Calibrating and emulating multivariate computer models

Richard Wilkinson and Nathan Urban (Princeton)

r.d.wilkinson@nottingham.ac.uk
School of Mathematical Sciences
University of Nottingham

Université catholique de Louvain
January 2011

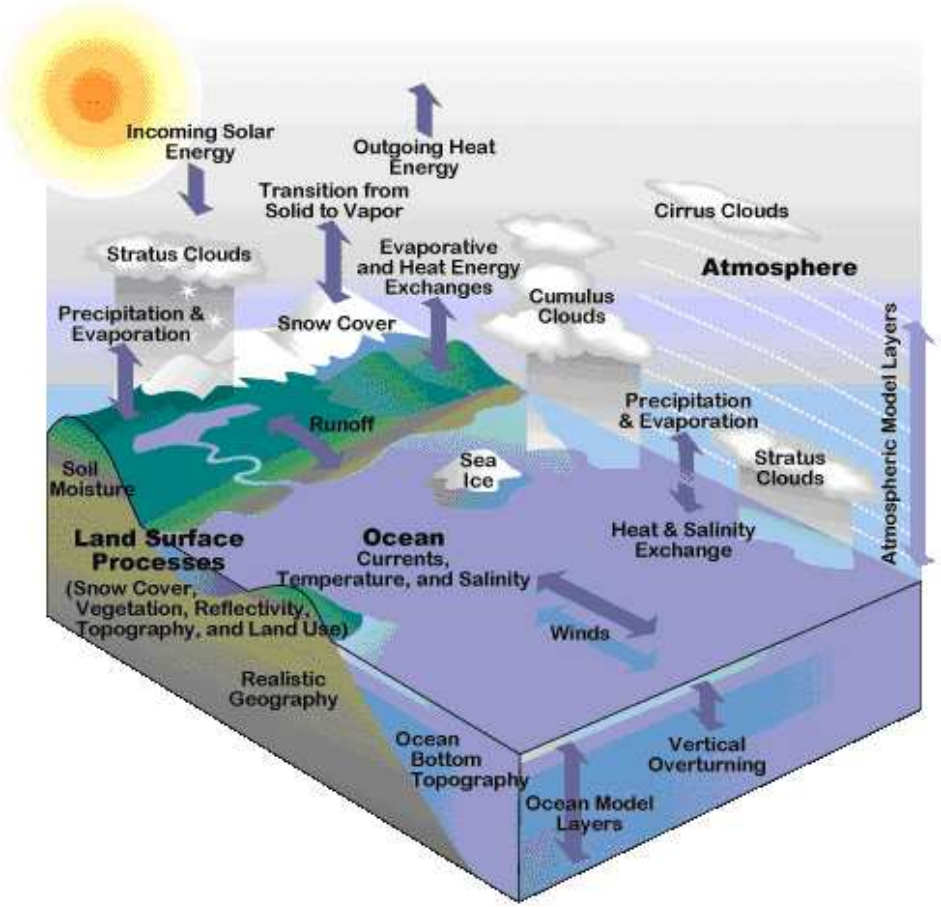
Calibration

The inverse problem

Most models are forwards models, i.e., specify parameters θ and i.c.s and the model $\eta()$ generates output $\mathcal{D}$. Often, we are interested in the inverse-problem, i.e., observe data, want to estimate parameter values.

Different terminology:

- Calibration
- Data assimilation
- Parameter estimation
- Inverse-problem
- Bayesian inference



Representation of uncertainty

Probability (Bayesian) can be used to represent uncertainty.

- Under minimal rationality assumptions, probability can be shown to be the only rational way to represent uncertainty.

Representation of uncertainty

Probability (Bayesian) can be used to represent uncertainty.

- Under minimal rationality assumptions, probability can be shown to be the only rational way to represent uncertainty.
- Probability is subjective probability - distributions represent degrees of belief of individuals. There is no escaping this interpretation in many applications!

Representation of uncertainty

Probability (Bayesian) can be used to represent uncertainty.

- Under minimal rationality assumptions, probability can be shown to be the only rational way to represent uncertainty.
- Probability is subjective probability - distributions represent degrees of belief of individuals. There is no escaping this interpretation in many applications!
- All uncertainty quantities θ can be given distributions $\pi(\theta)$ that represent our (an experts?) uncertainty about the value - this doesn't mean that they are random quantities, just that we don't know their value.
  - Even unknown functions will be described by probability distributions across a class of unknown functions

Representation of uncertainty

Probability (Bayesian) can be used to represent uncertainty.

- Under minimal rationality assumptions, probability can be shown to be the only rational way to represent uncertainty.
- Probability is subjective probability - distributions represent degrees of belief of individuals. There is no escaping this interpretation in many applications!
- All uncertainty quantities θ can be given distributions $\pi(\theta)$ that represent our (an experts?) uncertainty about the value - this doesn't mean that they are random quantities, just that we don't know their value.
  - Even unknown functions will be described by probability distributions across a class of unknown functions
- Bayesian approach uses the principle of conditionality, and always (where possible) conditions on our data.

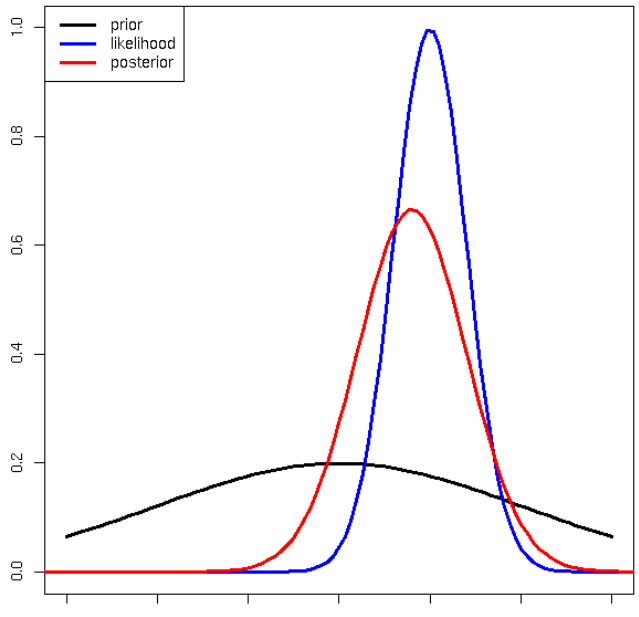
Basic Bayesian approach to calibration

The basic Bayesian approach is to specify

- a prior distribution $\pi(\theta)$ for unknown parameter θ
- a likelihood $\pi(\mathcal{D}|\theta)$

We then aim to find the posterior distribution

$$\pi(\theta \mid \mathcal{D}) = \frac{\pi(\theta)\pi(\mathcal{D} \mid \theta)}{\pi(\mathcal{D})}$$



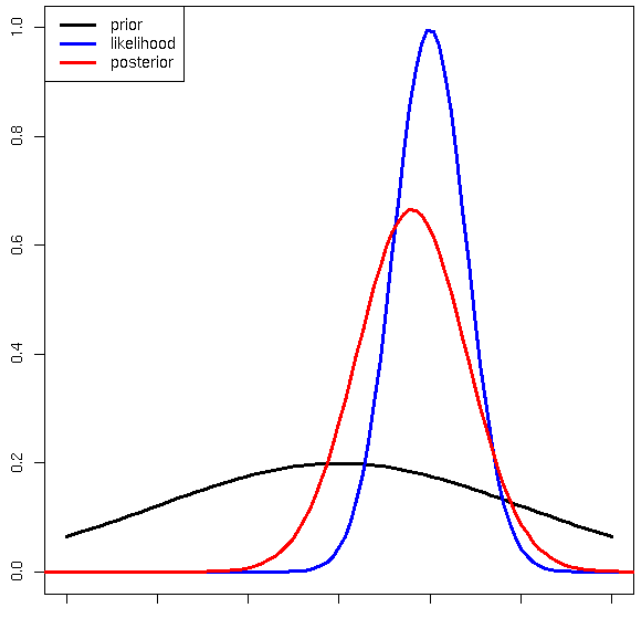
Basic Bayesian approach to calibration

The basic Bayesian approach is to specify

- a prior distribution $\pi(\theta)$ for unknown parameter θ
  - The prior captures our state of knowledge about θ before we see the data or the simulator output.
  - In practice, we're often pragmatic and let $\pi(\theta)$ be a reasonably uninformative distribution over some range of interest.
- a likelihood $\pi(\mathcal{D}|\theta)$

We then aim to find the posterior distribution

$$\pi(\theta \mid \mathcal{D}) = \frac{\pi(\theta)\pi(\mathcal{D} \mid \theta)}{\pi(\mathcal{D})}$$



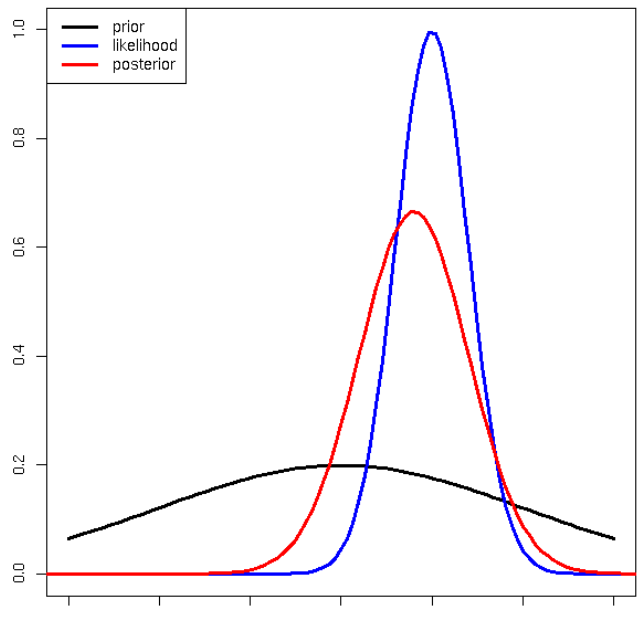
Basic Bayesian approach to calibration

The basic Bayesian approach is to specify

- a prior distribution $\pi(\theta)$ for unknown parameter θ
  - The prior captures our state of knowledge about θ before we see the data or the simulator output.
  - In practice, we're often pragmatic and let $\pi(\theta)$ be a reasonably uninformative distribution over some range of interest.
- a likelihood $\pi(\mathcal{D}|\theta)$
  - The likelihood is a pdf relating the parameter to the observation.
  - This is complex - we must relate the parameter to simulator output, the simulator output to climate, and then climate to the observations. More on this later.

We then aim to find the posterior distribution

$$\pi(\theta \mid \mathcal{D}) = \frac{\pi(\theta)\pi(\mathcal{D} \mid \theta)}{\pi(\mathcal{D})}$$



Basic Bayesian approach to calibration

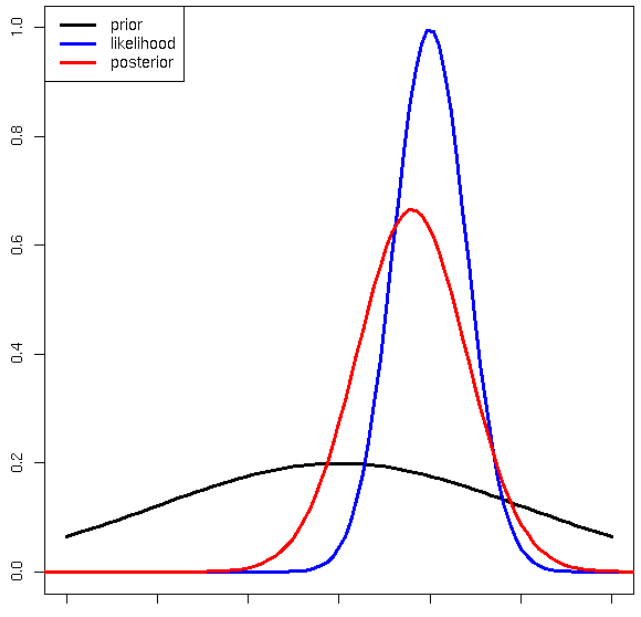
The basic Bayesian approach is to specify

- a prior distribution $\pi(\theta)$ for unknown parameter θ
  - The prior captures our state of knowledge about θ before we see the data or the simulator output.
  - In practice, we're often pragmatic and let $\pi(\theta)$ be a reasonably uninformative distribution over some range of interest.
- a likelihood $\pi(\mathcal{D}|\theta)$
  - The likelihood is a pdf relating the parameter to the observation.
  - This is complex - we must relate the parameter to simulator output, the simulator output to climate, and then climate to the observations. More on this later.

We then aim to find the posterior distribution

$$\pi(\theta \mid \mathcal{D}) = \frac{\pi(\theta)\pi(\mathcal{D} \mid \theta)}{\pi(\mathcal{D})}$$

For all but the simplest problems, this calculation is hard!



Calibrated prediction

Reasons why we may want the posterior $\pi(\theta|\mathcal{D})$:

1. Calibrated prediction
2. Calibrated sensitivity/uncertainty analysis
3. Parameter estimation

Calibrated prediction

Reasons why we may want the posterior $\pi(\theta|\mathcal{D})$:

1. Calibrated prediction
   - Suppose we wish to predict $\mathcal{D}_f$ given $\mathcal{D}_p$, taking account of parametric uncertainty:

$$\pi(\mathcal{D}_f \mid \mathcal{D}_p) = \int \pi(\mathcal{D}_f \mid \theta)\pi(\theta \mid \mathcal{D}_p)\mathrm{d}\theta$$

2. Calibrated sensitivity/uncertainty analysis

3. Parameter estimation

Calibrated prediction

Reasons why we may want the posterior $\pi(\theta|\mathcal{D})$:

1. Calibrated prediction
   - Suppose we wish to predict $\mathcal{D}_f$ given $\mathcal{D}_p$, taking account of parametric uncertainty:

$$\pi(\mathcal{D}_f \mid \mathcal{D}_p) = \int \pi(\mathcal{D}_f \mid \theta)\pi(\theta \mid \mathcal{D}_p)\mathrm{d}\theta$$

2. Calibrated sensitivity/uncertainty analysis
   - Assign uncertainty in our predictions to various uncertainties in the inputs? If we are given \$$X$ to measure/model some aspect of the climate - what would most decrease our uncertainty?
3. Parameter estimation

Calibrated prediction

Reasons why we may want the posterior $\pi(\theta|\mathcal{D})$:

1. Calibrated prediction
   - Suppose we wish to predict $\mathcal{D}_f$ given $\mathcal{D}_p$, taking account of parametric uncertainty:

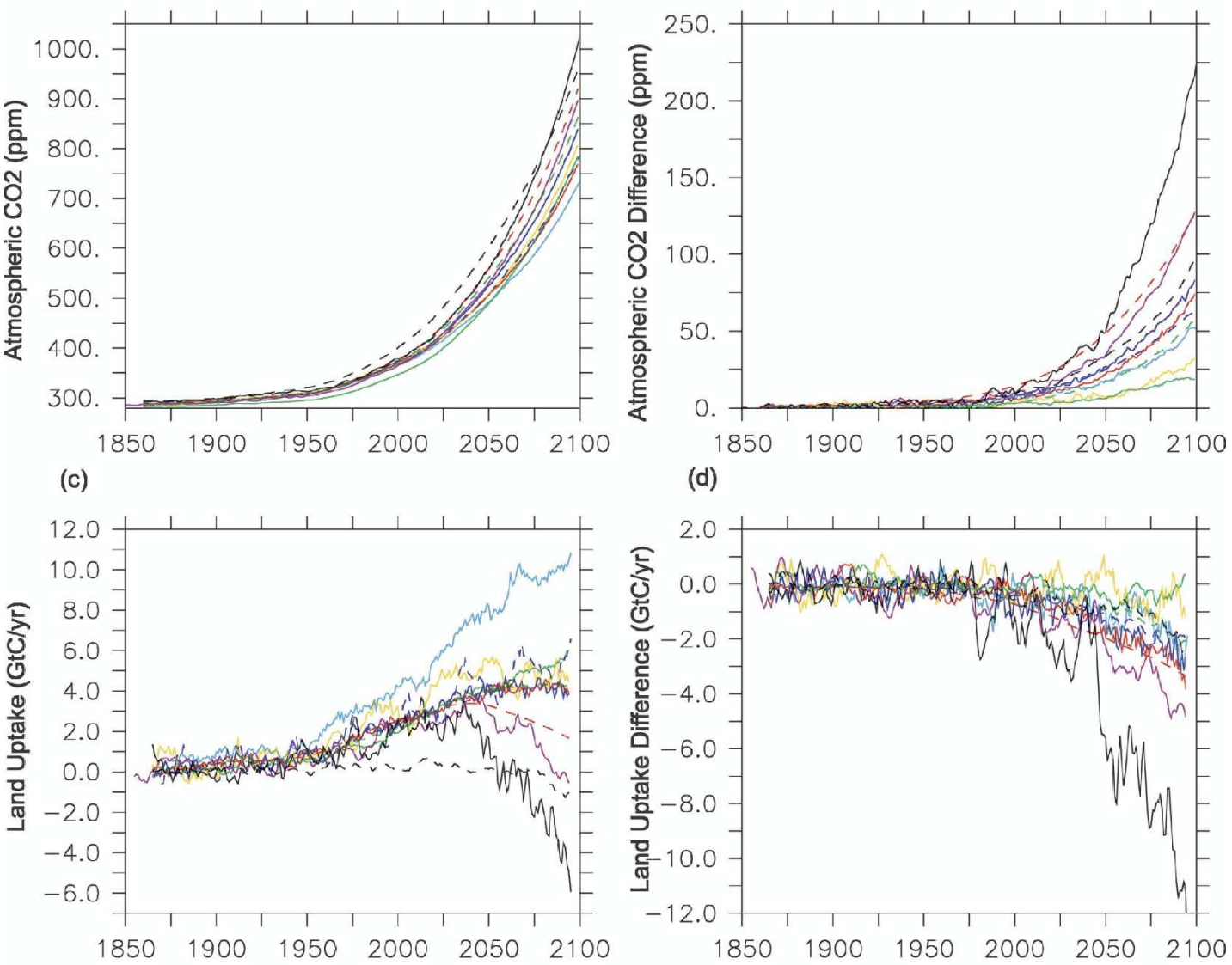
$$\pi(\mathcal{D}_f \mid \mathcal{D}_p) = \int \pi(\mathcal{D}_f \mid \theta)\pi(\theta \mid \mathcal{D}_p)\mathrm{d}\theta$$

2. Calibrated sensitivity/uncertainty analysis
   - Assign uncertainty in our predictions to various uncertainties in the inputs? If we are given \$$X$ to measure/model some aspect of the climate - what would most decrease our uncertainty?
3. Parameter estimation

The amount of care we take on various aspects of the statistical modelling will depend on our aim.

Friedlingstein *et al.* 2006 - 'uncalibrated' GCM carbon cycle predictions

Climate simulators tend to be 'tuned' rather than calibrated, due to their complexity.



Carbon feedbacks

- Terrestrial ecosystems currently absorb a considerable fraction of anthropogenic carbon emissions.
- However, the fate of this sink is highly uncertain due to insufficient knowledge about key feedbacks.
  - We are uncertain about the sensitivity of soil respiration to increasing global temperature.
  - GCM predictions don't agree on the sign of the net terrestrial carbon flux.

The figure shows inter-model spread in uncalibrated GCM model predictions.

- How much additional spread is there from parametric uncertainty? (as opposed to model structural uncertainty?)
- Would calibration reduce the range of the ensemble predictions? Or would it increase our uncertainty?

We can't answer these questions with full GCMs at present, but we can begin to investigate with simplified EMIC models.

UVic Earth System Climate Model

With Nathan Urban, Klaus Keller and group

UVic ESCM is an intermediate complexity model with a general circulation ocean and dynamic/thermodynamic sea-ice components coupled to a simple energy/moisture balance atmosphere. It has a dynamic vegetation and terrestrial carbon cycle model (TRIFFID) as well as an inorganic carbon cycle.

- Inputs: Q_{10} = soil respiration sensitivity to temperature (carbon source) and $K_c = CO_2$ fertilization of photosynthesis (carbon sink).
- Output: time-series of CO_2 values, cumulative carbon flux measurements, spatial-temporal field of soil carbon measurements.

UVic Earth System Climate Model

With Nathan Urban, Klaus Keller and group

UVic ESCM is an intermediate complexity model with a general circulation ocean and dynamic/thermodynamic sea-ice components coupled to a simple energy/moisture balance atmosphere. It has a dynamic vegetation and terrestrial carbon cycle model (TRIFFID) as well as an inorganic carbon cycle.

- Inputs: Q_{10} = soil respiration sensitivity to temperature (carbon source) and $K_c = CO_2$ fertilization of photosynthesis (carbon sink).
- Output: time-series of CO_2 values, cumulative carbon flux measurements, spatial-temporal field of soil carbon measurements.

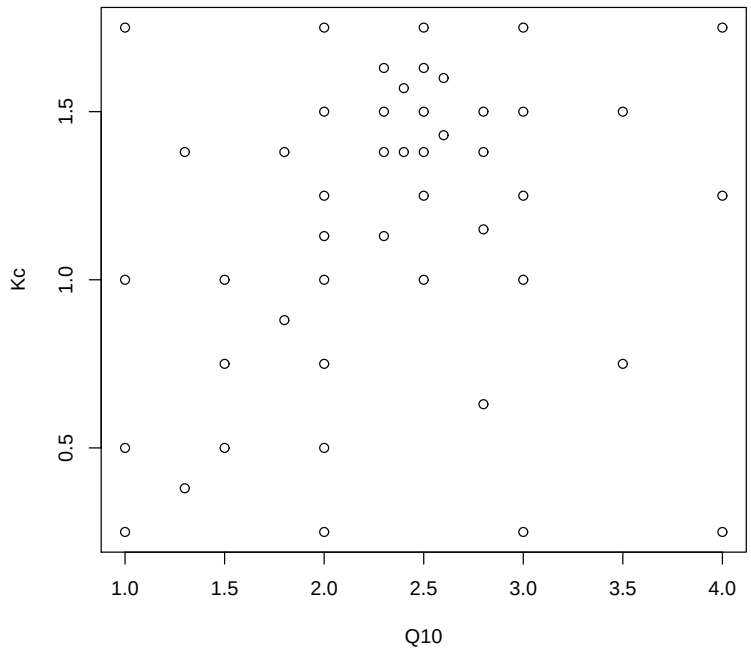
The observational data are limited, and consist of 60 measurements $\mathcal{D}_{field}$:

- 40 instrumental CO_2 measurements from 1960-1999 (from Keeling's Mauna Loa data)
- 17 ice core CO_2 measurements
- 3 cumulative ocean carbon flux measurements

Calibration

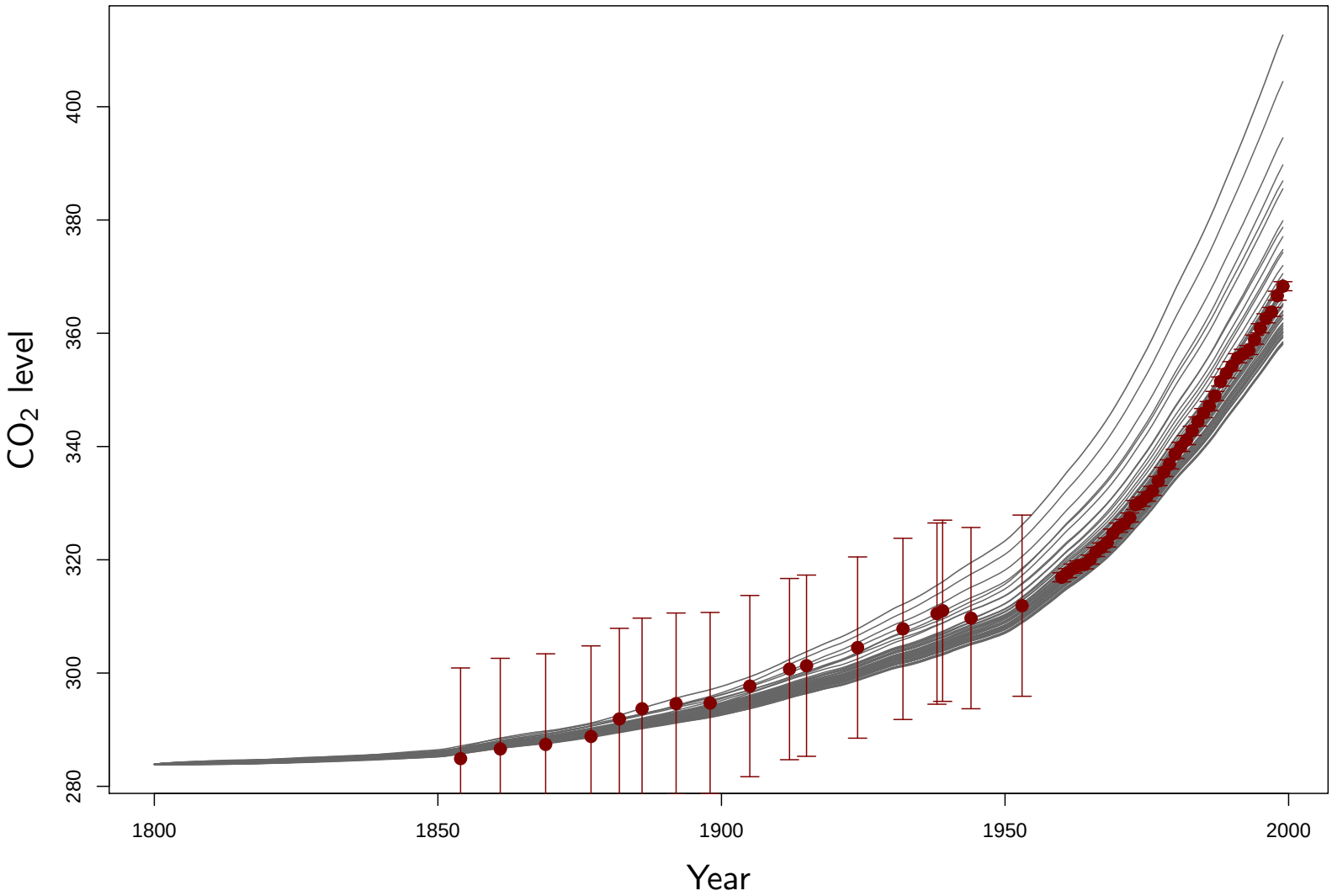
The aim is to combine the physics coded into UVic with the empirical observations to learn about the carbon feedbacks.
However, UVic takes approximately two weeks to run for a single input configuration. Consequently, all inference must be done from a limited ensemble of model runs.

- 48 member ensemble, grid design D, output $\mathcal{D}_{sim}$ $(48 \times n)$.



- this is overkill for this model - benefit of sequential designs

Model runs and data



Approaches to calibration

There are various approaches used to calibrate models:

- Monte Carlo - brute force
- Ad hoc manual tuning
- Component-wise tuning
- Emulation

Only the first and last options can be considered acceptable statistical calibration schemes, and the first option is ruled out if the simulator has a long run time.

The focus here is on emulation.

- See Wilkinson 2010, or Riccuito *et al.* 2009 for full details on this approach
- See Guillas *et al.* 2009 or Sanso *et al.* 2008 for more (challenging and impressive) examples of the emulation of climate simulators.

Code uncertainty

For complex simulators, run times might be long, ruling out brute-force approaches such as Monte Carlo methods.

Code uncertainty

For complex simulators, run times might be long, ruling out brute-force approaches such as Monte Carlo methods.

Consequently, we will only know the simulator output at a finite number of points.

- We call this *code uncertainty*.
- All inference must be done using a finite ensemble of model runs

$$\mathcal{D}_{sim} = \{(\theta_i, \eta(\theta_i))\}_{i=1,\ldots,N}$$

- If θ is not in the ensemble, then we are uncertainty about the value of $\eta(\theta)$.

Code uncertainty

For complex simulators, run times might be long, ruling out brute-force approaches such as Monte Carlo methods.

Consequently, we will only know the simulator output at a finite number of points.

- We call this *code uncertainty*.
- All inference must be done using a finite ensemble of model runs

$$\mathcal{D}_{sim} = \{(\theta_i, \eta(\theta_i))\}_{i=1,\ldots,N}$$

- If θ is not in the ensemble, then we are uncertainty about the value of $\eta(\theta)$.

If θ is multidimensional, then even short run times can rule out brute force approaches

- $\dim(\theta) \in \mathbb{R}^{10}$ then 1000 simulator runs is only enough for one point in each corner of the design space.

The design of computational experiments is an active field in statistics.

Meta-modelling

Idea: If the simulator is expensive, build a cheap model of it and use this in any analysis.

'a model of the model'

We call this meta-model an *emulator* of our simulator.

Meta-modelling

Idea: If the simulator is expensive, build a cheap model of it and use this in any analysis.

'a model of the model'

We call this meta-model an *emulator* of our simulator.

There are many types of emulator.

- ideally an emulator should come with an assessment of its accuracy
- rather than just predicting $\eta(\theta)$ it should predict $\pi(\eta(\theta)|\mathcal{D}_{sim})$ - our uncertainty about the simulator value given the ensemble $\mathcal{D}_{sim}$.

Gaussian process emulators are most popular choice for emulator. Built using

- an ensemble of model runs $\mathcal{D}_{sim} = \{(\theta_i, \eta(\theta_i))\}_{i=1,\ldots,N}$
- expert opinion about the simulator output.

Emulator possibilities

Linear Regression

+ve's:

- v. easy to use and understand

Neural networks

- non-parametric function fitting
- good existing software

Gaussian processes

- non-parametric function fitting
- explicitly incorporates a quantification of uncertainty

Emulator possibilities

Linear Regression

+ve's:

- v. easy to use and understand

-ve's:

- need to specify a parametric form
- error structure is white and ignores information

Neural networks

- non-parametric function fitting
- good existing software
- non-probabilistic - implicit specification of error (if at all)

Gaussian processes

- non-parametric function fitting
- explicitly incorporates a quantification of uncertainty
- can be difficult to implement
- requires careful user-specified inputs to work well

Meta-modelling

Gaussian Process Emulators

Gaussian processes provide a flexible nonparametric distributions for our prior beliefs about the functional form of the simulator:

$$\eta(\cdot) \sim GP(m(\cdot), \sigma^2 c(\cdot, \cdot))$$

where $m(\cdot)$ is the prior mean function, and $c(\cdot, \cdot)$ is the prior covariance function (semi-definite).

Meta-modelling

Gaussian Process Emulators

Gaussian processes provide a flexible nonparametric distributions for our prior beliefs about the functional form of the simulator:

$$\eta(\cdot) \sim GP(m(\cdot), \sigma^2 c(\cdot, \cdot))$$

where $m(\cdot)$ is the prior mean function, and $c(\cdot, \cdot)$ is the prior covariance function (semi-definite).

Definition If $f(\cdot) \sim GP(m(\cdot), c(\cdot, \cdot))$ then for any collection of inputs $x_1, \ldots, x_n$ the vector

$$(f(x_1), \ldots, f(x_n))^T \sim MVN(m(\mathbf{x}), \sigma^2 \mathbf{\Sigma})$$

where $\Sigma_{ij} = c(x_i, x_j)$.

Meta-modelling

Gaussian Process Emulators

Gaussian processes are invariant under Bayesian updating.

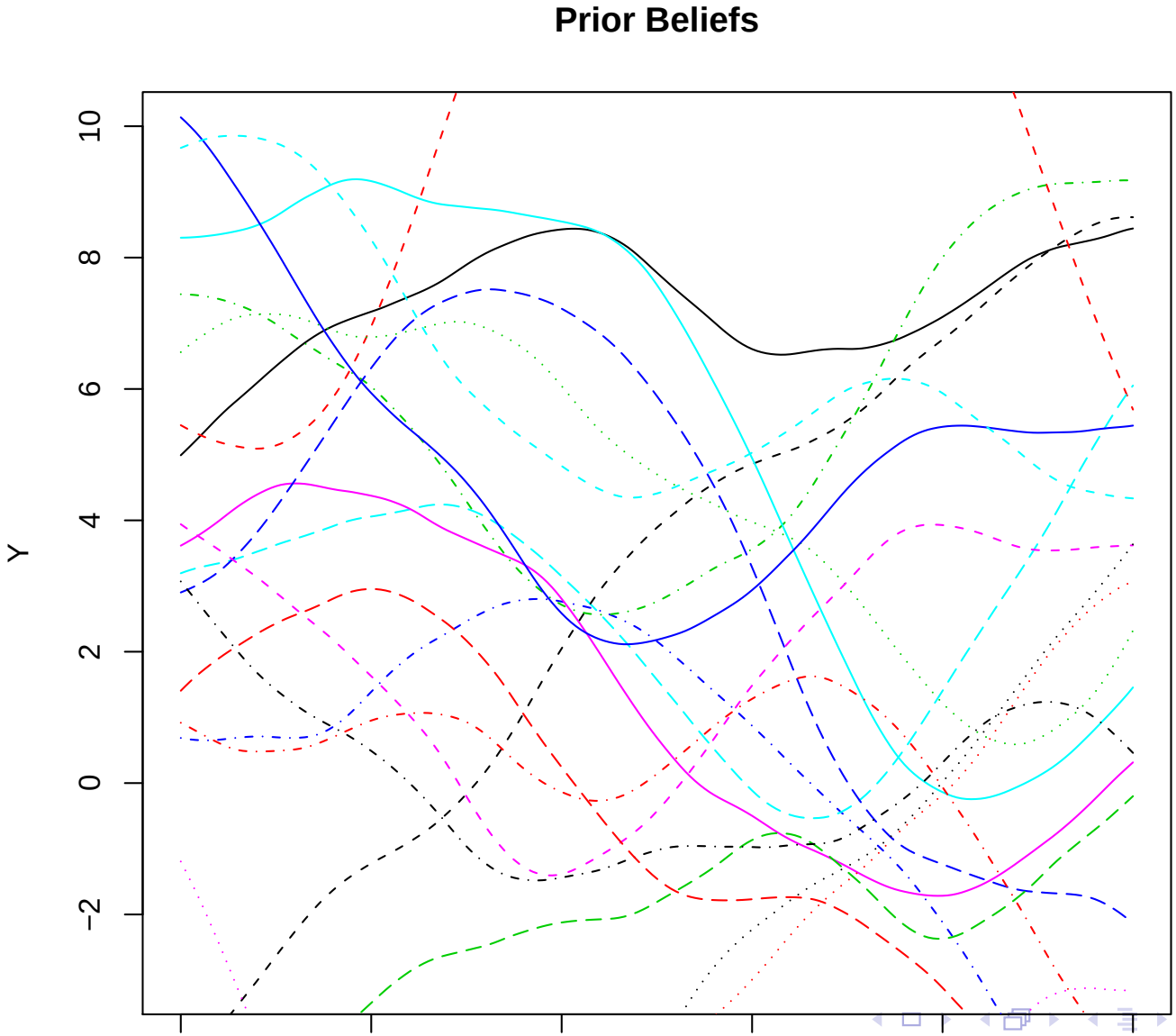
If we observe the ensemble of model runs $\mathcal{D}_{\text{sim}}$, then update our prior belief about η in light of the ensemble of model runs:

$$\eta(\cdot)|\mathcal{D}_{sim} \sim GP(m^*(\cdot), \sigma^2 c^*(\cdot,\cdot))$$

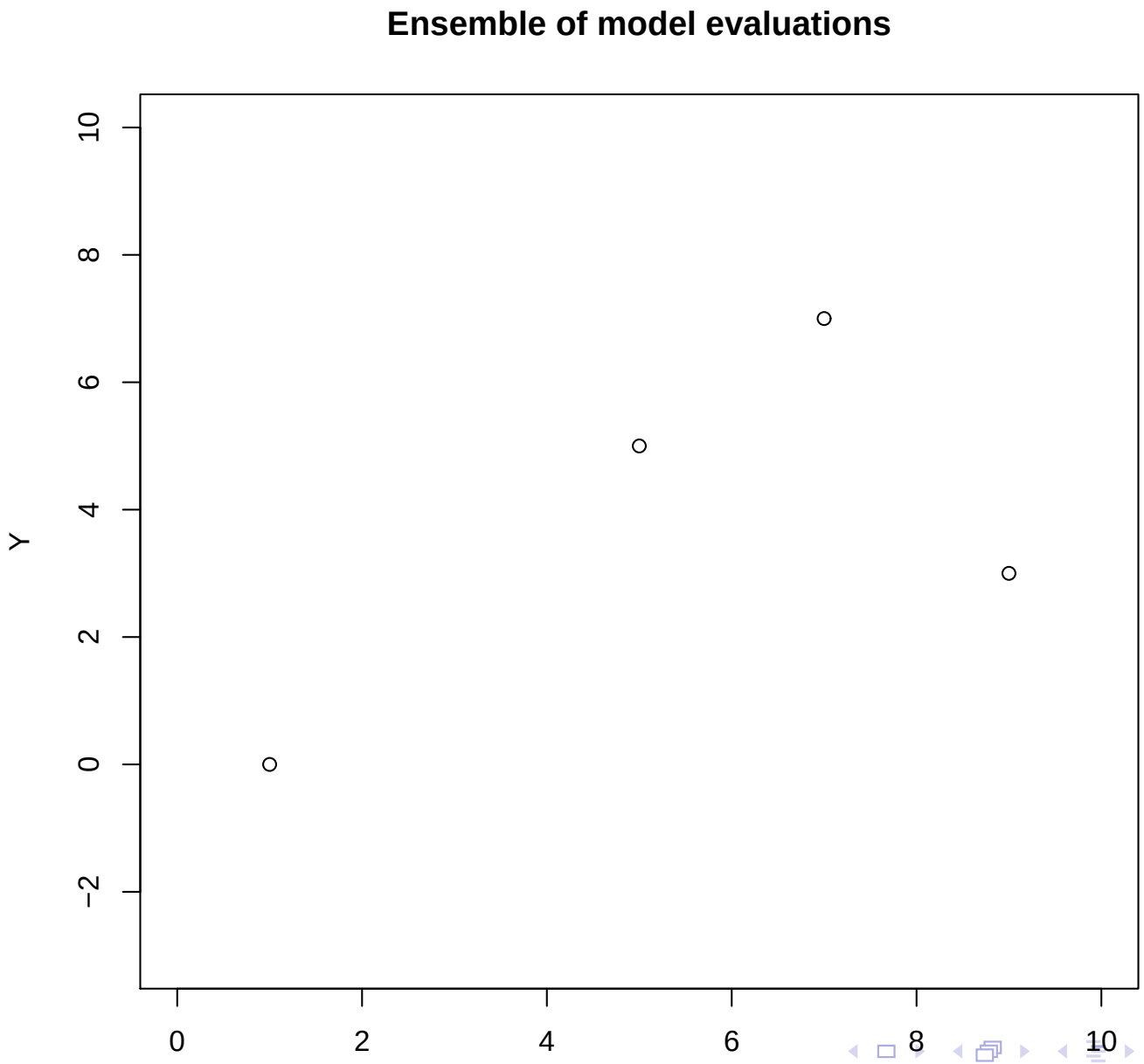
where m^* and c^* are the posterior mean and covariance functions (simple functions of $\mathcal{D}_{\text{sim}}$, m and c).

Gaussian Process Illustration

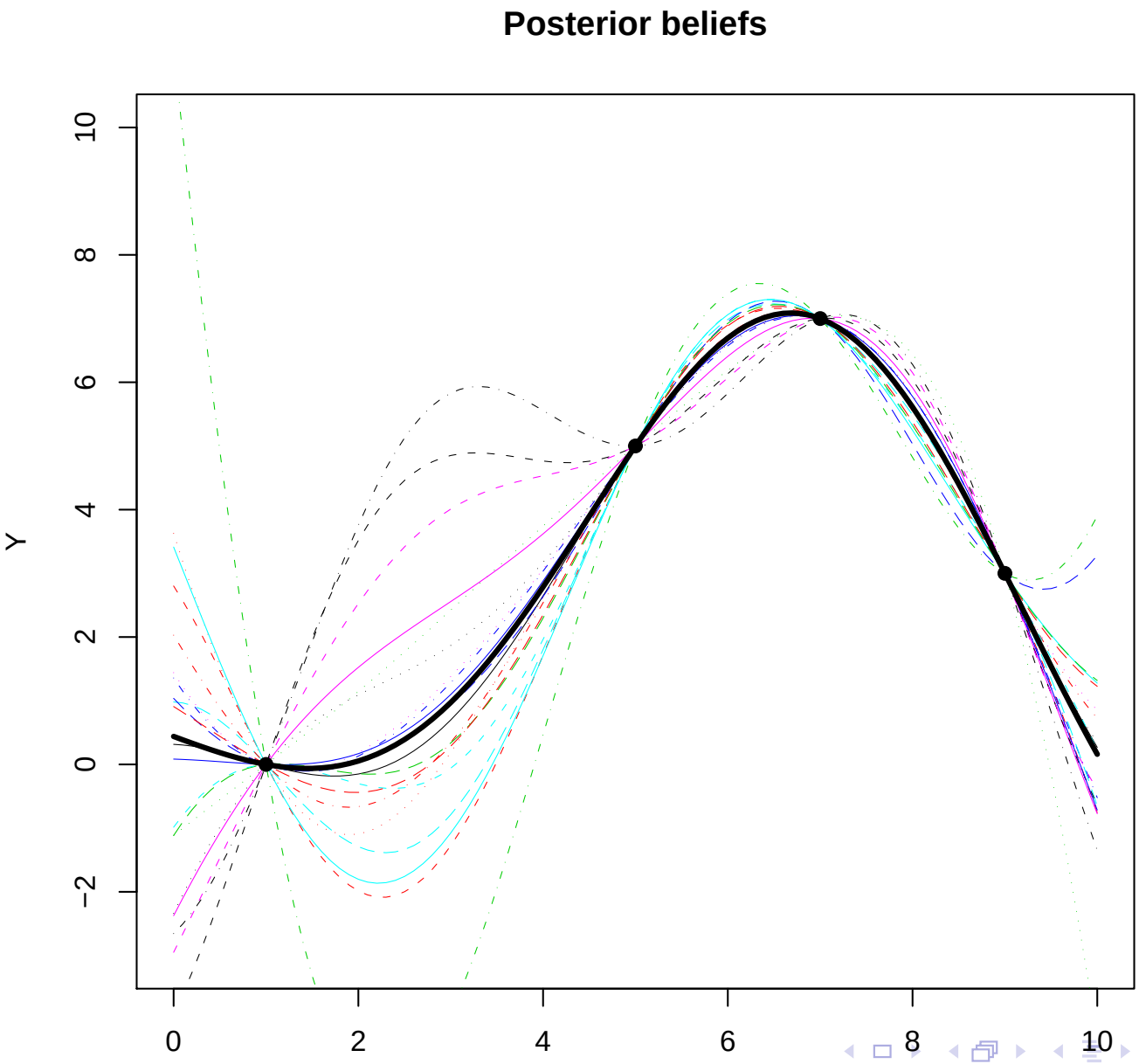
Zero mean



Gaussian Process Illustration



Gaussian Process Illustration



Emulator choices

$$\eta(x) = h(x)\beta + u(x)$$

$$\text{emulator} = \text{mean structure} + \text{residual}$$

Emulator choices

$$\eta(x) = h(x)\beta + u(x)$$

$$\text{emulator} = \text{mean structure} + \text{residual}$$

$u(x)$ can be taken to be a zero-mean Gaussian process

$$u(\cdot) \sim GP(0, c(\cdot, \cdot))$$

Emulator choices

$$\eta(x) = h(x)\beta + u(x)$$
$$\text{emulator} = \text{mean structure} + \text{residual}$$

$u(x)$ can be taken to be a zero-mean Gaussian process

$$u(\cdot) \sim GP(0, c(\cdot, \cdot))$$

Emulator choices:

- mean structure $h(x)$
  - $1, x, x^2, \ldots$, Legendre polynomials?

Emulator choices

$$\eta(x) = h(x)\beta + u(x)$$
$$\text{emulator} = \text{mean structure} + \text{residual}$$

$u(x)$ can be taken to be a zero-mean Gaussian process

$$u(\cdot) \sim GP(0, c(\cdot, \cdot))$$

Emulator choices:

- mean structure $h(x)$
  - $1, x, x^2, \ldots$, Legendre polynomials?
- covariance function $c(\cdot, \cdot)$
  - Stationary? Smooth?
  - Length-scale?

Multivariate Emulation

Higdon *et al.* 2008

How can we deal with multivariate ouput?

- Build independent or separable multivariate emulators,
- Outer product emulators,
- Linear model of coregionalization?

Multivariate Emulation

Higdon *et al.* 2008

How can we deal with multivariate ouput?

- Build independent or separable multivariate emulators,
- Outer product emulators,
- Linear model of coregionalization?

Instead, if the outputs are highly correlated we can reduce the dimension of the data by projecting the data onto some lower dimensional manifold $\mathcal{Y}^{pc}$.

Multivariate Emulation

Higdon *et al.* 2008

How can we deal with multivariate ouput?

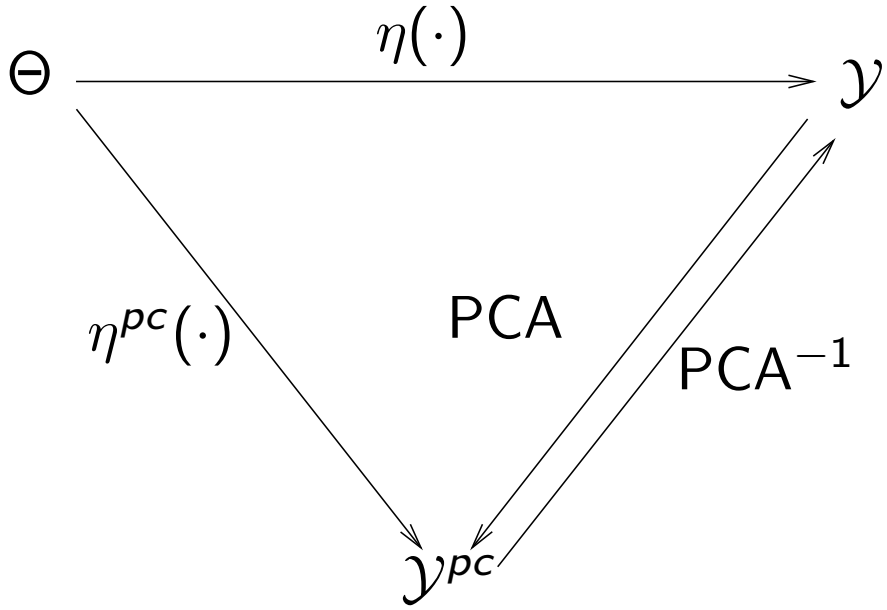
- Build independent or separable multivariate emulators,
- Outer product emulators,
- Linear model of coregionalization?

Instead, if the outputs are highly correlated we can reduce the dimension of the data by projecting the data onto some lower dimensional manifold $\mathcal{Y}^{pc}$.

We can use any dimension reduction technique as long as

- we can reconstruct to the original output space
- we can quantify the reconstruction error.

We can then emulate the function that maps the input space Θ to the reduced dimensional output space $\mathcal{Y}^{pc}$, i.e., $\eta_{pc}(\cdot) : \Theta \rightarrow \mathcal{Y}^{pc}$



It doesn't matter what dimension reduction scheme we use, as long as we can reconstruct from $\mathcal{Y}^{pc}$ and quantify the error in the reconstruction.

Principal Component Emulation (EOF)

We use principal component analysis to project the data onto a lower dimensional manifold, as it is the optimal linear projection (in terms of minimizing reconstruction error).

1. Centre and scale $\mathcal{D}_{\text{sim}}$ so that each column has mean 0 and variance 1. Scaling the columns makes specification of prior distributions for the emulators simpler.
2. Find the singular value decomposition of $\mathcal{D}_{\text{sim}}$.

$$\mathcal{D}_{\text{sim}} = U\Gamma V^*.$$

   Γ contains the singular values (eigenvalues), and V the principal components (eigenvectors).
3. Decide on the dimension of the principal subspace, n^* say, and throw away all but the n^* leading principal components. An orthonormal basis for the principal subspace is given by the first n^* columns of V, denoted V_1. Let V_2 be the matrix of discarded columns.
4. Project $\mathcal{D}_{\text{sim}}$ onto the principal subspace to find $\mathcal{D}_{\text{sim}}^{pc} = \mathcal{D}_{\text{sim}} V_1$

PCA emulation

We then emulate the reduced dimension model

$$\eta_{pc}(\cdot) = (\eta_{pc}^1(\cdot), \ldots, \eta_{pc}^{n^*}(\cdot)).$$

- Each component η_{pc}^i will be uncorrelated (in the ensemble) but not necessarily independent. We use independent Gaussian processes for each component, which seems to be an adequate approximation in all the examples examined.
- The output can be reconstructed from the principal component space to the original full space, accounting for reconstruction error, by a simple matrix multiplication and modelling the discarded components as Gaussian noise with variance equal to the corresponding eigenvalue:

$$\eta(\theta) = V_1 \eta_{pc}(\theta) + V_2 \text{diag}(\Lambda)$$

where $\Lambda_i \sim N(0, \Gamma_{ii})$ ($\Gamma_{ii} = i^{th}$ eigenvalue).

Comments

- This approach (PCA emulation) requires that the outputs are highly correlated.
- We are assuming that the output $\mathcal{D}_{\text{sim}}$ is really a linear combination of a smaller number of variables,

$$\eta(\theta) = \mathbf{v}_1 \eta_{pc}^1(\theta) + \ldots + \mathbf{v}_{n^*} \eta_{pc}^{n^*}(\theta)$$

  which may be a reasonable assumption in many situations, eg, temporal spatial fields.
- Although PCA is a linear method, the method can be used on highly non-linear models as we are still using non-linear Gaussian processes to map from Θ to $\mathcal{Y}^{pc}$ – the linear assumption applied only to the dimension reduction.
- This method accounts for code uncertainty and automatically accounts for the reconstruction error caused by reducing the dimension of the data.

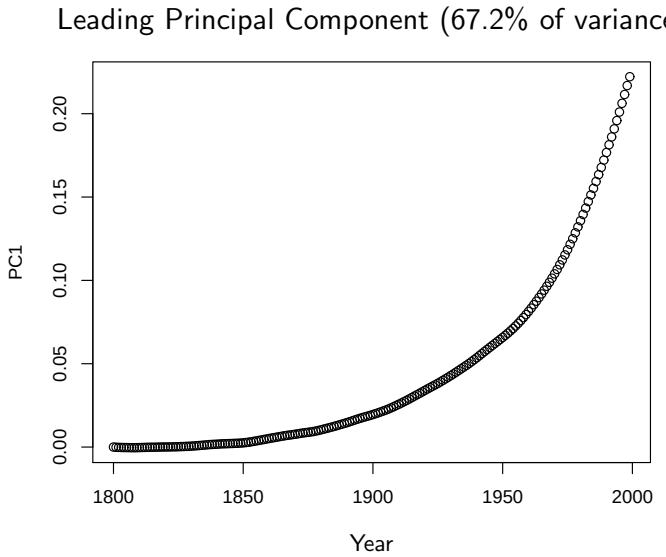
Comments

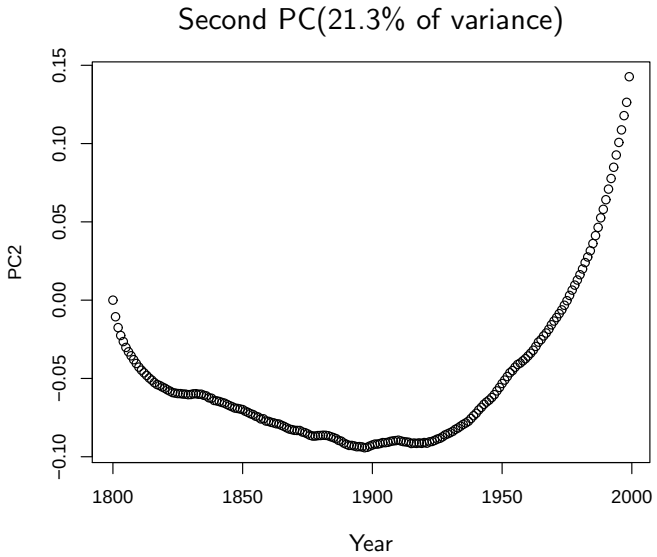
- A concern expressed about PC emulators is that we train the principal components using the simulator. If we then project the real observations on to these principal components, we are making the strong assumption that these PCs are informative about the true system, which may not be true.

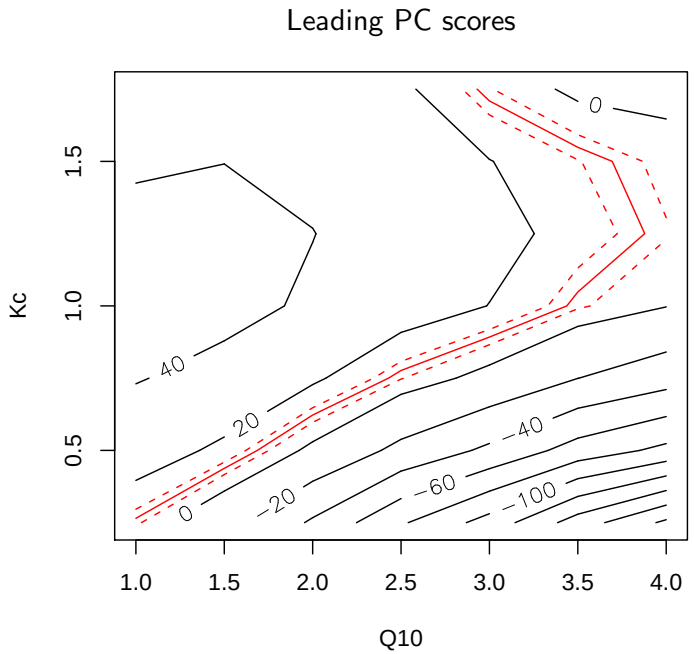
  In practice, this problem is avoided in two ways

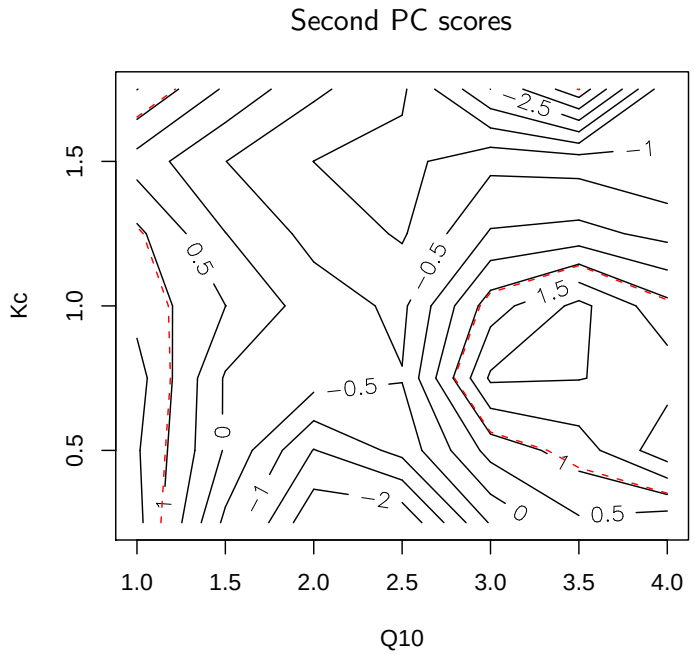
  1. reconstructing to the original space and using lots of diagnostic checks that the PC emulator works well in the original space
  2. checking that the principal components found from the simulator are physically meaningful.

PC Plots


Leading Principal Component (67.2% of variance)
PC1
0.00
0.05
0.10
0.15
0.20
1800
1850
1900
1950
2000
Year


Second PC(21.3% of variance)
PC2
−0.10
−0.05
0.00
0.05
0.10
0.15
1800
1850
1900
1950
2000
Year


Leading PC scores
Kc
0.5
1.0
1.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Q10
0
40
20
−20
−40
−60
−100


Second PC scores
Kc
0.5
1.0
1.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Q10
−2.5
−1
0.5
−0.5
1.5
0
1

GP Choices

Choice of regressors

- We use products of Legendre polynomials on $[-1, 1]$ (Rougier 2007) - an orthonormal basis. We allow up to quadratic terms

There is debate about how much effort to put into m and c.

GP Choices

Choice of regressors

- We use products of Legendre polynomials on $[-1, 1]$ (Rougier 2007) - an orthonormal basis. We allow up to quadratic terms

Covariance function

- Matern with $\nu = 5/2 \Rightarrow$ twice differentiable output

$$c_{5/2}(r) = \tau \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}r}{l}\right)$$

- Give τ a $\Gamma(1.5, 6)$ prior distribution in all the principal component emulators - benefit of scaling first!

There is debate about how much effort to put into m and c.

GP Choices

Choice of regressors

- We use products of Legendre polynomials on $[-1, 1]$ (Rougier 2007) - an orthonormal basis. We allow up to quadratic terms

Covariance function

- Matern with $\nu = 5/2 \Rightarrow$ twice differentiable output

$$c_{5/2}(r) = \tau \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}r}{l}\right)$$

- Give τ a $\Gamma(1.5, 6)$ prior distribution in all the principal component emulators - benefit of scaling first!

Length scales l

- Estimate and fix the length scales using their maximum likelihood estimates.

There is debate about how much effort to put into m and c.

GP Choices

Choice of regressors

- We use products of Legendre polynomials on $[-1, 1]$ (Rougier 2007) - an orthonormal basis. We allow up to quadratic terms

Covariance function

- Matern with $\nu = 5/2 \Rightarrow$ twice differentiable output

$$c_{5/2}(r) = \tau \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}r}{l}\right)$$

- Give τ a $\Gamma(1.5, 6)$ prior distribution in all the principal component emulators - benefit of scaling first!

Length scales l

- Estimate and fix the length scales using their maximum likelihood estimates.
- τ and l are often not both identifiable. Instead, fix l using Addler's theorem by considering the expected number of up-crossings by the residual.

There is debate about how much effort to put into m and c.

Diagnostics

Diagnostic checks are vital if we are to trust the use of the emulator in place of the simulator.

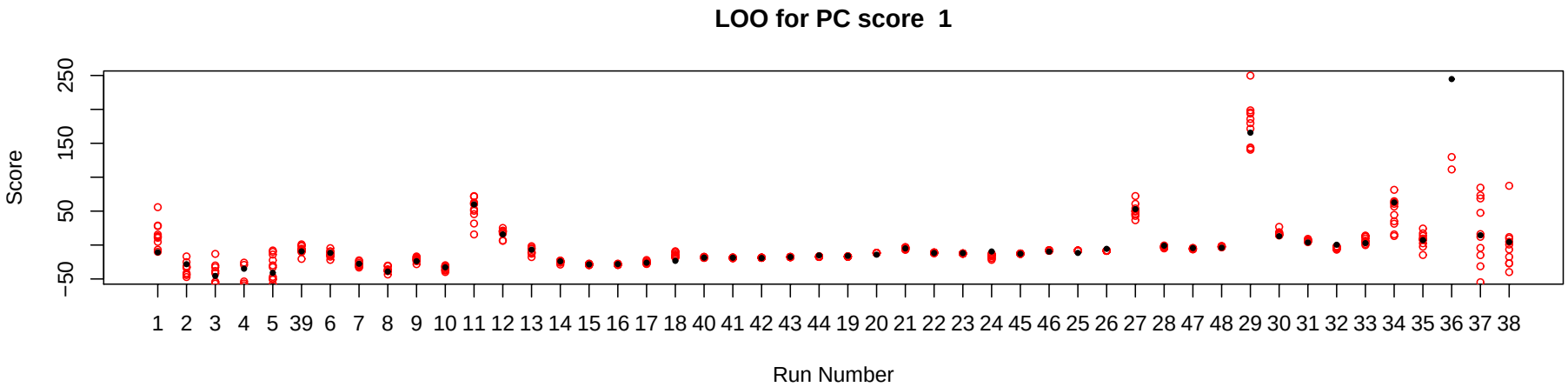
For the PC emulator, we ultimately want to predict the spatial field - so most diagnostic effort should be spent on the reconstructed emulator.

Looking only at the percentage of variance explained by the principal components can be misleading, even if the emulators are perfect, as we can find that PCs that have small eigenvalues (so explain a small amount of variance) can play an important role in prediction.

Leave-one-out (LOA) plots for PC1

Leave-one-out plots are a type of cross-validation to asses whether the final emulator is working well both in terms of the mean prediction, and the uncertainty estimates.

We leave each ensemble member, we leave it out of the training set and build a new emulator. We then predict the left-out ensemble member using the emulator

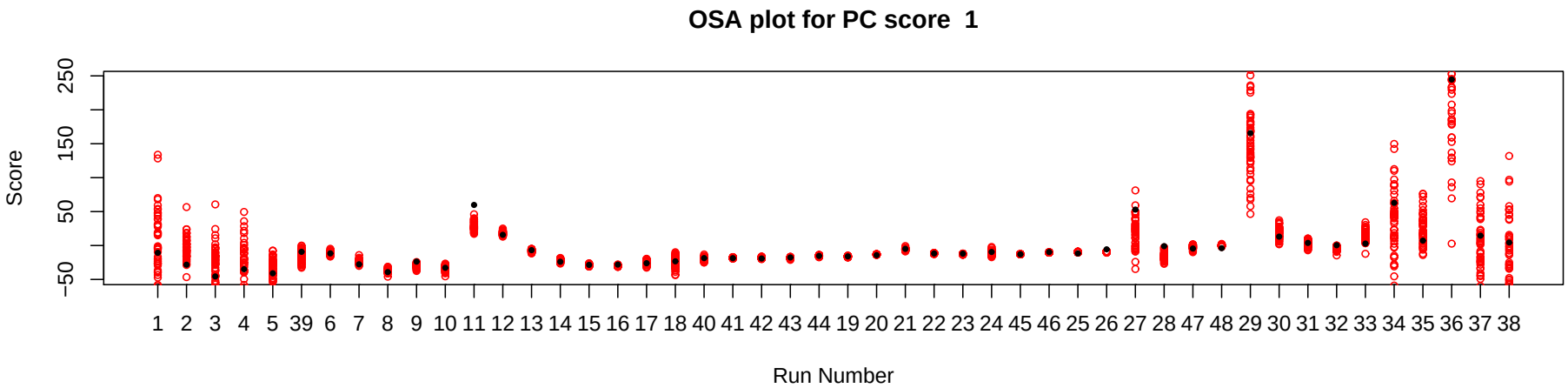


We would like accurate coverage.

One-step-ahead (OSA) plots for PC1

One-step-ahead diagnostics are created by first ordering the ensemble according to one of the input variables, in this case θ_1. We then train an emulator using only the first $n-1$ ensemble members, before predicting the nth ensemble member.

One-step-ahead diagnostics primarily test whether the uncertainty estimates of the emulator are accurate. Because the size of the ensemble grows, we can check more easily whether the length-scale and covariance structure of the emulator are satisfactory.



Other diagnostic tools

Bastos and O'Hagan 2010 suggested a range of statistical tests to diagnose problems with GP emulators.

These can be generalised further by the consideration of scoring rules (see Gneiting and Raftery 2008 for a recent survey). These can be used to test both the *calibration* and *sharpness* of your emulator.

In particular, the continuously-ranked-probability-score (CRPS) looks like a useful measure of emulator performance.

Calibration Framework

Kennedy and O'Hagan 2001

We have two sources of information:

- Computer model $\eta(t, \theta)$
  - with a limited ensemble of model runs $\mathcal{D}_{sim} = \{\eta(t_i, \theta_i), i = \ldots\}$.
- Field data $\mathcal{D}_{\text{field}}$: a collection of noisy measurements of reality at a variety of t values.

Calibration Framework

Kennedy and O'Hagan 2001

We have two sources of information:

- Computer model $\eta(t, \theta)$
  - with a limited ensemble of model runs $\mathcal{D}_{sim} = \{\eta(t_i, \theta_i), i = \ldots\}$.
- Field data $\mathcal{D}_{\text{field}}$: a collection of noisy measurements of reality at a variety of t values.

Many assimilation approaches assume that measurements represent the *computer model* run at its best input value plus independent random noise. If the model is wrong, this is assumption is false. At best, we observe *reality* plus independent random noise.

Calibration Framework

Kennedy and O'Hagan 2001

We have two sources of information:

- Computer model $\eta(t, \theta)$
  - with a limited ensemble of model runs $\mathcal{D}_{sim} = \{\eta(t_i, \theta_i), i = \ldots\}$.
- Field data $\mathcal{D}_{\text{field}}$: a collection of noisy measurements of reality at a variety of t values.

Many assimilation approaches assume that measurements represent the *computer model* run at its best input value plus independent random noise. If the model is wrong, this is assumption is false. At best, we observe *reality* plus independent random noise.
Instead, include an additional model error term.

- Measurement error ϵ
- Model discrepancy $\delta(t)$

Calibration Framework

Assume that reality $\zeta(t)$ is the computer model run at the 'true' value of the parameter $\hat{\theta}$ plus model error:

$$\zeta(t) = \eta(t, \hat{\theta}) + \delta(t)$$

Calibration Framework

Assume that reality $\zeta(t)$ is the computer model run at the 'true' value of the parameter $\hat{\theta}$ plus model error:

$$\zeta(t) = \eta(t, \hat{\theta}) + \delta(t)$$

We observe reality plus noise:

$$\mathcal{D}_{field}(t) = \zeta(t) + \epsilon(t)$$

so that

$$\mathcal{D}_{field}(t) = \eta(t, \hat{\theta}) + \delta(t) + \epsilon(t).$$

Calibration Framework

Assume that reality $\zeta(t)$ is the computer model run at the 'true' value of the parameter $\hat{\theta}$ plus model error:

$$\zeta(t) = \eta(t, \hat{\theta}) + \delta(t)$$

We observe reality plus noise:

$$\mathcal{D}_{field}(t) = \zeta(t) + \epsilon(t)$$

so that

$$\mathcal{D}_{field}(t) = \eta(t, \hat{\theta}) + \delta(t) + \epsilon(t).$$

We then aim to find $\pi(\hat{\theta}|\mathcal{D}_{sim}, \mathcal{D}_{field})$.

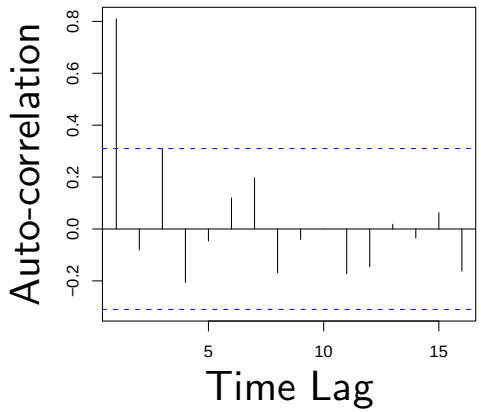
Model Discrepancy

The calibration framework used is:

$$\mathcal{D}_{field}(t) = \eta(\theta, t) + \delta(t) + \epsilon(t)$$

The model predicts the underlying trend, but real climate fluctuates around this. We model

- discrepancy as an AR1 process: $\delta(0) \sim N(0, \sigma_\delta^2)$, and $\delta(t) = \rho\delta(t-1) + N(0, \sigma_\delta^2)$.
- Measurement error as heteroscedastic independent random noise $\epsilon(t) \sim N(0, \lambda(t))$.



How should we better model this discrepancy?

MCMC

Metropolis-within-Gibbs Sampler

Prior distributions: $\rho \sim \Gamma(5, 1)$, $\sigma_\delta^2 \sim \Gamma(4, 0.6)$, $\sigma^2 \sim \Gamma(1.5, 6)$, $\theta = (Q_{10}, K_c)$, $Q_{10} \sim U[1, 4]$, $K_c \sim U[0.25, 1.75]$.

MCMC

Metropolis-within-Gibbs Sampler

Prior distributions: $\rho \sim \Gamma(5,1)$, $\sigma_\delta^2 \sim \Gamma(4,0.6)$, $\sigma^2 \sim \Gamma(1.5,6)$, $\theta = (Q_{10}, K_c)$, $Q_{10} \sim U[1,4]$, $K_c \sim U[0.25,1.75]$.
We can then use a Metropolis-within-Gibbs sampler to find the posterior distribution

$$\pi(\theta, \sigma^2, \rho, \sigma_\delta^2 | \mathcal{D}_{\text{field}}, \mathcal{D}_{\text{sim}})$$

using the following steps

$$\pi(\sigma^2 | \theta, \rho, \sigma_\delta^2, \mathcal{D}_{\text{sim}}, \mathcal{D}_{\text{field}}) \text{ - Gibbs update}$$
$$\pi(\theta | \sigma^2, \rho, \sigma_\delta^2, \mathcal{D}_{\text{sim}}, \mathcal{D}_{\text{field}}) \text{ - MH step}$$
$$\pi(\rho, \sigma_\delta^2 | \theta, \sigma^2, \mathcal{D}_{\text{sim}}, \mathcal{D}_{\text{field}}) \text{ - MH step}$$

Reparameterizing in terms of $\log(\rho)$ and using a block update for ρ and σ_δ^2 helps with the convergence.

A note on calibrating emulators

When calibrating emulators, we should explicitly use the likelihood function of the emulator, rather than simulating from it. In other words, if we have that

$$D = \eta(\theta) + e$$

where $e \sim N(0, \Sigma)$ then we should use the integrated likelihood (the convolution of $\eta(\theta)$ and e), namely

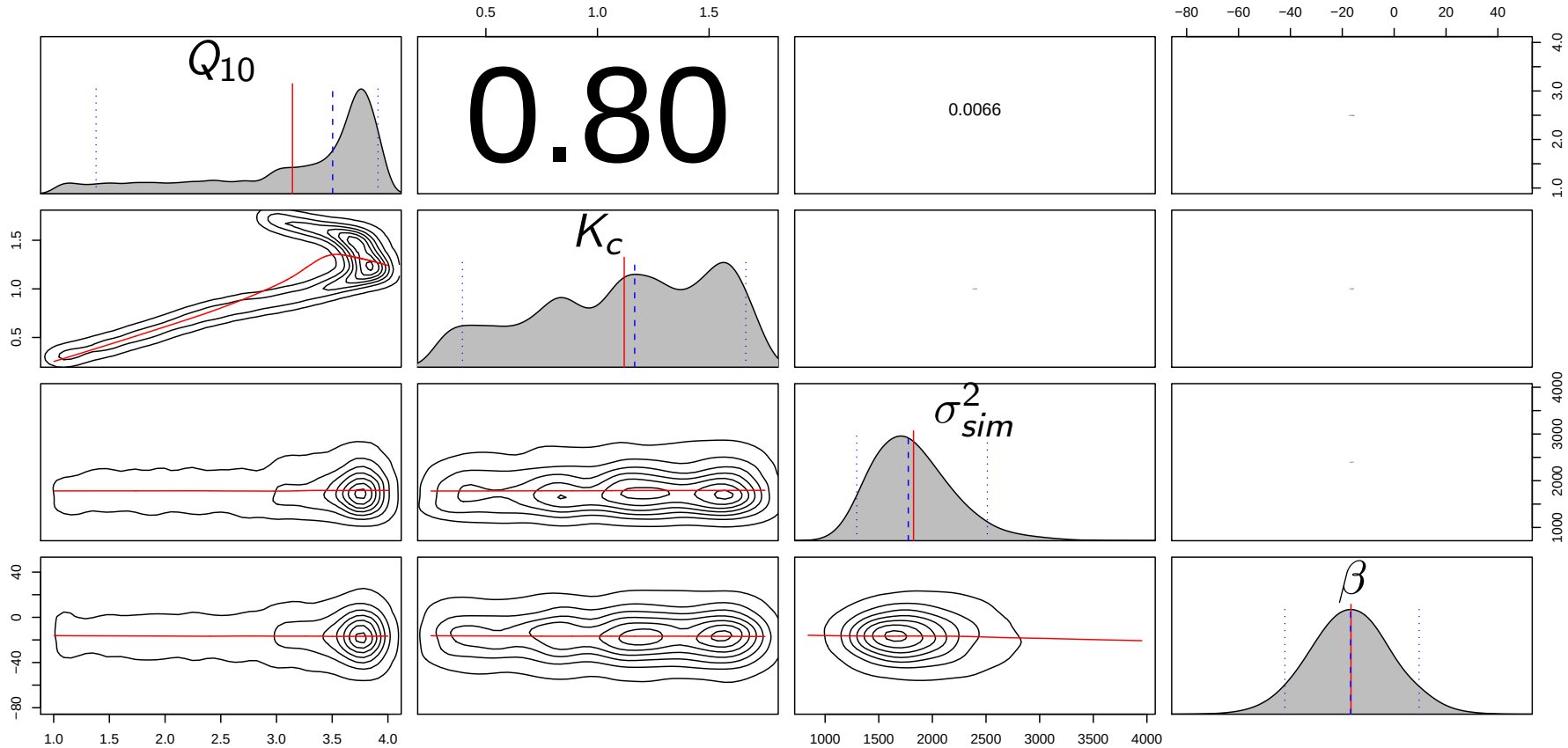
$$D|\theta \sim N(m^*(\theta), \Sigma + \sigma^2 c^*(\theta, \theta))$$

in any MCMC algorithm. The alternative is for each θ, first simulating a value $\eta(\theta)$ from the emulator, and then using likelihood

$$D|\eta(\theta) \sim N(\eta(\theta), \Sigma)$$

Both methods will work, but the former will be more efficient than the latter. Of course, if the distribution of e is not Gaussian then it unlikely that you will be able to calculate the convolution and so we will be forced to draw from the emulator.

Results



Conclusions

- For highly correlated multivariate output, principal component emulation can work well and is computationally cheap and easy to implement.
- A large number of output dimensions can be reduced to a smaller number of principal component scores which can then be emulated, accounting for any error in the compression.
- Given the model, forcing data, constraints and uniform priors, we cannot precisely constrain the two parameters K_c and Q_{10} - there is a ridge of values all of which are well supported by the data.
- Acceptable parameter combinations produce similar responses of the carbon cycle during the years 1800-1999 but produce divergent future predictions!