Gaussian Processes for Approximating Complex Models

Murali Haran

Department of Statistics
Penn State University

(references Rasmussen and Williams (2006) and others)

Yahoo! Machine Learning Tutorials
Computer Science, University of Washington, Seattle
March 2012

Modeling with Gaussian Processes

- Constructing flexible models for space-time processes. Model-based kriging, continuous space-time models
- GPs may be used to define a distribution on a space of functions. In Bayesian inference, used as a prior distribution on function space. Bayesian nonparametrics
- Useful for modeling complex computer codes, nonparametric regression and classification. Machine learning, models for complex computer models

The above uses are all very closely related.

Complex Models

This tutorial will be about approximating/emulating and calibrating complicated models.

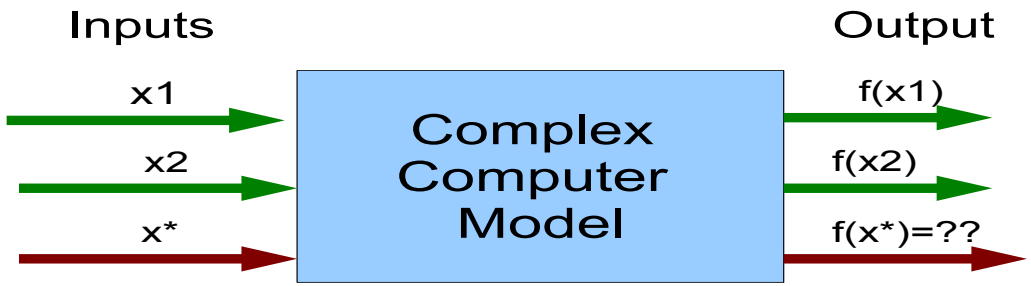
- Scientists working in the physical and natural sciences are often interested in learning about the mechanisms (processes) underlying physical phenomena.
- Systems of differential equations, state-space models
- Translated into computer code to study process simulations under different conditions
- Examples:
  - Modeling ocean-atmospheric systems (deterministic)
  - Disease dynamics (deterministic or stochastic)

Statistical Problems

- Emulation: Models are expensive to run. Want fast approximation for the model to see how it behaves at settings (e.g. parameter values) where it was not run.
- Want to study discrepancies between model and reality
- Calibration: Often want to fit the model to data. “Tune” the model to better represent reality and to learn about parameters that are of scientific interest.

Complex Computer Models

Statistical interpolation or nonparametric regression problem.



Green inputs/output are the training data.

Red = the input where predictions are desired.

Input, output need not be scalars

Computer Model Emulation via GPs

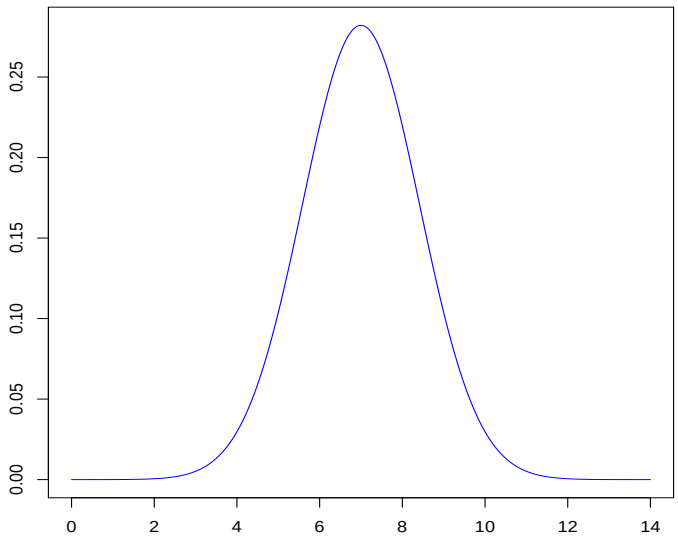
- Emulator: approximation of a complex computer model.
- An emulator is constructed by fitting a model to a training set of runs from the complex computer model.
- Simplicity allows for fast simulations from emulator even if the original model is very slow/expensive.
- The advantage of doing it in a probabilistic framework:
  - Uncertainties associated with interpolation, i.e., greater uncertainty where there is less training data.
  - Probability model: useful for statistical inference.
  - “Without any quantification of uncertainty, it is easy to dismiss computer models.” (A.O’Hagan)

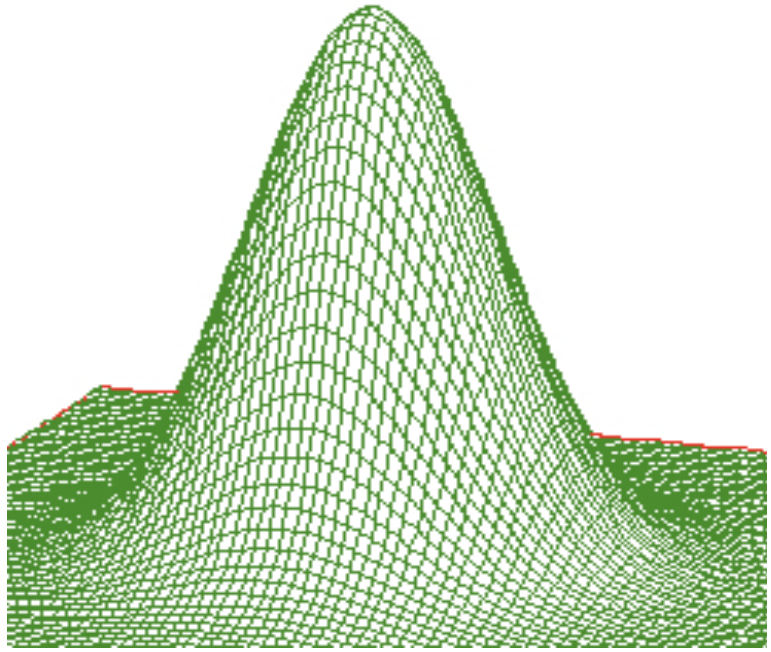
Outline

1. Gaussian process basics
2. GPs for computer model emulation
3. GPs for computer model calibration

Preliminaries: Gaussian Distributions

Gaussian (“Normal”) distribution: univariate and bivariate





Preliminaries: Multivariate Gaussian Distributions

A joint distribution for continuous random variables at 3 locations, $Z(\mathbf{s}_1), Z(\mathbf{s}_2), Z(\mathbf{s}_3)$

$$\begin{bmatrix} Z(\mathbf{s}_1) \\ Z(\mathbf{s}_2) \\ Z(\mathbf{s}_3) \end{bmatrix} \sim N\left(\begin{bmatrix} \mu(\mathbf{s}_1) \\ \mu(\mathbf{s}_2) \\ \mu(\mathbf{s}_3) \end{bmatrix}, \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \right).$$

- Want dependence (covariances σ_{ij}s) to be related to the distance between the locations.
- One possibility: $\sigma_{ij} = \psi \exp(-(\|s_i - s_j\|)/\phi), \psi > 0, \phi > 0$ where s_i is the location of the ith observation. If distance between locations $\mathbf{s}_i$ and $\mathbf{s}_j$ is large, σ_{ij} will be small.

Basic Gaussian Process (Linear) Model

- Process is modeled as $Z(\mathbf{s}) = \mu(\mathbf{s}) + w(\mathbf{s})$ where:
  - $\mathbf{s} \in D \subset \mathbb{R}^d$
  - $\mu(\mathbf{s})$ is the mean. Often $\mu(\mathbf{s}) = X(\mathbf{s})\beta$, $X(\mathbf{s})$ are covariates at $\mathbf{s}$ and β is a vector of coefficients.
- Model dependence among spatial random variables by modeling $\{w(\mathbf{s}) : \mathbf{s} \in D\}$ as a Gaussian process

Basic Gaussian Process (Continued)

- For any n locations, $\mathbf{s}_1, \ldots, \mathbf{s}_n$, $\mathbf{w} = (w(\mathbf{s}_1), \ldots, w(\mathbf{s}_n))^T$ is multivariate normal with covariance specified by a parametric covariance function with parameters Θ.
- Let $\mathbf{Z} = (Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n))^T$, so

$$\mathbf{Z}|\Theta, \beta \sim N(\mathbf{X}\beta, \Sigma(\Theta)).$$

  Example of covariance, $\Sigma(\Theta)_{ij} = \psi \exp(-(\|s_i - s_j\|)/\phi)$ so $\Theta = (\psi, \phi)$
- Computer models/machine learning: let locations $\mathbf{s}$ correspond to inputs so distances are no longer physical but in "input space" and $Z(\mathbf{s})$ are "outputs."

Fitting a GP Linear Model to Computer Model Runs

- "Data": $\mathbf{Z} = Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_N)$
- This specifies the likelihood function, $\mathcal{L}(\Theta, \beta; \mathbf{Z})$ or $\mathcal{L}(\mathbf{Z}|\Theta, \beta)$ (notation may vary)
  - Maximum likelihood: optimize $\mathcal{L}(\Theta, \beta; \mathbf{Z})$ with respect to Θ, β to obtain $\hat{\Theta}, \hat{\beta}$, maximum likelihood estimate (MLE)
  - Bayesian inference: specify prior distributions for Θ, β, $p(\Theta, \beta)$. Inference is based on posterior distribution

$$\pi(\Theta, \beta \mid \mathbf{Z}) \propto \mathcal{L}(\mathbf{Z}|\Theta, \beta) p(\Theta, \beta)$$

- Inference is routine in principle: optimization and Markov chain Monte Carlo respectively.
- But matrix computations involving $\Sigma(\Theta)$ are of order N^3. Inference is computationally expensive for large N.

GP Prediction

- We are interested in predicting $\mathbf{Z}^* = (Z(\mathbf{s}_1^*), \ldots, Z(\mathbf{s}_m^*))^T$, $\mathbf{s}_1^*, \ldots, \mathbf{s}_m^* \in D$ using $\mathbf{Z} = (Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_N))$
- Using GP assumption

$$\begin{bmatrix} \mathbf{Z} \\ \mathbf{Z}^* \end{bmatrix} \mid \Theta, \beta \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right),$$

- Using basic multivariate normal theory, we know the conditional distribution of $\mathbf{Z}^* \mid \mathbf{Z}$ is also multivariate normal with mean and covariance determined by $\mathbf{Z}$ and Θ, β.
- Maximum likelihood: "plug-in" MLE of Θ, β to obtain conditional normal distribution $\mathbf{Z}^* \mid \mathbf{Z}$.
- Bayesian: use posterior predictive distribution,

$$\pi(\mathbf{Z}^*|\mathbf{Z}) = \int \pi(\mathbf{Z}^*|\mathbf{Z}, \Theta, \beta)\pi(\Theta, \beta|\mathbf{Z}) d\Theta d\beta.$$

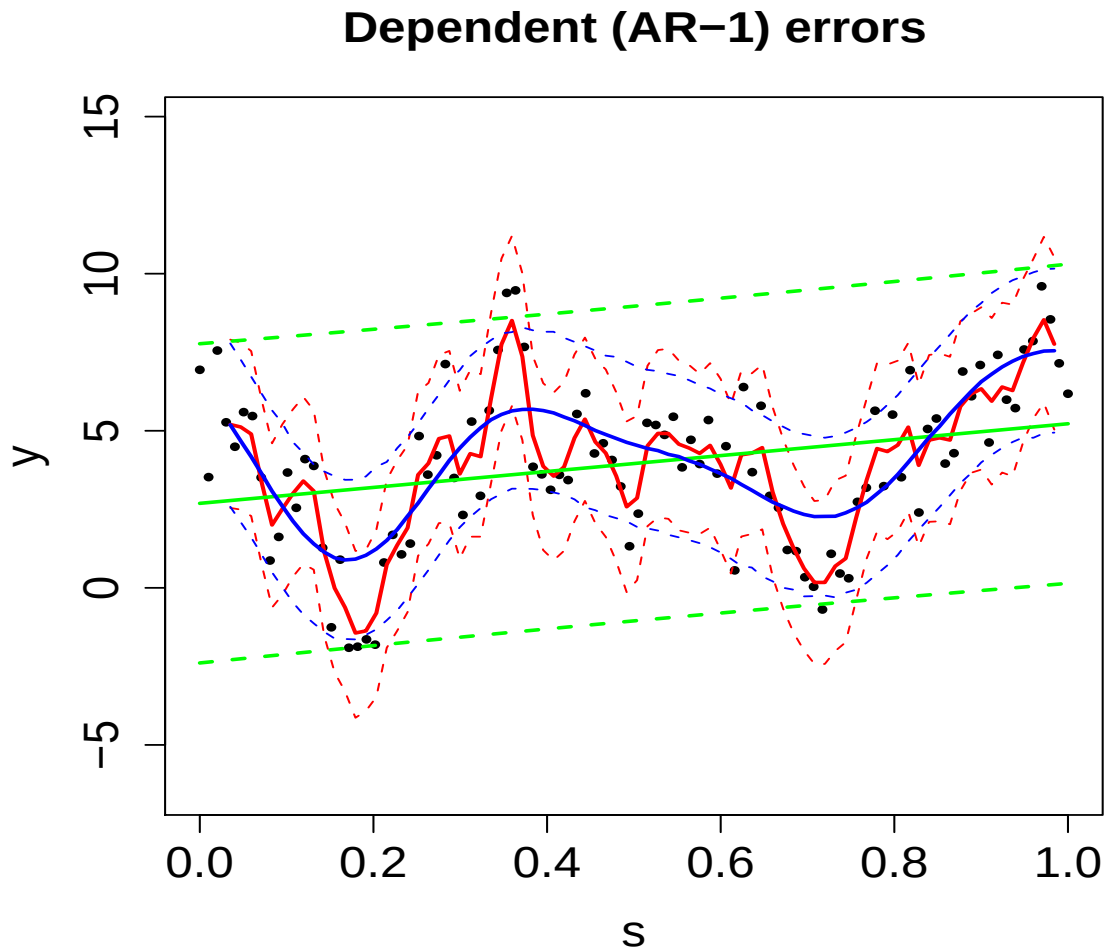
Simulating from Posterior Predictive

Draws from the posterior predictive distribution may be obtained in two steps:

1. Simulate $\Theta', \beta' \sim \pi(\Theta, \beta|\mathbf{Z})$ by MCMC.
2. Simulate $\mathbf{Z}^*|\Theta', \beta', \mathbf{Z}$ from conditional multivariate normal density and using Θ', β' above.

This approach incorporates uncertainties about Θ, β into predictions.

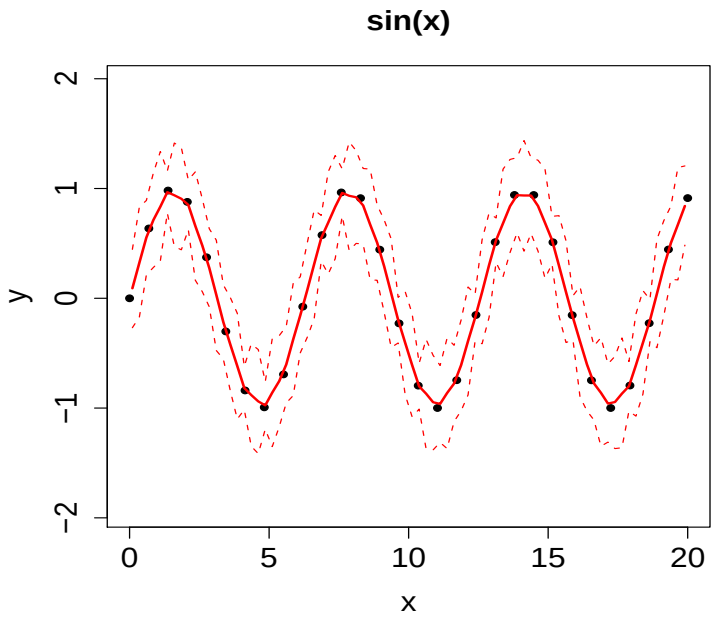
GP Model for Dependence: Example

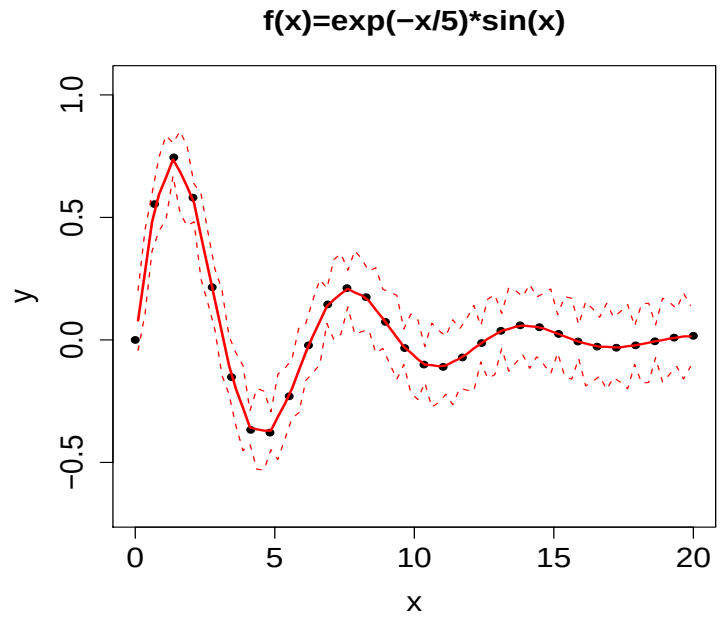


Black: 1-D AR-1 process simulation. Green: independent error.

Red: GP with exponential, Blue: GP with gaussian covariance.

GP for Function Approximation: Example





Run the two models at input values x equally spaced between 0 and 20 to obtain output (black dots). Can we predict/interpolate between black dots?

Red curves are interpolations using same, simple model:

$y(x) = \mu + w(x)$, $\{w(x), x \in (0, 20)\}$ is a zero-mean GP.

GPs for Function Approximation

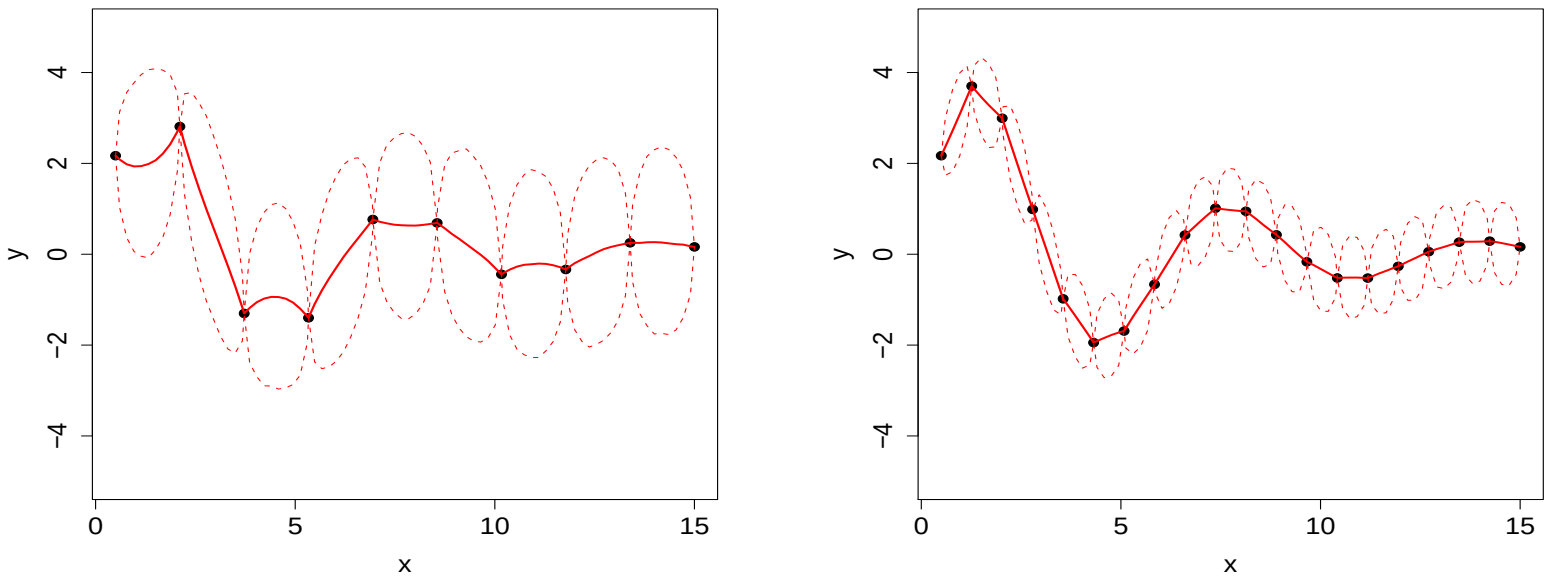
- Spatial models discussion of GPs largely focuses on accounting for dependence.
- But GPs are a flexible model for functions. Well known observation, summarized as follows:
  - “What is one person’s (spatial) covariance structure may be another person’s mean structure.” (Cressie, 1993, pg.25).
- GP models allow a simple covariance to substitute for a complicated mean with an unknown functional form.

GP Function Approximation: Uncertainties

x-axis: input locations $(\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n)$

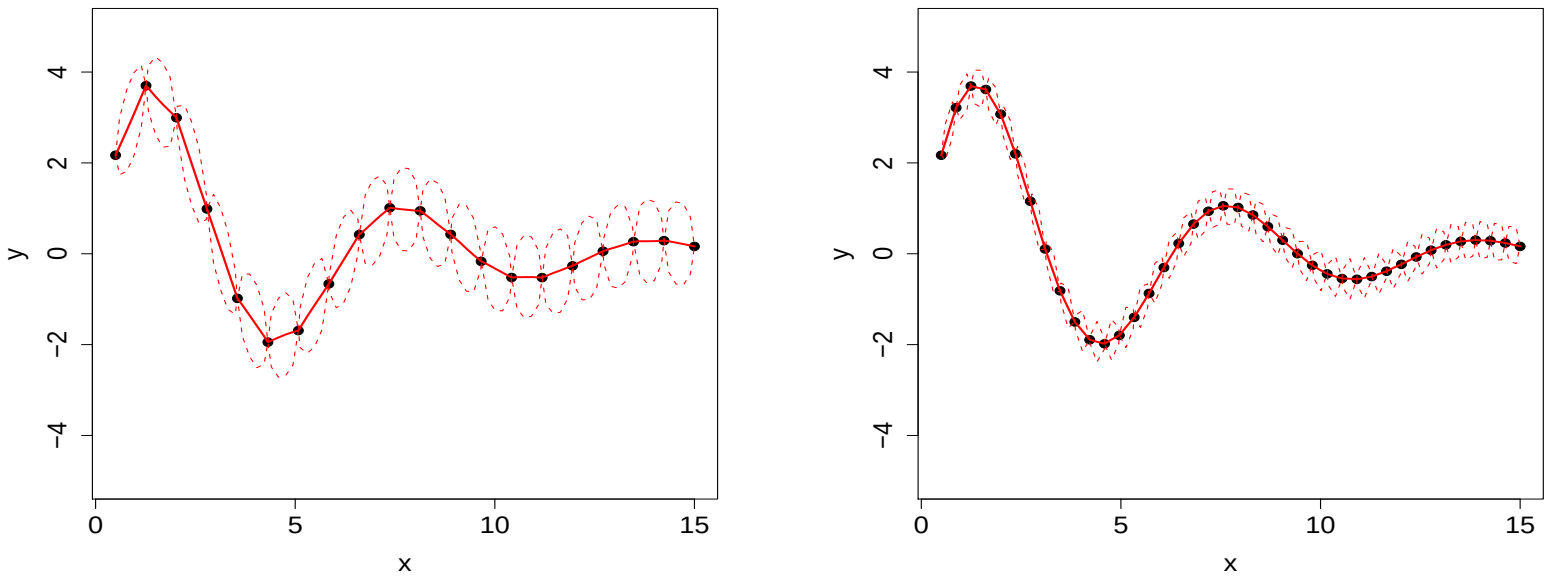
y-axis: output $(Z(\mathbf{s}_1), Z(\mathbf{s}_2), \ldots, Z(\mathbf{s}_n))$

Black dots: “data”; Red curves: interpolations



The effect of predictions as well as prediction intervals when data points are increased from 10 to 20.

GP Function Approximation: Uncertainties



The effect of predictions as well as prediction intervals when data points are increased from 20 to 40.

GP Complex Model Emulation

- Gaussian processes are extremely useful for emulating complicated models in situations where:
  - Simulator output varies smoothly in response to changing its inputs.
  - The number of inputs is relatively small.
- Output is often multivariate so much more complicated than 1-D examples.
- Often interested in using observations (real data) to improve simulations and to learn about parameters.

Examples of deterministic model emulation

Vehicle crash model (Bayarri et al., 2007):

- Non-linear dynamics analysis code using a finite element representation of the vehicle.
- E.g. of input: materials used in the components of the vehicle. Many other uncertain inputs, some fixed by modelers, some controllable.
- E.g. of output: velocity changes after impact at key positions on the vehicle, e.g. driver seat. Computer model takes 1-5 days for each run. Computationally expensive.
- Field data: crashing of prototype vehicles. Expensive!

Note: field data may not always be available.

Examples of deterministic model emulation

Climate models (Sanso et al., 2008; Bhat et al., 2012):

- E.g. of input: Parameters that describe key characteristics of the climate. For instance, climate sensitivity = the change in global mean temperature in response to a doubling of atmospheric CO_2.
- E.g. of output: Climate characteristics around the world. For instance, temperature fields (output on a spatial grid). General circulation models (GCMs): *very* expensive ($\approx$ 1-2 months). Earth climate models of intermediate complexity (EMICs): much faster, weeks or days.
- Field data: temperature measurements over the past century. May have errors, not on the same locations as model output, may be aggregates/averages.

GP Emulator Specification

We need:

- Mean function at any input x, $\mu(x)$

  Usually keep this simple, often just linear.

- Covariance function that specifies $\text{Cov}(x_i, x_j)$ for any pair of inputs x_i, x_j

  Covariance picks up non-linearities.

The GP emulator covariance function

- Assume standard parametric covariance functions, say from Matérn class or power-exponential family.
- The covariance function is often assumed to be separable in the different input dimensions. For example for inputs $\mathbf{x}, \mathbf{x}^* \in \mathbb{R}^d$, covariance

$$\kappa \exp\left(-\sum_{j=1}^{d} |x_j - x_j^*|^{\alpha_j}/\phi_j\right), \ \alpha_j \in [1,2], \ \phi_j \in (1,2), \ \kappa > 0.$$

- If not enough information to learn about α_j fix it at some value, say 1.9 (α_j =2 results in over-smooth processes. Also leads to numerical instabilities).
- Above: not most flexible but usually adequate. Simple, fast.

Emulation to Calibration

- So far have discussed GP-based emulation. Outline of methodology for approximating a complicated model (with possibly multivariate output), using just a few simulations from the model. Fit the GP model using maximum lkelihood or Bayesian approaches.
- Now: computer model calibration.

Computer Models and Parameter Inference

- Suppose scientists present a deterministic model to us and give us observations (field data) to go along with them. How do we infer the values of the parameters in their models?
- Several issues:
  1. the model may be deterministic, not even a probability model!
  2. complicated, impossible to write down in closed form
  3. may take very long to run

Computer Model Calibration: Outline

- Notation:
  - Computer model output $\mathbf{Y} = (Y(\theta_1), \ldots, Y(\theta_n))$.
  - Observation Z, assumed to be a realization of computer model at 'true' θ + discrepancy + measurement error.
  - Want to perform inference for θ (find θ that best "fits" the observations)
- Bayesian approach:
  - easily model discrepancy term
  - incorporate prior information about θ
  - easier to find multimodalities in likelihood/posterior surface for θ
  - conveniently look at marginal distributions if θ is multivariate

Two-stage Approach to Calibration/Inference

1. Find probability model for Z (data) using $\mathbf{Y}$ (simulations)
   - Create flexible emulator using $\mathbf{Y}$, say $\eta(\mathbf{Y}, \boldsymbol{\theta})$
   - Add model discrepancy and measurement error:

$$Z = \eta(\mathbf{Y}, \boldsymbol{\theta}) + \delta(\mathbf{Y}) + \epsilon$$

     where $\delta(\mathbf{Y})$ is a model discrepancy term, also modeled as a GP. ϵ is the observation/measurement error.

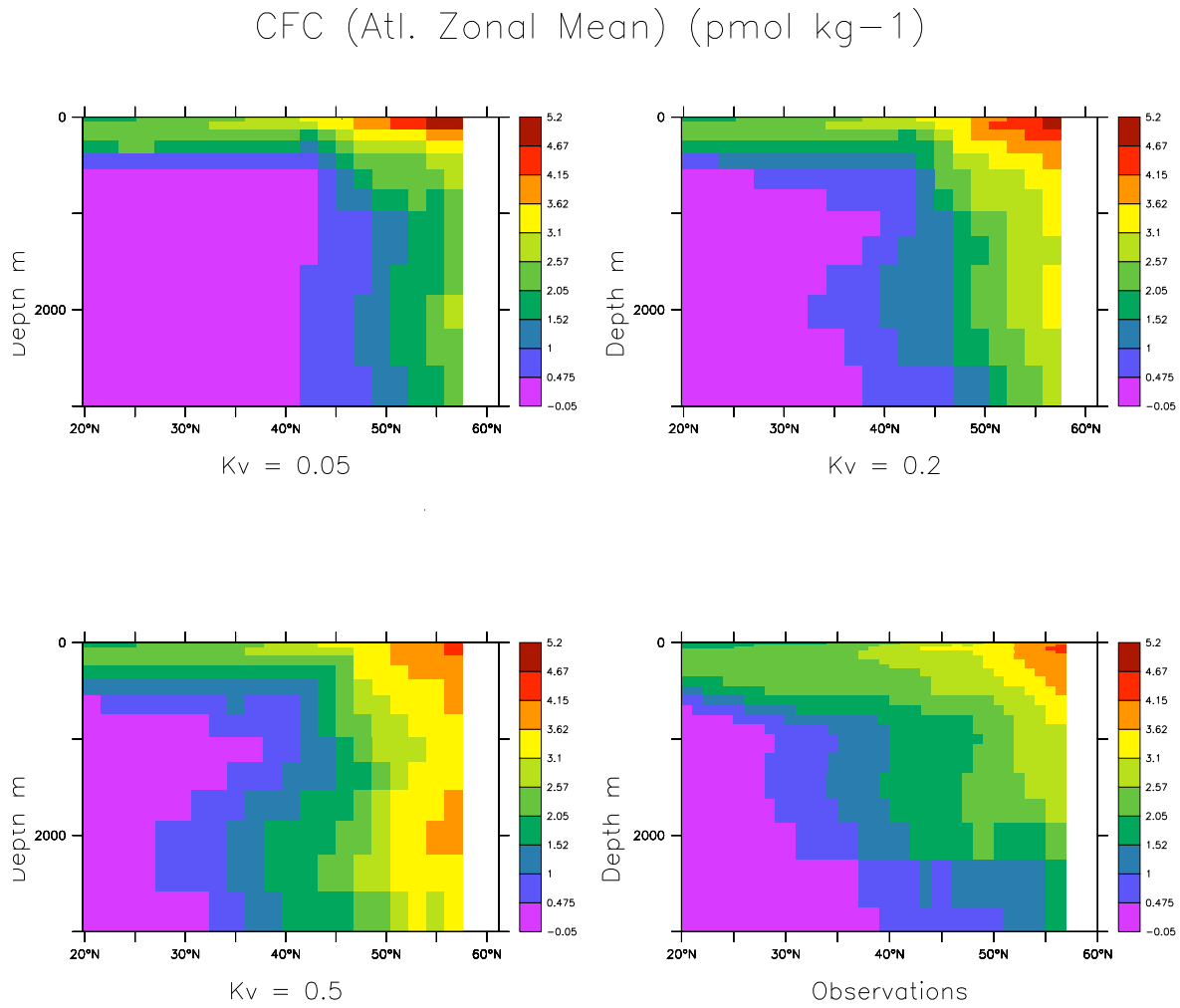
2. Posterior distribution $\pi(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{Z})$ derived from likelihood based on above model and prior on $\boldsymbol{\theta}$.

Also possible to just fit the entire model at once but above approach has some computational and inferential advantages (cf. Liu, Bayarri and Berger 2009; Bhat et al, 2012)

Climate Science Example: Learning about K_v

- K_v is a model parameter which quantifies the intensity of vertical mixing in the ocean, cannot be measured directly.
- Observations of two tracers: Carbon-14 (^{14}C) and Trichlorofluoromethane (CFC11) collected in the 1990s (latitude, longitude, depth), zonally averaged.
- Second source of information: climate model output at six different values of K_v. Model used: UVIC, an earth system model.
- Data size: 3706(observations); 5926(model) per tracers.
- K_v is interesting because it is a key parameter in modeling the Atlantic Meridional Overturning Circulation, related to climate change

CFC Example



- Bottom right: observations
- Remaining plots: climate model output at 3 settings of K_v.

GP model for emulation: climate model

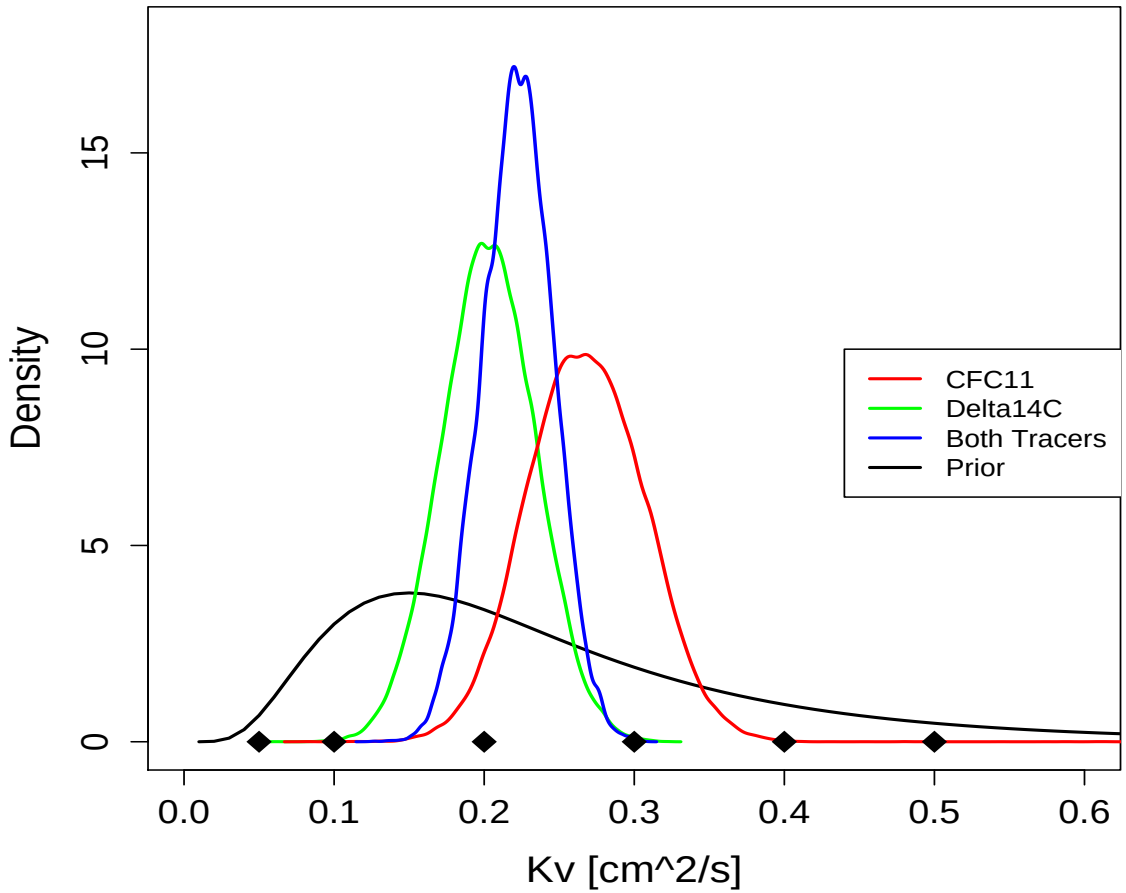
- Unlike the toy example, the output from the climate model is much more complicated — for each $\mathbf{K_v}$ we have two related spatial fields (not a single point).
- In principle: spatial locations are just treated as one more dimension (parameter locations + spatial locations)
- We fit a more sophisticated Gaussian process model to the climate model output.
- Need to construct a flexible model but construct the GP such that it is computationally tractable. (We use kernel convolutions and a hierarchical modeling approach.)

Computational Issues

- Matrix computations are $\mathcal{O}(N^3)$, where N is the number of observations. If we are not careful about modeling, N could be on the order of tens of thousands.
- Need long MCMC runs since there may be multimodality issues, and the algorithm mixes slowly.
- Used reduced rank approach based on kernel mixing (Higdon, 1998): continuous process created by convolving a discrete white noise process with a kernel function.
- Special structure + Sherman-Woodbury-Morrison identity used to reduce matrix computations.
- In MLE (optimization) step: take advantage of structure of hierarchical model to reduce computations.

Details in Bhat, Haran, Tonkonojenkov, Keller (2012)

Results for K_v inference



Probability density functions (pdfs): the prior pdf (assumption *before* using data), and posterior pdfs (*after* using the tracers.)

Summary

GPs are not just important models for dependent processes.

Lots of uses:

- Emulating complex computer models and performing inference based on these models.
- Flexible models for non-Gaussian data. Using a generalized linear model approach, GPs can be used for modeling non-Gaussian data as well. For this reason they are particularly useful for classification problems.

There are lots of open research problems, e.g. computational challenges presented by dimensionality of data; dimensionality of input space; adaptive design problems; dynamic emulation/calibration; optimization-specific problems

Some References

- Kennedy, M.C. and O'Hagan, A.(2001), Bayesian calibration of computer models, *J of Royal Stastitical Society (B)*.
- Sanso, B. and Forest, C.E. and Zantedeschi, D (2008) , Inferring Climate System Properties Using a Computer Model, *Bayesian Analysis (with discussion)*.
- Bhat, K.S., Haran, M., Tonkonojenkov, R., Keller, K. (2012) "Inferring likelihoods and climate system characteristics using climate models and multiple tracers."
- Bhat, K.S., Haran, M., Goes, M. (2010) "Computer model calibration with multivariate spatial output"

More References

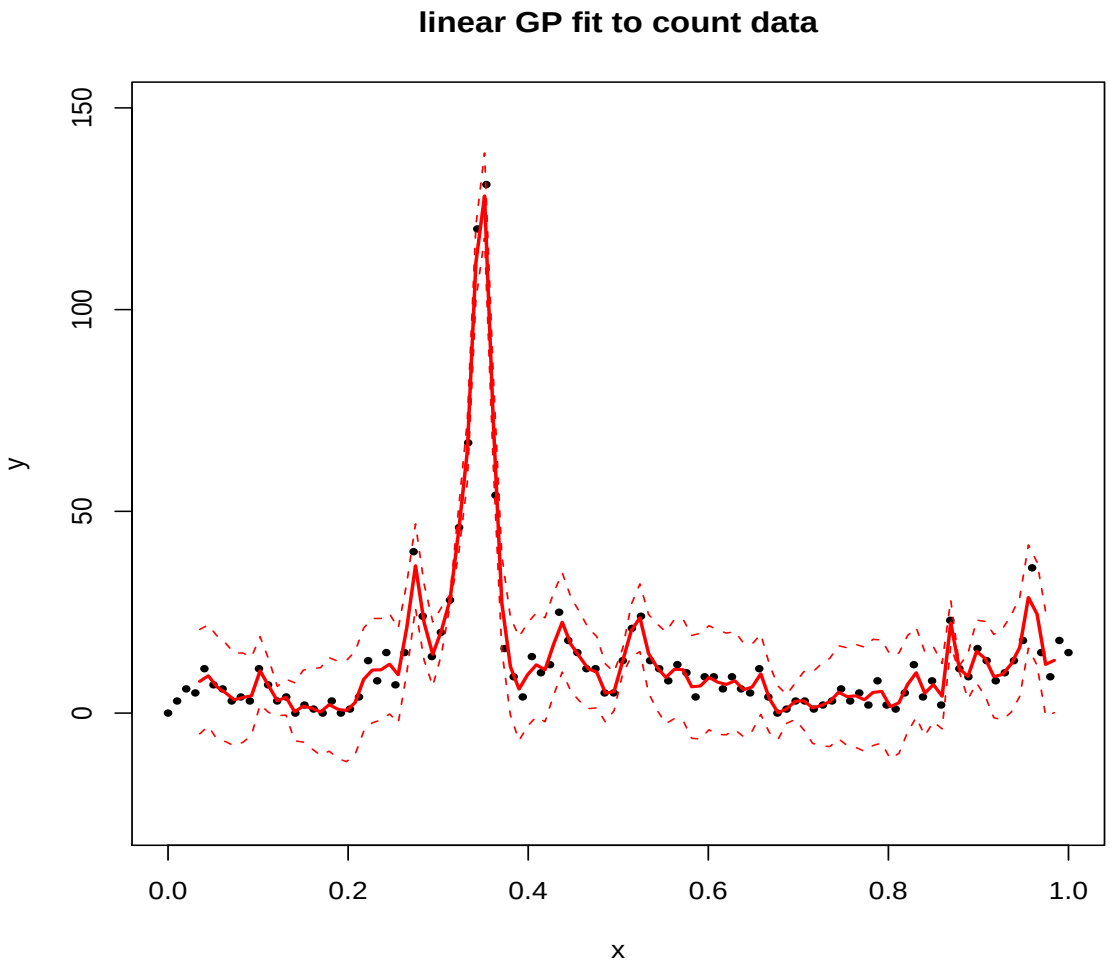
- Rasmussen and Williams (2006) “Gaussian processes for machine learning”. Available online for free!
- Gaussian processes website
- Papers/tutorial related to SAMSI (Statistical and Applied Mathematical Sciences Institute in Durham, NC) program on uncertainty quantification
- Talks/papers on MUCM projects (Managing Uncertainty in Computer Models).

Many references: can find these by searching for authors mentioned here, and following links provided in their papers.

GPs and Generalized Linear Mixed Models

First, note that linear GP models can work surprisingly well even for some (marginally) non-Gaussian data. E.g.

$Y_1, \ldots, Y_n \mid \lambda \sim \text{Poisson}(\lambda)$ with $\log(\lambda) \sim$ zero-mean GP.



Spatial Generalized Linear Models (SGLMs)

A general framework for modeling non-Gaussian output using the generalized linear model framework, following Diggle et al. (1998):

- Model $Y(\mathbf{x})$ conditionally independent with distribution f given parameters β, Θ, latent variable $w(\mathbf{x})$,

$$f(Y(\mathbf{x})|\beta, \Theta, w(\mathbf{x})),$$

  with $g(E(Y(\mathbf{x}))) = \eta(\mathbf{x}) = X(\mathbf{x})\beta + w(\mathbf{x})$, η is a link function (for example the logit link).
- Now model $\{w(\mathbf{x}), \mathbf{x} \in D\}$ as a Gaussian process.
- Bayesian approach: specify priors for Θ, β.
- Inference based on $\pi(\Theta, \beta, \mathbf{w} \mid \mathbf{X})$.

Classification: logistic regression

- GP model for real-valued output is analogous to linear regression.
- GP model for binary classification is analogous to logistic (or probit) regression.
- Model:
  - $p(\mathbf{x})$= P($y = 1$)=1-P($y = -1$).
  - logit($p(\mathbf{x})$) $= \log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \beta x + w(\mathbf{x})$.
  - $\{w(\mathbf{x}), \mathbf{x} \in D\}$ is assumed to be a Gaussian process, say GP with mean zero and covariance function with parameters Θ.
  - Priors for β and GP covariance parameters, Θ.
- Alternative: probit-GP or 'clipped GP' (De Oliveira, 2007).
- Bayesian inference: posterior distribution $\pi(\Theta, \beta, \mathbf{w} \mid \mathbf{y}, \mathbf{x})$.
- Prediction: posterior predictive distribution at inputs $\mathbf{x}^*$.

Classification with GPs: computing

- Dimensionality of posterior distribution grows according to the number of data points.
  - Linear GP: $\pi(\boldsymbol{\beta}, \Theta \mid \mathbf{X})$.
  - SGLM for binary data: $\pi(\boldsymbol{\beta}, \Theta, \mathbf{w} \mid \mathbf{X})$ where $\mathbf{w}$ is typically n dimensional where n=number of data points (size of training data).
- Computing is much harder for SGLM for binary data than for linear GP. For MCMC, two-pronged problem:
  - Computing time per iteration of the algorithm is much more expensive due to large number of parameters and expensive matrix operations.
  - The strong dependence among parameters lead to 'slow mixing' Markov chain — it takes many more iterations of the algorithm to get good estimates.

Classification with GPs: computing

- Sophisticated MCMC algorithms (cf. Neal, 2001).
- Laplace approximation: replace posterior distribution by a multivariate normal distribution centered at its mode, with variance given by its Hessian evaluated at the mode.
  Issue: symmetric approximation to a skewed posterior.
- Expectation propogation algorithm (Minka, 2001).
- A study in Rasmussen and Williams (2006) suggests that the expectation-propogation algorithm is more reliable than the Laplace approximation.
- Reasonably well constructed MCMC algorithm run for a long time still seems safest, though computationally very expensive. As in the linear GP case, for large n need to induce sparsity.