FAST SPATIAL INTERPOLATION USING SPARSE GAUSSIAN PROCESSES

Ben Ingram

Neural Computing Research Group, Aston University
Aston Triangle, Birmingham. B4 7ET, The United Kingdom
Correspondence to: ingrambr@aston.ac.uk

Lehel Csató

Dept. of Empirical Inference, Max Planck Inst. for Biological Cybernetics
Spemannstrasse 38, 72076 Tübingen, Germany
Correspondence to: csatol@tuebingen.mpg.de

David Evans

Neural Computing Research Group, Aston University
Aston Triangle, Birmingham. B4 7ET, The United Kingdom
Correspondence to: d.j.evans@aston.ac.uk

The estimation of the natural ambient radioactivity in this entry to the Spatial Interpolation Comparison 2004 (SIC2004) uses Gaussian processes (GP's) to predict the underlying dispersal process. GP's enable us to predict easily levels of radioactivity at previously unseen locations and in addition they allow us to assess the uncertainty in the predicted value. To speed up computation time, which is cubic in the number of examples, a sequential, sparse implementation of the Gaussian process inference (SSGP) was used together with a Gaussian observational noise assumption. The examination of the available data led to a covariance function which is a mixture of exponential and squared-exponential functions. The mixture was chosen so that it incorporates both the local ambiguity in the data and also at the same time it captures the larger-scale variation of the observations.

A further characteristic of the competition was the availability of 10 days of "prior observations". The individual sets comprised of data that was recorded at the same time but at different locations across Germany. Consequently in the modelling stage we assumed that the underlying process governing the observations was invariant across the 10 days of "prior observations", but that each day, i.e. each dataset, should be treated independently. These prior observations were used to infer the parameters of the covariance function, which are called hyper-parameters. Conforming to the Bayesian inference method, the exact values of the hyper-parameters were not fixed; rather we assigned a probability distribution to these parameters, giving a two level inference scheme. The inferred values of the hyper-parameters were used to set up a prior GP, and the same two level inference was employed when evaluating the method on the released SIC2004 dataset.

We show that the sparse approximation with the SSGP method can be used instead of the conventional GP's without significant loss in accuracy leading to a greatly reduced calculation time. We also compare the SSGP performance with

standard machine learning techniques: the SSGP results compare favourably to the other benchmark techniques.

1. INTRODUCTION

In this article we use Gaussian process (GP) inference, also known as Kriging methods in geostatistics (Cressie 1993). The advantage of GP inference lies primarily in its non-parametric nature and the intrinsic ability to produce probabilistic predictions at any location, seen or unseen. The theoretical flexibility is hindered by the amount of computation needed to obtain the predictions: in the GP algorithm we need to compute the inverse covariance matrix of the data, which is equal in size to the size of the data. The rapid analysis of data in geostatistics is problematic because of the computation of this inversion. The matrix is not only prohibitively slow to invert but also requires a significant amount of computer memory storage. A number of different solutions have been adopted to overcome this problem, such as partitioning the data into regions or sub-sampling the data. These methods are rather ad-hoc in their nature, lack a sound theoretical basis and it is unclear what information is being lost by their application.

In this submission we employ the Bayesian GP framework – also known as Kriging – to estimate the radioactivity level. These estimates are obtained from a latent global posterior GP: the mean function of the posterior process is the estimate of the radioactivity level and the variance of the GP marginal is the uncertainty associated to this predicted value (Cressie 1993).

One of the objectives of SIC2004 is to generate results in the shortest amount of time. Instead of the conventional GP method with a single large matrix inversion, we propose to use a *sequential estimation* scheme. In this sequential scheme we consider a single element of the dataset at a particular time and build a sequence of intermediate posteriors so that the last one is an approximation to the global posterior. The speed improvement is achieved by a subsequent modification to the above sequential inclusion: at the end of a step the possibility of removing some input locations from the representation of the posterior process. The removal is such that it guarantees that the posterior *still* includes the maximum amount of "information" about the current data item. This iterative removal leads to a modified *sparse* posterior which relies merely on a subset of the training inputs, making the operations to scale cubically only with the size of this subset (Cornford 2004).

The model selection is an important issue in the GP framework. Ideally one should choose a model – the class of priors – which reflects the underlying phenomenon. Similarly, knowledge about the data collection process should be encoded in the likelihood function. There was no specific information about the data collection method, thus we assumed additive Gaussian noise i.e. a Gaussian likelihood function with the exact value of the noise variance being inferred in the algorithm. With the Gaussian noise assumption GP inference becomes analytically tractable.

Within the GP inference (and geostatistics) a covariance function, also known as a kernel function in the machine learning community, is chosen to model the variation in the data. Kernel functions specify the relation between neighbouring input locations: how much these two observations resemble each other, or how much influence a given data point has on its "neighbours". They are analogous with covariance functions as used in classical geostatistics for describing spatial variation.

We aim at modelling the natural ambient radioactivity measured in the environment. According to existing studies, the radioactivity typically displays noisy and complex spatial patterns related to altitude, geology and weather (Hall 1984; UIC Melborne 2004). No additional knowledge relating to these environmental factors or their relationship to the data were given to participants. Since there is no principled way in the machine learning community to determine the "best" kernel function; and since a kernel function is usually selected from a small subset of pre-specified functions where often intuition and experience are employed, we decided that we will use the variogram of the "prior datasets" to help determine an appropriate choice of kernel function that could best model the data.

Our choice is a mixture of kernel functions which best represents the structure of the sample variogram shown in Figure 2. We see that the squared exponential kernel function alone is not an appropriate model to fit this sample variogram. One reason to discard this covariance function is that within the datasets there are insufficient observations at short distances for us to infer the process behaviour at this range. Since the locations where estimates are required for the contest are often at smaller separation intervals, and since we believe that there is very little variation at this level, we decided a squared exponential kernel is most appropriate to model the local behaviour. However, the long range variation of the data appears to be modelled more accurately by a simple exponential kernel function, hence a mixture of these two kernel function has been used.

2. METHODOLOGY

In this section we describe the Bayesian methodology used in obtaining the results for the SIC2004 exercise. We are using Gaussian processes in our experiments since they provide elegant non-parametric solutions to the interpolation problem. First we present the general framework to obtain a posterior process, the difficulties arising when dealing with large datasets and then we present a solution to overcome these difficulties. We then turn to the specifics of the SIC2004 data and the modelling issues we investigated, namely the choice of the kernel function for the radiation data.

The Bayesian approach to modelling is an attempt to utilise all available information in order to form a realistic model. In doing so, prior knowledge such as experience, expert knowledge or previous datasets can all be taken into account. We start with a brief introduction of the probabilistic Bayesian inference.

We assume that we aim at estimating a function $f(x, w)$ where $x \in R^n$ is the set of inputs to the function and *w* is a set of parameters that are unknown to us prior to seeing the data. We nevertheless have some information about the range of these parameters and also know that, within this range, some values of *w* are more probable than others; we encode this knowledge into a "prior distribution" of the parameters, which will be denoted by $p_0(w)$. The next step in Bayesian inference is the specification of the likelihood function, i.e. the noise model when collecting the data. Thus, for the data set containing *N* examples – sampled independently where $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ we have the following likelihood:

$$P(D \mid w)=\prod_{n=1}^{N} P\left(y_n \mid f(x_n, w)\right) \qquad (1)$$

where $P\left(y_n \mid f(x_n, w)\right)$ is the distribution of the noise. We further assume that the observation noise is Gaussian, with zero mean and variance sigma squared, hence we have:

$$P\left(y_n \mid f(x_n, w), \sigma^2\right)=\exp\left(-\frac{1}{2\sigma^2}\left(y_n - f(x_n, w)\right)^2\right). \qquad (2)$$

The computation of the posterior process is done according to Bayes' rule (Bernardo and Smith 1994), thus the a-posteriori probability of the parameters *w* becomes:

$$p_p(w \mid D)=\frac{P(D \mid w) p_0(w)}{\int P(D \mid w) p_0(w)\, dw} \qquad (3)$$

where the denominator – also called the evidence $p(D)=\int P(D \mid w) p_0(w)\, dw$ – is the normalising factor to the posterior distribution. To solve equation (3) apart from knowing the prior $p_0(w)$, we have to know the function *family* $f(x, w)$, i.e. the parametric form of the approximator. This is similar to fixing the degree of the polynomial, the number of coefficients in the Fourier expansion, or the architecture of the neural network. This choice is a rather difficult one to make.

Within the Gaussian process (GP) framework the previous two steps are unified: we specify priors over the functions themselves. This leads to a non-parametric setup and gives more freedom in making predictions. The advantage is that often – if the prior is properly chosen – the complexity of the resulting GP is driven automatically by the data without need for user intervention. Technically, GP's can be viewed as assigning a *joint Gaussian* distribution for every finite collection of function values (Papoulis and Pillai 2002). If we assume input locations $X=\left\{x_1, \ldots, x_N\right\}$, then the joint probability of $f_X=\left[f(x_1), \ldots, f(x_N)\right]^T$ is:

$$p_0(f_X)=\frac{1}{\left(\sqrt{2\pi}\right)^{\frac{N}{2}} \left|K_N\right|^{\frac{1}{2}}} \exp\left[-\frac{1}{2} f_X^T K_N^{-1} f_X\right] \qquad (4)$$

where $K_N=\left\{K_0(x_i, x_j)\right\}_{i,j=1}^{N}$ is the kernel function, which acts as the main parameter to the Gaussian process Schoelkopf and Smola 2002.

To obtain the predictive probability at a test location x_*, one first considers the joint distribution of the function values at the inputs and the test point, builds the joint posterior of these values using Bayes' rule, and finally integrates out the function values at the training locations – which are random variables – to arrive at the probability of the function value at x_*. Since both the prior and the likelihood is Gaussian, the *predictive* distribution of $f_* = f\left(x_*\right)$ is a Gaussian with mean and variance:

$$\begin{aligned}\mu_* &= \sum_{i=1}^{N} K_0(x_*, x_i)\,\alpha_i \\ \sigma_*^2 &= K_0(x_*, x_*) - \sum_{i,j=1}^{N} K_0(x_*, x_i)\, c_{ij}\, K_0(x_j, x_*)\end{aligned} \tag{5}$$

with a_i and c_{ij} being scalar coefficients and they are computed as $\underline{\alpha} = \left(\sigma_0^2 I_N + K_N\right)^{-1} \underline{y}$, $C = \left(\sigma_0^2 I_N + K_N\right)^{-1}$ where we use underlining to define a vector and capital letters for matrix quantities. We mention that the above equations are simply a different expression of the best linear predictor for the mean and variance within the Kriging framework, as used by geostatisticians (Cressie 1993 pp. 105-120).

As it was mentioned, the choice of the kernel function replaces the choice of the function family. A commonly used kernel is the squared-exponential kernel which has the following form:

$$K_0(x, x') = A \exp\left(-\frac{1}{2\theta^2}\left\|x - x'\right\|^2\right) \tag{6}$$

where hyper-parameters (A, θ) characterise the resulting process and they have to be chosen to provide "best fit" to the data. The amplitude, i.e. the typical range of variation in the output values is specified by A and θ specifies the length-scale of the process or the distance to which two observations would influence each other. In this article we employ the maximum likelihood (MLII) framework to set the kernel parameters (MacKay 1992; MacKay 1999). The presentation of the automated choice of the kernel parameters is not the aim of this submission; here we only give an intuitive picture – one can consult (e.g.) Williams and Rasmussen (1996), Gibbs and MacKay (1997) for details.

In the MLII framework the (log-) of the evidence in equation (3) is used to find the optimal kernel parameters. To see this, let us consider θ, the length-scale of the kernel function. With a fixed length-scale the computation of the evidence is a convolution of Gaussians, thus analytically computable. Having an analytic expression for the (log-) evidence $P(D \mid \theta)$, we maximise it with respect to θ. In MLII we consider the evidence term as the "likelihood" to the hyper-parameters and we perform a maximum a-posteriori parameter optimisation by adding a prior and differentiating the resulting expression.

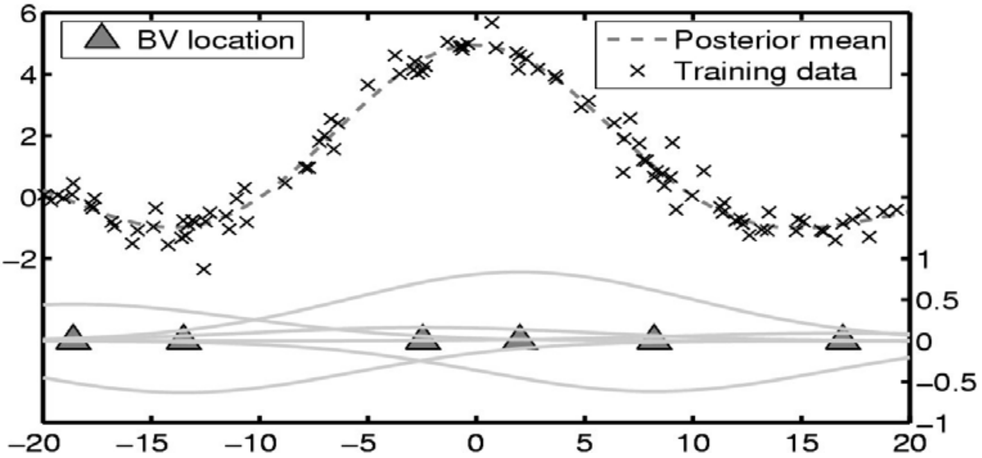

Figure 1
The sparse GP approximation to the posterior process: with conventional GP the approximated posterior mean function is a linear sum of RBF functions at all training points (x signs), with SSGP the sum is relative to the BV set (triangles) and the sum of the 6 RBF functions leads to an accurate approximation of the function – the dashed line on the top is the approximation provided by the posterior mean.

Using a fixed set of hyper-parameters we compute the posterior process. Once computed, we readily have estimates for both the function itself and the uncertainty in its value at a given point. The predictive mean is used in a maximum a-posteriori setting to estimate the unknown function, i.e. taking the mean of the posterior distribution as the approximation to the function (Gibbs and MacKay 1997). We highlight the main drawback of the GP inference: although analytically appealing, it cannot be used on large datasets due to the high computational cost of the matrix inversion.

In recent years there has been an increased of interest in techniques for approximation that take advantage of the concept of sparsity (Williams and Seeger 2001; Csató and Opper 2002). In the sequential sparse approximation to Gaussian processes we provide a principled approach to approximate the GP posterior. The approximation is such that in the end, for predicting, we do not require the whole set, rather we only require a "sparse" subset thereof.

In sparse approximation to the posterior we aim at "minimising" the cost of storing the GP in computer memory and implicitly decreasing the computation time. To achieve it, we add examples to the posterior GP one by one using Bayes' rule in a sequential manner. At each step we use the posterior from the preceding iteration as the current prior. The decrease in computation time is achieved when we *remove an input location* from the posterior – the removal being done such that the difference between the current posterior and its trimmed approximation is minimal. We take into consideration the probabilistic nature of the Bayesian inference, i.e. that we have access to the posteriors *distribution* of the latents, consequently we minimise in the Kullback-Leibler (KL) divergence between the two distributions (Csató 2002; Cover and Thomas 1991). This divergence measures the similarity of two probability distributions and can be thought of as *jointly* matching the first two moments of the Gaussians.

Often one can remove, whilst performing the sequential update, a large proportion of the input locations and at the end of this modified learning there is a small subset left to represent the approximated GP, we call this subset the set of "basis vectors", or BV set and we have called the procedure SSGP – sequential sparse GP algorithm. The resulting approximation uses only the BV set in computing both the posterior mean and the (co)variance. We provide an illustration in Figure 1 where we can see the sparse GP estimate of the mean function together with the training set (x signs) and the location of the BV's (triangles).

The significant difference thus between conventional Gaussian processes and the SSGP algorithm is the need for large matrix inversion operations: the SSGP algorithm scales cubically ***only*** with the size of the smaller BV set. Since one of the main objectives of SIC2004 is fast interpolation, we think that SSGP is an attractive estimation method.

We next summarise the main steps of the SSGP algorithm, for more information one can consult (Csató 2002). The algorithm controls the "novelty" corresponding to each new input location by computing the loss between the GP after adding the observation to the GP – via the likelihood function – and a second GP obtained from the previous posterior when only the location corresponding to the new input has been removed, the "distance" measure is the Kullback-Leibler divergence (Cover and Thomas 1991) between the Gaussian distributions derived from the respective GP's taken at the input locations. We will call this quantity the "score" associated to a specific input and this score can be computed irrespective of the order of input addition. In the algorithm one has to set a threshold value $\varepsilon(\text{ e.g.} = 10^{-3})$ and at the end of every step the scores associated to the inputs in the BV set are recomputed. If any of the scores falls below the threshold value, then the respective BV is removed from the BV set (since the algorithm is sequential, if by mistake a BV is removed from an important region in the input space, at a later occasion BV's will appear which will fill in the respective region). When an estimate of the posterior GP is available, one proceeds to the estimation of the hyper-parameters to the model by optimising the log-evidence of the data. There is no analytic solution and we employ a scaled conjugate gradient method to find the correct parameters. When the parameters are found, the SSGP algorithm is rerun using the new hyper-parameter values. We observed how the hyper-parameters varied as a function of the number of re-estimation cycles. This is to see how quickly the hyper-parameters converged. We used these plots to decide the number of times we should use the SSGP algorithm. We decided to use 6 cycles, each cycle through the algorithm taking approximately 30 seconds.

2.1 USE OF PRIOR INFORMATION

The prior information given to the participants of the "Spatial Interpolation Comparison" exercise was 10 datasets sampled at specified locations and at 10 distinct times, at each time there were 200 data points (x_i, y_i) sampled, thus the full prior dataset D has been given separated into 10 parts as $D = \{D_1, \dots, D_{10}\}$. Since there was no further information that would connect the individual sets, we assumed that they are independent but the samples across each individual set are connected in that they were observed at the same time.

Before we performed any calculations, we normalised the inputs and outputs to zero mean and unit variance. However, in order to maintain the directional structure of the data, we calculated the variance globally. When showing the results, the normalised data is transformed and rescaled back to the original coordinates.

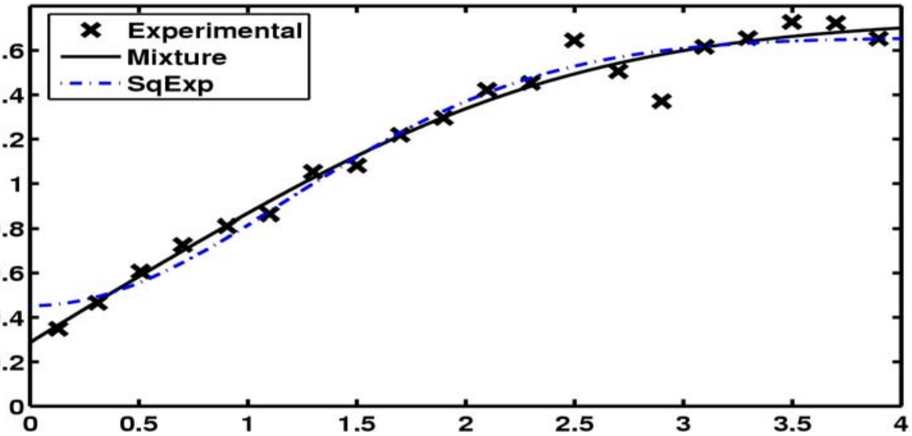

Figure 2
Variogram of the prior data, plotted for 20 lags and the different fits to the empirical one. It can be seen that the variogram corresponding to the RBF provides bad estimates at the origin and the mixture kernel has a more accurate fit to the experimental variogram.

To examine the structure of the data, we plotted the variogram for each of the prior datasets, depicting the variation between observations at increasing lag separation distances on Figure 2: the closer points in the input space are (X-axis), the more likely the values of the observed radiation are the same (semi variance on the Y-axis), this functional relation is depicted in the variogram. We first calculated the experimental variogram from the data (crosses). We estimated the variogram for each of the days individually. Looking at the empirical variogram we fitted several variogram models, a squared exponential, or RBF, from equation (6) and a simple exponential having the same form as the RBF except that the instead of the squared distance it uses the absolute value in the exponential. These models fitted the data well in certain regions, as we can see. We used the variogram as a means to assist in the most appropriate choice of kernel function for the SSGP algorithm. We decided that a mixture of kernel functions is appropriate to model the radiation data. In addition to this, inspecting the locations of the prior data, it could be seen that there were insufficient data points at short intervals for our model to learn the structure at short ranges accurately. Based on studies about how ambient radiation disperses through the atmosphere (Hall 1984; UIC Melborne 2004), we decided at the short range level that a squared-exponential model would be appropriate. For longer ranges an exponential model would fit the data best. We employed a linear combination of these two kernel functions each with different range and amplitude:

$$K_{\text{mix}}(x,x') = \pi A_1 \exp\left[-\frac{1}{2}\sum_{i=1}^{2}\frac{(x_i - x_i')^2}{\theta_{1,j}^2} \right] + (1-\pi) A_2 \exp\left[-\frac{1}{2}\sum_{i=1}^{2}\frac{\left|x_i - x_i'\right|}{\theta_{2,j}^2} \right] \quad (7)$$

with π the mixing ratio of the squared exponential kernel with parameters $\left(A_1, \theta_{1,1}, \theta_{1,2}\right)$ and that of the exponential one with parameters $\left(A_2, \theta_{2,1}, \theta_{2,2}\right)$. Although we are not using the variogram to estimate the initial parameters for the kernels we use in our model, certain observable features of the variogram relate to the parameters of the kernel functions. The parameter *A* relates to the amplitude of the kernel function, which equates to the plateau that the variogram reaches, called the sill. The parameters *θ* relate to the length-scale which corresponds to the range feature of the variogram: this is the point where an increase in separation distance between pairs no longer causes an increase in the average squared difference between the pairs. Observe that now we have more parameters relating to the kernel function: $\left(A_1, \theta_{1,1}, \theta_{1,2}, A_2, \theta_{2,1}, \theta_{2,2}, \pi\right)$ and one has to set these parameters before the objective datasets are seen, i.e. based on the 10 validation sets alone. We have used non-isotropic covariance functions since we believe that the variation in the data is not uniform in all directions. The increase in computational complexity by using non-isotropic covariance functions is that we now have an additional parameter to optimise. The optimisation of these parameters is described in the next section.

2.2 TUNING THE ALGORITHMS

To estimate the hyper-parameters to use with the competition data, we trained a GP for each of the 10 datasets; the SSGP algorithm was used for training. Since the 200 training points for each dataset were independent and identically distributed, the posteriors obtained are also independent conditioned on the priors. The optimisation of hyper-parameters used all of the posterior GP's in the computation of the gradients. We assumed that the "true gradient" is a linear combination of individual gradients and since there was no information about the individual datasets, we assumed an equal weight to all of them, thus the gradient was:

$$\frac{\partial \log P(D)}{\partial s} = \frac{1}{10} \sum_{k=1}^{10} \frac{\partial \log P(D_k)}{\partial s} \quad (8)$$

where *s* is a parameter of the mixture kernel from equation (7) which we want to optimise. In the experiments we used the scaled conjugate gradient algorithm from the NETLAB package Nabney 2002. In performing the gradient optimisation we added priors to the parameters of the mixture kernel. The priors were factorising and effectively served as penalties, similar to the penalised regression framework, preventing the kernel parameters from going to "abnormal" values, e.g. that would make the posterior mean collapsing to a sum of delta functions. Since all parameters have to be positive, we chose to put a prior on their log-value and we choose to have wide Gaussian distributions, meaning that the hyper-priors over the log-values of GP were Gaussians with a large variance.

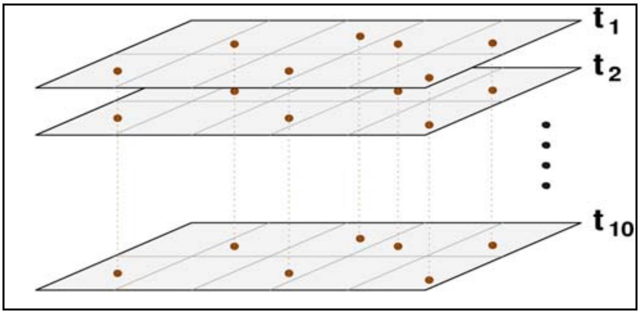

Figure 3
Illustration of combining the observations. The dots on the same horizontal planes show the observations made at the same time. An SSGP fit has been performed for each set of observations on separate planes and the hyper-parameter selection is based on optimising a weighted sum of gradients at each individual level.

To evaluate the combined SSGP performance, we performed leave-one-out cross validation on the 10 datasets: nine days of data were used to estimate the GP parameters and the one day left out used to measure the error related to the parameters we found. For the left out day we used the hyper-parameters resulting from the nine days as the mean value for the hyper-priors of the GP and decreased their corresponding variances, acknowledging that some information has already been acquired about the parameters themselves but we are still uncertain about their actual values. This setup enables the re-estimation of the hyper-parameters to use with the competition data.

We mention an important advantage of the Bayesian methodology exploited in this contribution, namely the fact that there is no need to have a separate validation set to assess the performance of the method, and **all** data can be used for accurate hyper-parameter estimation using the marginal data likelihood (MacKay 1992). This scenario was used for the SIC2004 “Spatial Interpolation Comparison.” It has been the done in the following manner: we used the 10 “prior” datasets to train the ten GP’s and then combined the individual gradients as in equation (8) which were fed into a conjugate gradient optimisation algorithm and new hyper-parameters were obtained. The previous steps were repeated a number of times (30), and the parameters found this way have been inserted as means in the hyper-priors for the GP we used for the competition datasets 1 and 2. Having no information about the joker dataset, we decided that the most appropriate approach for this dataset would be to loosen our prior beliefs about the hyper-parameters we were using in our model since we had absolutely no idea about what this file would contain.

3. RESULTS

This section contains the results of our experiments and compares a variety of benchmark methods to see the effectiveness of the SSGP algorithm. Using Matlab with the SSGP toolbox (Csató 2003) modified to minimise the combined gradient, we predicted the values at the unknown locations using the methodology described above. Netlab (Nabney 2002), a Matlab toolbox was used to provide benchmark results using a variety of other machine learning techniques (Multi-later Perceptron (MLP) and Radial Basis Functions (RBF) (Bishop 1995)). The training of the MLP was achieved using early stopping to control the effective complexity of the network. Standard Gaussian processes were also used so that the choice to use a mixture of kernel functions could be validated. It should be noted that the scales on individual uncertainty maps are different, so that the uncertainty can be more easily gauged, however for all of the Isoline level maps, the contour scale is constant.

3.1. OVERALL RESULTS

N = 808	Min	Max	Mean	Median	Std. dev.
Observed	57.00	180.00	98.01	98.80	20.02
SSGP	68.82	125.41	96.75	98.96	14.41
GP (mixture)	67.07	127.20	96.58	98.65	14.98
GP (sqexp)	69.26	123.77	96.87	99.19	14.32
RBF	68.22	129.55	96.85	98.66	14.58
MLP	66.05	129.13	96.80	98.07	14.65

Table 1
Comparison of the estimated and measured values (nSv/h) 1st Dataset.

N = 808	Min	Max	mean	Median	Std. dev.
Observed	57.00	1528.20	105.42	98.95	83.71
SSGP	74.25	634.49	100.78	95.68	39.22
GP (mixture)	87.09	150.81	106.13	102.46	15.51
GP (sqexp)	80.80	161.73	108.51	101.28	22.53
RBF	82.22	160.73	108.22	101.77	21.91
MLP	-129.18	760.02	102.41	94.71	80.03

Table 2
Comparison of the estimated and measured values (nSv/h) 2nd Dataset.

	MAE	ME	Pearson’s r	RMSE
SSGP	9.10	-1.27	0.788	12.46
GP (mixture)	9.08	-1.44	0.787	12.47
GP (sqexp)	9.47	-1.15	0.776	12.75
RBF	9.49	-1.19	0.776	12.71
MLP	9.48	-1.22	0.775	12.73

Table 3
Comparison of the errors for 1st Dataset.

	MAE	ME	Pearson’s r	RMSE
SSGP	18.55	-4.64	0.856	54.22
GP (mixture)	21.77	0.72	0.350	79.57
GP (sqexp)	22.53	3.09	0.331	79.16
RBF	22.73	3.21	0.334	79.31
MLP	48.41	-3.01	0.384	90.89

Table 4
Comparison of the errors for 2nd Dataset.

ESTIMATIONS FOR SSGP METHOD

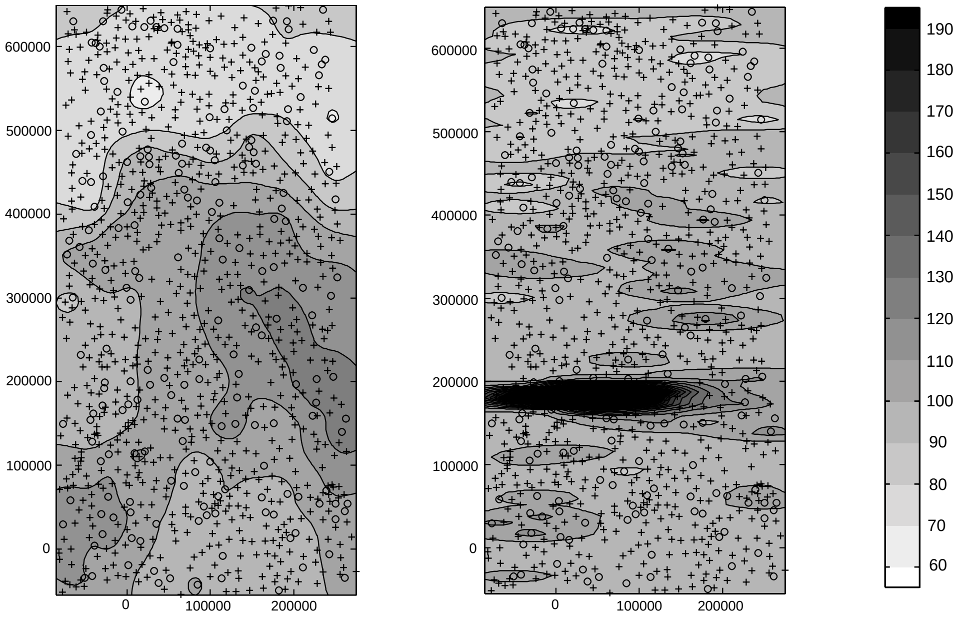

Figure 4
Isoline levels (nSv/h) for the 1st set (left) and the 2nd set (right) using SSGP.

UNCERTAINTY IN SSGP METHOD

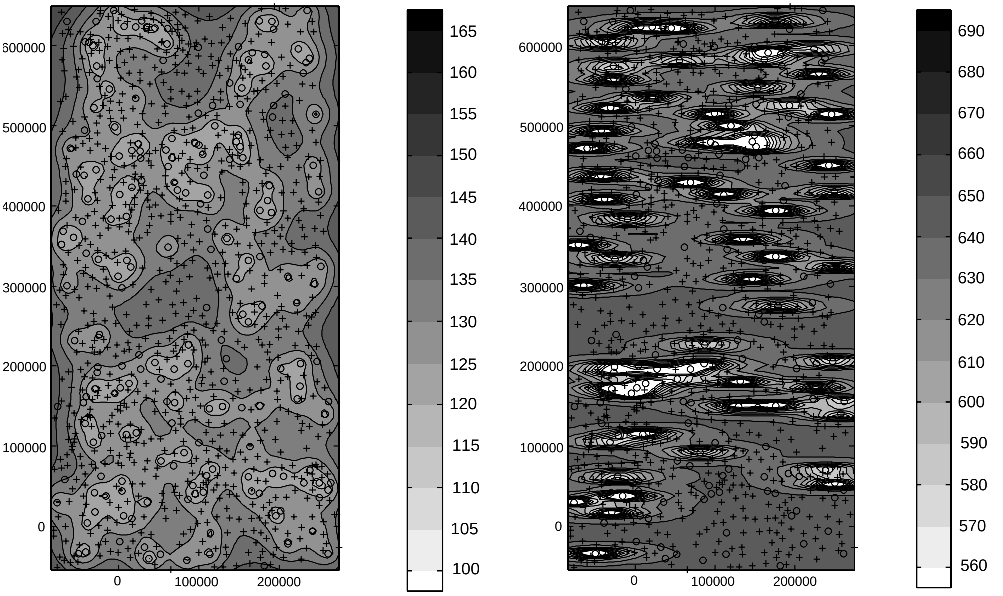

Figure 5
Isoline levels showing the uncertainty in the estimations obtained for the 1st set (left) and the 2nd set (right) using SSGP. Notice that the grey-levels are different in the two subfigures.

ESTIMATIONS FOR GP (MIXTURE) METHOD

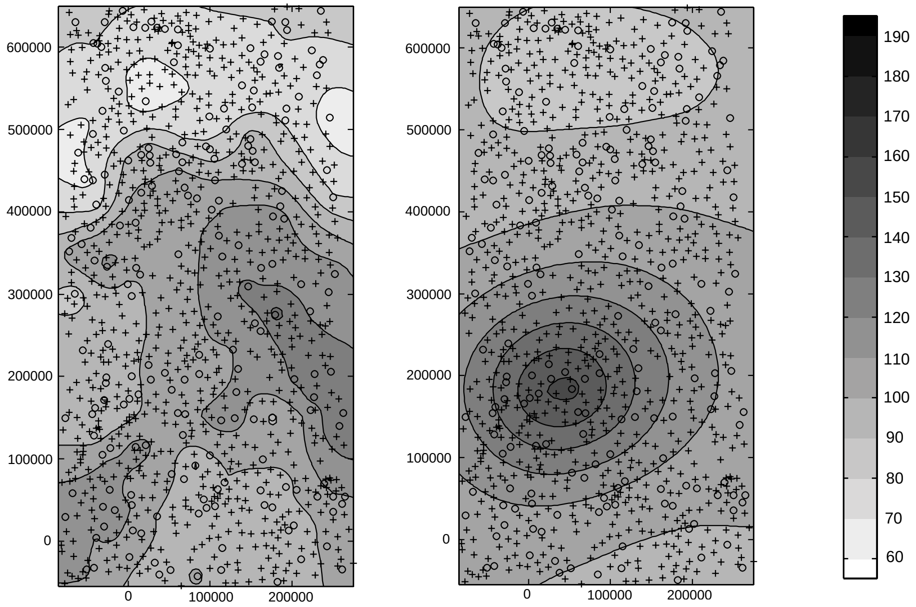

Figure 6
Isoline levels (nSv/h) for the 1st set (let) and the 2nd set (right) Using GP.

UNCERTAINTY IN GP (MIXTURE) METHOD

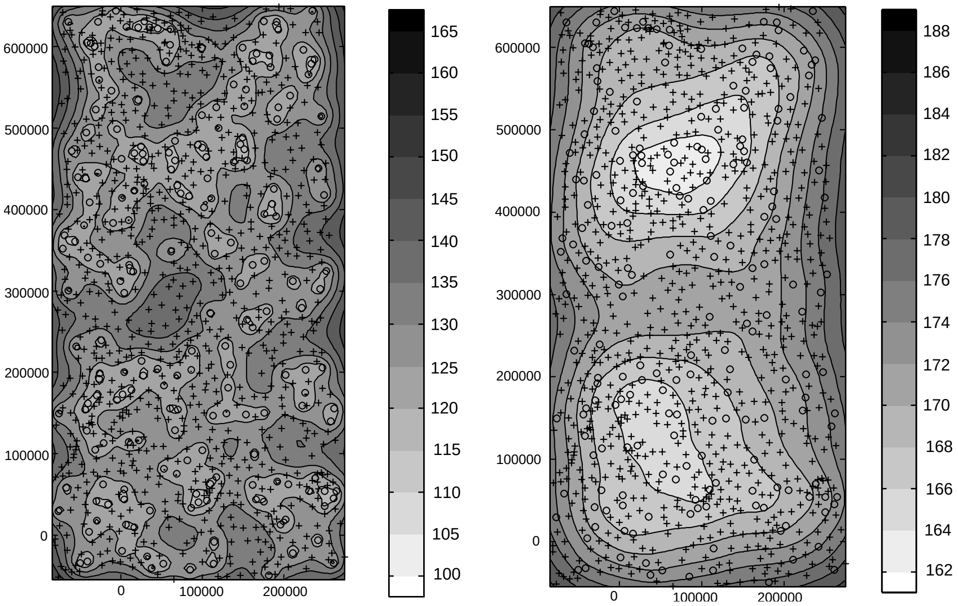

Figure 7
Isoline levels showing the uncertainty associated to the estimations obtained for the 1st set (left) and the 2nd set (right) using GP. Notice that the grey-levels are different in the two subfigures.

3.2. DETECTING ANOMALIES AND OUTLIERS

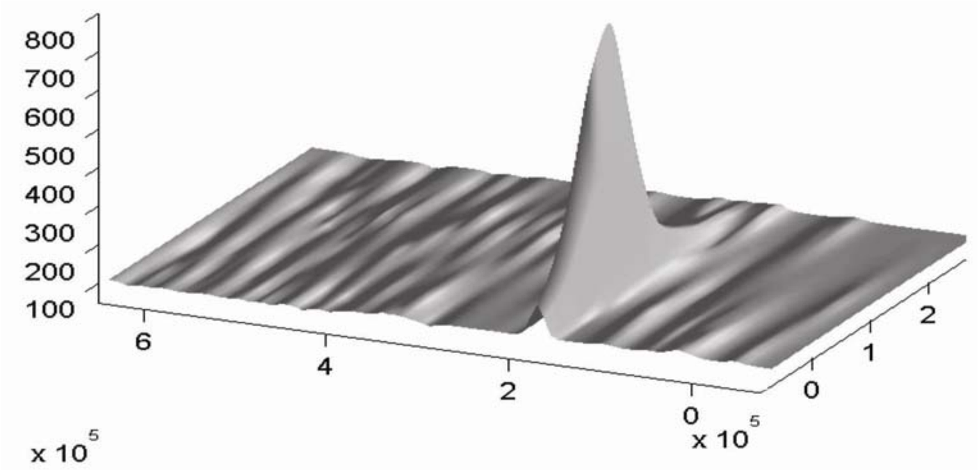

Figure 8
3D map showing extreme values found in the 2nd set (vertical scale in nSv/h)

3.3. CALCULATION TIME

	Time taken
1st Dataset	3m28s
2nd Dataset	4m27s

Table 5
Calculation Time.

3.4. LEARNED HYPER-PARAMETERS FOR SSGP

	1st Dataset	2nd Dataset
SqExp x range	0.02	0.03
SqExp y range	0.24	0.24
SqExp amplitude	0.05	0.05
Exp x range	1.89	0.10
Exp y range	1.11	0.00
Exp amplitude	0.90	60.12
Noise	0.30	41.59

Table 6
SSGP Learnt Hyper-parameters

4. DISCUSSION

The results for the normal dataset in Table 1 show that the range between the minimum and maximum values to be much shorter in all of our methods than the observed data and the estimates are less varied. All the maximum estimates are of a similar magnitude which would suggest a limited amount of information in the training dataset. Comparing the errors (*Table 3*) between the SSGP and the GP (mixture) method, it can be seen that these methods have a similar performance. It is clear from this that little information has been lost by imposing sparsity in representing the GP. Selecting the number of basis vectors to represent the GP by is another task one must take into consideration. For this exercise we choose to use 80 BV's. Using the prior datasets, a minimal number of BV's was chosen that would optimally represent the GP with minimal error.

When comparing the two GP methods that were tested it can be seen that the kernel function mixture method predicts the true values more accurately than the squared exponential, which supports our underlying assumptions about the structure of the data. Having prior knowledge about the covariance structure of ambient background radiation clearly is advantageous in prediction. The GP (sqexp) method was used as a benchmark and does not take into account any prior information about the dispersal process except hyper-parameters that fit a squared exponential model to the data. This method has similar performance to the MLP and RBF methods which also do not take into account any assumptions about the dispersal process.

The nature of the second dataset was unknown prior to the interpolation exercise. The 2nd dataset estimates (*Tables 2 & 4*) are poor, but this particular algorithm is a smoother and is not designed to cope with extreme values since it's based on the assumption that observations closer together are more likely to be similar than those further apart. The MLP method showed the worst prediction error, giving negative estimates at some of locations; this is not surprising since it is a machine learning method that does not take into account any prior knowledge about the underlying process we are predicting. The MAE error for these two benchmarks is a lot higher than any of the GP models; however the predictions with the MLP are slightly more correlated with the observed data than the two standard GP models. The SSGP method shows the smallest MAE and the predictions are correlated. The SSGP still did not manage to come close to predicting the maximum as seen in the observed data, but this is to be expected since GP's are poor at predicting extreme values.

Looking at the MAE and RMSE error for the two datasets, it can be seen in the first dataset that the RMSE is slightly larger than the MAE. However for the second dataset, RMSE is much higher than the MAE, this is because the RMSE is more sensitive to high residuals. The estimates are all slightly negatively biased for the first dataset. For the second dataset it can be seen the bias in estimates has grown much larger depend on the algorithm used.

The contour plots (*Figure 4*) show the estimates over the whole area on the two datasets. The second dataset plot clearly shows bizarre behaviour in one direction with the prediction of the dispersal of the contaminant introduced. The hyper-parameters learnt by the SSGP as it cycles through the new data are shown in *Table 6*. The learnt hyper-parameters refer to the normalised data, since during the algorithm the data used has been normalised. Looking at the learnt hyper-parameters it can be seen that the kernel function range in the y direction has collapsed to zero. This accounts for the elliptic shape of the outliers and the features in the plot. It is unknown why the hyper-parameter optimisation has resulted in these parameters. In *Figure 8* the narrow dispersal is clearly visible.

The learnt hyper-parameters show an increased noise level. The noise corresponds to the nugget feature of the variogram. This learnt noise parameter will cause the interpolation to not only become more uncertain, but also cause there to be a smaller range of variation between the predicted points. Looking at the uncertainty maps (*Figure 5*) it is clear that predictions with the second dataset are extremely uncertain. The range and pattern of uncertainty for the first data set are similar to the uncertainty in the results calculated in the GP model (*Figure* 7). The uncertainty measured in the GP for the second dataset shows a similar, but less defined pattern to that shown in the first dataset. *Figure* 6 shows the prediction of the dispersed contaminant using the GP. The pattern of the dispersal is less elliptic and closer to the pattern of dispersal one would expect.

The calculation time is shown in *Table 5*. For the purposes of this contest, the algorithm was set to cycle through the SSGP algorithm seven times to achieve improved estimates. Since much of the algorithm and hyper-parameters were tuned using the prior data before the actual contest data was made available, predicting at unknown locations could be done after one or two cycles through the algorithm without a great loss in accuracy and hence further reducing computation time. All experiments were carried out on a 2 Ghz PC using Matlab 6.5 and the SSGP and Netlab Matlab toolboxes (Csató 2003; Nabney 2002).

5. CONCLUSIONS

The sparse extension to the Gaussian processes provides a fast, statistically principled approach to spatial interpolation. As with many methods of spatial interpolation, outliers are particularly difficult to model. There have been various approaches at modelling more accurately outliers such as using combinations of prediction models, one such model uses a linear combination of a number of methods where by the large scale discontinuous variation is modelled using fast wavelet interpolation and the residuals are interpolated using standard kriging techniques (Demyanov et al. 2001). This entry was to show the advantages of the SSGP model. The SSGP could be coupled with existing algorithms to predict more effectively where extreme values are more likely.

Once the nature of the covariance structure is understood, this method is entirely automatic. There are a large number of tuneable parameters that one can set to more accurately predict unknown data. In emergency conditions the SSGP algorithm could be used to predict, with accuracy commensurate with an ordinary GP algorithm and within a significantly shorter period of time. Since the computational complexity only scales with the number of basis vectors, predication with ordinarily prohibitively large datasets can become manageable. For situations where there are extreme values, tighter variances on the prior beliefs about the hyper-parameters or turning off hyper-parameter re-estimation may allow a more accurate estimate of the data, however prediction in locations where there are extreme values will still perform poorly.

Another advantage of a sparse model as previously mentioned is the compact way in which the GP is represented by a subset of basis vectors. In effect, these basis vectors compress the data so that transmission would require significantly less bandwidth. The real advantage of this compression can be appreciated more if one considers the transmission of such data from a satellite.

Further work that is currently being completed is an implementation of the SSGP algorithm for parallel processor computer systems. Further computational speed increases will make SSGP's an extremely attractive solution in emergency situations where quick predictions are needed and the dataset is extremely large. For scenarios whereby there is a natural contaminant introduced into the area, as in the joker set, one cannot really predict with any great degree of certainty how the data will be distributed, but an assumption has to be make in advance. Further investigation could look into the use of different noise models which would allow for example robust estimation of the radioactivity level. Cressie states that there is no real test regarding the non-Gaussianity of the underlying phenomenon, however he does suggest the use of different loss functions. The consideration of robust models is important since it could lead to a better predictive model. For this one could consider the Laplace noise model or other additive noise models having a heavy tail like the Student-t noise model (Rousseeuw and Leroy 1987).

REFERENCES

Bernardo, J.M, Smith A.F (1994), *Bayesian Theory*. John Wiley & Sons.

Bishop C.M (1995), *Neural Networks for Pattern Recognition*. New York, N.Y. Oxford University Press.

Cornford D, Csató L, Opper M (2004), ‘Sequential, Bayesian Geostatistics: A Principled Method for Large Data Sets’, Technical Report, Aston University.

Cover T.M, Thomas J.A (1991), *Elements of Information Theory*, John Wiley & Sons.

Cressie N.A (1993), *Statistics for Spatial Data*. John Wiley and sons.

Csató L, Opper M (2002), *Sparse On-Line Gaussian Processes. Neural Computation* vol. 14, pp. 641-669. http://www.ncrg.aston.ac.uk/Papers

Csató L. (2002), ‘Gaussian Processes - Iterative Sparse Approximations’. PhD thesis, Aston University.

Csató L (2003), SSGP Toolbox. http://www.ncrg.aston.ac.uk/Projects/SSGP

Demyanov V, Soltani S, Kanevski M, Canu S, Maignan M, Savelieva E, Timonin V, Pisarenko V (2001), ‘Wavelet analysis residual kriging vs. neural network residual kriging’. Stochastic Environmental Research and Risk Assessment 15.

Gibbs M, MacKay D.J.C (1997), ‘Efficient implementation of Gaussian processes’, Technical Report, Cavendish Laboratory, Cambridge University.

Hall E.J (1984), *Radiation and Life* 2nd Ed. Pergamon Press, New York.

Isaaks E.H, Srivastava R.M (1989), *An Introductions to Applied Geostatistics*, Oxford University Press.

MacKay D.J.C (1992), *Bayesian interpolation. Neural Computation* vol. 4, pp. 415-447, The MIT Press.

MacKay D.J.C (1999), *Comparison of Approximate Methods for Handling Hyper-parameters. Neural Computation* vol. 11, pp. 1035-1068, The MIT Press.

Nabney I.T (2002), NETLAB: Algorithms for Pattern Recognition, Springer Verlag http://www.ncrg.aston.ac.uk/netlab/, Aston University.

Papoulis A, Pillai S.U (2002), *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill.

Rousseeuw P.J, Leroy A.M (1987) *Robust Regression and Outlier Detection*, John Wiley & Sons.

Schoelkopf B, Smola A.J (2002), *Learning with Kernels*, The MIT Press.

UIC, Melbourne, Australia. Nuclear Issues Briefing Paper 17, March 2004. http://www.uic.com.au/nip17.html

Williams C.K.I, Rasmussen C.E (1996), ‘Gaussian processes for regression’. In Proceedings of the Neural Information Processing Systems, vol. 8, The MIT Press. www.nips.cc

Williams C.K.I, Seeger M (2001), ‘Using the Nystrom method to speed up Kernel Machines’. In *Proceedings of the Neural Information Processing Systems*, vol. 13, The MIT Press. www.nips.cc

Cite this article as: Ingram, Ben; Csató, Lehel; Evans, David. ‘Fast spatial interpolation using sparse Gaussian processes’. *Applied GIS*, Vol 1, No 2, 2005. pp. 15-01 to 15-17. DOI: 10.2104/ag050015