Statistical Inference for Complex Models

(with connections to computer model emulation and calibration)

Murali Haran

Department of Statistics
Penn State University

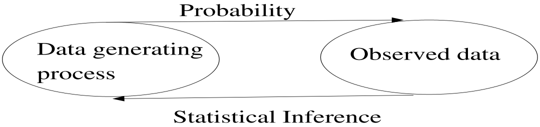
Operations Research Seminar
Penn State University
April 2011

Statistical Inference: Review of Basics

Basics: Probability and Inference

Scientific research based on data, whether from experimental studies or observational studies, can be summarized as follows:

Given the data (what we have observed), what can we infer about the process that generated the data?



Probability

Given a data generating process, what are the properties of the outcomes (observations)?

- If you know probability p of "Heads" on a coin toss, and X= # of heads on 100 tosses, what is the distribution of X?
- With a stochastic model for the spread of an infectious disease: How will the disease spread over space and time for different initial conditions/parameter values of the model? How likely is an epidemic?
- With a *deterministic* model for climate: What are the projections for climate based on different initial conditions/parameter values? What are the uncertainties associated with these projections?

Statistical Inference

Given the outcomes (our observations) and a model for the observations, what can we say about the model parameters and the model that generated the data?

- If you know X (# of heads on 100 independent coin tosses) what can you say about p?
- Given space-time data on the spread of an infectious disease, what are the (unknown) parameters of the stochastic model? Does the model fit (explain/match) the observations?
- Given data on relevant climate characteristics, what are the unknown parameters of the deterministic model for climate? *In engineering: "computer model calibration"*.

A statistical model

- A (parametric) statistical model for **data** x can be described as a set of distributions $\{f(x, \theta),$ for some set of permissible values of $\theta\}$. θ **is the parameter** or parameters of the model.
- Simple example: You observe n data points, $x_1, \ldots, x_n$ and you assume that these data values are independent of each other and have the same distribution. For instance, assume $x_1, \ldots, x_n$ have a normal distribution with some mean and some variance, i.e., $x_1, \ldots, x_n \overset{iid}{\sim} N(\mu, \sigma^2)$.
- μ, σ^2 are parameters (θ) for this simple model.

Maximum likelihood inference

- Frequentist inference: assume the parameters θ are fixed and unknown.
- Write down the distribution of the data and plug in the observations (Y) to obtain the **likelihood**, $\mathcal{L}(\theta; Y)$. The result is a function of only the parameters θ.
- Inference is based on maximizing this function: find θ that maximizes $\mathcal{L}(\theta; Y)$. To describe uncertainty about θ, use asymptotic approximations or simulation methods like the bootstrap.

Uncertainty about θ: primarily has to do with sampling distribution of $\hat{\theta}$.

Bayesian inference

- Assume the parameters θ are random variables. Uncertainty about θ captured in the distribution of θ.
- Write down prior distribution for θ. This represents uncertainty regarding θ before the data are observations and may come from scientific knowledge.
- Inference is based on the posterior distribution:

  $$\pi(\theta \mid Y) \propto \mathcal{L}(Y \mid \theta) p(\theta).$$

  Given what we have observed (Y), what is our updated information (posterior distribution) for θ?
- Often use Markov chain Monte Carlo (MCMC) to learn about $\pi(\theta \mid Y)$, which is not available in closed form.

Uncertainty about θ is automatically described by its posterior distribution.

Challenges posed by complex models

- There is extensive theory justifying the usage of maximum likelihood and Bayesian inference and prediction. They are both very general and powerful approaches to performing statistical inference.
- Models are becoming increasingly scientifically plausible and complex. Inference often becomes challenging with such models for the following reasons:
  - Models may be deterministic. May not be able to run the model quickly, nor write down closed form expressions relating input (parameters) to output.
  - The likelihood might be very expensive to evaluate. Hard to optimize, or do MCMC.

Complex Computer Models

Deterministic models

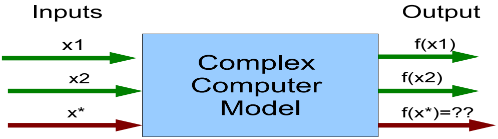
- Re-examining our notion of a model:
  - Scientists working in the physical and natural sciences are often interested in learning about the mechanisms or ‘laws’ and processes underlying physical phenomena.
    (Not enough to simply fit standard statistical models to observations.)
- To learn about the mechanisms and processes, scientists build complicated models.
  - Usually numerical solutions of complex mathematical models.
  - Translated into computer code so that they can study simulations of their physical processes under different conditions.

Deterministic models (cont'd)

- In each case, in addition to the model being less than perfect, parameters of the model may be uncertain.
- 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/extrapolations.
- 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: computer model emulation or nonparametric regression or machine learning.



Green inputs/output are the training data.

Red = the input where predictions are desired.

Input and output need not be scalars.

Computer model emulation via Gaussian Processes

- 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 (predictions), for example greater uncertain 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)

Gaussian processes

Modeling with Gaussian processes

- Gaussian processes (GPs) are useful models for dependent processes, e.g. time series, spatial data.
- The use of GPs here is as generally discussed in the statistics for computer models, computer science and engineering literature. Gets listed under some of these labels: machine learning, models for complex computer models, and computer model calibration.

The above two uses are all *very* closely related.

Basic Gaussian process (linear) model

- Process at location $\mathbf{s} \in D$ is $Z(\mathbf{s}) = \mu(\mathbf{s}) + w(\mathbf{s})$. Location may be physical or in "input space".
  - $\mu_{\boldsymbol{\beta}}(\mathbf{s})$ is the mean. Generally keep this simple, though if more is known about the form of the mean, it is useful to add that information. For generality, assume μ is a function of $\boldsymbol{\beta}$.
- Model dependence among spatial random variables by modeling $\{w(\mathbf{s}) : \mathbf{s} \in D\}$ as a Gaussian process (infinite-dimensional).
- 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

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

GP linear model: [cont'd.]

- For this model, once priors for $\Theta, \boldsymbol{\beta}$ specified, inference is based on posterior $\pi(\Theta, \boldsymbol{\beta} \mid \mathbf{Z})$.
- Θ has low dimensions, while dimensions of $\mathbf{Z}$ can be large.
- Relatively easy to construct MCMC algorithm to sample from $\pi(\Theta, \boldsymbol{\beta} \mid \mathbf{Z})$.
- Alternatively, maximum likelihood framework: maximize likelihood with respect to $\Theta, \boldsymbol{\beta}$.
- Note: matrix computations involving $\Sigma(\Theta)$ ($n \times n$) are of order n^3. Expensive for large n. (Many strategies for dealing with this, e.g. in Rasmussen and Williams, 2006).

GP linear model prediction

- Predictions $\mathbf{Z}^* = (Z(\mathbf{s}_1^*), \ldots, Z(\mathbf{s}_m^*))^T$, $\mathbf{s}_1^*, \ldots, \mathbf{s}_m^* \in D$, obtained from the posterior predictive distribution,

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

  Under the GP assumption ($\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma$ depend on $\boldsymbol{\beta}, \Theta$):

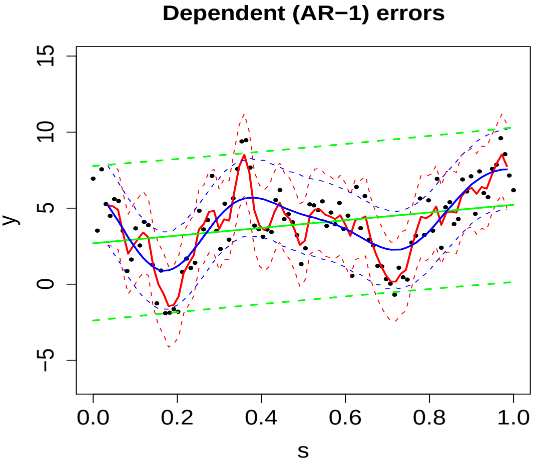
  $$\begin{bmatrix} \mathbf{Z} \\ \mathbf{Z}^* \end{bmatrix} \mid \Theta, \boldsymbol{\beta} \sim N\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right), \qquad (1)$$

- Draws from the posterior predictive distribution are obtained in two steps:
  1. Simulate $\Theta', \boldsymbol{\beta}' \sim \pi(\Theta, \boldsymbol{\beta}|\mathbf{Z})$ by Metropolis-Hastings.
  2. Simulate $\mathbf{Z}^*|\Theta', \boldsymbol{\beta}', \mathbf{Z}$ from conditional multivariate normal density (from (1) and basic normal theory) using $\Theta', \boldsymbol{\beta}'$ above.

GP linear model prediction: ML

- In maximum likelihood framework, same idea as before except fix $\Theta, \boldsymbol{\beta}$ at maximum likelihood estimates (MLEs) and then do prediction based on the MLE (instead of based on the posterior distribution of the parameters.)

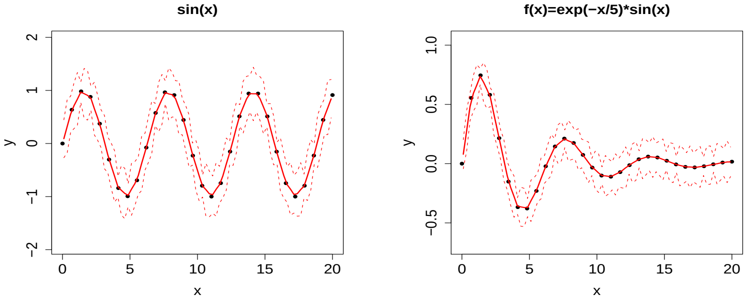
GP model for dependence: toy 1-D example



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

(Red, blue): GP with (exponential, gaussian) covariances.

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.

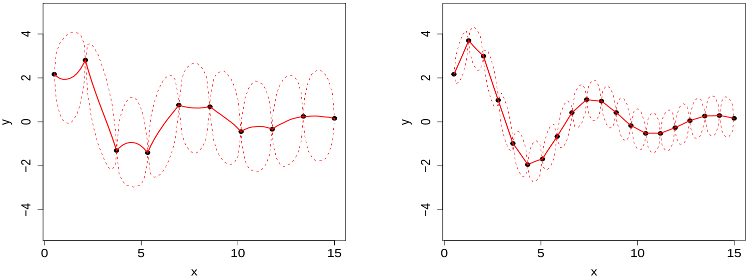
GPs for function approximation

- The usual spatial models discussion of GPs largely focuses on accounting for dependence (first toy example).
- But GPs are a flexible model for functions (second toy example). 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.

GPs for modeling complicated functions

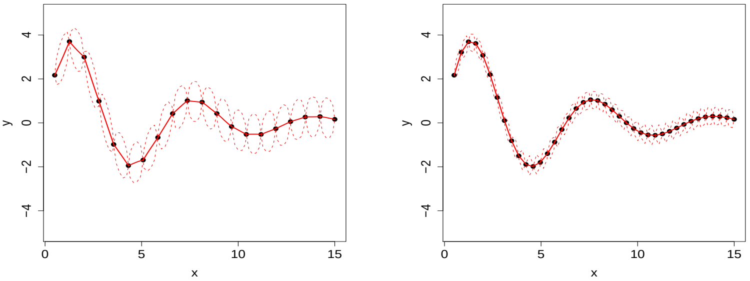
- Consider the following problem: We are interested in modeling the response y as a function of a predictor x so $y = f(x)$.
- We have observations in terms of (response,predictor) or (input, output) pairs: $(x_1, y_1), \ldots, (x_n, y_n)$.
- Based on the observations, called a ‘training set’ in machine 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.

Computer model emulation with Gaussian processes

GP model emulation

- GP model emulation: 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.

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., 2010):

- 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.

Computer model calibration

Computer models and 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 deterministic models? No standard notions of probability and statistical inference apply here.
- 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.

Complex stochastic models and likelihood inference

- Consider the general case that the probability model for Z depends on some parameter θ.
- If the likelihood function for this probability model is explicit, we have $\mathcal{L}(Z \mid \theta)$ and we can perform likelihood-based inference, either finding the maximum likelihood estimator of θ or the posterior distribution for $\theta \mid Z$ after specifying a prior for θ.
- If the (assumed) mechanism/process to simulate Z is provided, but no likelihood corresponding to it is available, the likelihood is *implicit* and hence likelihood-based inference may be challenging. Advantage of this: scientists build models that correspond to their scientific interests and goals.

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?
- 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 'true' θ.
- 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).

Computer model calibration: background

- Non-statistical calibration: search input space for best fit to the data, using a crude measure of fit (e.g. least squares).
- If model runs are very expensive, this is infeasible.
- Does not provide a framework for obtaining probability distributions for θ, which is often of great interest.
- Kennedy and O'Hagan (2001) laid out the basic framework for Bayesian model calibration.
- Series of papers by Bayarri et al., Higdon, Rougier, O'Hagan, Craig, Goldstein and co-authors.

Computer model calibration [cont'd]

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

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

  x: controllable input, θ is unknown input.

- It is important to model $\delta(x, \theta)$ (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 $\delta(\theta)$. 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.

Example: climate science

- Future climate predictions usually made using climate models.
- Climate models can be extremely computationally expensive. Cannot see what happens at all interesting input settings. Input settings=boundary conditions, key model parameters. Parameters may have important physical meaning or may be tuning parameters or representations of unresolved physics,
- If we have observations, we may want to learn about the computer model inputs most ‘compatible’ with reality. e.g. we can compare measurements of temperature values across the world to the climate model output to infer inputs/parameters.

Recap

We have discussed how Gaussian processes can be very useful.

- Can emulate complex computer models and performing inference based on these models, largely due to their flexibility as a prior for functions.
- Lots of open research problems (e.g. multivariate output, dynamic models), applications to interdisciplinary research (most of which generate new methodological problems), and many computational challenges.

Calibration with Multiple Spatial Fields

Calibration with multiple spatial fields

Two stage approach to obtain posterior of $\boldsymbol{\theta}$:

1. Model relationship between $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2)$ and $\boldsymbol{\theta}$ via emulation of model output $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2)$.

   Emulation done via a Gaussian process model.

2. Use observations $\mathbf{Z}$ to infer $\boldsymbol{\theta}$ (parameter of interest).

Calibration with multiple spatial fields

- 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.)

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

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

- $\mathbf{B}(\gamma)$ is a matrix relating $\mathbf{Y}_1$ and $\mathbf{Y}_2$, with parameters γ.
- The covariances of the Gaussian processes depend on both $\mathbf{s}$ (spatial distance) and $\boldsymbol{\theta}$ (distance in parameter space).
- $\boldsymbol{\beta}$s, $\boldsymbol{\xi}$s are regression, covariance parameters.

Very flexible relationship between $\mathbf{Y}_1$ and $\mathbf{Y}_2$.

Calibration with multiple spatial fields [cont'd]

- Emulation: Fit GP via maximum likelihood, then obtain predictive distribution at locations of observations.
- We then model the observations by adding measurement error and a model discrepancy term to the GP emulator:

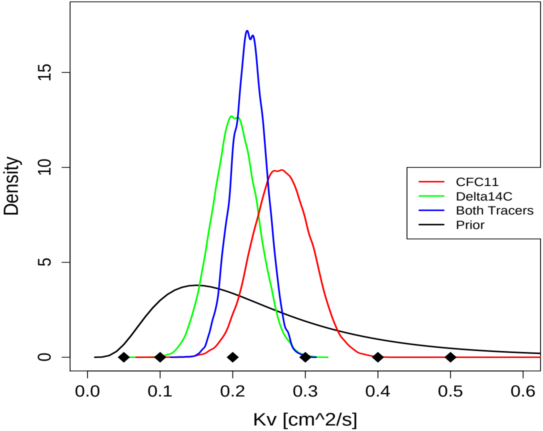
  $$\mathbf{Z} = \boldsymbol{\eta}(\mathbf{Y}, \boldsymbol{\theta}) + \boldsymbol{\delta}(\mathbf{Y}) + \boldsymbol{\epsilon}$$

  where $\boldsymbol{\delta}(\mathbf{Y}) = (\boldsymbol{\delta}_1\ \boldsymbol{\delta}_2)^T$ is the model discrepancy, $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_1\ \boldsymbol{\epsilon}_2)^T$ is the observation error.
- Model discrepancy term can make crucial adjustment to $\boldsymbol{\theta}$ estimates (Bayarri et al. 2007; Bhat et al., 2010).
- Use Markov chain Monte Carlo (MCMC) to estimate $\pi(\boldsymbol{\theta} \mid \mathbf{Z}, \mathbf{Y})$, integrating 'out' remaining parameters.
- Separating stages: 'modularization' (e.g. Liu et al., 2009).

Computational issues

- Matrix computations are $\mathcal{O}(N^3)$, where N is the number of observations. Naive approach: N is in tens of thousands.
- Need long MCMC runs since there may be multimodality issues, and the chain mixes slowly.
- We use 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 step: take advantage of structure of hierarchical model to reduce computations.

Results for K_v inference



posteriors: only CFC-11, only $\Delta^{14}C$, both CFC-11 & $\Delta^{14}C$.

Result: $\mathbf{K_v}$ pdf suggests weakening of MOC in the future.

Summary

1. Our approach is to perform inference in two stages:
   - Obtain a probability model connecting CFC-11, $\Delta^{14}C$ tracer observations to $\mathbf{K_v}$ by fitting a Gaussian process model to climate model runs.
   - Using this probability model, infer a posterior density for $\mathbf{K_v}$ from the observations.
2. We model multivariate spatial data via a flexible hierarchical structure.
3. We use kernel mixing to obtain patterned covariances, making computations tractable for large data sets.

We can use inferred $\mathbf{K_v}$ in the climate model to project the MOC.

Some references

- Kennedy, M.C. and O’Hagan, A.(2001), Bayesian calibration of computer models, *JRSS(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. (2010) “Inferring likelihoods and climate system characteristics using climate models and multiple tracers.”