Bayes Linear Emulation of Computer Models

Ian Vernon*
Department of Mathematical Sciences
Durham University.

*work done in collaboration with Michael Goldstein (Dept. Mathematical Sciences); Richard Bower and Carlos Frenk’s group at the Institute for Computational Cosmology, Durham University. Funding: MUCM and EPSRC.
Overview

- Bayes Linear Methods.
Overview

- Bayes Linear Methods.
- The Bayes Linear Update.
Overview

- Bayes Linear Methods.
- The Bayes Linear Update.
- Bayes Linear Approach to Emulation.
Overview

• Bayes Linear Methods.
• The Bayes Linear Update.
• Bayes Linear Approach to Emulation.
• Galform: a complex Computer Model of Galaxy Formation.
Overview

- Bayes Linear Methods.
- The Bayes Linear Update.
- Bayes Linear Approach to Emulation.
- Searching for ‘acceptable’ inputs...
Overview

- Bayes Linear Methods.
- The Bayes Linear Update.
- Bayes Linear Approach to Emulation.
- Searching for ‘acceptable’ inputs...
- Bayes Linear Implausibility Measures.
Overview

- Bayes Linear Methods.
- The Bayes Linear Update.
- Bayes Linear Approach to Emulation.
- Searching for ‘acceptable’ inputs...
- Bayes Linear Implausibility Measures.
- History Matching: Learning about ‘acceptable’ inputs via Implausibility.
Bayes Linear Methods.

- An alternative to fully probabilistic Bayesian Statistics.
Bayes Linear Methods.

- An alternative to fully probabilistic Bayesian Statistics.
- Has the same Bayesian form: we update our prior beliefs in the light of new data.
Bayes Linear Methods.

- An alternative to fully probabilistic Bayesian Statistics.
- Has the same Bayesian form: we update our prior beliefs in the light of new data.
- Instead of dealing with probabilities, we deal with expectations and variances: $\mathbb{E}(X)$ and $\text{Var}(X)$. 
Bayes Linear Methods.

- An alternative to fully probabilistic Bayesian Statistics.
- Has the same Bayesian form: we update our prior beliefs in the light of new data.
- Instead of dealing with probabilities, we deal with expectations and variances: $E(X)$ and $\text{Var}(X)$.
- In fact, following de Finetti we treat expectation as the primitive quantity in the analysis.
Bayes Linear Methods.

- An alternative to fully probabilistic Bayesian Statistics.
- Has the same Bayesian form: we update our prior beliefs in the light of new data.
- Instead of dealing with probabilities, we deal with expectations and variances: $\mathbb{E}(X)$ and $\text{Var}(X)$.
- In fact, following de Finetti we treat expectation as the primitive quantity in the analysis.
- Probabilities can be obtained by examining the expectation of indicator functions.
The Bayes Linear Update.

- We replace Bayes Theorem with the Bayes Linear Update.
The Bayes Linear Update.

- We replace Bayes Theorem with the Bayes Linear Update.
- Say we are interested in a vector of random quantities $B$, and we are going to measure some related vector of quantities $D$. 

The Bayes Linear Update.

- We replace Bayes Theorem with the Bayes Linear Update.
- Say we are interested in a vector of random quantities $B$, and we are going to measure some related vector of quantities $D$.
- We specify the prior $E[B], E[D], \text{Var}(B), \text{Var}(D), \text{Cov}(B, D)$ and we can now update our beliefs about $E(B)$ after measuring $D$:
The Bayes Linear Update.

- We replace Bayes Theorem with the Bayes Linear Update.
- Say we are interested in a vector of random quantities $B$, and we are going to measure some related vector of quantities $D$.
- We specify the prior $E[B]$, $E[D]$, $\text{Var}(B)$, $\text{Var}(D)$, $\text{Cov}(B, D)$ and we can now update our beliefs about $E(B)$ after measuring $D$:

$$E_D[B] = E[B] + \text{Cov}(B, D)\text{Var}(D)^{-1}(D - E[D]),$$
The Bayes Linear Update.

- We replace Bayes Theorem with the Bayes Linear Update.
- Say we are interested in a vector of random quantities $B$, and we are going to measure some related vector of quantities $D$.
- We specify the prior $E[B]$, $E[D]$, $\text{Var}(B)$, $\text{Var}(D)$, $\text{Cov}(B, D)$ and we can now update our beliefs about $E(B)$ after measuring $D$:

\[
\begin{align*}
E_D[B] &= E[B] + \text{Cov}(B, D)\text{Var}(D)^{-1}(D - E[D]), \\
\text{Var}_D[B] &= \text{Var}(B) - \text{Cov}(B, D)\text{Var}(D)^{-1}\text{Cov}(D, B)
\end{align*}
\]
The Bayes Linear Update.

- We replace Bayes Theorem with the Bayes Linear Update.
- Say we are interested in a vector of random quantities $B$, and we are going to measure some related vector of quantities $D$.
- We specify the prior $E[B]$, $E[D]$, $\text{Var}(B)$, $\text{Var}(D)$, $\text{Cov}(B, D)$ and we can now update our beliefs about $E(B)$ after measuring $D$:

$$E_D[B] = E[B] + \text{Cov}(B, D)\text{Var}(D)^{-1}(D - E[D]),$$

$$\text{Var}_D[B] = \text{Var}(B) - \text{Cov}(B, D)\text{Var}(D)^{-1}\text{Cov}(D, B)$$

- $E_D[B]$, $\text{Var}_D[B]$ are the expectation and variance for $B$ adjusted by $D$. 
The Bayes Linear Update.

• We replace Bayes Theorem with the Bayes Linear Update.
• Say we are interested in a vector of random quantities $B$, and we are going to measure some related vector of quantities $D$.
• We specify the prior $E[B]$, $E[D]$, $\text{Var}(B)$, $\text{Var}(D)$, $\text{Cov}(B, D)$ and we can now update our beliefs about $E(B)$ after measuring $D$:

$$
E_D[B] = E[B] + \text{Cov}(B, D)\text{Var}(D)^{-1}(D - E[D]),
$$

$$
\text{Var}_D[B] = \text{Var}(B) - \text{Cov}(B, D)\text{Var}(D)^{-1}\text{Cov}(D, B)
$$

• $E_D[B]$, $\text{Var}_D[B]$ are the expectation and variance for $B$ adjusted by $D$.
• Bayes linear adjustment may be viewed as:
The Bayes Linear Update.

- We replace Bayes Theorem with the Bayes Linear Update.
- Say we are interested in a vector of random quantities $B$, and we are going to measure some related vector of quantities $D$.
- We specify the prior $E[B]$, $E[D]$, $\text{Var}(B)$, $\text{Var}(D)$, $\text{Cov}(B, D)$ and we can now update our beliefs about $E(B)$ after measuring $D$:

\[
\begin{align*}
E[D][B] &= E[B] + \text{Cov}(B, D)\text{Var}(D)^{-1}(D - E[D]), \\
\text{Var}_D[B] &= \text{Var}(B) - \text{Cov}(B, D)\text{Var}(D)^{-1}\text{Cov}(D, B)
\end{align*}
\]

- $E[D][B]$, $\text{Var}_D[B]$ are the expectation and variance for $B$ adjusted by $D$.
- Bayes linear adjustment may be viewed as:
  
  [1] an approximation to a full probabilistic Bayesian analysis;
The Bayes Linear Update.

- We replace Bayes Theorem with the Bayes Linear Update.
- Say we are interested in a vector of random quantities $B$, and we are going to measure some related vector of quantities $D$.
- We specify the prior $E[B], E[D], \text{Var}(B), \text{Var}(D), \text{Cov}(B, D)$ and we can now update our beliefs about $E(B)$ after measuring $D$:
  \[
  E_D[B] = E[B] + \text{Cov}(B, D)\text{Var}(D)^{-1}(D - E[D]), \\
  \text{Var}_D[B] = \text{Var}(B) - \text{Cov}(B, D)\text{Var}(D)^{-1}\text{Cov}(D, B)
  \]
- $E_D[B], \text{Var}_D[B]$ are the expectation and variance for $B$ adjusted by $D$.
- Bayes linear adjustment may be viewed as:
  [1] an approximation to a full probabilistic Bayesian analysis;
  [2] the “appropriate” analysis given a partial specification based on expectation (with methodology for modelling, interpretation and diagnostic analysis).
Bayes Linear and Emulation.

- Bayes Linear techniques fit naturally into the area of analysis of Computer Models: for a model that maps inputs $x$ to outputs $f_i(x)$ we emulate as follows.
Bayes Linear and Emulation.

- Bayes Linear techniques fit naturally into the area of analysis of Computer Models: for a model that maps inputs $x$ to outputs $f_i(x)$ we emulate as follows.
- For each output $i$, we identify a subset of Active inputs $x_A$, then emulate with:
Bayes Linear and Emulation.

- Bayes Linear techniques fit naturally into the area of analysis of Computer Models: for a model that maps inputs $x$ to outputs $f_i(x)$ we emulate as follows.
- For each output $i$, we identify a subset of Active inputs $x_A$, then emulate with:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$
Bayes Linear and Emulation.

- Bayes Linear techniques fit naturally into the area of analysis of Computer Models: for a model that maps inputs $x$ to outputs $f_i(x)$ we emulate as follows.
- For each output $i$, we identify a subset of Active inputs $x_A$, then emulate with:

  $$f_i(x) = \sum_j \beta_{ij} g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij} g_{ij}(x^A)$ is a polynomial in the active inputs.
Bayes Linear and Emulation.

- Bayes Linear techniques fit naturally into the area of analysis of Computer Models: for a model that maps inputs $x$ to outputs $f_i(x)$ we emulate as follows.
- For each output $i$, we identify a subset of Active inputs $x_A$, then emulate with:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij} g_{ij}(x^A)$ is a polynomial in the active inputs.
- $u(x^A)$ is a random process with expectation zero and covariance structure:

$$\text{Cov}(u_i(x^A_1), u_i(x^A_2)) = \sigma^2_i \exp[-\theta_i|x^A_1 - x^A_2|^2]$$
Bayes Linear and Emulation.

- Bayes Linear techniques fit naturally into the area of analysis of Computer Models: for a model that maps inputs \( x \) to outputs \( f_i(x) \) we emulate as follows.

- For each output \( i \), we identify a subset of Active inputs \( x_A \), then emulate with:

\[
  f_i(x) = \sum_j \beta_{ij} g_{ij}(x^A) + u_i(x^A) + \delta_i(x)
\]

- The \( \sum_j \beta_{ij} g_{ij}(x^A) \) is a polynomial in the active inputs.

- \( u(x^A) \) is a random process with expectation zero and covariance structure:

\[
  \text{Cov}(u_i(x_1^A), u_i(x_2^A)) = \sigma_i^2 \exp[-\theta_i |x_1^A - x_2^A|^2]
\]

- The nugget \( \delta_i(x) \) models the effects of inactive variables as random noise.
• Bayes Linear techniques fit naturally into the area of analysis of Computer Models: for a model that maps inputs $x$ to to outputs $f_i(x)$ we emulate as follows.

• For each output $i$, we identify a subset of Active inputs $x_A$, then emulate with:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

• The $\sum_j \beta_{ij} g_{ij}(x^A)$ is a polynomial in the active inputs.

• $u(x^A)$ is a random process with expectation zero and covariance structure:

$$\text{Cov}(u_i(x_1^A), u_i(x_2^A)) = \sigma_i^2 \exp[-\theta_i |x_1^A - x_2^A|^2]$$

• The nugget $\delta_i(x)$ models the effects of inactive variables as random noise.

• The Emulators give the expectation $E[f_i(x)]$ and variance $\text{Var}(f_i(x))$ at point $x$ for each output $f_i(x)$. 
Andromeda Galaxy and Hubble Deep Field View

- Andromeda Galaxy: closest large galaxy to our own milky way, contains 1 trillion stars.
- Hubble Deep Field: one of the furthest images yet taken. Covers 2 millionths of the sky but contains over 3000 galaxies.
Andromeda Galaxy and Hubble Deep Field View

- Andromeda Galaxy: closest large galaxy to our own milky way, contains 1 trillion stars.
- Hubble Deep Field: one of the furthest images yet taken. Covers 2 millionths of the sky but contains over 3000 galaxies.
- Cosmologists want to understand the creation and evolution of Galaxies in the presence of large amounts of Dark Matter.
Galform

- Galform attempts to simulate the creation and evolution of approximately 1 million galaxies, from the big bang until now.
Galform

- Galform attempts to simulate the creation and evolution of approximately 1 million galaxies, from the big bang until now.
- First a Dark Matter simulation is performed over a volume of $(1.63 \text{ billion light years})^3$. This takes 3 months on a supercomputer and cannot be easily repeated.
Galform

- Galform attempts to simulate the creation and evolution of approximately 1 million galaxies, from the big bang until now.

- First a Dark Matter simulation is performed over a volume of $(1.63 \text{ billion light years})^3$. This takes 3 months on a supercomputer and cannot be easily repeated.

- This volume is then split into 512 sub-volumes, and Galform is run independently on each sub-volume using the output of the Dark matter simulation.
Galform

- Galform attempts to simulate the creation and evolution of approximately 1 million galaxies, from the big bang until now.

- First a Dark Matter simulation is performed over a volume of \((1.63\text{ billion light years})^3\). This takes 3 months on a supercomputer and cannot be easily repeated.

- This volume is then split into 512 sub-volumes, and Galform is run independently on each sub-volume using the output of the Dark matter simulation.

- Galform models the more complicated physics of ‘normal’ matter: gas cloud formation, radiation, cooling, star formation and black hole creation, and takes approximately 30 minutes to run.
Galform

- Galform attempts to simulate the creation and evolution of approximately 1 million galaxies, from the big bang until now.
- First a Dark Matter simulation is performed over a volume of $(1.63 \text{ billion light years})^3$. This takes 3 months on a supercomputer and cannot be easily repeated.
- This volume is then split into 512 sub-volumes, and Galform is run independently on each sub-volume using the output of the Dark matter simulation.
- Galform models the more complicated physics of ‘normal’ matter: gas cloud formation, radiation, cooling, star formation and black hole creation, and takes approximately 30 minutes to run.
- Galform has 17 input parameters that are to vary...
Galform

- Galform attempts to simulate the creation and evolution of approximately 1 million galaxies, from the big bang until now.

- First a Dark Matter simulation is performed over a volume of \((1.63 \text{ billion light years})^3\). This takes 3 months on a supercomputer and cannot be easily repeated.

- This volume is then split into 512 sub-volumes, and Galform is run independently on each sub-volume using the output of the Dark matter simulation.

- Galform models the more complicated physics of ‘normal’ matter: gas cloud formation, radiation, cooling, star formation and black hole creation, and takes approximately 30 minutes to run.

- Galform has 17 input parameters that are to vary...

- Want to determine the set of inputs that will give rise to an acceptable match between the model output and observed data on Galaxies in the real Universe.
**Input Parameters**

- Due to expert judgements we attempt to History Match Galform over 8 of the input parameters (while taking into account the possible effects of the remaining 9).
Input Parameters

- Due to expert judgements we attempt to History Match Galform over 8 of the input parameters (while taking into account the possible effects of the remaining 9).

- The input parameters and their initial ranges are:

  vhotdisk: 100 - 550  
aReheat: 0.2 - 1.2  
alphacool: 0.2 - 1.2  
vhotburst: 100 - 550  
epsilonStar: 0.001 - 0.1  
stabledisk: 0.65 - 0.95  
alphahot: 2 - 3.7  
yield: 0.02 - 0.05

  What values should I choose to get 'good' outputs?

- The other 9 parameters are: V CUT, Z CUT, alphastar, tau0mrg, fellip, fburst, FSMBH, epsilonSMBHEddington and tdisk.
• Galform provides multiple output data sets.
• Initially we only analyse the luminosity functions which give the number of galaxies per unit volume, for each luminosity.
Galform Outputs: The Luminosity Functions

- Galform provides multiple output data sets.
- Initially we only analyse the luminosity functions which give the number of galaxies per unit volume, for each luminosity.
- Bj Luminosity: corresponds to density of young (blue) galaxies
- K Luminosity: corresponds to density of old (red) galaxies
Galform provides multiple output data sets.

Initially we only analyse the luminosity functions which give the number of galaxies per unit volume, for each luminosity.

- Bj Luminosity: corresponds to density of young (blue) galaxies
- K Luminosity: corresponds to density of old (red) galaxies
Galform Outputs: The Luminosity Functions

- Galform provides multiple output data sets.
- Initially we only analyse the luminosity functions which give the number of galaxies per unit volume, for each luminosity.
- Bj Luminosity: corresponds to density of young (blue) galaxies
- K Luminosity: corresponds to density of old (red) galaxies
We take only 11 outputs to use for the calibration process.
11 Outputs Chosen

- We take only 11 outputs to use for the calibration process.
- Outputs chosen to be informative enough to allow us to cut down the parameter space, but simple enough to be emulated easily.
Linking Model to Reality

- We represent the simulator (Galform) as a function, which maps the input parameters \( x \) to the outputs \( f(x) \).
Linking Model to Reality

- We represent the simulator (Galform) as a function, which maps the input parameters $x$ to the outputs $f(x)$.
- We use the “Best Input Approach”, where we assume there exists a value $x^*$ independent of the function $f$ such that the value of $f^* = f(x^*)$ summarises all the information the simulator conveys about the system.
Linking Model to Reality

- We represent the simulator (Galform) as a function, which maps the input parameters $x$ to the outputs $f(x)$.
- We use the “Best Input Approach”, where we assume there exists a value $x^*$ independent of the function $f$ such that the value of $f^* = f(x^*)$ summarises all the information the simulator conveys about the system.
- We then link the real system denoted by $y$ to the simulator by the equation:

$$y = f^* + \epsilon_{md},$$

where we define $\epsilon_{md}$ to be the model discrepancy and assume that $\epsilon_{md}$ is independent of $f, x^*$. (Here, and onwards, all probabilistic statements relate to the uncertainty judgements of the analyst.)
Linking Model to Reality

- We represent the simulator (Galform) as a function, which maps the input parameters $x$ to the outputs $f(x)$.
- We use the “Best Input Approach”, where we assume there exists a value $x^*$ independent of the function $f$ such that the value of $f^* = f(x^*)$ summarises all the information the simulator conveys about the system.
- We then link the real system denoted by $y$ to the simulator by the equation:

$$y = f^* + \epsilon_{md},$$

where we define $\epsilon_{md}$ to be the model discrepancy and assume that $\epsilon_{md}$ is independent of $f, x^*$. (Here, and onwards, all probabilistic statements relate to the uncertainty judgements of the analyst.)
- Finally, we relate the true system $y$ to the observational data $z$ by,

$$z = y + \epsilon_{obs},$$

where $\epsilon_{obs}$ represent the observational errors.
General Strategy

- **Design** Evaluate a space filling batch of runs of the model using a Latin Hypercube design.
General Strategy

- **Design** Evaluate a space filling batch of runs of the model using a Latin Hypercube design.

- **Emulation** We build an emulator that mimics the behaviour of each of the 11 outputs of the Galform model, and is fast to evaluate.
General Strategy

- **Design** Evaluate a space filling batch of runs of the model using a Latin Hypercube design.

- **Emulation** We build an emulator that mimics the behaviour of each of the 11 outputs of the Galform model, and is fast to evaluate.

- **Implausibility** Construct Implausibility Measures that describe which parts of the input space are unlikely to give rise to 'acceptable' fits between model output and observed data.
General Strategy

- **Design** Evaluate a space filling batch of runs of the model using a Latin Hypercube design.

- **Emulation** We build an emulator that mimics the behaviour of each of the 11 outputs of the Galform model, and is fast to evaluate.

- **Implausibility** Construct Implausibility Measures that describe which parts of the input space are unlikely to give rise to ‘acceptable’ fits between model output and observed data.

- **History Match** Discard from further analysis, regions of input space deemed Implausible.
General Strategy

- **Design** Evaluate a space filling batch of runs of the model using a Latin Hypercube design.

- **Emulation** We build an emulator that mimics the behaviour of each of the 11 outputs of the Galform model, and is fast to evaluate.

- **Implausibility** Construct Implausibility Measures that describe which parts of the input space are unlikely to give rise to ‘acceptable’ fits between model output and observed data.

- **History Match** Discard from further analysis, regions of input space deemed Implausible.

- **Refocussing** Repeat above process, but now start with the non-implausible input volume.
General Strategy

- **Design** Evaluate a space filling batch of runs of the model using a Latin Hypercube design.

- **Emulation** We build an emulator that mimics the behaviour of each of the 11 outputs of the Galform model, and is fast to evaluate.

- **Implausibility** Construct Implausibility Measures that describe which parts of the input space are unlikely to give rise to ‘acceptable’ fits between model output and observed data.

- **History Match** Discard from further analysis, regions of input space deemed Implausible.

- **Refocussing** Repeat above process, but now start with the non-implausible input volume.

- We will use Bayes Linear Analysis which treats expectation as primitive, and only requires specification of expectations, variances and covariances.
Design: Latin Hypercubes

- **Design** Construct a batch of runs of the model using a Latin Hypercube design:
Design: Latin Hypercubes

- **Design** Construct a batch of runs of the model using a Latin Hypercube design:

- We evaluated 1000 runs of the model for the first Wave.
Galform: Emulation

- For each of the 11 outputs we pick active variables $x^A$ then emulate univariately (at first) using:

$$f_i(x) = \sum_j \beta_{ij} \, g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$
For each of the 11 outputs we pick active variables $x^A$ then emulate univariately (at first) using:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij} g_{ij}(x^A)$ is a 3rd order polynomial in the active inputs.
Galform: Emulation

- For each of the 11 outputs we pick active variables $x^A$ then emulate univariately (at first) using:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij} g_{ij}(x^A)$ is a 3rd order polynomial in the active inputs.
- $u(x^A)$ is a Gaussian process.
Galform: Emulation

- For each of the 11 outputs we pick active variables $x^A$ then emulate univariately (at first) using:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij} g_{ij}(x^A)$ is a 3rd order polynomial in the active inputs.
- $u(x^A)$ is a Gaussian process.
- The nugget $\delta_i(x)$ models the effects of inactive variables as random noise.
Galform: Emulation

- For each of the 11 outputs we pick active variables $x^A$ then emulate univariately (at first) using:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij} g_{ij}(x^A)$ is a 3rd order polynomial in the active inputs.
- $u(x^A)$ is a Gaussian process.
- The nugget $\delta_i(x)$ models the effects of inactive variables as random noise.
- The $u_i(x^A)$ have covariance structure given by:

$$\text{Cov}(u_i(x_1^A), u_i(x_2^A)) = \sigma_i^2 \exp[-\theta_i |x_1^A - x_2^A|^2]$$
Galform: Emulation

- For each of the 11 outputs we pick active variables $x^A$ then emulate univariately (at first) using:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij} g_{ij}(x^A)$ is a 3rd order polynomial in the active inputs.
- $u(x^A)$ is a Gaussian process.
- The nugget $\delta_i(x)$ models the effects of inactive variables as random noise.
- The $u_i(x^A)$ have covariance structure given by:

$$\text{Cov}(u_i(x_1^A), u_i(x_2^A)) = \sigma_i^2 \exp[-\theta_i |x_1^A - x_2^A|^2]$$

- The Emulators give the expectation $E[f_i(x)]$ and variance $\text{Var}(f_i(x))$ at point $x$ for each output given by $i = 1, .., 11$, and are fast to evaluate.
Model Discrepancy

Before calculating the implausibility we need to assess the Model Discrepancy and Measurement error.

Model Discrepancy $MD = \text{Var}(\epsilon_{md}) = \Phi_40 + \Phi_9 + \Phi_E$

- $\Phi_{40}$: Discrepancy term due to choosing first 40 sub-volumes from full 512 sub-volumes. Assess this by repeating 100 runs but now choosing 40 random regions.
Model Discrepancy

Before calculating the implausibility we need to assess the Model Discrepancy and Measurement error.

Model Discrepancy $MD = \text{Var}(\epsilon_{md}) = \Phi_{40} + \Phi_{9} + \Phi_{E}$

- $\Phi_{40}$: Discrepancy term due to choosing first 40 sub-volumes from full 512 sub-volumes. Assess this by repeating 100 runs but now choosing 40 random regions.

- $\Phi_{9}$: As we have neglected 9 parameters (due to expert advice) we need to assess effect of this (by running latin hypercube design across all 17 parameters)
**Model Discrepancy**

Before calculating the implausibility we need to assess the Model Discrepancy and Measurement error.

Model Discrepancy \( MD = \text{Var}(\epsilon_{md}) = \Phi_{40} + \Phi_{9} + \Phi_{E} \)

- \( \Phi_{40} \): Discrepancy term due to choosing first 40 sub-volumes from full 512 sub-volumes. Assess this by repeating 100 runs but now choosing 40 random regions.

- \( \Phi_{9} \): As we have neglected 9 parameters (due to expert advice) we need to assess effect of this (by running latin hypercube design across all 17 parameters)

- \( \Phi_{E} \): Expert assessment of model discrepancy of full model with 17 parameters and using 512 sub-volumes

It is straightforward to find the multivariate expressions for \( \Phi_{40} \) and \( \Phi_{9} \), but \( \Phi_{E} \) requires more careful thought.
Measurement Error

Observational Errors $OE = \text{Var}(\epsilon_{obs})$ are composed of 4 parts:

- Normalisation Error: correlated vertical error on all luminosity output points
Observational Errors $OE = \text{Var}(\epsilon_{\text{obs}})$ are composed of 4 parts:

- Normalisation Error: correlated vertical error on all luminosity output points
- Luminosity Zero Point Error: correlated horizontal error on all luminosity points
Measurement Error

Observational Errors $OE = \text{Var}(\epsilon_{obs})$ are composed of 4 parts:

- Normalisation Error: correlated vertical error on all luminosity output points
- Luminosity Zero Point Error: correlated horizontal error on all luminosity points
- $k + e$ Correction Error: Outputs have to be corrected for the fact that galaxies are moving away from us at different speeds (light is red-shifted), and for the fact that galaxies are seen in the past (as light takes millions of years to reach us)
**Measurement Error**

Observational Errors $OE = \text{Var}(\epsilon_{obs})$ are composed of 4 parts:

- **Normalisation Error**: correlated vertical error on all luminosity output points.
- **Luminosity Zero Point Error**: correlated horizontal error on all luminosity points.
- $k + e$ **Correction Error**: Outputs have to be corrected for the fact that galaxies are moving away from us at different speeds (light is red-shifted), and for the fact that galaxies are seen in the past (as light takes millions of years to reach us).
- **Poisson Error**: assumed Poisson process to describe galaxy production.
Observational Errors $OE = \text{Var}(\epsilon_{\text{obs}})$ are composed of 4 parts:

- **Normalisation Error**: correlated vertical error on all luminosity output points
- **Luminosity Zero Point Error**: correlated horizontal error on all luminosity points
- $k + e$ **Correction Error**: Outputs have to be corrected for the fact that galaxies are moving away from us at different speeds (light is red-shifted), and for the fact that galaxies are seen in the past (as light takes millions of years to reach us)
- **Poisson Error**: assumed Poisson process to describe galaxy production

The multivariate form for each of these quantities is straightforward(!) to calculate.
Implausibility (Univariate)

We can now calculate the **Implausibility** at any input parameter point $x$ for each of the 11 outputs. This is given by:

$$I_{(i)}^2(x) = \frac{|E[f_i(x)] - z_i|^2}{(\text{Var}(f_i(x)) + MD + OE)}$$
Implausibility (Univariate)

We can now calculate the **Implausibility** at any input parameter point $x$ for each of the 11 outputs. This is given by:

$$I^2_{(i)}(x) = |E[f_i(x)] - z_i|^2 / (\text{Var}(f_i(x)) + MD + OE)$$

- $E[f_i(x)]$ and $\text{Var}(f_i(x))$ are the emulator expectation and variance.
Implausibility (Univariate)

We can now calculate the **Implausibility** at any input parameter point $x$ for each of the 11 outputs. This is given by:

$$I^2_{(i)}(x) = |E[f_i(x)] - z_i|^2 / (\text{Var}(f_i(x)) + MD + OE)$$

- $E[f_i(x)]$ and $\text{Var}(f_i(x))$ are the emulator expectation and variance.
- $z_i$ are the observed data and $MD$ and $OE$ are the (univariate) Model Discrepancy and Observational Errors.
Implausibility (Univariate)

We can now calculate the **Implausibility** at any input parameter point $x$ for each of the 11 outputs. This is given by:

$$I^{2}_{(i)}(x) = |E[f_i(x)] - z_i|^2 / (\text{Var}(f_i(x)) + MD + OE)$$

- $E[f_i(x)]$ and $\text{Var}(f_i(x))$ are the emulator expectation and variance.
- $z_i$ are the observed data and $MD$ and $OE$ are the (univariate) Model Discrepancy and Observational Errors.
- Large values of $I_{(i)}(x)$ imply that we are highly unlikely to obtain acceptable matches between model output and observed data at input $x$. 
Implausibility (Univariate)

We can now calculate the **Implausibility** at any input parameter point $x$ for each of the 11 outputs. This is given by:

$$I_{(i)}^2(x) = |E[f_i(x)] - z_i|^2 / (\text{Var}(f_i(x)) + MD + OE)$$

- $E[f_i(x)]$ and $\text{Var}(f_i(x))$ are the emulator expectation and variance.
- $z_i$ are the observed data and $MD$ and $OE$ are the (univariate) Model Discrepancy and Observational Errors.
- Large values of $I_{(i)}(x)$ imply that we are highly unlikely to obtain acceptable matches between model output and observed data at input $x$.
- We can combine the implausibilities across outputs by maximizing over outputs:

$$I_M(x) = \max_i I_{(i)}(x)$$
Implausibility (Univariate)

We can now calculate the **Implausibility** at any input parameter point $x$ for each of the 11 outputs. This is given by:

$$I^2_{(i)}(x) = |E[f_i(x)] - z_i|^2 / (\text{Var}(f_i(x)) + MD + OE)$$

- $E[f_i(x)]$ and $\text{Var}(f_i(x))$ are the emulator expectation and variance.
- $z_i$ are the observed data and $MD$ and $OE$ are the (univariate) Model Discrepancy and Observational Errors.
- Large values of $I_{(i)}(x)$ imply that we are highly unlikely to obtain acceptable matches between model output and observed data at input $x$.
- We can combine the implausibilities across outputs by maximizing over outputs:

$$I_M(x) = \max_i I_{(i)}(x)$$

- We can then impose a cutoff $I_M(x) < 3$ in order to discard regions of input parameter space.
Calibration via Implausibility: a 1D Example
Calibration via Implausibility: a 1D Example
Calibration via Implausibility: a 1D Example
Calibration via Implausibility: a 1D Example

![Graph showing Emulator of Galform Output and Implausibility](image-url)
Calibration via Implausibility: a 1D Example
Calibration via Implausibility: a 1D Example

Emulator of Galform Output

Implausibility

Input Parameter x

Galform Output

Implausibility = f(x)
Calibration via Implausibility: a 1D Example

Emulator of Galform Output

Implausibility

Input Parameter x

Galform Output

Implausibility = f(x)
Calibration via Implausibility: a 1D Example
2D Implausibility Projections: Stage 1 (8%)
2D Implausibility Projections: Stage 1 (8%) to Stage 4 (0.12%)
2D Implausibility Projections: Stage 1 (8%) to Stage 4 (0.12%)
2D Implausibility Projections: Stage 1 (8%) to Stage 4 (0.12%)
2D Implausibility Projections: Stage 1 (8%) to Stage 4 (0.12%)
2D Implausibility Projections: Stage 1 (8%) to Stage 4 (0.12%)
2D Implausibility Projections: Stage 1 (8%) to Stage 4 (0.12%)
2D Implausibility Projections: Stage 1 (8%) to Stage 4 (0.12%)
2D Implausibility Projections: Stage 1 (8%) to Stage 4 (0.12%)
2D Implausibility Projections: Stage 4 (0.12%)
Summary of Results

- We have completed Four Stages:

<table>
<thead>
<tr>
<th>Stage</th>
<th>No. Model Runs</th>
<th>No. Active Vars</th>
<th>Adjusted $R^2$</th>
<th>Space Remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 1</td>
<td>1000</td>
<td>5</td>
<td>0.58 - 0.90</td>
<td>8.0 %</td>
</tr>
<tr>
<td>Stage 2</td>
<td>1916</td>
<td>8</td>
<td>0.83 - 0.98</td>
<td>2.9 %</td>
</tr>
<tr>
<td>Stage 3</td>
<td>1487</td>
<td>8</td>
<td>0.79 - 0.99</td>
<td>1.2 %</td>
</tr>
<tr>
<td>Stage 4</td>
<td>2000</td>
<td>10</td>
<td>0.75 - 0.99</td>
<td>0.21 %</td>
</tr>
</tbody>
</table>

- In Stages 3 and 4 we used a Multivariate Implausibility measure to help reduce space further.
- In Stage 4 we included 2 more active input variables that had previously been inactive.
bj Luminosity Output of Waves 1, 2, 3 and 5

bj Luminosity Function Wave 1

log(No. Galaxies per unit Volume) vs. bj Luminosity
bj Luminosity Output of Waves 1, 2, 3 and 5
bj Luminosity Output of Waves 1, 2, 3 and 5

bj Luminosity Function Wave 1

bj Luminosity Function Wave 2

bj Luminosity Function Wave 3
bj Luminosity Output of Waves 1, 2, 3 and 5

bj Luminosity Function Wave 1

bj Luminosity Function Wave 2

bj Luminosity Function Wave 3

bj Luminosity Function Wave 5
K Luminosity Output of Waves 1,2,3 and 5

K Luminosity Function Wave 1
K Luminosity Output of Waves 1, 2, 3 and 5

K Luminosity Function Wave 1

K Luminosity Function Wave 2
K Luminosity Output of Waves 1, 2, 3 and 5

K Luminosity Function Wave 1

K Luminosity Function Wave 2

K Luminosity Function Wave 3
K Luminosity Output of Waves 1, 2, 3 and 5
Conclusions and Further Issues

- Have introduced the Bayes Linear methodology.
Conclusions and Further Issues

- Have introduced the Bayes Linear methodology.
- We have discussed how to Emulate a complex physical model such as Galform using a Bayes Linear approach.
Conclusions and Further Issues

- Have introduced the Bayes Linear methodology.
- We have discussed how to Emulate a complex physical model such as Galform using a Bayes Linear approach.
- Described the Implausibility measure and how to discard input space by imposing cutoffs on $I_M(x)$. 
Conclusions and Further Issues

- Have introduced the Bayes Linear methodology.
- We have discussed how to Emulate a complex physical model such as Galform using a Bayes Linear approach.
- Described the Implausibility measure and how to discard input space by imposing cutoffs on $I_M(x)$.
- Outlined the iterative procedure we used to perform History Matching via Implausibility.
Conclusions and Further Issues

- Have introduced the Bayes Linear methodology.
- We have discussed how to Emulate a complex physical model such as Galform using a Bayes Linear approach.
- Described the Implausibility measure and how to discard input space by imposing cutoffs on $I_M(x)$.
- Outlined the iterative procedure we used to perform History Matching via Implausibility.
- Have reduced the input parameter space to less than 0.12% of its original volume.
Conclusions and Further Issues

- Have introduced the Bayes Linear methodology.
- We have discussed how to Emulate a complex physical model such as Galform using a Bayes Linear approach.
- Described the Implausibility measure and how to discard input space by imposing cutoffs on $I_M(x)$.
- Outlined the iterative procedure we used to perform History Matching via Implausibility.
- Have reduced the input parameter space to less than 0.12% of its original volume.
- (We now have a large set of ‘acceptable’ runs that can be analysed by the Cosmologists and used to explore other features of their model: already we have found good matches to additional data sets that the cosmologist have been unable to match for the last 7 years.)
References


