Gaussian Processes and Complex Computer Models

Astroinformatics Summer School, Penn State University

June 2018

Murali Haran

Department of Statistics, Penn State University

Modeling with Gaussian processes

1. Gaussian processes (GPs) are useful models for spatial data and for constructing flexible models for space-time processes. Kriging, continuous space-time models
2. GPs are very useful for analyzing complex computer models Emulation and calibration of computer models
3. GPs are commonly used for nonparametric regression and classification. Machine learning
4. 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

Lots of overlap among above uses. Focus on (2), (3)

Outline

Gaussian processes intro

Computer model emulation and calibration

Computer model calibration

Classification

What are Gaussian processes?

Infinite-dimensional stochastic process, $\{Y(\mathbf{s}), s \in D\}$ $D \subset \mathbb{R}^d$

- Infinite-dimensional $\Rightarrow$ Y lives everywhere in domain D, even if it is only observed at finite set of locations
- Every finite-dimensional $(Y(\mathbf{s}_1), \ldots, Y(\mathbf{s}_n))$ for $s_i \in D$, $i = 1, \ldots, n$, has a multivariate normal distribution
- E.g. exponential covariance function, $C(\mathbf{s}_i, \mathbf{s}_j) = \sigma^2 \exp(-|\mathbf{s}_i - \mathbf{s}_j|/\phi)$ where $\sigma^2, \phi > 0$, resulting in $n \times n$ covariance matrix, $\Sigma(\theta)$.
- This is a very popular model for dependent data, spatial or temporal. Covariance of random variables that are near each other (in space or time) is higher than random variables that are far apart. Dependence decreases according to covariance function.

Reasons to use Gaussian processes for spatial data

- Better interpolation, while accounting for errors appropriately
- Can lead to superior statistical analysis (e.g. lower mean squared error).
- Fitting an inappropriate model for the data, say by ignoring dependence, may lead to incorrect conclusions. e.g. underestimated variances
- Also: Very difficult to find other ways to construct a process that allows for dependence, particularly dependence that varies as a function of distance.
  - This is why Gaussian processes are used for modeling non-Gaussian spatial data as well: done by embedding a GP within another model ("latent Gaussian process models" using generalized linear mixed models framework)

Specifying Gaussian processes

- A general statistical model for observations:

  mean (deterministic) process + **error** (stochastic) process.

- Basic Gaussian process-based model at any location $\mathbf{s}$,
  $Y(\mathbf{s}) = \mu_{\boldsymbol{\beta}}(X(\mathbf{s})) + \epsilon_{\boldsymbol{\theta}}(\mathbf{s})$,
  - Mean function $\mu_{\boldsymbol{\beta}}(X(\mathbf{s}))$ is a function of covariates $X(\mathbf{s})$, for example $\mu_{\boldsymbol{\beta}}(X(\mathbf{s})) = \boldsymbol{\beta} X(\mathbf{s})$, linear function of a spatial coordinate ($X(\mathbf{s}) = \mathbf{s}_1$)
  - Mean function could represent a deterministic physical model
  - Error/noise $\epsilon_{\boldsymbol{\theta}}(\mathbf{s})$ is specified by Gaussian process with a covariance function that depends on parameters $\boldsymbol{\theta}$
- Fit the above model to observations $Y(\mathbf{s}_1), \ldots, Y(\mathbf{s}_n)$ to obtain estimates of $\boldsymbol{\beta}, \boldsymbol{\theta}$ via MLE or Bayes or other methods

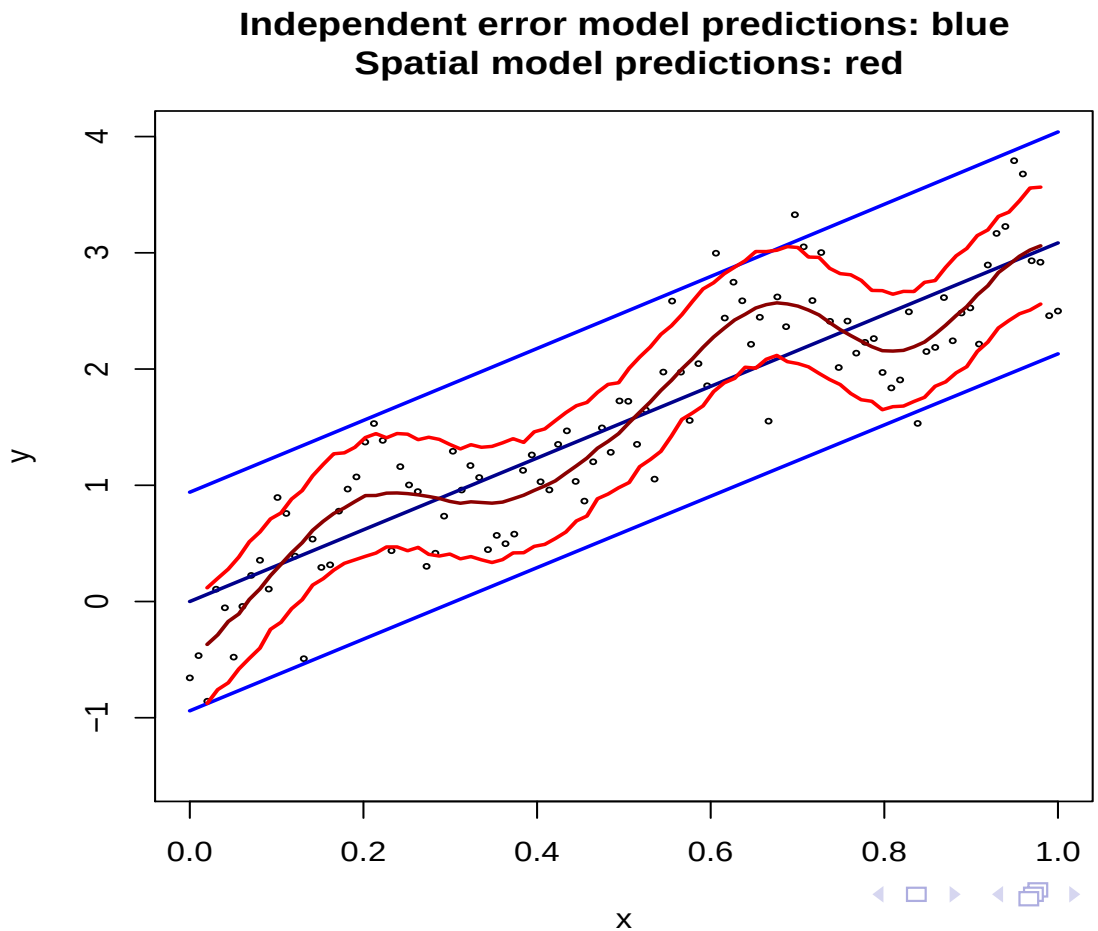
Predictions ("kriging") with Gaussian processes

- We have observations $\mathbf{Y} = (Y(\mathbf{s}_1), \ldots, Y(\mathbf{s}_n))$
- We want to predict at a new set of locations $(\mathbf{s}_1^*, \ldots, \mathbf{s}_m^*) = \mathbf{s}^*$
- Predict using conditional expectation, $E(\mathbf{Y}(\mathbf{s}^*)|\mathbf{Y})$
- This prediction minimizes mean squared prediction error
- Multivariate normal distributions make this easy: conditional distributions are all multivariate normal. Just need to calculate the new mean and covariance of the conditional distributions

Prediction example

Simulated data from time series (dependent) model.

Independent error model misses wiggles. GP error model picks up wiggles: better interpolator

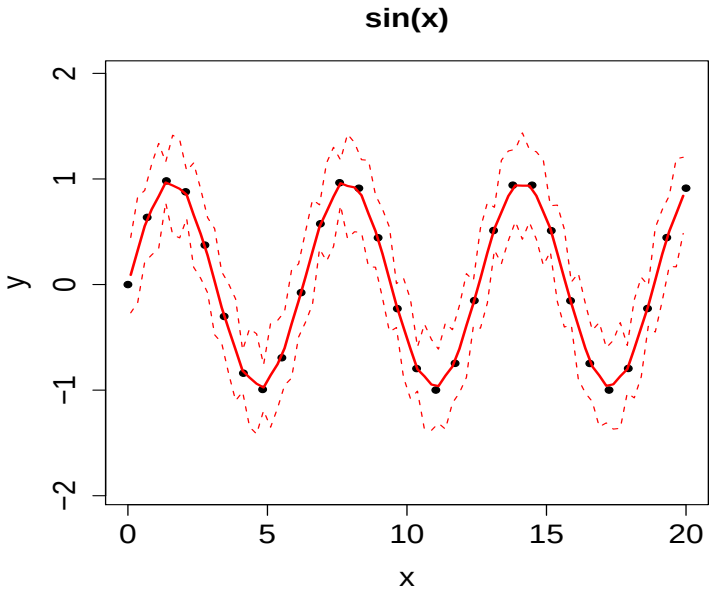


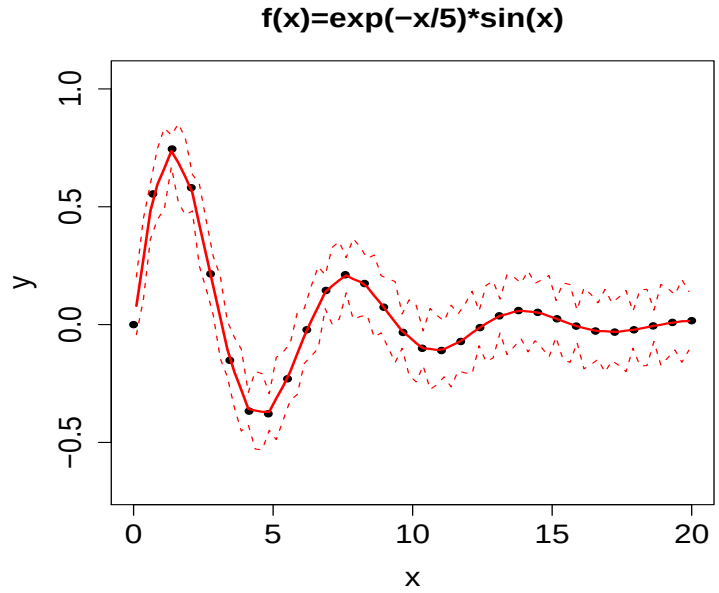
Outline

Gaussian processes for non-spatial settings

- Simple spatial model setting: $Y(\mathbf{s}) = \beta X(\mathbf{s}) + \epsilon(\mathbf{s})$ where $\mathbf{s}$ vary over index set $D \subset \mathbb{R}^d$. The stochastic process is therefore $\{Y(\mathbf{s}) : s \in D\}$. Often $d = 2$ or 3. Mainly concerned with *modeling spatial dependence* among Ys although well known that this model is also useful for protecting against model (mean function) misspecification, unmeasured spatially-varying covariates etc. (cf. Cressie, 1993).
- In the machine learning setting: let locations $\mathbf{s}$ correspond to inputs so distances are no longer physical but in 'input space,' and $Y(\mathbf{s})$ are 'outputs'. Interpolate assuming input values close to each other result in outputs that are similar.

GP for function approximation: toy 1-D example





Suppose we ran the two toy computer models at 'input' values x equally spaced between 0 and 20 to evaluate the function (black dots). Can we predict between black dots?

Pretend we don't know the model (functions). The red curves are interpolations using the same, simple GP model:

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

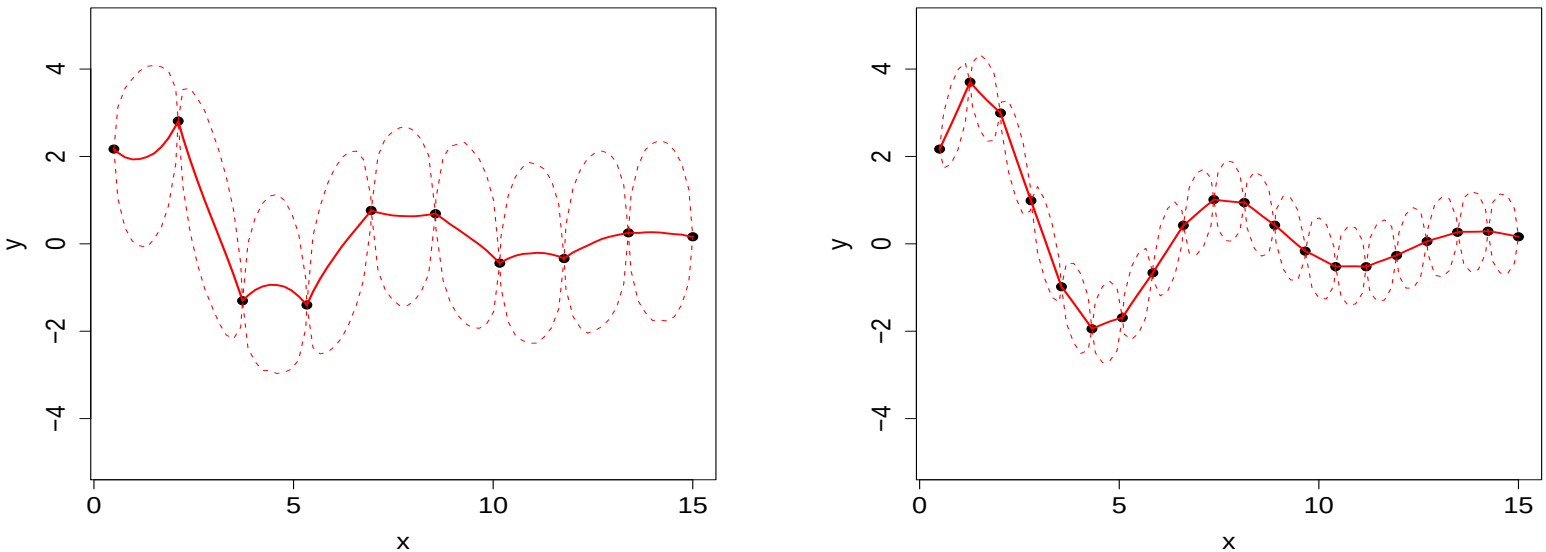
GPs for function approximation

- The usual 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).
- **"Nonparametric"**
  - GP models allow a simple covariance to substitute for a complicated mean with an unknown functional form.

GPs for modeling complicated functions

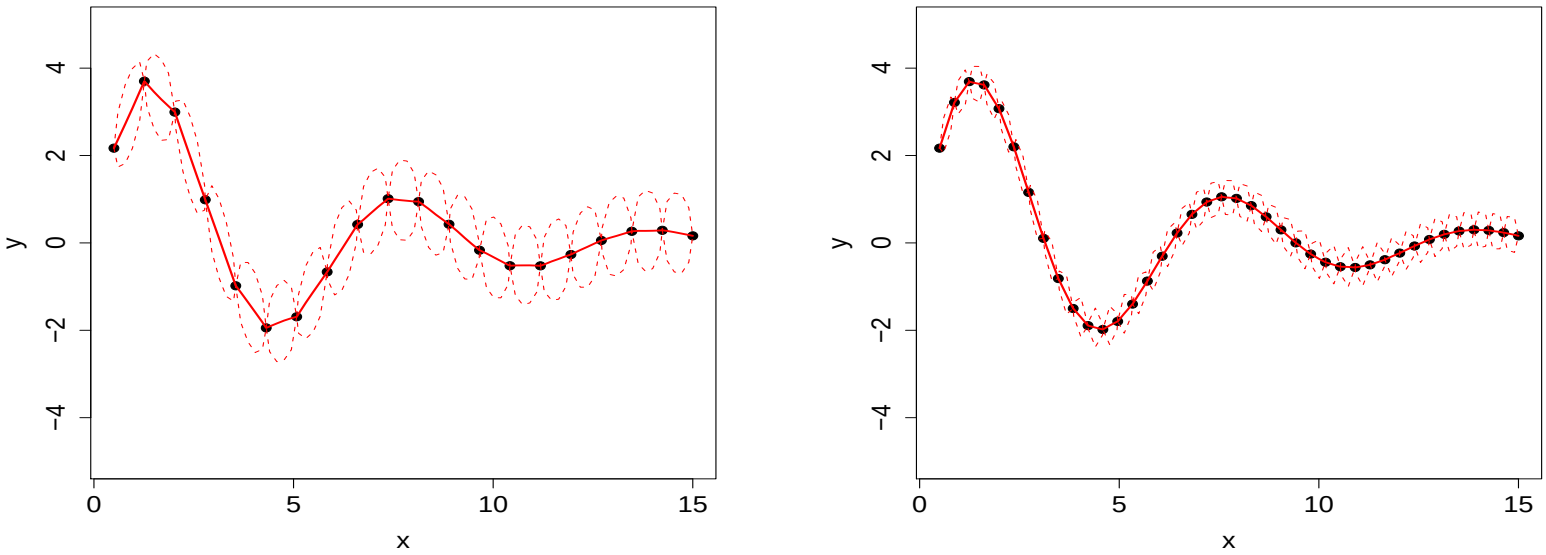
- Consider the following problem: We are interested in modeling the response y as a function of a predictor x so $y = f(x)$.
- We have observations in terms of (response,predictor) or (input, output) pairs: $(x_1, y_1), \ldots, (x_n, y_n)$.
- Based on the observations, called a 'training set' in machine learning, want to build a model that will predict y for a new set of inputs $(x_1^*, \ldots, x_n^*)$.
- May not want to assume a particular functional form for relationship between x and y. Use a GP prior on $f(x)$.
- With GPs: *statistical interpolation*, obtain uncertainty estimates.

GP function emulation: toy example



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

GP function emulation: toy example



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

GP benefit: flexible and at the same time, get prediction uncertainties

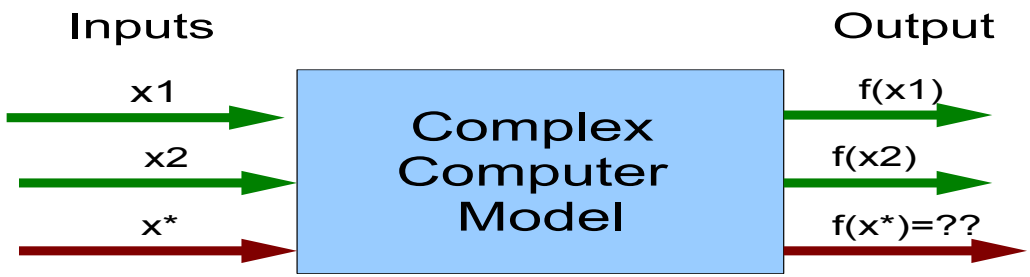
Nice app to visualize this: https://chi-feng.github.io/gp-demo/

Gaussian processes and deterministic models

- Scientists are building increasingly complex models
- Numerical solutions of complex mathematical models
- Translated into computer code to study simulations of their physical processes under different conditions
- There is often great interest in learning about the parameters of the physical model that best explain observations of the phenomena, either to 'tune' the model or because the parameters are of scientific interest.
- These 'fitted' models may be useful for predictions
- The notion of 'uncertainty' — many modelers talk about uncertainty in terms of lack of knowledge about the 'best' input. Fits nicely into a Bayesian formulation.
- In some cases, the models may be stochastic.

Deterministic 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

- An emulator (or 'meta-model') is an 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.
- The emulator serves as a surrogate for the computer model, and is much faster/simpler. Hence, it is possible to simulate output from the emulator very quickly.
- The advantage of doing it in a probabilistic framework:
  - Uncertainties associated with interpolation, e.g. greater uncertainty where there is less training data information.
  - Probability model: useful for statistical inference.
  - "Without any quantification of uncertainty, it is easy to dismiss computer models." (A.O'Hagan)

GP model emulation

- GP model emulation is just like prediction using a GP for spatial data: fit a GP model to the training data (model runs), then make predictions at new inputs based on fitted model, conditioning on training data.
- Gaussian processes are extremely useful for emulating complicated models in situations where:
  - Simulator output varies smoothly in response to changing its inputs.
  - There are no discontinuities or very rapid changes in responses to inputs.
  - The number of inputs is relatively small.

Example: ice sheet model emulation and calibration

Takes $\approx$ 48 hours at each parameter setting, so build a fast **emulator** (Chang, Haran et al., 2014, 2016a,b).

- Input (parameters) describe key characteristics of the dynamics of the West Antarctic ice sheet. e.g. ocean melt coefficient: sensitivity of ice sheet to ocean temperature
- Output: Model equations predict ice flow, thickness, temperatures, and bedrock elevation, through thousands to millions of years.
- Field data: ice sheet satellite images, reconstructed past
- **Calibration**: use field data to learn about parameters. Use parameters (with uncertainties) to project ice sheet future
- "With uncertainties" = probability distribution of parameters
- Bayesian approach: distribution of parameters *given* data

GPs for computer models: background

- Basic theory for prediction with GPs (Wiener, Kolmogorov, 1940s). Regression applications (Whittle, 1963).
- Used in time series, spatial statistics, geostatistics (Journel and Huijbregts, 1978; Cressie, 1993).
- Parallel literature in splines (Kimeldorf and Wahba, 1970), connection between splines and GPs (Wahba, 1990).
- O'Hagan (1978): Gaussian processes for fitting curves.
- Emulators based on GPs: Sacks et al. (1989), Welch et al. (1992), Morris et al. (1993), Higdon et al. (2004),...
- Currin et al. (1991) is the first Bayesian approach to GP-based emulation.
- Some books: Santner et al. (2003), Fang et al. (2005), Rasmussen & Williams (2006) (machine learning).

The GP emulator mean function

- GP model needs the specification of:
  - Mean function at any input x, $\mu(x)$.
  - Covariance function (often assuming stationarity, isotropy), specify $\text{Cov}(x_i, x_j)$ for any pair of inputs x_i, x_j using a parametric family.
- Since the idea of GP modeling is to pick up non-linearities via the covariance rather than the mean function, it makes sense to keep the mean fairly simple.
- In practice, useful to capture important trends/relationships between the inputs and outputs to the extent possible. By default, include a simple linear regression mean function.
  - The emulator can make predictions with small variance.
  - Helps keep residuals reasonably small.
  - Important in producing reasonable predictions far from training data.

The GP emulator mean function

- The GP is then essentially acting as a smooth interpolator of the residuals $y(x) - \mu(x)$, where the smoothness of the interpolation is controlled by the covariance of the GP.
- Strong dependence here says the input has a very smooth and predictable effect on the output.
- Weak dependence here says the input has a more variable effect on the output. Training inputs that are not very close to the input at which predictions are desired are not very informative.

The GP emulator covariance function

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

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

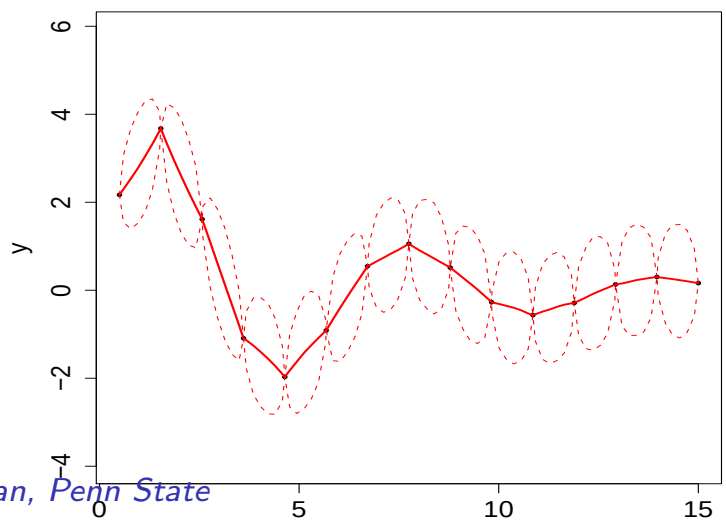
- Usually not enough information to learn about α_j so fix it at some value, say 1.9 (α_j =2 results in over-smooth processes. Also leads to numerical instabilities.)
- Above: convenient, especially for multiple inputs.

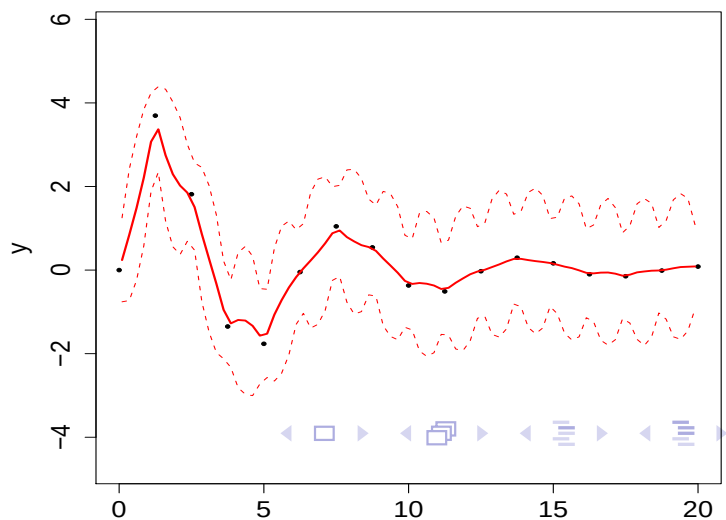
Whether to include 'nuggets'

Covariance function when including nugget:

$$\psi 1(\mathbf{x}_i = \mathbf{x}_j) + \kappa \exp\left(-\sum_{j=1}^{d} |x_j - x_j^*|^{\alpha_j}/\phi_j\right), \psi > 0.$$

Whether to include measurement or micro-scale error ('nugget') term is a modeling choice. Depends on whether it is appropriate to have a perfect interpolator (left) or not (right).





Whether to include 'nuggets' [cont'd]

- For stochastic model emulation: makes sense to include the nugget so that prediction at the training data set will have some variability.
- For deterministic model emulation: intuitively want to return observed model output at training input with zero variance. However, there may be reasons to still include it:
  - Surrogate for micro-scale error.
  - Computational considerations — (i) numerical stability for matrix operations, (ii) helpful structure for fast computing, for e.g. use of Sherman-Morrison-Woodbury matrix identity.

Summary of emulation/nonparametric regression

- Gaussian processes are very useful for emulating (approximating) complex computer models
- The ideas discussed here are closely related to ideas for doing nonparametric regression with GPs
- Replace parameters (θ) with predictors and model output $Y(\theta)$ with responses, and we have a regression problem with essentially the same ideas
- Major differences: Nonparametric regression problems typically do not involve an expensive step to produce data. Hence, common to have lots of data. This can make some things easier (more information, handle higher-dimensional predictors/parameters), and some things harder (higher dimensions can make computing challenging)

High-dimensional emulation/calibration

- This is an important problem and active research area.
- I will not attempt to summarize things here, but instead point out that it is worth looking at literature in different areas:
  - Spatial statistics methods for Gaussian processes with large data sets
  - Machine learning methods (start with Rassmussen and Williams GP book online)
  - Engineering/statistics literature

Outline

Computer models and inference

- Suppose we have a deterministic model and some field observations (field data) to go along with it. How do we infer the values of the parameters in the deterministic models?
- Several issues: (a) the model is deterministic, not even a statistical model! (b) the model may be very complicated, impossible to write down in closed form, (c) the model may be so complicated that it takes *very* long to simulate values from it.
- Issues (b) and (c) arise even when the model is stochastic.

A two-stage approach to emulation-calibration

1. Emulation step: Find fast approximation for computer model using a Gaussian process (GP).

2. Calibration step: Infer climate parameter using emulator and observations, while accounting for data-model discrepancy

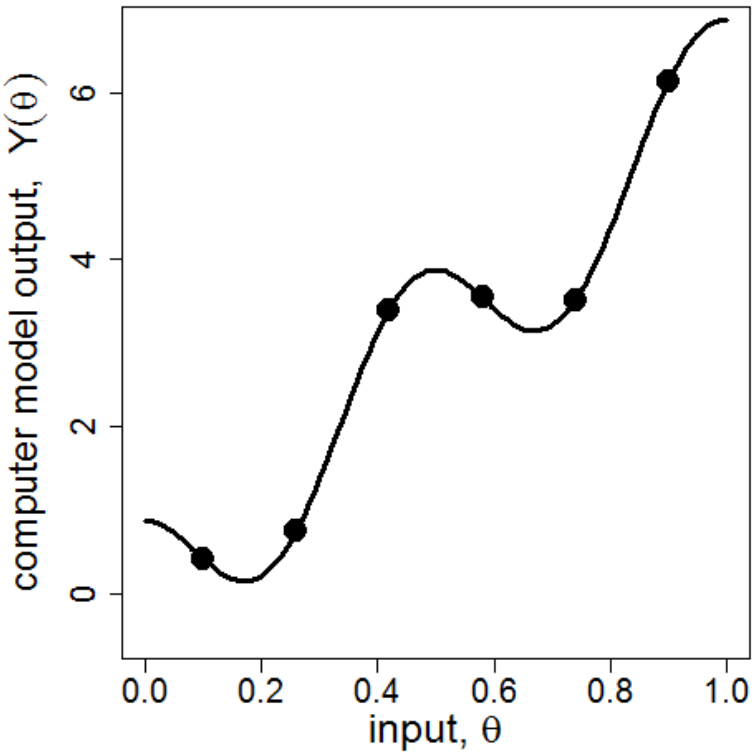
   Modularization

   - Liu, Bayarri and Berger (2009)
   - Bhat, Haran, Olson, Keller (2012)
   - Chang, Haran, Applegate, Pollard (2016a; 2016b)

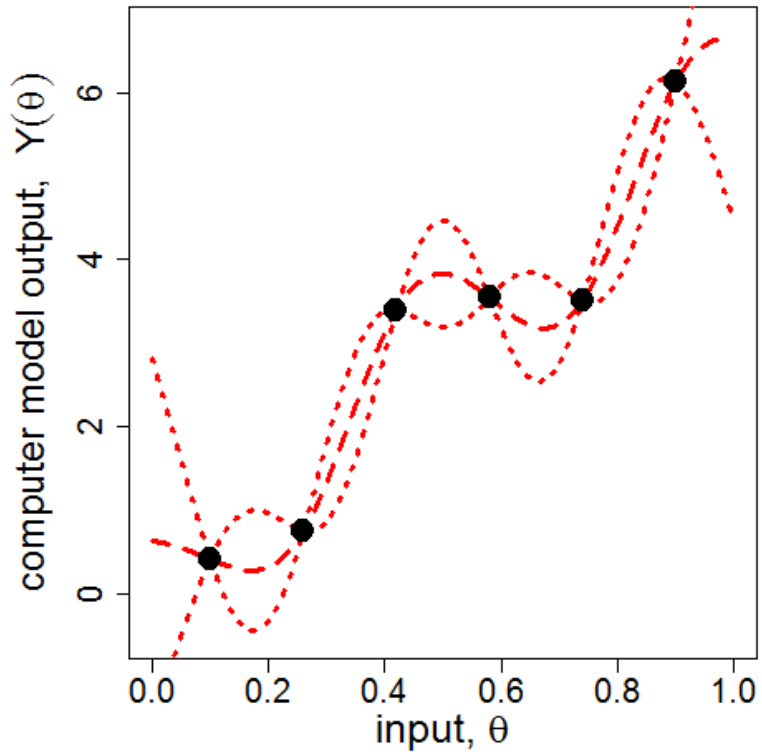
Non-modular approaches: Higdon et al. (2008), Sanso et al. (2008),..

Emulation step

Toy example: model output is a scalar, and continuous.



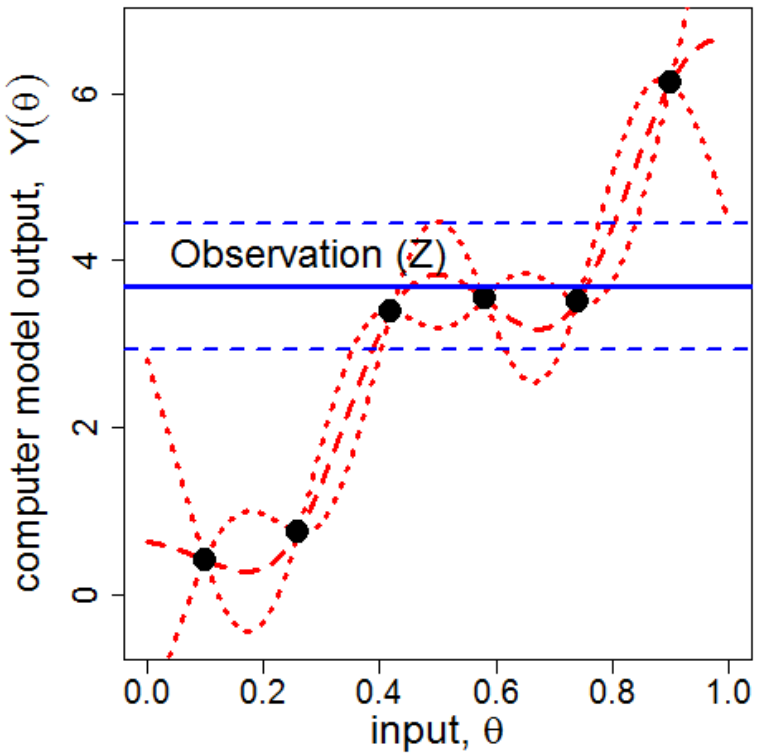
Computer model output (y-axis) vs. input (x-axis)



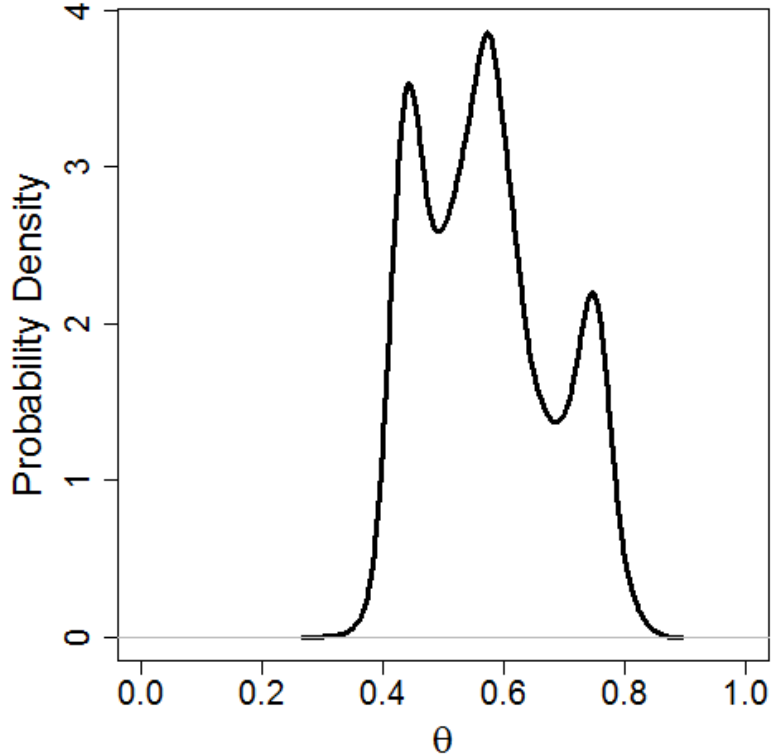
Emulation (approximation) of computer model using GP

Calibration step

Toy example: model output, observations are scalars

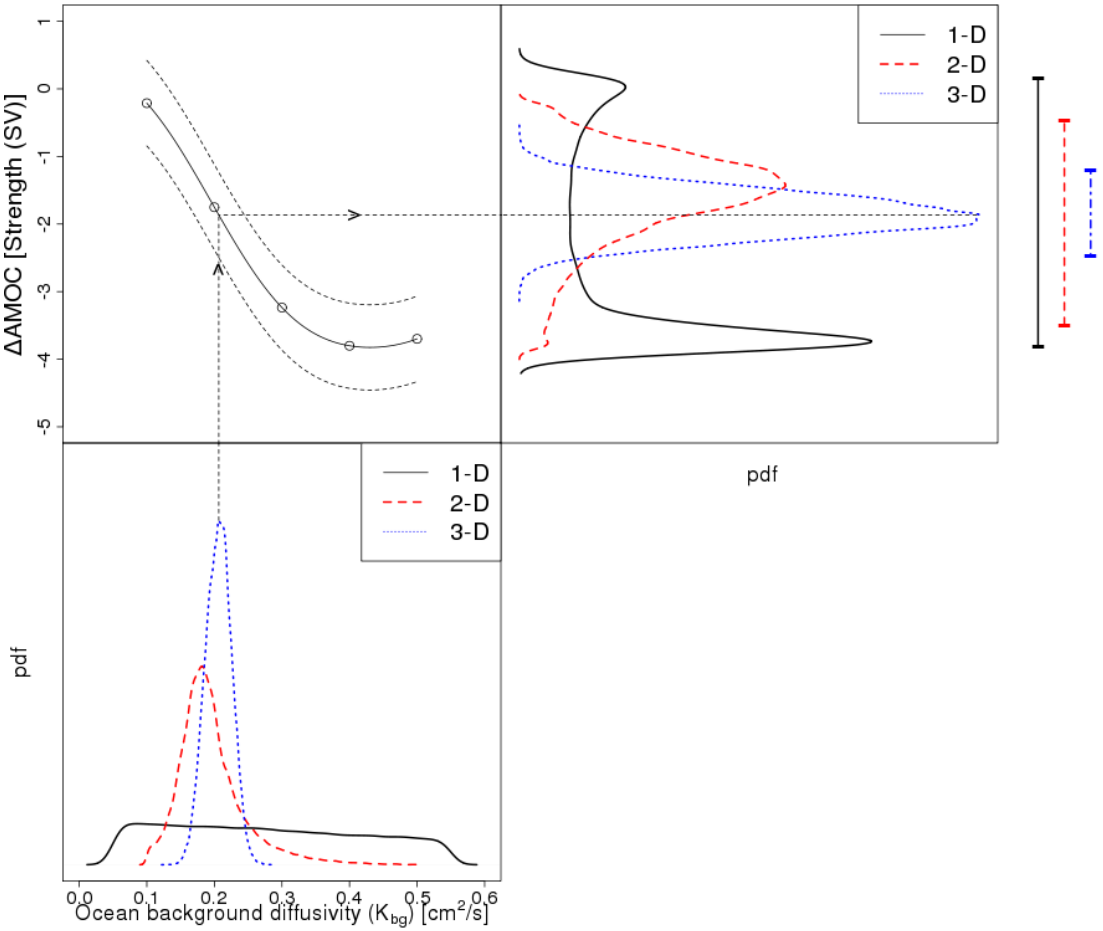


Combining observation and emulator



Posterior PDF of θ given model output and observations

How does all this affect scientific conclusions?



Parametric uncertainties ⇒ model output/prediction uncertainty

Can see how different data sets, resolutions, aggregation-levels impact uncertainties (Chang, Haran, et al., 2014)

Computer model calibration

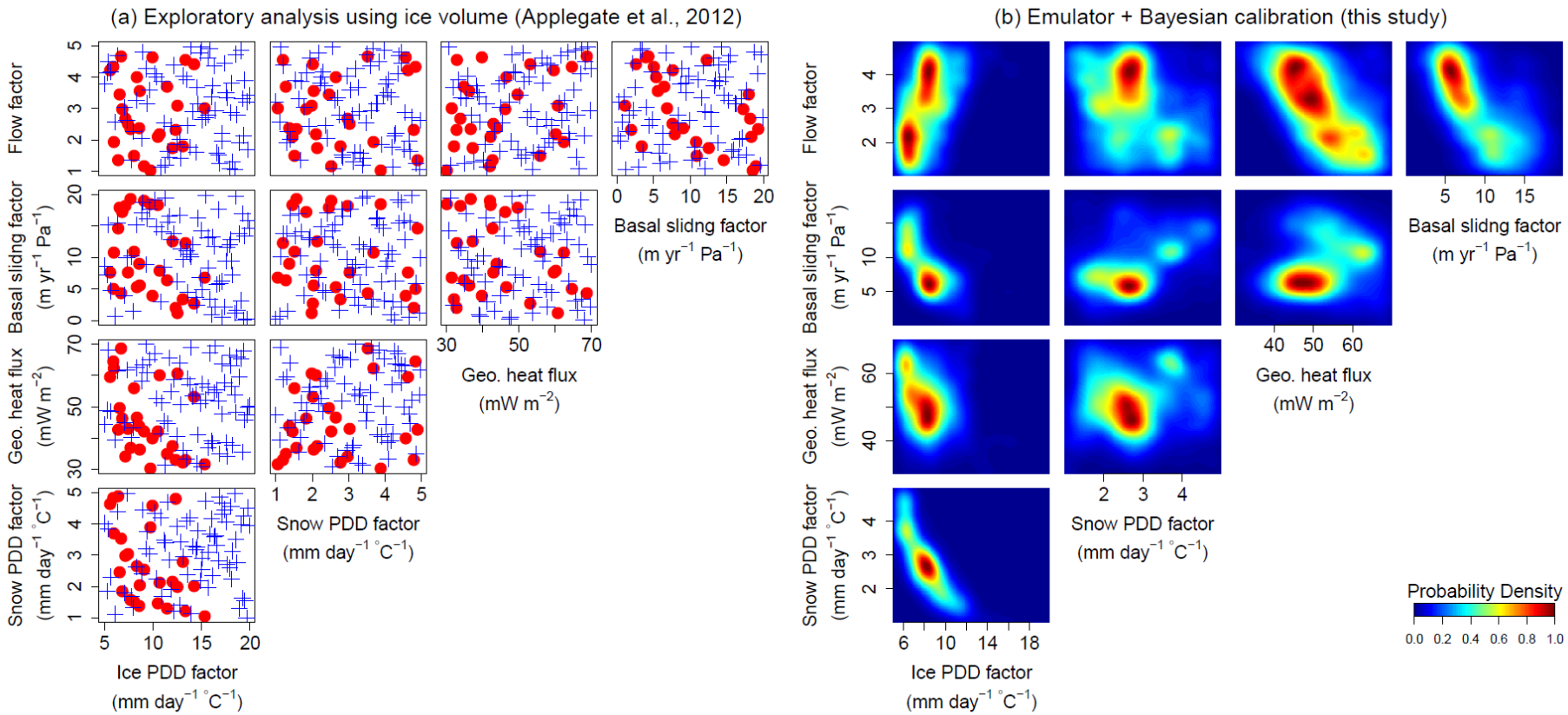
- Statistical problem: Given data sources (i) computer model output at several inputs, and (ii) observations of the real process being modeled by the computer code, what is the value of the input that best 'fits' the observations?
- Want to infer θ
  - Computer model output $\mathbf{Y} = (Y(\theta_1), \ldots, Y(\theta_n))$.
  - Observation Z, assumed to be a realization of computer model at 'true' θ + discrepancy + measurement error.
- Ideally done in a Bayesian setting:
  - There is often real prior information about θ.
  - The likelihood surface for θ may often be highly multimodal; useful to have access to the full posterior distribution.
  - If θ is multivariate, may be important to look at bivariate and marginal distributions (easier w/ sample-based approach).

How does statistical rigor help scientists?

- Non-statistical calibration: search input space for best fit to the data, using a crude measure of fit (e.g. least squares), many other reasonable methods
- Does not provide a framework for obtaining probability distributions for θ, which is often of great interest.

1. We account for (epistemic) uncertainties in emulation
2. We provide *real* probability distributions, very important for impacts/risk quantification.
3. We use all available information (no aggregation): often reduces uncertainties.
4. We provide sharper/more useful results.

Example of sharper, interpretable results



Left: ad-hoc method. Right: **probability distributions**

Chang, Haran, Olson, Keller (2014), *Annals of Applied Stats*

Computer model calibration outline

- Field data = computer model + model discrepancy (structural error, biases) + measurement error

$$Z(x) = Y(x, \theta) + \delta(x) + \epsilon(x).$$

x: controllable input, θ is unknown input.

- It is important to model $\delta(x)$ (not appropriate to assume i.i.d. error), as this may result in over-fitting/biased θ as it tries to compensate for model inadequacy.
  - GP model for $Y(\theta)$ since it is an unknown function.
  - GP model for $\delta(x)$. It is also an unknown function.
  - $\epsilon(x) \overset{iid}{\sim} N(0, \psi), \psi > 0$.
  - Replications (multiple field output at same x) are useful.
- Obvious that there are a lot of identifiability issues.

Computer model calibration [cont'd]

- Scientists can often provide strong prior information for θ.
- Priors for model discrepancy, Gaussian process covariance may not be obvious. Work on reference priors (Berger et al., 2001; Paulo, 2004; De Oliveira, 2007), though these can be computationally expensive.
- Markov chain Monte Carlo (MCMC) for sampling from posterior distribution, $\pi(\Theta_Y, \boldsymbol{\beta}_Y, \Theta_\delta, \boldsymbol{\beta}_\delta, \theta \mid Z, \mathbf{Y})$. Covariance, regression parameters $\Theta_Y, \boldsymbol{\beta}_Y$ for emulator and $\Theta_\delta, \boldsymbol{\beta}_\delta$ for discrepancy; variance of i.i.d. error ψ.
- Posterior distribution is likely to be multimodal in many cases: need well designed MCMC algorithm that escapes local modes, e.g. slice sampler. Run long chains, assess MCMC s.errors.

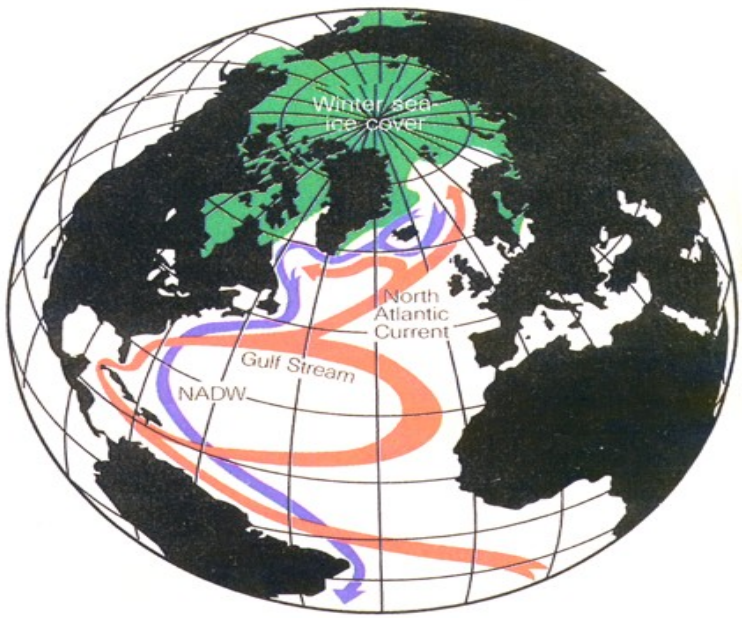
Uncertainty quantification

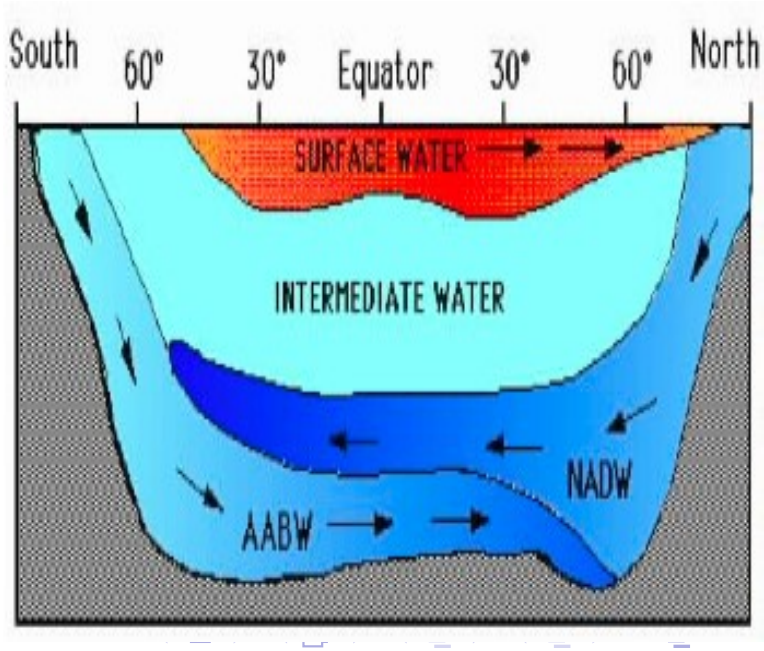
Let input be x, output be $y = f(x)$ and real world value that model is trying to predict be z. Different kinds of uncertainty:

- Structural uncertainty:
  - Discrepancy between model and reality, for example due to incomplete physics. $z - f(x^*)$ where x^* is 'best input'.
  - Numerical issues, coding errors (computing issues).
- Input (or parameter) uncertainty:
  - Initial values/boundary conditions unknown.
  - Forcing inputs (perturbations to the system) uncertain.
  - Key model parameters unknown.
- Goal is often to provide a probability distribution for z while incorporating the uncertainties above.
- Hard to assess uncertainties, especially structural uncertainty, due to identifiability issues, lack of data.

Example from climate science

- What is the risk of human induced climate change?
- Example of climate change: potential collapse of meridional overturning circulation (MOC).
- An MOC collapse may result in drastic changes in temperatures and precipitation patterns.





(plots: Rahmstorf (Nature, 1997) and Behl and Hovan)

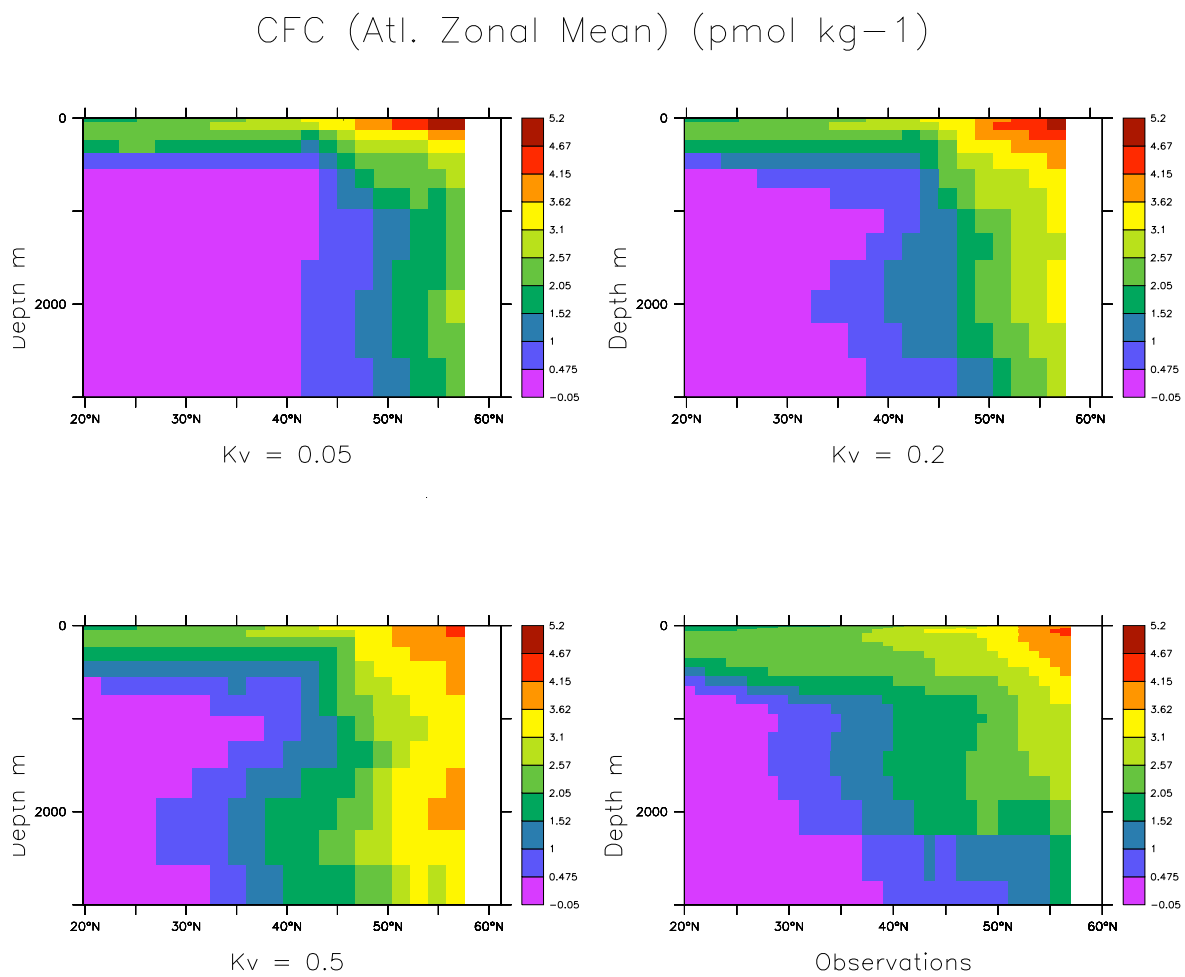
Motivation-MOC

- MOC phenomenon: Movement of water from equator to higher latitudes, deep water masses created by cooling of water in Atlantic, resulting in sea ice formation. Result is denser salt water, which sinks, causing ocean circulation.
- MOC weakening results in disruptions in the equilibrium state in the climate, may lead to major temperature and precipitation changes and shifts in terrestrial ecosystems.
- Key source of uncertainty in AMOC projections is uncertainty about background ocean vertical diffusivity, $\mathbf{K_v}$.

Information sources

- $\mathbf{K_v}$ 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 $\mathbf{K_v}$.
- Data size: 3706(observations); 5926(model) per tracers.

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). We fit a more sophisticated Gaussian process model to the climate model output.
- We can now use this GP model instead of the very complicated climate model — this provides a connection between $\mathbf{K_v}$ and the climate model output, in this case the tracers CFC-11 and C-14.
- Model for the observed CFC-11 and C-14: can use the GP model + allow for additional sources of structural uncertainty and bias.

Statistical Inference

- Notation: $Z(\mathbf{s})$: physical observations, $Y(\mathbf{s}, \boldsymbol{\theta})$: model output at location $\mathbf{s}$=(latitude, depth), and climate parameter $\boldsymbol{\theta}$.
- Climate model: Complex and requires long time to run.
- This is a computer model calibration problem.
- **Data Sources**: Observations for ^{14}C/CFC11: $\mathbf{Z}_1, \mathbf{Z}_2$ (locations at $\mathbf{S}$).
- Climate model runs at several values of $\mathbf{K_v}$: $\mathbf{Y}_1, \mathbf{Y}_2$.
- **Goal**: Inference for climate parameter $\boldsymbol{\theta}$.

Calibration with multiple spatial fields

- How can we combine information from multiple tracers (^{14}C, CFC11) in a flexible manner to infer $\mathbf{K_v}$?
- Two stage approach to obtain posterior of $\boldsymbol{\theta}$:
  - Model relationship between $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2)$ and $\boldsymbol{\theta}$ via emulation of model output $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2)$.
  - Use observations $\mathbf{Z}$ to infer $\boldsymbol{\theta}$ (parameter of interest).
- Model $(\mathbf{Y}_1, \mathbf{Y}_2)$ as a hierarchical model: $\mathbf{Y}_1|\mathbf{Y}_2$ and $\mathbf{Y}_2$ as Gaussian processes. (following Royle and Berliner (1999)).

$$\mathbf{Y}_1 \mid \mathbf{Y}_2, \beta_1, \xi_1, \gamma \sim N(\mu_{\beta_1}(\boldsymbol{\theta}) + \mathbf{B}(\gamma)\mathbf{Y}_2, \Sigma_{1.2}(\xi_1))$$

$$\mathbf{Y}_2 \mid \beta_2, \xi_2 \sim N(\mu_{\beta_2}(\boldsymbol{\theta}), \Sigma_2(\xi_2))$$

- $\mathbf{B}(\gamma)$ is a matrix relating $\mathbf{Y}_1$ and $\mathbf{Y}_2$, with parameters γ.
- βs, ξs are regression, covariance parameters.

Calibration with multiple spatial fields [cont'd]

- Based on fitted GP, obtain predictive distribution at locations of observations. This serves as the emulator.
- We then model the observations by adding measurement error and a model discrepancy term to the GP emulator:

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

where $\boldsymbol{\epsilon} = (\epsilon_1 \ \epsilon_2)^T$ is the observation error, $\boldsymbol{\delta} = (\delta_1 \ \delta_2)^T$ is the model discrepancy.

- Inference on $\boldsymbol{\theta}$ performed using MCMC (integrating out other parameters).

Computational Issues

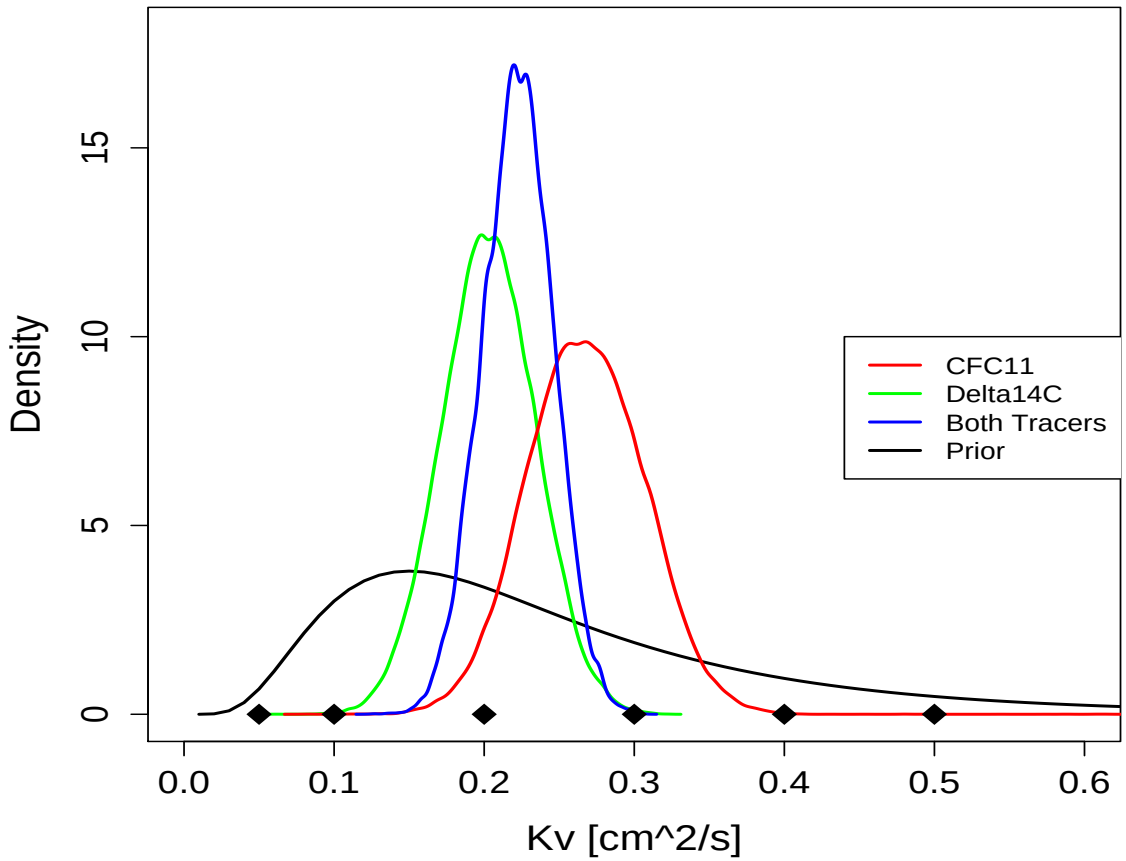
- Matrix computations are $\mathcal{O}(N^3)$, where N is the number of observations. If we are not careful about modeling, N could be on the order of tens of thousands.
- Need long MCMC runs since there may be multimodality issues, and the algorithm mixes slowly.
- Used reduced rank approach based on 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 (2009).

K_v inference: summary

- Taken the climate model output, CFC-11 and C14 spatial fields at several values for $\mathbf{K_v}$, and fit a very flexible GP model for bivariate spatial fields.
- Now, assume this GP model + model for error, discrepancy is the model for the observations of CFC-11 and C14.
- Since we have a probability model, we can perform inference for $\mathbf{K_v}$ based on the data. That is, we can use statistical techniques to learn about the values of $\mathbf{K_v}$ most compatible with all the information we have. This information will be in the form of a (posterior) probability distribution for $\mathbf{K_v}$.
- Computational considerations are important in modeling (hierarchical structure + kernel mixing approach).

Results for K_v inference



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

General challenges with emulation-calibration

- Complicated output: multivariate, time series, spatial, multivariate-spatial, qualitative and quantitative etc.
- Design (parameter values at which model is run), sequential design.
- Dynamic emulation.
- Combining information from multiple models.
- How to characterize model discrepancy, whether it is possible to learn about it.

An active area of research

Some references for emulation and calibration

- Kennedy, M.C. and O'Hagan, A.(2001), Bayesian calibration of computer models, *J Royal Stat Society (B)*.
- RB Gramacy, HKH Lee (2008) Bayesian treed Gaussian process models with an application to computer modeling , *Journal of the American Stat. Assoc*
- D Higdon, J Gattiker, B Williams (2008) Computer model calibration using high-dimensional output, *Journal of the American Stat. Assoc*
- Chang, W., M. Haran, R. Olson, and K. Keller (2014): Fast dimension-reduced climate model calibration, *Annals of Applied Statistics*

Outline

Gaussian processes for classification

- So far, we have discussed using GPs for modeling real valued output. What if we are interested in classification, where we wish to assign an input x to one of k classes, $C_1, \ldots, C_k$?
- Gaussian processes are often useful for classification problems as well.
- This is like a pure emulation problem, except the output is non-Gaussian.

Binary classification problem

- Here, we will consider binary classification so there are only two classes, C_1 and C_2.
- Example: based on several characteristics of a tissue (when performing a biopsy), we may wish to automatically classify it as being benign or malignant.
  - Training data: we measure the characteristics of several tissues and are told whether they are benign or malignant, so $\mathbf{x}$ is the vector of characteristics and $f(\mathbf{x}) \in \{-1, 1\}$ is the classification (benign, malignant). Pairs: $((\mathbf{x}_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, f(\mathbf{x}_n)))$
  - For new tissue sample with characteristics $\mathbf{x}^*$, want to automatically classify it as benign or malignant.

Classification

Green inputs/output are the training data.

Red = the input where *classifications* are desired.

Classification as a statistical problem

- Statistical problem: logistic regression.
- Given training data: inputs (say a vector of characteristics) and the corresponding classifications, what is the classification of a new input?
  - Training data: $((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))$, pairs of inputs and their 'labels' (classification).
  - For a new input $\mathbf{x}^*$, what is our prediction for y? This will involve building a model for the probability $y^* = f(\mathbf{x}^*) = 1$ based on the training data.
- The need to go beyond logistic regression: if a more flexible model is necessary to model the relationship between the input (predictors) and the logit of the probability $y = 1$. As in the linear model case, a GP-based classifier is a very flexible approach.

Spatial generalized linear models (SGLMs)

However, in many cases, a linear GP model will not work. E.g. binary output (or binary spatial data). 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 $\boldsymbol{\beta}, \Theta$, latent variable $w(\mathbf{x})$,

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

  with $g(E(Y(\mathbf{x}))) = \eta(\mathbf{x}) = X(\mathbf{x})\boldsymbol{\beta} + w(\mathbf{x})$, 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 $\Theta, \boldsymbol{\beta}$.
- Inference based on $\pi(\Theta, \boldsymbol{\beta}, \mathbf{w} \mid \mathbf{X})$.

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($y = -1$).
  - $\text{logit}(p(\mathbf{x})) = \log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \beta x + w(\mathbf{x})$.
  - $\{w(\mathbf{x}), \mathbf{x} \in D\}$ 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: $\pi(\beta, \Theta \mid \mathbf{X})$.
  - SGLM for binary data: $\pi(\beta, \Theta, \mathbf{w} \mid \mathbf{X})$ 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.

Recap

- We have discussed how Gaussian processes are very useful in non-spatial contexts. In particular, they can be used for:
  - Emulating complex computer models and performing inference based on these models, largely due to their flexibility as a prior for functions.
  - Classification problems, though this was only discussed briefly.
- Lots of open research problems, applications to interdisciplinary research (most of which generate new methodological problems), and many computational challenges.

References

- Rasmussen and Williams (2006) "Gaussian processes for machine learning". Available online for free!
- Gaussian processes website.
- Tony O'Hagan's talks/papers on MUCM projects (Managing Uncertainty in Computer Models).
- Uncertainty in Computer Models (UCM or MUCM) organization/conferences
- SAMSI program 2018-2019 (Model Uncertainty Mathematical and Statistical)
- Computer code: R packages BACCO, SAVE...