\mathrm{E}^*[V]$ and $\mathrm{Var}^*[M]$ this is all that we require and it is not necessary for the emulator actually to be a GP or a t process (tP). However, the formulae for $\mathrm{Var}^*[V]$ require fourth order moments, and in this case we assume that the emulator is a GP or tP.

2.1 Moments of M

First

$$\mathrm{E}^*[M] = \mathrm{E}^*[\int f(x)\ dg(x)] = \int \mathrm{E}^*[f(x)]\ dg(x) = \int m^*(x)\ dg(x)\ .$$

This can be evaluated with a single Monte Carlo computation, sampling values of x from $g(x)$ and averaging the resulting values of $m^*(x)$.

For the variance of M we have

$$\begin{aligned}\mathrm{Var}^*[M] &= \mathrm{Var}^*[\int f(x)\ dg(x)] = \int\int \mathrm{Cov}^*[f(x), f(x')]\ dg(x)dg(x') \\ &= \int\int v^*(x, x')\ dg(x)dg(x')\ .\end{aligned}$$

This can also be evaluated by a single Monte Carlo computation, but requires pairs of values to be sampled from $g(x)$.

2.2 Mean of V

Next consider

$$\mathrm{E}^*[V] = \mathrm{E}^*[M_2] - \mathrm{E}^*[M^2] = \mathrm{E}^*[M_2] - \mathrm{Var}^*[M] - \mathrm{E}^*[M]^2\ ,$$

where

$$M_2 = \mathrm{E}[f(X)^2] = \int f(x)^2\ dg(x)\ .$$

We have already obtained $\mathrm{E}^*[M]$ and $\mathrm{Var}^*[M]$, so we now need only consider

$$\begin{aligned}\mathrm{E}^*[M_2] &= \mathrm{E}^*[\int f(x)^2\ dg(x)] = \int \mathrm{E}^*[f(x)^2]\ dg(x) \\ &= \int v^*(x, x)\ dg(x) + \int m^*(x)^2\ dg(x)\ .\end{aligned}$$

Therefore, just two more Monte Carlo evaluations are required to complete the computation of $\mathrm{E}^*[V]$.

Before leaving this particular computation, it is of interest to look more deeply at the resulting computation.

$$\begin{aligned}\mathrm{E}^*[V] &= \left[\int m^*(x)^2\ dg(x)\ - \left\{\int m^*(x)\ dg(x)\right\}^2\right] \\ &\quad + \left[\int v^*(x,x)\ dg(x) - \int\int v^*(x,x')\ dg(x)dg(x')\right]\ .\end{aligned}$$

The first term in brackets on the right hand side is a computation of V obtained by substituting the emulator mean $m^*(x)$ for the simulator $f(x)$. This is perhaps a natural, if naive, way to obtain an estimate of V. It would be a correct computation if the emulator were perfect, but we see that in general the correct calculation has another term. Furthermore, the second term in brackets is necessarily positive. Any emulator in practice is imperfect in the sense that its variance is not zero everywhere, and the consequent code uncertainty will on average increase the variability of the function. This increase is the second term in brackets.

2.3 Variance of V — the GP case

Now we address the main purpose of this note. Making use again of the fact that $V = M_2 - M^2$ we have

$$\mathrm{Var}^*[V] = \mathrm{Var}^*[M_2] + \mathrm{Var}^*[M^2] - 2\mathrm{Cov}^*[M_2, M^2]\ . \tag{1}$$

Each of these terms requires the evaluation of fourth-order moments of $f(\cdot)$.

First suppose that the emulator is a GP, which for our purposes means that any set of points $\{f(x), f(x'), f(x''), f(x''')\}$ has a 4-dimensional multivariate normal (MVN) distribution. Fourth-order MVN moments are not very well known to statisticians but can be readily derived. In particular, the Wikipedia MVN page (http://en.wikipedia.org/wiki/Multivariate_normal_distribution) provides the results we need to start our derivation. Suppose that (W, X, Y, Z) have a MVN distribution with mean vector μ having elements μ_w, μ_x, μ_y and μ_z and with covariance matrix Σ having elements σ_{ww}, σ_{wx} etc. Then

$$\begin{aligned}E\{(W-\mu_w)^4\} &= 3\sigma_{ww}^2\ , \\ E\{(W-\mu_w)^3(X-\mu_x)\} &= 3\sigma_{ww}\sigma_{wx}\ , \\ E\{(W-\mu_w)^2(X-\mu_x)^2\} &= \sigma_{ww}\sigma_{xx} + 2\sigma_{wx}^2\ , \\ E\{(W-\mu_w)^2(X-\mu_x)(Y-\mu_y)\} &= \sigma_{ww}\sigma_{xy} + 2\sigma_{wx}\sigma_{wy}\ , \\ E\{(W-\mu_w)(X-\mu_x)(Y-\mu_y)(Z-\mu_z)\} &= \sigma_{wx}\sigma_{yz} + \sigma_{wy}\sigma_{xz} + \sigma_{wz}\sigma_{xy}\ .\end{aligned}$$

The last three of these can be used to produce the following (after a little algebra):

$$\begin{aligned}
Cov(X^2, Y^2) &= 2\sigma_{xy}^2 + 4\mu_x\mu_y\sigma_{xy} \ , \\
Cov(WX, Y^2) &= 2\sigma_{wy}\sigma_{xy} + 2\mu_w\mu_y\sigma_{xy} + 2\mu_x\mu_y\sigma_{wy} \ , \\
Cov(WX, YZ) &= \sigma_{wy}\sigma_{xz} + \sigma_{wz}\sigma_{xy} + \mu_w\mu_y\sigma_{xz} \\
&\quad + \mu_x\mu_z\sigma_{wy} + \mu_w\mu_z\sigma_{xy} + \mu_x\mu_y\sigma_{wz} \ .
\end{aligned}$$

We can now return to equation (1) and consider each term in turn. First

$$\begin{aligned}
\mathrm{Var}^*[M_2] &= \mathrm{Var}^*[\int f(x)^2 \ dg(x)] \\
&= \int\int \mathrm{Cov}^*[f(x)^2, f(x')^2] \ dg(x)dg(x') \\
&= 2\int\int v^*(x,x')^2 \ dg(x)dg(x') \\
&\quad +4\int\int m^*(x)m^*(x')v^*(x,x') \ dg(x)dg(x') \ .
\end{aligned}$$

This gives us two more integrals to evaluate. The next term is

$$\begin{aligned}
\mathrm{Var}^*[M^2] &= \mathrm{Var}^*[\{\int f(x) \ dg(x)\}^2] = \mathrm{Var}^*[\int\int f(x)f(x') \ dg(x)dg(x')] \\
&= \int\int\int\int \mathrm{Cov}^*[f(x)f(x'), f(x'')f(x''')] \ dg(x)dg(x')dg(x'')dg(x''') \\
&= 2\int\int\int\int v^*(x,x')v^*(x'',x''') \ dg(x)dg(x')dg(x'')dg(x''') \\
&\quad +4\int\int\int\int m^*(x)m^*(x')v^*(x'',x''') \ dg(x)dg(x')dg(x'')dg(x''') \\
&= 2\left[\int\int v^*(x,x') \ dg(x)dg(x')\right]^2 \\
&\quad +4\left[\int m^*(x) \ dg(x)\right]^2 \left[\int\int v^*(x,x') \ dg(x)dg(x')\right] \\
&= 2\mathrm{Var}^*[M]^2 + 4\mathrm{E}^*[M]^2\mathrm{Var}^*[M] \ .
\end{aligned}$$

Finally,

$$\begin{aligned}
\mathrm{Cov}^*[M_2, M^2] &= \mathrm{Cov}^*[\int f(x)^2 \; dg(x), \int\int f(x)f(x') \; dg(x)dg(x')] \\
&= \int\int\int \mathrm{Cov}^*[f(x)^2, f(x')f(x'')] \; dg(x)dg(x')dg(x'') \\
&= 2\int\int\int v^*(x,x')v^*(x,x'') \; dg(x)dg(x')dg(x'') \\
&\quad +4\int\int\int m^*(x)m^*(x')v^*(x,x'') \; dg(x)dg(x')dg(x'') \\
&= 2\int\int\int v^*(x,x')v^*(x,x'') \; dg(x)dg(x')dg(x'') \\
&\quad +4\mathrm{E}^*[M]\int\int m^*(x)v^*(x,x') \; dg(x)dg(x') \; ,
\end{aligned}$$

which introduces another two integrals to evaluate.

2.4 Summary of integrals required

Here we list the integrals required by the above formulae, naming them for reference in subsequent sections.

$$\begin{aligned}
\mathrm{E}^*[M] &= \int m^*(x) \; dg(x) \; , \\
\mathrm{Var}^*[M] &= \int\int v^*(x,x') \; dg(x)dg(x') \; , \\
I_1 &= \int v^*(x,x) \; dg(x) \; , \\
I_2 &= \int m^*(x)^2 \; dg(x) \; , \\
I_3 &= \int\int v^*(x,x')^2 \; dg(x)dg(x') \; , \\
I_4 &= \int\int m^*(x)m^*(x')v^*(x,x') \; dg(x)dg(x') \; , \\
I_5 &= \int\int\int v^*(x,x')v^*(x,x'') \; dg(x)dg(x')dg(x'') \; , \\
I_6 &= \int\int m^*(x)v^*(x,x') \; dg(x)dg(x') \; .
\end{aligned}$$

In terms of these integrals, we have

$$\mathrm{E}^*[V] = (I_1 - \mathrm{Var}^*[M]) + (I_2 - \mathrm{E}^*[M]^2) \; , \tag{2}$$

$$\begin{aligned}
\mathrm{Var}^*[V] &= 2(I_3 - 2I_5 + \mathrm{Var}^*[M]^2) \\
&\quad +4(I_4 - 2\mathrm{E}^*[M]I_6 + \mathrm{E}^*[M]^2\mathrm{Var}^*[M]) \; .
\end{aligned} \tag{3}$$

2.5 Variance of V — the tP case

The last of these formulae was derived under the assumption that the emulator (the distribution of $f(\cdot)$) is a GP. This will be appropriate if there were no uncertain hyperparameters in the prior GP specification, or if all hyperparameters have been fixed at plug-in estimates. It also applies if the prior mean function has the linear form $h(x)^T\beta$ and the hyperparameter vector β has a MVN prior distribution, as long as any unknown hyperparameters of the covariance function are fixed at plug-in estimates. However, if the covariance function has an unknown scalar multiplier σ^2 with a vague or inverse-gamma prior (which is part of the usual GP specification in the toolkit) then integrating out σ^2 makes the emulator a tP.

In order to identify the amendments required to the above analysis for the tP case, first notice that no change is necessary to the computations of $\mathrm{E}^*[M]$, $\mathrm{Var}^*[M]$ or $\mathrm{E}^*[V]$, save only that the covariance function $v^*(\cdot,\cdot)$ is the actual covariance function, which is accommodated in the MUCM toolkit through the estimate $\hat{\sigma}^2$ being the posterior mean of σ^2. In particular it is defined as a sum of squared residuals with a divisor $n-q-2$ rather than $n-q$.

It is only in respect of $\mathrm{Var}^*[V]$ that any modifications are required, because the fourth-order moments of a multivariate t distribution are not the same as for a MVN; the t distribution has higher kurtosis than the normal. We will address this by considering the $\mathrm{Var}^*[V]$ operation being conducted in two steps. Let $\mathrm{E}^+[\cdot]$ and $\mathrm{Var}^+[\cdot]$ denote expectation and variance with respect to the GP conditional on the scalar variance factor σ^2. Then $\mathrm{E}^+[V]$ and $\mathrm{Var}^+[V]$ have the forms given above for $\mathrm{E}^*[V]$ and $\mathrm{Var}^*[V]$ but with the interpretation that every occurrence of the covariance function $v^*(\cdot,\cdot)$ is now read as including the unknown σ^2 factor. Let $\mathrm{E}^\sigma[\cdot]$ and $\mathrm{Var}^\sigma[\cdot]$ denote expectation and variance with respect to σ^2, which we suppose to have an inverse chi-square distribution with d degrees of freedom. Then

$$\mathrm{Var}^*[V] = \mathrm{E}^\sigma[\mathrm{Var}^+[V]] + \mathrm{Var}^\sigma[\mathrm{E}^+[V]] \ . \tag{4}$$

Consider the first term of this expression. Wherever any of the individual terms that make up the formula (3) has a single σ^2 factor it should be replaced by $\hat{\sigma}^2$, the posterior expectation (using the divisor $d-2$), which simply means interpreting $v^*(\cdot,\cdot)$ again as the actual covariance function. However, the terms in the first part of (3), namely the I_3, I_5 and $\mathrm{Var}^*[M]^2$, all have σ^4 as a factor. The expectation of σ^4 is not $\mathrm{E}^\sigma[\sigma^2]^2 = \hat{\sigma}^4$ but $\hat{\sigma}^4(d-2)/(d-4) = \hat{\sigma}^4(1+\frac{2}{d-4})$.

Now consider the second term in (4). In equation (2) the first part has a σ^2 factor but the second does not. So $\mathrm{Var}^\sigma[\mathrm{E}^+[V]]$ has this first part squared with the resulting σ^4 term replaced by $\mathrm{Var}^\sigma[\sigma^2] = 2\hat{\sigma}^4/(d-4)$.

The result of these two findings is that to compute $\mathrm{Var}^*[V]$ in the tP case we have the sum of two parts. The first part is just (3) with $v^*(\cdot,\cdot)$ read as the actual covariance function of the tP (including the term $\hat{\sigma}^2$). The second part is due to the additional kurtosis, and is

$$\frac{2}{d-4}\left\{2(I_3 - 2I_5 + \mathrm{Var}^*[M]^2) + (I_1 - \mathrm{Var}^*[M])^2\right\} \ .$$

In practice this is likely to be a small correction term as long as a reasonable sized training dataset has been used, but it should be included in order to yield a correct computation of $\mathrm{Var}^*[V]$ in the tP case.

3 Linear mean and vague prior

The preceding section has expressed the UA quantities that we wish to compute in terms of the eight integrals listed in subsection 2.4. All of these can be evaluated by Monte Carlo sampling, which provides a much more efficient computational approach than sampling realisations of $f(\cdot)$. The next steps in the development provide further computational efficiency by finding closed form expressions for these integrals. In order for this to be possible, we need to make further assumptions about the form of the emulator.

We now assume that the emulator is built using a linear prior mean function $h(x)^T\beta$ and a prior covariance function $\sigma^2 c(x,x')$. Furthermore, we assign weak prior distributions to β and σ^2 and also assume that any hyperparameters δ in the correlation function $c(x,x')$ have known values (or in practice have been assigned values using plug-in estimation). Then after applying training data $y_i = f(x_i)$, $i = 1,2,\ldots,n$, the posterior distribution of $f(x)$ is a t process with $d = n - q$ degrees of freedom (where q is the dimension of β).

The posterior mean function is

$$\mathrm{E}^*[f(x)] = m^*(x) = h(x)^T\hat{\beta} + t(x)^T e \ , \tag{5}$$

where $\hat{\beta} = WG^Ty$, $W = (H^TA^{-1}H)^{-1}$, $G = A^{-1}H$, $e = A^{-1}(y - H\hat{\beta})$, A is the training data correlation matrix with elements $c(x_k, x_l)$, $t(x)$ is the correlation vector between x and the training data having elements $c(x, x_k)$ and H is the design matrix with rows $h(x_k)^T$.

The posterior covariance function is

$$\begin{aligned}\mathrm{Cov}^*[f(x), f(x')] &= v^*(x,x') \\ &= \hat{\sigma}^2[c(x,x') - t(x)^TA^{-1}t(x') + \{h(x) - G^Tt(x)\}^T W\{h(x') - G^Tt(x')\}] \ ,\end{aligned} \tag{6}$$

where $\hat{\sigma}^2 = (n-q-2)^{-1}y^T(I - GWG^T)y$ is the posterior mean of σ^2.

In this section, we substitute these expressions for $m^*(\cdot)$ and $v^*(\cdot,\cdot)$ into the eight integrals in order to express them in terms of a number of simpler integrals, which in turn will be evaluated in closed form in the next section.

3.1 The new integrals

We will evaluate the eight integrals in terms of the following:

$$
\begin{aligned}
R_h &= \int h(x) \ dg(x) \ , \\
R_t &= \int t(x) \ dg(x) \ , \\
R_{hh} &= \int h(x)h(x)^T \ dg(x) \ , \\
R_{ht} &= \int h(x)t(x)^T \ dg(x) \ , \\
R_{tt} &= \int t(x)t(x)^T \ dg(x) \ , \\
U &= \int\int c(x,x') \ dg(x)dg(x') \ , \\
U_h &= \int\int h(x)c(x,x') \ dg(x)dg(x') \ , \\
U_t &= \int\int t(x)c(x,x') \ dg(x)dg(x') \ , \\
U_{hh} &= \int\int h(x)c(x,x')h(x')^T \ dg(x)dg(x') \ , \\
U_{ht} &= \int\int h(x)c(x,x')t(x')^T \ dg(x)dg(x') \ , \\
U_{tt} &= \int\int t(x)c(x,x')t(x')^T \ dg(x)dg(x') \ , \\
\tilde{U} &= \int c(x,x) \ dg(x) \ , \\
S &= \int\int\int c(x,x')c(x,x'') \ dg(x)dg(x')dg(x'') \ , \\
\tilde{S} &= \int\int c(x,x')^2 \ dg(x)dg(x') \ .
\end{aligned}
$$

Some of these integrals were required in the toolkit up to release 7. The old release 7 symbols for these are: R_h is the old R^T; R_t is the old T^T; R_{hh} is the old Q_p; R_{ht} is the old S_p; R_{tt} is the old P_p; U is the old U (the only unchanged symbol!); $\tilde{U}$ is the old U_p. Although there are now fourteen integrals in place of eight, the significance of this step is that these integrals can be evaluated in closed form under the conditions in the next section.

3.2 Expressing eight integrals in terms of fourteen

Using (5) and (6) and expanding all the eight integral formulae, simple algebra yields the following results.

$$
\begin{aligned}
\mathrm{E}^*[M] &= R_h^T\hat{\beta} + R_t^T e \ , \\
\mathrm{Var}^*[M] &= \hat{\sigma}^2[U - R_t^T A^{-1} R_t + (R_h - G^T R_t)^T W (R_h - G^T R_t)] \ , \\
I_1 &= \hat{\sigma}^2[\tilde{U} - \mathrm{tr}\, A^{-1} R_{tt} + \mathrm{tr}\, W(R_{hh} - 2R_{ht}G + G^T R_{tt} G)] \ , \\
I_2 &= \hat{\beta}^T R_{hh}\hat{\beta} + 2\hat{\beta}^T R_{ht} e + e^T R_{tt} e \ ,
\end{aligned}
$$

$$
\begin{aligned}
I_3 = \hat{\sigma}^4[&\tilde{S} - 2\,\mathrm{tr}\, A^{-1}U_{tt} + \mathrm{tr}\, A^{-1}R_{tt}A^{-1}R_{tt} + 2\,\mathrm{tr}\, W(U_{hh} - 2U_{ht}G + G^T U_{tt} G) \\
&-2\,\mathrm{tr}\, A^{-1}(R_{ht} - G^T R_{tt})^T W (R_{ht} - G^T R_{tt}) \\
&+\mathrm{tr}\, W(R_{hh} - 2R_{ht}G + G^T R_{tt}G) W (R_{hh} - 2R_{ht}G + G^T R_{tt} G)] \ ,
\end{aligned}
$$

$$
\begin{aligned}
I_4 = \hat{\sigma}^2[&\hat{\beta}^T U_{hh}\hat{\beta} + 2\hat{\beta}^T U_{ht} e + e^T U_{tt} e \\
&-\hat{\beta}^T R_{ht} A^{-1} R_{ht}^T \hat{\beta} - 2\hat{\beta}^T R_{ht} A^{-1} R_{tt} e - e^T R_{tt} A^{-1} R_{tt} e \\
&+(R_{hh}\hat{\beta} - G^T R_{ht}^T \hat{\beta} + R_{ht} e - G^T R_{tt} e)^T \\
&\quad W(R_{hh}\hat{\beta} - G^T R_{ht}^T\hat{\beta} + R_{ht} e - G^T R_{tt} e)] \ ,
\end{aligned}
$$

$$
\begin{aligned}
I_5 = \hat{\sigma}^4[&S - 2R_t^T A^{-1} U_t + R_t^T A^{-1} R_{tt} A^{-1} R_t \\
&+2(U_h - G^T U_t)^T W (R_h - G^T R_t) \\
&-2R_t^T A^{-1}(R_{ht}^T - R_{tt}G) W (R_h - G^T R_t) \\
&+(R_h - G^T R_t)^T W (R_{hh} - 2R_{ht}G + G^T R_{tt} G) W (R_h - G^T R_t)] \ ,
\end{aligned}
$$

$$
\begin{aligned}
I_6 = \hat{\sigma}^2[&\hat{\beta}^T U_h - \hat{\beta}^T R_{ht} A^{-1} R_t + \hat{\beta}^T (R_{hh} - R_{ht} G) W (R_h - G^T R_t) \\
&+e^T U_t - e^T R_{tt} A^{-1} R_t + e^T (R_{ht} - G^T R_{tt})^T W (R_h - G^T R_t)] \ .
\end{aligned}
$$

4 Gaussian forms

In the toolkit pages for UA as they stand in release 7, two special cases are developed for evaluating the fourteen integrals defined above, i.e. the R, U and S series of integrals. The first special case is based on the following two assumptions:

- The correlation function $c(x, x')$ has the Gaussian form with nugget,

  $$c(x, x') = \nu I(x = x') + (1 - \nu)\exp\{-(x - x')C(x - x')\} \ ,$$

  where $I(x = x')$ is the indicator function that is 1 if $x = x'$ and is otherwise 0, C is in general a positive-definite matrix of correlation parameters and $\nu \in [0, 1]$ is a nugget. Both C and ν are assumed to be known or to have been specified by plug-in estimation. The case of no nugget is simply $\nu = 0$.

- The uncertainty in X is expressed in a normal distribution, $dg(x) = \omega(x)dx$ with

  $$\omega(x) = (2\pi)^{-p/2}\,|B|^{1/2}\exp\{-\tfrac{1}{2}(x-m)^T B(x-m)\} = (2\pi)^{-p/2}\,|B|^{1/2}\,\varphi(x)\ ,$$

  so that X has mean m and positive-definite precision matrix B.

4.1 The R integrals

The integrals R_h, R_t, R_{hh}, R_{ht} and R_{tt} are all one-dimensional and are the simplest to derive.

The simplest of all are R_h and R_{hh}. Let $\mathrm{E}_X[\cdot\,|\,m, B]$ denote the expectation with respect to a random variable X having the MVN distribution with mean m and *precision* matrix B. Then

$$\begin{aligned} R_h &= \mathrm{E}_X[h(X)\,|\,m, B]\ , \\ R_{hh} &= \mathrm{E}_X[h(X)h(X)^T\,|\,m, B]\ . \end{aligned}$$

Since we do not specify any form for $h(\cdot)$ at this point, these integrals cannot be simplified further.

For R_t and R_{ht}, we need to do some algebra. Consider

$$c(x, x_k)\varphi(x) = \nu I(x = x_k) + (1-\nu)\exp\{-Q_k(x)/2\}\ ,$$

where

$$\begin{aligned} Q_k(x) &= 2(x-x_k)^T C(x-x_k) + (x-m)^T B(x-m) \\ &= (x-m_k')^T(2C+B)(x-m_k') + Q_k(m_k') \end{aligned}$$

and where

$$m_k' = (2C+B)^{-1}(2Cx_k + Bm)\ .$$

Then the k-th element $R_t(k)$ of the vector R_t and the k-th column $R_{ht}(k)$ of the matrix R_{ht} are

$$\begin{aligned} R_t(k) &= (1-\nu)\,|B|^{1/2}\,|2C+B|^{-1/2}\exp\{-Q_k(m_k')/2\}\ , \\ R_{ht}(k) &= R_t(k)\mathrm{E}_X[h(X)\,|\,m_k', 2C+B]\ . \end{aligned}$$

Notice that the nugget term disappears in these integrals because the distribution $g(x)$ gives zero probability that $x = x_k$.

For the last of the R integrals, consider

$$c(x, x_k)\varphi(x) = a + (1-\nu)^2\exp\{-Q_{kl}(x)/2\}\ ,$$

where a is a term involving the nugget ν and the indicator functions for $x = x_k$ and $x = x_l$, which will disappear when we integrate with respect to $g(x)$, where

$$\begin{aligned} Q_{kl}(x) &= 2(x-x_k)^T C(x-x_k) + 2(x-x_l)^T C(x-x_l) + (x-m)^T B(x-m) \\ &= (x-m_{kl}')^T(4C+B)(x-m_{kl}') + Q_{kl}(m_{kl}') \end{aligned}$$

and where

$$m'_{kl} = (4C + B)^{-1}(2Cx_k + 2Cx_l + Bm) \ .$$

Then the (k, l)-th element of R_{tt} is

$$R_{tt}(k,l) = (1-\nu)^2 \, |B|^{1/2} \, |4C+B|^{-1/2} \exp\{-Q_{kl}(m'_{kl})/2\} \ .$$

4.2 The U integrals

For U, U_h and U_{hh} consider

$$c(x,x')\varphi(x)\varphi(x') = \nu I(x = x') + (1-\nu)\exp\{-Q/2\} \ ,$$

where

$$\begin{aligned} Q &= 2(x-x')^T C(x-x') + (x-m)^T B(x-m) + (x'-m)^T B(x'-m) \\ &= (\mathbf{x}-\mathbf{m})^T \mathbf{B}(\mathbf{x}-\mathbf{m}) \ , \end{aligned}$$

and where

$$\mathbf{x} = \begin{pmatrix} x \\ x' \end{pmatrix} \ , \quad \mathbf{m} = \begin{pmatrix} m \\ m \end{pmatrix} \ , \quad \mathbf{B} = \begin{pmatrix} 2C+B & -2C \\ -2C & 2C+B \end{pmatrix} \ .$$

Again the nugget term disappears because there is zero probability that $x = x'$. We have

$$\begin{aligned} U &= (1-\nu)\,|B|\,|\mathbf{B}|^{-1/2} \ , \\ U_h &= U \ \mathrm{E}_{\mathbf{X}}[h(X)\,|\,\mathbf{m},\mathbf{B}] \ , \\ U_{hh} &= U \ \mathrm{E}_{\mathbf{X}}[h(X)h(X')^T\,|\,\mathbf{m},\mathbf{B}] \ . \end{aligned}$$

For U_t and U_{ht} the next step in complexity is

$$c(x',x_k)c(x,x')\varphi(x)\varphi(x') = a + (1-\nu)^2 \exp\{-Q^u_k(\mathbf{x})/2\} \ ,$$

where a is a term involving the nugget ν and the indicator functions for $x' = x_k$ and $x = x'$, which will disappear when we integrate,

$$\begin{aligned} Q^u_k(\mathbf{x}) &= 2(x'-x_k)^T C(x'-x_k) + 2(x-x')^T C(x-x') \\ &\quad +(x-m)^T B(x-m) + (x'-m)^T B(x'-m) \\ &= (\mathbf{x}-\mathbf{m}'_k)^T \mathbf{B}_k(\mathbf{x}-\mathbf{m}'_k) + Q^u_k(\mathbf{m}'_k) \ , \end{aligned}$$

and where

$$\mathbf{m}'_k = \mathbf{B}_k^{-1} \begin{pmatrix} Bm \\ 2Cx_k + Bm \end{pmatrix} \ , \quad \mathbf{B}_k = \begin{pmatrix} 2C+B & -2C \\ -2C & 4C+B \end{pmatrix} \ .$$

Then the k-th element of U_t and the k-th column of U_{ht} are

$$\begin{aligned} U_t(k) &= (1-\nu)^2\,|B|\,|\mathbf{B}_k|^{-1/2} \exp\{-Q^u_k(\mathbf{m}'_k)/2\} \ , \\ U_{ht}(k) &= U_t \ \mathrm{E}_{\mathbf{X}}[h(X)\,|\,\mathbf{m}'_k,\mathbf{B}_k] \ . \end{aligned}$$

For U_{tt},

$$c(x, x_k)c(x', x_l)c(x, x')\varphi(x)\varphi(x') = a + (1-\nu)^3 \exp\{-Q^u_{kl}(\mathbf{x})/2\} ,$$

where a is a term involving the nugget ν and the indicator functions for $x = x_k$, $x' = x_l$ and $x = x'$, which will disappear when we integrate,

$$\begin{aligned} Q^u_{kl}(\mathbf{x}) &= 2(x-x')^T C(x-x') + Q_k(x) + Q_l(x') \\ &= (\mathbf{x} - \mathbf{m}'_{kl})^T \mathbf{B}_{kl} (\mathbf{x} - \mathbf{m}'_{kl}) + Q_k(m'_k) + Q_l(m'_l) , \end{aligned}$$

and where

$$\mathbf{m}'_{kl} = \begin{pmatrix} m'_k \\ m'_l \end{pmatrix} , \quad \mathbf{B}_{kl} = \begin{pmatrix} 4C+B & -2C \\ -2C & 4C+B \end{pmatrix} .$$

Then the (k,l)-th element of U_{tt} is

$$U_{tt}(k,l) = (1-\nu)^3 \, |B| \, |\mathbf{B}_{kl}|^{-1/2} \exp\{-(Q_k(m'_k) + Q_l(m'_l))/2\} .$$

The last of the U integrals is $\tilde{U}$, but since $c(x,x) = 1$ for all x we simply have

$$\tilde{U} = 1 .$$

This is the only one of the integrals for which the nugget term does not disappear.

4.3 The S integrals

The last two integrals are S and $\tilde{S}$. For S,

$$c(x, x')c(x, x'')\varphi(x)\varphi(x')\varphi(x'') = a + (1-\nu)^2 \exp\{-Q/2\} ,$$

where a is a term involving the nugget ν and the indicator functions for $x = x'$ and $x = x''$, which will disappear when we integrate,

$$\begin{aligned} Q &= 2(x-x')^T C(x-x') + 2(x-x'')^T C(x-x'') \\ &\quad +(x-m)^T B(x-m) + (x'-m)^T B(x'-m) \\ &\quad +(x''-m)^T B(x''-m) \\ &= \begin{pmatrix} x-m \\ x'-m \\ x''-m \end{pmatrix}^T \begin{pmatrix} 4C+B & -2C & -2C \\ -2C & 2C+B & 0 \\ -2C & 0 & 2C+B \end{pmatrix} \begin{pmatrix} x-m \\ x'-m \\ x''-m \end{pmatrix} . \end{aligned}$$

Then

$$S = (1-\nu)^2 \, |B|^{3/2} \begin{vmatrix} 4C+B & -2C & -2C \\ -2C & 2C+B & 0 \\ -2C & 0 & 2C+B \end{vmatrix}^{-1/2} .$$

Finally, the $\tilde{S}$ integral is the same as U, except that the first term in the quadratic form Q is doubled. Therefore

$$\tilde{S} = (1-\nu) \, |B| \begin{vmatrix} 4C+B & -4C \\ -4C & 4C+B \end{vmatrix}^{-1/2} .$$

5 Further simplifications

A common choice for the $h(\cdot)$ vector is $h(x)^T = (1, x^T)$. In this case we have

$$\mathrm{E}_X[h(X) \,|\, m, B] = \begin{pmatrix} 1 \\ m \end{pmatrix}, \quad \mathrm{E}_X[h(X)h(X)^T \,|\, m, B] = \begin{pmatrix} 1 & m^T \\ m & mm^T + B^{-1} \end{pmatrix}.$$

In the slightly more complex case where X is just the upper $p \times 1$ subvector of $\mathbf{X}$ then for $\mathrm{E}_{\mathbf{X}}[h(X) \,|\, \mathbf{m}, \mathbf{B}]$ and $\mathrm{E}_{\mathbf{X}}[h(X)h(X)^T \,|\, \mathbf{m}, \mathbf{B}]$ we use the above formulae but replace m by the corresponding upper $p \times 1$ subvector of $\mathbf{m}$ and B^{-1} by the upper-left $p \times p$ submatrix of $\mathbf{B}^{-1}$.

The formulae for the integrals can be evaluated even more explicitly when both B and C are diagonal. A diagonal B matrix corresponds to the elements of the input vector X being independent, while a diagonal C is a common choice for a GP emulator in the MUCM toolkit. The matrix computations in the formulae of Section 4 all involve $p \times p$ (or $2p \times 2p$ or $3p \times 3p$) matrices and so are of order p^3. When B and C are diagonal they can all be reduced to order p computations. However, unless p is larger than we usually try to build GP emulators for, even the order p^3 computations are quick. (In contrast, the order n^3 computation of A^{-1} requires much more computing resource.) Although explicit formulae for this case (called special case 2) are given in the toolkit up to release 7, the formulae are messy and become even messier in the new integrals needed for $\mathrm{Var}^*[V]$. I think it is probably not worth giving these formulae in release 8 because in practice it's probably not worth anyone coding them up when the matrix versions are easier and run fast enough.