Gaussian process emulation of stochastic models: developments and application to rabies modelling

Keywords: stochastic emulation, Gaussian processes, sensitivity analysis, disease modelling

Abstract

Complex simulators, based on mechanistic and physical process driven models, are widely used in many scientific and engineering problems. Simulators are incomplete representations of the systems they represent, and thus their rational treatment requires probabilistic methods. The complexity of simulators necessitates Monte Carlo methods in most cases, however their computational complexity renders this problematic. Recent work in statistics has developed methods for constructing meta-models, or emulators, for simulators. Emulators are typically implemented as Gaussian process regression models which approximate the simulator mapping using a finite set of design points. Very little work has been carried out on the emulation of stochastic simulators. In this paper, motivated by a stochastic simulator of Rabies in a two species disease model, we extend existing machine learning Gaussian process methods and apply these to stochastic emulation. We extend the most likely heteroscedastic Gaussian process regression method. We go on to discuss the emulation of a probabilistic outcome from the simulator, the probability of disease extinction within 5 years of an outbreak, which is a key disease management indicator. We also show how feature selection methods can help us better understand the importance of different factors in controlling the probability of disease survival. We conclude with speculation on where machine learning, statistics and simulation modelling can benefit from interaction to further our understanding of complex simulators and systems.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction and Motivation

In many scientific and engineering problems complex simulators, based on mechanistic and physical process driven models, are routinely used to solve complex problems (Law, 2007). Such simulators are often computationally expensive, and full uncertainty analysis, sensitivity analysis or other probabilistic analysis becomes extremely time consuming, if not impossible (Saltelli et al., 2000). The most commonly applied solution is to create a meta-model for the simulator (Sacks et al., 1989), often referred to as an *emulator* (Kennedy & O'Hagan, 2001; Oakley & O'Hagan, 2004). The role of the emulator can be related to approximating the, typically complex, prior defined by the simulator. In most existing work the emulator methods are applied to deterministic models, of the form $\mathbf{y} = \mathbf{f}(\mathbf{x})$ where $\mathbf{x}$ represents the inputs to the simulator, $\mathbf{y}$ represents the outputs for the simulator, or some summary of these, and $\mathbf{f}$ represents the mapping imposed by the simulator evaluation. The probabilistic nature of the emulator, which is typically modelled as a Gaussian process (Kennedy & O'Hagan, 2001), arises from the *approximation* of the simulator due to having a finite number of simulator runs. In this paper we develop methods for the emulation of a stochastic simulator, a relatively new field (Kleijnen, 2007), and propose several novel machine learning based solutions.

The paper starts with a review of emulation of stochastic simulators. We then describe the stochastic Rabies model we emulate in the paper. We present two aspects of emulating a stochastic simulator; emulating the first two moments of the simulator output, and emulating a summary statistic (the probability of disease extinction within a given time) using Gaussian Process (GP) methods. We provide summary results and show how input relevance determination can be employed in stochastic emulation. We conclude with a discussion and some suggestions for future work.

2. Stochastic emulation

Assuming we have a deterministic simulator, $\mathbf{y} = \mathbf{f}(\mathbf{x})$, emulation can be understood by following the algorithm:

1. choose a set of n experimental design points, $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, covering the input space $\mathbf{x}$; typically Latin Hyper-cube is used, but others are possible;
2. evaluate the simulator at the n design points to produce the 'training set', $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$;
3. fit a (GP) regression model (the emulator) to the training data, without noise;
4. use the emulator as a surrogate for the simulator to perform uncertainty analysis, sensitivity analysis, etc.

The essence of the idea is to replace the computationally expensive simulator with the more efficient emulator, and use this to undertake, the typically Monte Carlo based, probabilistic analyses. The main distinction between emulation and classical regression is that is becomes far easier to consider active design, and sequential refinement, to the training set in a manner very similar to query learning (Campbell et al., 2000). After training, the GP posterior (emulator) is an exact (noise-free) interpolator at the design points (O'Hagan, 2006). Away from the design points the GP variance models the additional uncertainty that arises from the use of the emulator, and should be factored into any subsequent analysis (Oakley & O'Hagan, 2004).

Stochastic emulation is a less well explored field. Kleijnen and co-workers (Kleijnen, 2007; van Beers & Kleijnen, 2008) have studied the problem closely, investigating queuing models which are inherently stochastic, that is random numbers are used internally in the simulator. In the work of Kleijnen a relatively simple approach to emulating stochastic simulators is developed which uses m repetitions of the simulator at each of the i design points. From this the mean response $\bar{y}_i = \frac{1}{m}\sum_{j=1}^{m} \mathbf{y}_{i,j}$ and the variance of the response $\mathbf{S}_i = \frac{1}{m-1}\sum_{j=1}^{m}(\bar{y}_i - \mathbf{y}_{i,j})^2$ are computed, where $\mathbf{y}_{i,j}$ is the j'th realisation from the stochastic simulator, at the i'th design point. Kleijnen and van Beers (2005) are mainly concerned with modelling the mean response of the stochastic simulator, and use the variance estimates, $\mathbf{S}_i$ to 'Studentize' the output with the transformation $\tilde{\mathbf{y}}_i = \frac{\bar{y}_i}{\sqrt{\mathbf{S}_i/m^2}}$, where they assume $\mathbf{y}$ has had any 'large scale' trend removed. A standard GP regression of the transformed output, $\tilde{\mathbf{y}}_i$, is then applied. A novel element in their approach is the use of a bootstrapping procedure that approximates a full Bayesian treatment of hyper-parameters in the covariance function. This bootstrap is used to obtain correct prediction variances, which are then used in an optimal sequential design method. The allowance for heteroscedastic variance is limited to a small number of simple parametric models.

Bates et al. (2006) approaches stochastic emulation from another aspect. Their concern is with so called 'robust design' where 'design' is used in the engineering context of product design and 'robust' in a quite general manner. The stochastic nature of the emulation arises from so called *noise factors* which are aspects of the engineering design problem that cannot be controlled, for example in simulating the impact of a collision on an object, the angle of collision cannot be controlled, and thus the design must be 'robust' with respect to this input. Robust is equated with minimum variance in the simulator output with respect to changes in the *noise factor* inputs. The modelling approach adopted is to create an emulator for the output of the simulator using a Latin hyper-cube design with a small number of points. This emulator includes both *design factors* and *noise factors* in the inputs. The emulator is then used to generate a series of realisations across a new set of design points which cover only the *design factors*, with replications being sampled from the *noise factors*. In this manner a set of estimates for the mean and variance of the output, $\bar{y}_i$ and $\mathbf{S}_i$, are determined for each input, $\mathbf{x}_i$, at the second level experimental design points. Emulators, based on zero mean GPs are developed for the mean and variance of the outputs independently and then used to solve an optimisation problem to minimise the variance of the output, given the mean is constrained to a certain value.

In all the work on stochastic emulation very little attention is paid to a rigorous treatment of heterogeneity of the output variance. In this paper we extend the recent work of (Kersting et al., 2007) and related work on GP regression with heteroscedastic noise (Goldberg et al., 1998) to enable improved stochastic emulation of a rabies disease simulator.

3. Stochastic Rabies Model

Wildlife rabies was eradicated from large parts of Europe, however it remains endemic in some Eastern European countries. Thus, contingency strategies for rabies elimination have to be developed in case of disease reintroduction into a currently rabies-free country. Such strategies will differ from those used to elimi-

nate endemic rabies (Smith et al., 2008). Due to a non-native but invading species – the raccoon dog (Nyctereutes procyonoides Gray) – rabies risk is enhanced in Eastern and Central Europe. Even areas where low densities of the traditional major European wildlife host – the red fox (Vulpes vulpes L.) – impeded rabies epizootics, the presence of both species could sustain epidemic or endemic rabies (Holmala & Kauhala, 2006; Singer et al., 2008). Models that demonstrate this increased risk may be very dependent on parameter values, which are notoriously difficult to estimate, and thus these models require rigorous evaluation if they are to inform rabies contingency planning.

We use a simulation model (Singer et al., 2008) to analyse the risk and strength of rabies spread in a community of raccoon dogs and foxes as a test case for different emulation and sensitivity analysis methods. The individual-based (agent-based) non-spatial model tracks population and disease dynamics on a seasonal time step (3 months). Reproduction and natural mortality (including hunting pressure) of host animals is simulated.

The processes are parametrised using data from field studies in Finland (Kauhala et al., 2006) and Poland (Goszczynski, 2002). For some of the parameters, the studies indicate a considerable range of possible values. This is typical for ecological processes and is caused by measurement uncertainty in difficult and often time-limited field studies, uncontrolled external parameters (such as weather) and intrinsic biological variability (e.g. individual variability or behaviour). Distinguishing between the sources of variation is not currently possible. To reflect natural variability, the population and disease processes are modelled as stochastic processes. Thus, variation in field data is captured as intrinsic model variability.

In terms of sensitivity analysis, however, this modelling approach ignores external parameter uncertainty, which would be partly hidden by internal variability, both in the natural, and the model system. Nevertheless, knowledge on effects of parameter variation remains important. The information can help to focus sampling effort in the field, improve understanding of model outcome or assess model applicability to different ranges of the parameter space. For the purpose of a sensitivity analysis, we can derive suitable parameter variation, even if external variation of parameters is not clearly quantifiable from field data.

In this example, we assumed potential shifts in parameter space that could be caused by a change in habitat conditions.All parameters (except for Fox Density and Rac Density) were varied by ±10% and are listed in Table 1. The variation was adjusted such that all parameters remain in their well-defined domain. For simplicity, a uniform distribution is assumed for variation of all parameters. Parameter Dummy is not used in the model. Thus, by definition, sensitivity analysis methods should find this parameter to be unimportant.

Table 1. Parameters of the fox raccoon dog rabies model.

Name	Standard	Min.	Max.
Area Size	5400	4860	5940
Fox Density	0.2	0.1	0.5
Rac Density	0.3	0.1	1
Rac Inf Prob	0.43	0.39	0.47
Dummy	1	0.9	1.1
Fox Death	1	0.9	1.1
Rac Death	1	0.9	1.1
Win Hunt Prop	1	0.9	1.1
Fox Birth	1	0.9	1.1
Rac Birth	1	0.9	1.1
Fox Inf	1	0.9	1.1
Fox Rabid	1	0.95	1.05
Rac Rabid	1	0.95	1.05
Cross Inf	1	0.9	1.1

4. Emulating the Stochastic Output

In the first instance we emulate a single output of the model, the number of time steps required for the disease to become extinct in the Raccoon Dog population. This output is important in deciding on the response to a potential rabies outbreak. We note this output has a rather complex, non-Gaussian, distribution; in this paper we emulate the log extinction time, which is more approximately Gaussian, as evidenced from visual inspection of Q-Q plots.

We base our approach on Kersting et al. (2007) where a coupled combination of GPs is constructed to model heteroscedastic noise, since exploratory analysis revealed that the level of variability in the output is input dependent. In Section 4.1 we present a modification of Kersting et al. (2007) to take into account the additional uncertainty of using a finite sample from the GP posterior to train the GP modelling the output variance. Additionally, in the context of stochastic emulation it is possible to produce replicate output observations, $\mathbf{y}_{i,j}$, at the price of running the simulator multiple times for the input $\mathbf{x}_i$. Thus a direct estimation of the output mean, $\bar{y}_i$, and the output variance, $\mathbf{S}_i$, can be used in building the emulator, as shown in Section 4.2.

Following Kersting et al. (2007), we define $\mathbf{G_1}$ and $\mathbf{G_3}$ as the GPs on the mean response and $\mathbf{G_2}$ the

Gaussian Process on the variance of the response. We do not present the full GP inference framework here but note that most experiments employed the GPML code (Rasmussen & Williams, 2006) with maximum marginal likelihood estimation of hyper-parameters.

4.1. Uncertainty in the variance estimation

In Kersting et al. (2007) the log variance is computed by sampling the posterior of $\mathbf{G_1}$ N times and computing the variance for each design point. Due to finite sample size effects this is a biased estimate. Standard theory (Cox & Solomon, 2003) allows us to estimate the bias and variance of the estimator:

$$r = log(\mathbf{S}) + (d + d\log(2) - \Psi(d/2))^{-1}, \qquad (1)$$

where r is the true log variance, $\mathbf{S}$ is the sample variance estimate, $d = N-1$ and Ψ the digamma function. The uncertainty of the estimate of the log variance can also be estimated using standard theory (Cox & Solomon, 2003):

$$\sigma_{\mathbf{S}} = \Psi_1(d/2), \qquad (2)$$

where Ψ_1 is the trigamma function, i.e. the derivative of the digamma function.

These corrections can be applied directly to the estimation of $\mathbf{G_2}$ by using (1) to correct the sample log variance for each design point. The corresponding uncertainty of the log variance estimates can be included in the likelihood of $\mathbf{G_2}$ using (2). The main rationale for suggesting these improvements is to make the method more robust to smaller sample sizes, N, and thus improve the efficiency of training the GPs.

4.2. Utilizing repeated observations

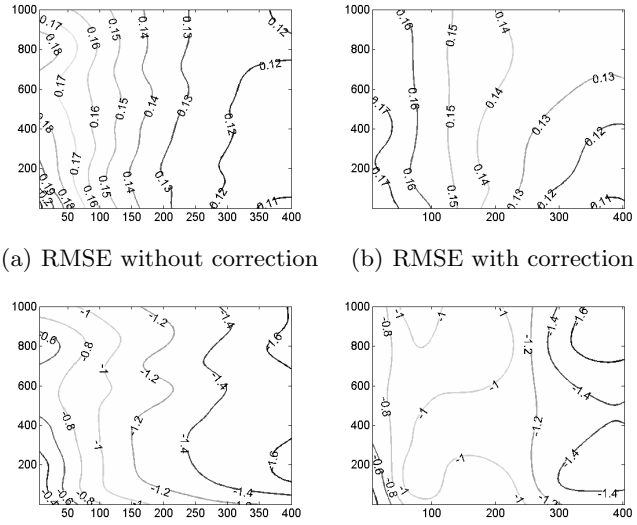
When repeated observations are available, sampling from the posterior of the GP is no longer necessary. Rather we use a loosely coupled pair of GPs. From the repeated simulator realisations, the sample mean output $\bar{y}_i$ and corrected sample variance, $\mathbf{S}_i$ is calculated for each design point i, as in Section 2. $\mathbf{G_1}$ is used to provide an initial estimate of the hyper-parameters for $\mathbf{G_3}$, although this is not strictly required. $\mathbf{G_2}$ is trained on the corrected log variance (Section 4.1) where we replace N with m, the number of simulator realisations at each design point. We note here that typically $m \ll N$ since the simulator is computationally expensive to run. The Rabies model used in this study is comparatively cheap requiring approximately 4 minutes per realisation for the configuration used. $\mathbf{G_2}$ is trained accounting for the noise on the variance computed using (2), which is particularly important in the small sample case where the second moment estimates can be quite sensitive. This allows $\mathbf{G_2}$ to smooth the variance estimates based on the prior GP specified, and produces more reliable estimates of the underlying noise variance. As in Kersting et al. (2007) we use the noise levels estimated by $\mathbf{G_2}$ when inferring $\mathbf{G_3}$. Conceptually we should then iterate, using $\mathbf{G_3}$ to compute the mean, when estimating the variance prior to fitting $\mathbf{G_2}$ again, however in practice we found this was not necessary.

4.3. Results

One of the most important issues in emulation is the significant expense of running the simulator, which dominates the computational cost. An interesting design question that we explore here is whether one should prefer a space filling design, with only a single simulator realisation at each point, or a less dense space filling design with a larger number of realisations at each design point. Table 2 shows that for this model a mixed strategy of a space filling design with a few realisations (HetGPD1 and HetGPDC1) achieves the best compromise in terms of test set performance and predicted variance. This is best explained by realising that the GP prior acts as a local smoother and is thus effective in combining local estimates to produce reliable estimates of the local mean and thus also variance. Using only 5 simulator realisations the use of the finite sample size correction (HetGPDC1) to the variance improves the fit of the emulator for the variance in particular, as would be expected. It also appears to improve the robustness of the GP posterior prediction when using repeated simulator realisations.

Table 2. Comparison of all the methods for 1000 simulator runs, using Kerstings heteroscedastic GP (HetGP) approach and corrected HetGP (HetGPC) with 200 samples used in the algorithm to estimate the variance, and various combinations of design points and simulator realisations for the coupled GPs using simple (HetGPD) and corrected estimators (HetGPDC). HetGPD1 and HetGPDC1 are developed using 200 design points $\times$ 5 simulator realisations on these points and HetGPD2 and HetGPDC2 are developed using 100 design points $\times$ 10 simulator realisations. In all cases we use 1000 simulator runs and average over 5 training set realisations.

Methods	RMSE	NLL	RMSEvar	Time (seconds)
HetGP	0.12 ± 0.02	-1.08 ± 0.98	0.13 ± 0.01	6×10^3
HetGPC	0.11 ± 0.02	-0.61 ± 0.15	0.12 ± 0.05	1×10^4
HetGPD1	0.12 ± 0.02	-0.83 ± 0.40	0.14 ± 0.10	30
HetGPDC1	0.12 ± 0.03	-0.72 ± 0.17	0.12 ± 0.05	83
HetGPD2	0.16 ± 0.01	-0.88 ± 0.36	0.16 ± 0.05	6
HetGPDC2	0.16 ± 0.01	-0.96 ± 0.15	0.16 ± 0.05	16



(a) RMSE without correction (b) RMSE with correction

(c) NLL without correction (d) NLL with correction

Figure 1. Comparison of coupled method (as described in Section 4.2) with and without variance correction when emulating the stochastic Rabies simulator. The plots show the response as a function of the number of design points (x-axis) and realisations of the simulator (y-axis).

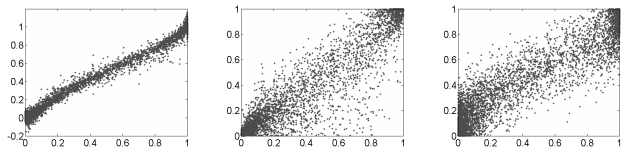
In Figure 1 we show the effect of the finite sample size corrections, together with correctly using the variance of the variance estimates to learn the emulator for the Rabies model. The contour plots show the Root Mean Square Error (RMSE) for the prediction of the mean, and the Negative Log Likelihood (NLL) both computed on a large independent test set (1500 design points), averaged over 5 training set samples. This shows that the dominant factor in our ability to emulate the stochastic output is the number of design points in the training set. For large numbers of design points, and also for large numbers of simulator realisations the results are similar, since the effect of the finite sample size corrections is small, but in the critical region for practical emulation, near the origin on the plots, the corrected process is far more stable.

5. Probabilistic Output

In Section 4 we described a framework to model the scalar output describing the extinction time of the Raccoon Dog population. Survival analysis also includes analysis of short term fixed horizon effects. In the particular example of the Rabies model, a key management metric is the probability that rabies will become extinct within a fixed time horizon within the simulator.

Modelling a probability requires a different approach since the output is constrained in the range $[0, 1]$ to be a valid probability. We have implemented three approaches. The first two approaches work directly on estimates of the probability of disease extinction $p_i = f_i/m$ where f_i is the number of times the disease died out in the m simulator realisations at input x_i. Using GP Regression (GPR) the output is unbounded but the approach is useful to benchmark other approaches. We also employ Logistic GP Regression (LGPR) which uses the logistic function to build a latent space representation of the probabilistic output. A GP is fit in the latent space and the output is projected into the probability space using the logit link function. For probabilities that exactly equal 0 or 1 a small ϵ is added to avoid infinities in the latent space.

The third approach is to treat the task as a classification problem. The outputs of the model are now binary, $\beta_i = \{-1, 1\}$, defining whether the disease survived or not, and we use a Sparse Probit GP (SPGP) model based on the online GP framework (Csato, 2002) with a probit link function. The probit link function is more appropriate here, since the classification problem lacks hard boundaries. The SPGP approach has the benefit that single realisations of the simulator can be used to better cover the input space, compared to the GPR and LGPR approaches which require several simulator realisations for each design point.



(a) GPR (b) LGPR (c) SPGP

Figure 2. Predicted probability of disease extinction (y-axis) against estimated probability, p_i (x-axis).

In the experiments we fit the GP for each method using 1250 training design points. The frequency estimate of the probability for the GPR and LGPR approaches was calculated using $m = 2000$ repetitions at each design point. For the SPGP, 10 realisations per design point were used, the total labelled data thus being 12,500 points. Thus the SPGP uses many orders of magnitude fewer simulator evaluations.

The GPR achieved the best overall performance in terms of RMSE, as can be seen from Figure 2. The LGPR and SPGP models behave similarly, despite the later using only a fraction of the simulator runs. In this model it seems that for many inputs the probabilities are one and zero, and in this case the small ϵ that is

Table 3. Ranking of input factors for the probabilistic output using the approaches described in the text.

Sobol	CCA	MAVE 2D	GPR
Fox Density	Fox Density	Fox Density	Area Size
Rac Density	Rac Density	Rac Density	Fox Density
Fox Death	Fox Death	Fox Death	Rac Density
Rac Death	Rac Death	Rac Death	Rac Inf Prob
Fox Birth	Fox Birth	Fox Birth	Dummy
Rac Rabid	Fox Inf	Fox Inf	Fox Death
Cross Inf	Rac Birth	Rac Birth	Rac Death
Area Size	Fox Rabid	Rac Rabid	Win Hunt Prop
Fox Rabid	Dummy	Fox Rabid	Fox Birth
Rac Birth	Win Hunt Prop	Win Hunt Prop	Rac Birth
Fox Inf	Rac Rabid	Cross Inf	Fox Inf
Dummy	Area Size	Area Size	Fox Rabid
Rac Inf Prob	Cross Inf	Rac Inf Prob	Rac Rabid
Win Hunt Prop	Rac Inf Prob	Dummy	Cross Inf

selected to represent the cut-off is rather critical. The SPGP approximation allows us to use a large number of design points, using the projected process approximation with expectation propagation (Csato, 2002) and retaining only 100 active points (or basis vectors), although training time is significantly high.

5.1. Input relevance

In many modelling applications a frequently asked question is about the relative importance of input factors. In machine learning this is typically called feature relevance (selection), and in statistics it is often called sensitivity analysis (screening). In this application the disease modellers are interested in better understanding the importance of the inputs for the probabilistic output in particular, to better understand the response of their Rabies model and plan management strategies.

The probability of disease extinction converges to a fixed value as the sample size increases and we therefore treat it here as a deterministic quantity. This allows the application of the Sobol sensitivity analysis method which assumes determinism. The Sobol method (Saltelli et al., 2000) allows the computation of main effects and total effects for each input factor. The latter include interaction effects across factors and are shown in Table 3. The main effects are not provided for brevity but the significant divergence in ranking that is observed indicates strong interaction effects between factors. The Sobol indices were calculated on a set of 20032 design points each using 4000 simulator runs, i.e using ∼80 million simulator evaluations. The ranking was in close agreement with expert knowledge and we use it later as the correct ranking to evaluate other methods. The Sobol method is computationally demanding and thus we have explored other methods to address the issue of feature relevance determination.

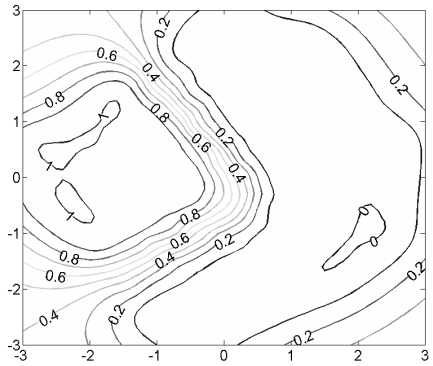


Figure 3. MAVE projection of the probabilistic response onto a 2D manifold.

Canonical correlation analysis (CCA) (Krzanowski, 1988) is a well known method, closely related to principal components analysis, where the association between two sets of variables is calculated by removing within set correlation through a linear transformation. In our single output application, CCA effectively projects the input space on a single hyperplane. The magnitudes of the factor loadings are used as a measure of input relevance. We have also applied the Minimum Average Variance Estimation (MAVE) supervised dimension reduction method (Xia et al., 2002) which is based on the sliced inverse regression method. Both CCA and MAVE used 2,500 design points with $m = 2000$ realisations, i.e. ∼5 million simulator evaluations. The MAVE projection of the 14 inputs to a two dimensional latent space is shown in Figure 3. This shows that the probabilistic response is structured and that this structure can be represented on a two dimensional latent space. This provides further confidence that emulation of the probabilistic output is realistic.

Lastly all the GP emulators used in our experiments implement automatic relevance determination (Neal, 1996) where each input factor is associated with a length scale hyper-parameter. In the rankings in Table 3 we show the result from the best fitting emulator, which was the GPR shown in Figure 2.

As can be seen in Table 3, the MAVE ranking using a two dimensional latent space best approximates the Sobol ranking, capturing the most important factors, despite the presence of strong interaction effects. Lower ranked features have effects of similar magnitude, and thus are expected to be less consistently ordered. The GPR based ranking is quite poor and demonstrates that caution should be exercised when

using length scales to determine the relevance of inputs with strong interaction effects, even though a reasonable emulator fit has been achieved. This requires further investigation to assess the impact of interaction effects and the reliability of automatic relevance determination based methods when non-trivial numbers of inputs are considered for complex response functions.

6. Discussion and Conclusions

Emulation of stochastic simulators is a complex problem. Although not reported here, significant exploratory analysis was initially carried out to determine the nature of the stochasticity. In the present work we are relatively fortunate that after a log transformation the simulator output is sufficiently well marginally approximated by a Gaussian that we could directly apply a GP. For other models, or outputs, this need not be the case, and in these situations a warped GP (Snelson et al., 2004) and related trans-Gaussian kriging methods (De Oliveira et al., 1997) might prove useful. The challenge in stochastic emulation is to maximise the accuracy of the emulator for a given number of simulator evaluations. The introduction of some known finite sample size corrections to the variance estimator, together with the corresponding uncertainty estimates allowed us to create coupled mean and variance GPs using only small numbers of simulator realisations which performed well.

A natural extension of this work would be to consider the problem from a sequential experimental design viewpoint where active learning, or query learning is used to select the next design point, for example using informative vector machine (Lawrence & Platt, 2004) like approaches. An additional complexity that is faced in stochastic emulation is the ability to replicate simulator realisations at design points. It would be quite simple to extended the coupled GP models to allow the inclusion of both direct simulator output and summary statistics derived from multiple simulator evaluations. It might be that such as design has desirable properties in terms of robustness and ability to estimate hyper-parameters accurately; we are currently working on this issue. In the original paper Kersting et al. (2007) had also extended the heteroscedastic model to include a projected process approximation and we are also exploring the use of the online GP algorithms developed by Csato (2002) with the aim to integrate the sequential experiment design into the training method.

Emulation of the probabilistic output has significant room for improvement. We believe part of the error of the classification based approaches (LGPR and SPGP) can be attributed to the squashing effect of the link function. In the Rabies model, a significant area of the input space behaves deterministically even under large number of repeated experiments, i.e. for specific ranges in the parameter space the disease almost surely dies out or respectively survives. When viewed from a classification view point, these design points are quite distant from the decision boundary and thus in the tails of the activation functions, be it either the probit or logistic. However the squashing effect in the latent space is quite disruptive for the smoothness and stationarity of the response thus requiring a more complex emulator model to fit accurately. We note that in the experiments reported here the squared exponential covariance function was used in all cases, and a more careful choice of covariance function could lead to better results, particularly given our beliefs about the latent space representation.

A sequential experimental design strategy is also possible for the probability output, starting with an initially space filling design and subsequently placing design points with an emphasis on the boundary region that is defined as being critical by the requirements for Rabies control. One approach would be to create a hybrid classification / regression emulator using single and repeated simulator evaluations in conjunction with appropriate mixed likelihood models.

When undertaking feature relevance determination the MAVE method proved effective. The applicability of such results reaches beyond sensitivity analysis since constraining the input space to a lower dimensional (linear) manifold allows more efficient experimental design. Thus by combining feature selection, or projection onto a subspace, we are able focus on regions of the model input space that are most relevant to determining the model outputs. This will result in a lower dimensional space in which to design our experiments and fewer hyper-parameters to estimate. Feature selection can be combined with all the methods discussed above to further improve the efficiency of stochastic emulation.

Overall we have presented several contributions toward the integration of machine learning approaches and emulation for stochastic models. This analysis presents the first results of the application and extension of GP methods developed in machine learning to stochastic emulation. In further work we intend to develop these methods into a more robust, practical set of tools that can be used widely across a range of models and applications.

References

Bates, R., Kennett, R., Steinberg, D., & Wynn, H. (2006). Achieving robust design from computer simulations. *Quality Technology and Quantitative Management*, *3*, 161–177.

Campbell, C., Cristianini, N., & Smola, A. (2000). Query learning with large margin classifiers. *Proc. 17th International Conf. on Machine Learning* (pp. 111–118). Morgan Kaufmann, San Francisco, CA.

Cox, D. R., & Solomon, P. J. (2003). *Components of variance*. Chapman and Hall CRC.

Csato, L. (2002). *Gaussian processes - iterative sparse approximations*. PhD Thesis, Aston University.

De Oliveira, V., Kedem, B., & Short, D. A. (1997). Bayesian prediction of transformed Gaussian random fields. *Journal of the American Statistical Association*, *92*, 1422–1433.

Goldberg, P. W., Williams, C. K. I., & Bishop, C. M. (1998). Regression with input-dependent noise: A Gaussian process treatment. *Advances in Neural Information Processing Systems*. The MIT Press.

Goszczynski, J. (2002). Home ranges in red fox: territoriality diminishes with increasing area. *Acta Theriologica*, *47*.

Holmala, K., & Kauhala, K. (2006). Ecology of wildlife rabies in Europe. *Mammal Review*, *36*, 17–36.

Kauhala, K., Holmala, K., Lammers, W., & Schregel, J. (2006). Home ranges and densities of medium-sized carnivores in southeast Finland with special reference to rabies spread. *Acta Theriologica*, *51*, 1–13.

Kennedy, M., & O'Hagan, A. (2001). Bayesian calibration of computer models (with discussion). *Journal of the Royal Statistical Society*, *B63*, 425–464.

Kersting, K., Plagemann, C., Pfaff, P., & Burgard, W. (2007). Most likely heteroscedastic Gaussian process regression. *Proc. 24th International Conf. on Machine Learning* (pp. 393–400). Omnipress.

Kleijnen, J., & van Beers, W. (2005). Robustness of Kriging when interpolating in random simulation with heterogeneous variances: Some experiments. *European Journal of Operational Research*, *165*, 826–834.

Kleijnen, J. P. C. (2007). Kriging metamodeling in simulation: a review. *European Journal of Operational Research*.

Krzanowski, W. (1988). *Principles of multivariate analysis*. Oxford Science Publications.

Law, A. (2007). *Simulation modeling and analysis*. McGraw Hill.

Lawrence, N. D., & Platt, J. C. (2004). Learning to learn with the informative vector machine. *Proc. 21st International Conf. on Machine Learning*. Omnipress.

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

Oakley, J., & O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society*, *B66*, 751–769.

O'Hagan, A. (2006). Bayesian analysis of computer code outputs: a tutorial. *Reliability Engineering and System Safety*, *91*, 1290–1300.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press.

Sacks, J., Welch, W., Mitchell, T., & Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, *4*, 409–435.

Saltelli, A., Chan, K., & Scott, E. (2000). *Sensitivity analysis*. Wiley: New York.

Singer, A., Kauhala, F., Holmala, K., & Smith, G. (2008). Rabies epidemiology in a community of foxes and raccoon dogs. In press.

Smith, G., Thulke, H., Fooks, A., Artois, M., Macdonald, D., Eisinger, D., & Selhorst, T. (2008). What is the future of rabies control in Europe? *Developments in Biologicals, In press.*

Snelson, E., Rasmussen, C. E., & Ghahramani, Z. (2004). Warped Gaussian processes. In S. Thrun, L. Saul and B. Schölkopf (Eds.), *Advances in neural information processing systems 16*. MIT Press.

van Beers, W., & Kleijnen, J. P. C. (2008). Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *European Journal of Operational Research*, *186*, 1099–1113.

Xia, Y., Tong, H., Li, W., & Zhu, L. (2002). An adaptive estimation of dimension reduction space. *Journal of the Royal Statistical Society B*, *64*, 363–410.