Bayesian model selection for computer model validation via mixture model estimation

Kaniav Kamary
ATER, CNAM

Joint work with É. Parent, P. Barbillon, M. Keller and N. Bousquet
Outline

Computer model validation

Introduction and Motivations

Testing problems via mixture estimation model

Mixture estimation

Illustrations

Conclusion and perspectives
What is a computer model validation?

Physical phenomena (real system):

- Question: how can we study a real phenomenon?
- What is a computer model?
What is a computer model validation?

Physical phenomena (real system) $\xrightarrow{\text{Physicien}}$ simplify by physical laws $\rightarrow$ Physical model

- Physicien use a part of reality, complicated
- then find a simple representation to which he applies a theory
- the simple representation help us to reproduce the functionality of the physical phenomena
- to explain, analyze and predict some aspects
What is a computer model validation?

Physical phenomena (real system) $\xrightarrow{\text{Physicien}}$ simplify by physical laws $\rightarrow$ Physical model

**Example** when we want to study movements of a car?

- do not consider the car itself as a representation
- BUT representing it a point in space to which we assign a masse and speed
- describe the movements based on the theory of point mechanics

From this simple representation we can evaluate the movements of the car
What is a computer model validation?

Physical model $\xrightarrow{\text{by applying the mathematical approximation techniques and theories}}$ Numerical code

or computer model
or simulator

$\xrightarrow{\text{the physical model is translated to a mathematical model}}$

We obtain therefore what we call numerical code.
What is a computer model validation?

Computer model → Statisticien

examine by statistical methods
to know if it display a reliable picture of “reality”?

Statisticien’s work in two steps:

▶ Verification: quantifying the errors produced by the approximate resolution of the mathematical problem
▶ Validation: try to answer the question on the validity of the mathematical modeling of physical system
What is a computer model validation?

Physical phenomena (real system) \[\xrightarrow{\text{Physicien}}\] simplify by physical laws \[\xrightarrow{\text{Model uncertainty}}\] Physical model

Physical model \[\xrightarrow{\text{Mathematicien}}\] numerical approximation and algorithms \[\xrightarrow{\text{Numerical uncertainty}}\] Numerical code or computer model or simulator

Computer model \[\xrightarrow{\text{Statisticien}}\] examine by statistical methods to know if it display a reliable picture of “reality”?
Physical system: Mathematical notation

For unknown real physical system $r(x)$

- physical parameters: $x \in X \subset \mathbb{R}^d$ observable and controllable inputs

the observation $y$ can be obtained as

$$x \in \mathbb{R}^d \rightarrow \text{r: physical system} \rightarrow y \in \mathbb{R}$$

or

$$y = r(x) + \epsilon$$

with $\epsilon \sim \mathcal{N}(0, \lambda)$: measure error that degrades the knowledge of $r(x)$. 
Computer model: Mathematical notation

Computer model is represented by a function $f$:

$$f : \mathbb{R}^d \times \mathbb{R}^p \rightarrow \mathbb{R}$$

$$(x, \theta) \rightarrow f(x, \theta)$$

in which

- The inputs $\theta$ are the calibration parameters of the computation code.

The function $f(x, \theta)$:

- can have a simple linear structure $f(x, \theta) = g(x)\theta$

Or challenges

- can be a complex function with large computation time
  - sparing computation time by meta-modelling
Validation problem: Statistical modelling

Suppose that
\[ \delta(x) = r(x) - f(x, \theta) \]
\( \delta(x) \) is a stochastic function and called “code error”. Then the observation \( y \) can be obtained as
\[ y = f(x, \theta) + \delta(x) + \epsilon \]

Remark
- Identifiability issue: How to define \( \theta \) and \( \delta \) w.r.t \( r(x) \) in a good way? It is not clear.

Does the computer code mimics the reality or
\[ y = f(x, \theta) + \epsilon? \]
Validation problem: Statistical modelling

We are interested in testing hypotheses

\[ H_0 : \delta(\mathbf{x}) = 0 \]
\[ H_1 : \delta(\mathbf{x}) \neq 0 \]

The code validation can be performed using Bayesian model selection methods Damblin et al. (2016).

\[ M_0 : y = f(\mathbf{x}, \theta_0) + \epsilon_0 \]
\[ M_1 : y = f(\mathbf{x}, \theta_1) + \delta(\mathbf{x}) + \epsilon_1. \]

where \( M_0 \) and \( M_1 \) are the pure code and the discrepancy-corrected code, respectively.
Bayesian model choice:

- a special case of testing hypotheses theory
- comparison of $k$ potential statistical models towards the selection of model that fits the data “best”
- not to seek to identify which model is “true”, but to indicate which fits data better

The most common approaches to Bayesian hypothesis testing in practice

- posterior probabilities of the model given the data
- Bayes factor and its approximations such as the Bayesian information criterion (BIC) and the Deviance information criterion (DIC) and posterior predictive tools and their variants
Suppose that two families of models under comparison are given by

\( M_0 : x \sim \pi_0(x|\theta_0), \ \theta_0 \in \Theta_0 \) and \( M_1 : x \sim \pi_1(x|\theta_1), \ \theta_1 \in \Theta_1, \)

and associate with each model a prior distribution,

\[
\begin{align*}
\theta_0 &\sim \pi_0(\theta_0) \quad \text{and} \quad \theta_1 \sim \pi_1(\theta_1), \\
\omega_0 &= \pi(M_0) \quad \text{and} \quad \omega_1 = \pi(M_1)
\end{align*}
\]
Standard Bayesian approach to testing

In order to compare the marginal likelihoods

\[ m_0(x) = \int_{\Theta_0} \pi_0(x|\theta_0) \pi_0(\theta_0) \, d(\theta_0) \quad \text{and} \quad m_1(x) = \int_{\Theta_1} \pi_1(x|\theta_1) \pi_1(\theta_1) \, d(\theta_1) \]

either through Bayes factor or posterior probability, respectively:

\[ B_{01} = \frac{m_0(x)}{m_1(x)}, \quad P(M_0|x) = \frac{\omega_0 m_0(x)}{\omega_0 m_0(x) + \omega_1 m_1(x)}; \]

the latter depends on the prior weights \( \omega_i \).
How to make a decision

Bayesian decision step in order to comparing two models
▶ comparing Bayes factor $\mathcal{B}_{01}$ with threshold value of one or
▶ comparing posterior probability $\mathbb{P}(M_0|x)$ with bound

When comparing more than two models
▶ selecting model with highest posterior probability but highly dependent on the prior modeling, even with large databases
How to make a decision

Bayesian decision step in order to comparing two models

▶ comparing Bayes factor $\mathcal{B}_{01}$ with threshold value of one or
▶ comparing posterior probability $\mathbb{P}(\mathcal{M}_0|x)$ with bound

When comparing more than two models

▶ selecting model with highest posterior probability but highly dependent on the prior modeling, even with large databases
Interpretation of Bayes Factor

If \( \omega_0 = \omega_1 = 0.5 \), then

\[
B_{01} = \frac{m_0(x)}{m_1(x)}, \quad P(M_0|x) = \frac{B_{01}}{B_{01} + 1};
\]

Jeffreys gave a scale for interpretation of \( B_{01} \):

- \( B_{01} < 1 \), negative, \( P(M_0|x) < 0.5 \)
- \( 1 < B_{01} < 10^{1/2} \), barely worth mentioning, \( 0.5 < P(M_0|x) < 0.75 \)
- \( 10^{1/2} < B_{01} < 10 \), substantial, \( 0.75 < P(M_0|x) < 0.9 \)
- \( 10 < B_{01} < 10^{3/2} \), strong, \( 0.9 < P(M_0|x) < 0.96 \)
- \( 10^{3/2} < B_{01} < 100 \), very strong, \( 0.96 < P(M_0|x) < 0.99 \)
- \( B_{01} > 100 \), decisive, \( P(M_0|x) > 0.99 \)
Some difficulties with traditional handling of Bayesian tests

\[ B_{01} = \frac{m_0(x)}{m_1(x)}, \quad P(M_0|x) = \frac{\omega_0 B_{01}}{\omega_0 B_{01} + \omega_1}; \]

- subsequent and delicate interpretation (or calibration) of the strength towards supporting a given hypothesis or model, because it is not a Bayesian decision rule
- long-lasting impact of the prior modeling, despite overall consistency proof for Bayes factor
Some more difficulties

- discontinuity in use of improper priors since they are not justified in most testing situations, leading to many alternative
- binary (accept vs. reject) outcome more suited for immediate decision (if any) than for model evaluation, in connection with rudimentary loss function
- lack of assessment of uncertainty associated with decision itself
- difficult computation of marginal likelihoods in most settings with further controversies about which computational solution to adopt
Testing problems via mixture estimation model

A new paradigm for testing: Simple representation of the testing problem as a two-component mixture estimation problem where the weights are formally equal to 0 or 1
  ▶ provides a convergent and naturally interpretable solution,
  ▶ allowing for a more extended use of improper priors
Inspired from consistency result of Rousseau and Mengersen (2011) on estimated overfitting mixtures
  ▶ over-parameterised mixtures can be consistently estimated

[Kamary & Mengersen & Robert & Rousseau, 2014]
Testing problems via mixture estimation model

Given two statistical models,

\[ M_0 : x \sim \pi_0(x|\theta_0), \ \theta_0 \in \Theta_0 \quad \text{and} \quad M_1 : x \sim \pi_1(x|\theta_1), \ \theta_1 \in \Theta_1, \]

embed both within an encompassing mixture

\[ M_\alpha : x \sim \alpha \pi_0(x|\theta_0) + (1 - \alpha) \pi_1(x|\theta_1), \ 0 \leq \alpha \leq 1 \quad (1) \]

- Both models correspond to special cases of (1), one for \( \alpha = 1 \) and one for \( \alpha = 0 \)
- Draw inference on mixture representation (1), as if each observation was individually and independently produced by the mixture model

[Kamary & Mengersen & Robert & Rousseau, 2014]
Advantages

- relying on a Bayesian estimate of the weight $\alpha$ rather than on posterior probability of model $M_1$ does produce an equally convergent indicator of which model is “true”
- interpretation of estimator of $\alpha$ at least as natural as handling the posterior probability, while avoiding zero-one loss setting
- highly problematic computations of the marginal likelihoods is bypassed, since standard algorithms are available for Bayesian mixture estimation
- allows to consider all models at once rather than engaging in pairwise costly comparisons
More advantages

- posterior distribution of $\alpha$ evaluates more thoroughly strength of support for a given model than the single figure outcome of a posterior probability

- additional feature missing from traditional Bayesian answers: a mixture model acknowledges possibility that, for a finite dataset, both models or none could be acceptable

- non-informative (improper) priors are manageable in this setting, provided both models first reparameterised towards shared parameters, e.g. location and scale parameters

- in special case when all parameters are common

\[ M_\alpha : x \sim \alpha \pi_0(x|\theta) + (1 - \alpha) \pi_1(x|\theta), \ 0 \leq \alpha \leq 1 \]

if $\theta$ is a location parameter, a flat prior $\pi(\theta) \propto 1$ is available
Mixture estimation using latent variable

Consider sample $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ from (1). Completion by latent component indicators $\zeta_i$ leads to completed likelihood

$$(\theta, \alpha_0, \alpha_1 | \mathbf{x}, \zeta) = \prod_{i=1}^{n} \alpha_{\zeta_i} \pi_{\zeta_i} (x_i | \theta_{\zeta_i})$$

$$= \alpha^{n_1} (1 - \alpha)^{n_2} \prod_{i=1}^{n} \pi_{\zeta_i} (x_i | \theta_{\zeta_i}),$$

where

$$(n_1, n_2) = \left( \sum_{i=1}^{n} \mathbb{I}_{\zeta_i=0}, \sum_{i=1}^{n} \mathbb{I}_{\zeta_i=1} \right)$$

under constraint

$$n = \sum_{j=1}^{1} \sum_{i=1}^{n} \mathbb{I}_{\zeta_i=j}$$

[Diebolt & Robert, 1990]
Mixture estimation using latent variable

Using natural Gibbs implementation
- under a $\text{Beta}(a_1, a_2)$, $\alpha$ is generated from a Beta $\text{Beta}(a_1 + n_1, a_2 + n_2)$
- Gibbs sampling scheme is valid from a theoretical point of view
- convergence difficulties in the current setting, especially with large samples
- due to prior concentration on boundaries of $(0,1)$ for the mixture weight $\alpha$
Metropolis-Hastings algorithms

- model parameters $\theta_i$ generated from respective full posteriors of both models (i.e., based on entire sample)

$$\pi(\theta_i|x, \alpha) = (\alpha \pi_0(x|\theta_0) + (1 - \alpha)\pi_1(x|\theta_1)) \pi(\theta_i); \quad i = 0, 1$$

- mixture weight $\alpha$ generated from a random walk proposal on $(0, 1)$
Gibbs versus MH implementation

(Left) Gibbs; (Right) Metropolis–Hastings sequences \( (\alpha_t) \) when the mixture model is \( \alpha \mathcal{N}(\mu, 1) + (1 - \alpha)\mathcal{N}(0, 1) \) for a \( \mathcal{N}(0, 1) \) sample of size \( N = 5, 10, 50, 100, 500, 10^3 \) (from top to bottom) based on \( 10^5 \) simulations. The y-range range for all series is \((0, 1)\).
Illustrations

Computer model validation

Introduction and Motivations

Testing problems via mixture estimation model

Mixture estimation

Illustrations
  Simple example: Poisson versus Geometric
  Code validation via mixture model estimation

Conclusion and perspectives
Poisson or Geometric

- Models under comparison:
  \[ M_0 : \mathcal{P}(\lambda) \quad \text{and} \quad M_1 : Geo(p) \]

- if both models share a common parameter \( \lambda \),
  \[ M_\alpha : \alpha \mathcal{P}(\lambda) + (1 - \alpha)Geo(1/1+\lambda) \]

- prior modeling:
  \[ \pi(\lambda) = 1/\lambda; \quad \alpha \sim Be(a_0, a_0) \]

- posterior simulation:
  - independent Metropolis–within–Gibbs
Parameter estimation

Posterior means of $\lambda$ and medians of $\alpha$ for 100 Poisson $\mathcal{P}(4)$ datasets of size $n = 1000$, for $a_0 = .0001, .001, .01, .1, .2, .3, .4, .5$. Each posterior approximation is based on $10^4$ Metropolis-Hastings iterations.
MCMC convergence

Dataset from a Poisson distribution $\mathcal{P}(4)$: Estimations of (top) $\lambda$ and (bottom) $\alpha$ via MH for 5 samples of size $n = 5, 50, 100, 500, 10,000$. 
Consistency

Evolution of mixture weight over sample size: Posterior means (sky-blue) and medians (grey-dotted) of $\alpha$, over 100 Poisson $\mathcal{P}(4)$ datasets for sample sizes from 1 to 1000.
Suppose that

- \( X = (x_1 \ldots x_n)' \): input physical experiment matrix of size \( n \times d \)
- \( Y = (y_1, \ldots, y_n) \): vector of related available measurements of size \( n \)

then for \( i = 1, \ldots, n \), we want to choose a model between

\[ M_0 : y_i = g(x_i)\theta_0 + \epsilon_i^{(0)} \]
\[ M_1 : y_i = g(x_i)\theta_1 + \delta(x_i) + \epsilon_i^{(1)} \]

where \( \epsilon_i^{(0)} \sim \mathcal{N}(0, \lambda_0) \) and \( \epsilon_i^{(1)} \sim \mathcal{N}(0, \lambda_1) \).
Code validation via mixture model estimation

\[ M_0 : y_i = g(x_i)\theta_0 + \epsilon_i^{(0)} \]
\[ M_1 : y_i = g(x_i)\theta_1 + \delta(x_i) + \epsilon_i^{(1)}. \]

Remark

- Discrepancy-corrected prediction is traditionally based on the Gaussian processes
  \[ \text{[Damblin et al., 2016]} \]

- difficulty item from the dependence of the data which was not the case in the initial implementation of mixture technique.
  \[ \text{[Kamary et al., 2014]} \]

We therefore define the mixture of the competing models under the condition of considering \( \delta(x_i) \) as a latent variable to be estimated with the other model parameters.
Code validation via mixture model estimation

\[ M_0 : y_i = g(x_i)\theta_0 + \epsilon_i^{(0)} \]

\[ M_1 : y_i = g(x_i)\theta_1 + \delta(x_i) + \epsilon_i^{(1)}. \]

→ after embedding both models within an encompassing mixture model

\[
\ell_{M_{\alpha}}(\theta_0, \lambda_0, \theta_1, \lambda_1, \delta; Y, X) = \prod_{i=1}^{n} \left( \alpha \ell_{M_0}(\theta_0, \lambda_0; y_i, x_i) + (1 - \alpha) \ell_{M_1}(\theta_1, \lambda_1, \delta; y_i, x_i) \right).
\]

where

\[
\ell_{M_0}(\theta_0, \lambda_0; y_i, x_i) = \exp\left( -\frac{1}{2\lambda_0^2} (y_i - g(x_i)\theta_0)^2 \right) / (2\pi\lambda_0^2)^{n/2}
\]

\[
\ell_{M_1}(\theta_1, \lambda_1, \delta; y_i, x_i) = \exp\left( -\frac{1}{2\lambda_1^2} (y_i - g(x_i)\theta_1 - \delta(x_i))^2 \right) / (2\pi\lambda_1^2)^{n/2}
\]
In order to able the non-informative priors to be used for mixture parameters, suppose that the parameters $\theta$ and $\lambda$ are shared:

$$M_0 : y_i = g(x_i)\theta + \epsilon_i$$
$$M_1 : y_i = g(x_i)\theta + \delta(x_i) + \epsilon_i.$$

then $\ell_{M_\alpha}$ equals

$$\frac{1}{(2\pi\lambda^2)^{n/2}} \prod_{i=1}^{n} \left( \alpha \exp\left(-\frac{1}{2\lambda^2} (y_i - g(x_i)\theta)^2\right) + (1 - \alpha) \exp\left(-\frac{1}{2\lambda^2} (y_i - g(x_i)\theta - \delta(x_i))^2\right) \right)$$

and

$$\pi(\theta, \lambda) = \frac{1}{\lambda}$$
Parameter prior choice

\[ \ell_{\beta \gamma} \] equals

\[
\frac{1}{(2\pi \lambda^2)^{n/2}} \prod_{i=1}^{n} \left( \alpha \exp\left( -\frac{1}{2\lambda^2} (y_i - g(x_i, \theta))^2 \right) + (1 - \alpha) \exp\left( -\frac{1}{2\lambda^2} (y_i - g(x_i, \theta) - \delta(x_i))^2 \right) \right)
\]

(5)

The discrepancy \( \delta \) has a Gaussian process prior as

\[ \delta(X) \sim \mathcal{G}\mathcal{P}(\mu, \Sigma); \quad \Sigma = \sigma^2_\delta \text{Corr}_{\gamma_\delta}(x_i, x_{i'}) \]

where

\[ \text{Corr}_{\gamma_\delta}(x_i, x_{i'}) = \exp \left( -\frac{|x_i - x_{i'}|}{\gamma_\delta} \right). \]
Theorem

Let $g : \mathbb{R}^d \rightarrow \mathbb{R}^d; d > 1$ be a finite-valued function and for any vector $x_i$ of size $d$, $g(x_{ij}) \neq 0; j = 1, \ldots, d$. The posterior distribution associated with the prior $\pi(\theta, \lambda) = 1/\lambda$ and with the likelihood $\ell_{\mathcal{M}_\alpha}$ is proper when

- for any $0 < k < 1$, the hyperparameter $\sigma_\delta^2$ of the discrepancy prior distribution is reparameterized as $\sigma_\delta^2 = \lambda^2/k$ and so $\Sigma_\delta = (\lambda^2/k) \text{Corr}_{\gamma_\delta}$ when $\text{Corr}_{\gamma_\delta}$ is the correlation function of $\delta$.
- the dimensionality, $d$, is less than $n$;
- the mixture weight $\alpha$ has a proper beta prior $\mathcal{B}(a_0, a_0)$;
- $\gamma_\delta$ has a proper Beta prior $\mathcal{B}(b_1, b_2)$.
- proper distribution is used on $k$. 
Simulation studies

Simulated data: If \( x = \{i/n\}_{i=1}^{n} \),

- \( g(x) \) with a degree 2 polynomial code in \( x \) as \( g(x) = (1, x, x^2) \)
- parameter true values, \( \theta^* = (4, 1, 2)' \), \( \lambda^* = 0.1 \) and \( k^* = 0.1 \)

We simulate samples of size \( n \) by considering that the true value of the parameter \( \gamma_\delta \) varies between \((0, 1)\). Under the priors

- \( \pi(\theta, \lambda) = 1/\lambda \)
- \( \gamma_\delta \sim Beta(b_1, b_2) \)
- \( k \sim Beta(2, 18) \)
- \( \alpha \sim Beta(0.5, 0.5) \)

we estimate mixture model \( M_\alpha \) by implementing Metropolis-within-Gibbs algorithm.
Sensitivity of the correlation length to the prior choice

(Top) Beta prior distribution $\text{Beta}(b_1, b_2)$ with (left) $b_1 = b_2 = 1$; (middle) $b_1 = b_2 = 0.5$; (right) different curves are related to the $b_1$ and $b_2$ indicated on the plot. (Bottom) Empirical posterior distributions of $\gamma_\delta$ obtained for each prior illustrated on the top when four datasets of size $n = 50$ are simulated from $\mathcal{M}_1$ for the true value of $\gamma_\delta^* = 0.01, 0.1, 0.5, 0.9$. The number of MCMC iterations is $2 \times 10^4$ with a burn-in of $10^3$ iterations.
Sensitivity of $\delta$ to the choice of $\delta$ prior

For three datasets of size $n = 50$ simulated from $\mathcal{M}_1$ when the true value $\gamma^*_\delta$ is (left) 0.1; (right) 0.9: Comparison between the posterior distributions of $\delta(x_i); i = 1, \ldots, n$ and the true value (black points). For all plots, the results are shown in (skyblue color) when the prior of $\delta(x)$ is $\mathcal{GP}(1_n, \Sigma_\delta)$ and in (gray color) when $\delta(x) \sim \mathcal{GP}(0_n, \Sigma_\delta)$. The number of MCMC iterations is $2 \times 10^4$ with a burn-in of $10^3$ iterations.
Sensitivity of $\theta$ to the choice of $\delta$ prior

For three datasets of size $n = 50$ simulated from $\mathcal{M}_1$ when the true value $\gamma^*_\delta$ is (left) 0.1; (right) 0.9: Comparison between the posterior distributions of $\theta_1$ and the true value (dotted line). For all plots, the results are shown in (skyblue color) when the prior of $\delta(x)$ is $\mathcal{GP}(1_n, \Sigma_\delta)$ and in (gray color) when $\delta(x) \sim \mathcal{GP}(0_n, \Sigma_\delta)$. The number of MCMC iterations is $2 \times 10^4$ with a burn-in of $10^3$ iterations.
Sensitivity of $\alpha$ to the choice of $\delta$ prior

For three datasets of size $n = 10, 30, 50$ are simulated from $\mathcal{M}_1$ when the true value $\gamma^*_\delta$ is (left) 0.1; (right) 0.9 : Comparison between the posterior distributions of $\alpha$, the weight of $\mathcal{M}_0$ in the mixture model. For different sample sizes, the results are shown in (skyblue color) when the prior of $\delta(x)$ is $\sim \mathcal{GP}(1_n, \Sigma_\delta)$ and in (gray color) when $\delta(x) \sim \mathcal{GP}(0_n, \Sigma_\delta)$. The number of MCMC iterations is $2 \times 10^4$ with a burn-in of $10^3$ iterations.
Sensitivity of $\lambda$ to the choice of $\delta$ prior

For a sample of size 50 simulated from $\mathcal{M}_1$ when $\gamma_\delta^* = 0.1$ and $\delta^*(x) \sim \mathcal{GP}(1_n, \Sigma_\delta)$: (Top) Point process representation of the posterior draws of $\alpha$ versus $\lambda$, (bottom) Comparison between data points $y_i$ versus $x_i$ (black points), the posterior estimate of $\mathcal{M}_1$ obtained by averaging over MCMC iterations (dotted line) with overlaying box plots (gold) of the last 1000 posterior draws and the true code (solid line). Mixture model parameters have been estimated when the prior distribution of $\delta(x)$ is (Left) $\mathcal{GP}(0_n, \Sigma_\delta)$; (Right) $\mathcal{GP}(1_n, \Sigma_\delta)$. 
Conclusion

The code validation problem

- is considered as a Bayesian model selection
- resort to a technique developed by Kamary et al., 2014 that rely on the encompassing mixture model
- common parameterisation allows for reference priors
- posterior distribution of the parameters is highly subjected to the choice of the discrepancy prior
- posterior distribution of the mixture component weights is used to make a decision about the data
- $\alpha$ can capture the true model only under the condition that the $\delta$ prior is informative enough
Perspectives

- what about the real datasets?
- what if the computer model has a complex structure?
Thank you