Análise Estatística de Simuladores
Planejamento e múltiplos outputs

Leo Bastos[1] Richard Wilkinson[2]

[1]Departamento de Estatística

[2]Department of Statistics

19o SINAPE

Outline

1. Planejamento para experimentos computacionais
   - Introdução
   - Protocolo de planejamento
   - Space filling designs
   - Outros tópicos

2. Emuladores com múltiplos outputs
   - Introdução
   - Emuladores separáveis
   - Emuladores independentes
   - Exemplo

Outline

1. Planejamento para experimentos computacionais
   - Introdução
   - Protocolo de planejamento
   - Space filling designs
   - Outros tópicos

2. Emuladores com múltiplos outputs
   - Introdução
   - Emuladores separáveis
   - Emuladores independentes
   - Exemplo

Construíndo um emulador gaussiano

1. Escolha os inputs de treinamento, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, cobrindo o espaço de inputs*
2. “Rode” o simulador para obter o outputs treinamento.

$$D = \{y_1 = \eta(\mathbf{x}_1), \ldots, y_n = \eta(\mathbf{x}_n)\}$$

3. Usando os dados de treinamento estime os parâmetro de correlação, $\delta = \tilde{\delta}$
4. Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$
   - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
   - A disitribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.

Construíndo um emulador gaussiano

1. Escolha os inputs de treinamento, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, cobrindo o espaço de inputs*
2. “Rode” o simulador para obter o outputs treinamento.

$$D = \{y_1 = \eta(\mathbf{x}_1), \ldots, y_n = \eta(\mathbf{x}_n)\}$$

3. Usando os dados de treinamento estime os parâmetro de correlação, $\delta = \tilde{\delta}$
4. Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$
    - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
    - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.

Construíndo um emulador gaussiano

1. Escolha os inputs de treinamento, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, cobrindo o espaço de inputs*
2. "Rode" o simulador para obter o outputs treinamento.

$$D = \{y_1 = \eta(\mathbf{x}_1), \ldots, y_n = \eta(\mathbf{x}_n)\}$$

3. Usando os dados de treinamento estime os parâmetro de correlação, $\delta = \tilde{\delta}$
4. Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$
   - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
   - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.

Construíndo um emulador gaussiano

1. Escolha os inputs de treinamento, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, cobrindo o espaço de inputs*
2. "Rode" o simulador para obter o outputs treinamento.

$$D = \{y_1 = \eta(\mathbf{x}_1), \ldots, y_n = \eta(\mathbf{x}_n)\}$$

3. Usando os dados de treinamento estime os parâmetro de correlação, $\delta = \tilde{\delta}$
4. Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$
   - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
   - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.

Construíndo um emulador gaussiano

1. Escolha os inputs de treinamento, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, cobrindo o espaço de inputs*
2. "Rode" o simulador para obter o outputs treinamento.

$$D = \{y_1 = \eta(\mathbf{x}_1), \ldots, y_n = \eta(\mathbf{x}_n)\}$$

3. Usando os dados de treinamento estime os parâmetro de correlação, $\delta = \tilde{\delta}$
4. Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$
   - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
   - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.

Construíndo um emulador gaussiano

1. Escolha os inputs de treinamento, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, cobrindo o espaço de inputs*
2. "Rode" o simulador para obter o outputs treinamento.

$$D = \{y_1 = \eta(\mathbf{x}_1), \ldots, y_n = \eta(\mathbf{x}_n)\}$$

3. Usando os dados de treinamento estime os parâmetro de correlação, $\delta = \tilde{\delta}$
4. Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$
   - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
   - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.

O Problema

- Em qualquer estudo científico o planejamento deve ter um papel essencial.
- Os planejamentos em experimentos surgiram (ou ganharam força) a partir dos experimentos agrícolas devidamente planejados por R A Fisher na decada de 1920.
- Aqui, iremos discutir métodos de planejamento de experimentos computacionais com o próposito de construir um emulador.
- Para quais valores do espaço de inputs $\mathcal{X} \subset \mathbb{R}^p$ devemos rodar o simulador para construírmos um bom emulador?

O Problema

- Em qualquer estudo científico o planejamento deve ter um papel essencial.
- Os planejamentos em experimentos surgiram (ou ganharam força) a partir dos experimentos agrícolas devidamente planejados por R A Fisher na decada de 1920.
- Aqui, iremos discutir métodos de planejamento de experimentos computacionais com o próposito de construir um emulador.
- Para quais valores do espaço de inputs $\mathcal{X} \subset \mathbb{R}^p$ devemos rodar o simulador para construírmos um bom emulador?

O Problema

- Em qualquer estudo científico o planejamento deve ter um papel essencial.
- Os planejamentos em experimentos surgiram (ou ganharam força) a partir dos experimentos agrícolas devidamente planejados por R A Fisher na decada de 1920.
- Aqui, iremos discutir métodos de planejamento de experimentos computacionais com o próposito de construir um emulador.
- Para quais valores do espaço de inputs $\mathcal{X} \subset \mathbb{R}^p$ devemos rodar o simulador para construírmos um bom emulador?

O Problema

- Em qualquer estudo científico o planejamento deve ter um papel essencial.
- Os planejamentos em experimentos surgiram (ou ganharam força) a partir dos experimentos agrícolas devidamente planejados por R A Fisher na decada de 1920.
- Aqui, iremos discutir métodos de planejamento de experimentos computacionais com o próposito de construir um emulador.
- Para quais valores do espaço de inputs $\mathcal{X} \subset \mathbb{R}^p$ devemos rodar o simulador para construírmos um bom emulador?

Alguns pontos importantes para planejar um experimentos computacional

1 Definir o objetivo do experimento. Existem vários:
   1. Compreender a relação input-output;
   2. Encontrar os fatores explicativos mais importantes
   3. Encontrar os inputs que geram um output em alguma região particular. (Problema inverso)
   4. Encontrar o vetor inputs que maximiza o output, e qual o valor desse máximo.

2 Váriaveis de input e output: Uma descrição completa é essencial
   1. para cada variável a deve se fazer um "CV completo"

Alguns pontos importantes para planejar um experimentos computacional

1 Definir o objetivo do experimento. Existem vários:
   1. Compreender a relação input-output;
   2. Encontrar os fatores explicativos mais importantes
   3. Encontrar os inputs que geram um output em alguma região particular. (Problema inverso)
   4. Encontrar o vetor inputs que maximiza o output, e qual o valor desse máximo.

2 Váriaveis de input e output: Uma descrição completa é essencial

Alguns pontos importantes para planejar um experimentos computacional

1 Definir o objetivo do experimento. Existem vários:
   1. Compreender a relação input-output;
   2. Encontrar os fatores explicativos mais importantes
   3. Encontrar os inputs que geram um output em alguma região particular. (Problema inverso)
   4. Encontrar o vetor inputs que maximiza o output, e qual o valor desse máximo.

2 Váriaveis de input e output: Uma descrição completa é essencial
   1. para cada variável a deve se fazer um "CV completo"

Alguns pontos importantes para planejar um experimentos computacional

1 Definir o objetivo do experimento. Existem vários:
   1. Compreender a relação input-output;
   2. Encontrar os fatores explicativos mais importantes
   3. Encontrar os inputs que geram um output em alguma região particular. (Problema inverso)
   4. Encontrar o vetor inputs que maximiza o output, e qual o valor desse máximo.

2 Váriaveis de input e output: Uma descrição completa é essencial
   1. para cada variável a deve se fazer um "CV completo"

Alguns pontos importantes para planejar um experimentos computacional

1 Definir o objetivo do experimento. Existem vários:
   1. Compreender a relação input-output;
   2. Encontrar os fatores explicativos mais importantes
   3. Encontrar os inputs que geram um output em alguma região particular. (Problema inverso)
   4. Encontrar o vetor inputs que maximiza o output, e qual o valor desse máximo.

2 Váriaveis de input e output: Uma descrição completa é essencial
   1. para cada variável a deve se fazer um "CV completo"

Alguns pontos importantes para planejar um experimentos computacional

1 Definir o objetivo do experimento. Existem vários:
   1. Compreender a relação input-output;
   2. Encontrar os fatores explicativos mais importantes
   3. Encontrar os inputs que geram um output em alguma região particular. (Problema inverso)
   4. Encontrar o vetor inputs que maximiza o output, e qual o valor desse máximo.

2 Váriaveis de input e output: Uma descrição completa é essencial
   1. para cada variável a deve se fazer um “CV completo”

Alguns pontos importantes para planejar um experimentos computacional

1. Definir o objetivo do experimento. Existem vários:
    1. Compreender a relação input-output;
    2. Encontrar os fatores explicativos mais importantes
    3. Encontrar os inputs que geram um output em alguma região particular. (Problema inverso)
    4. Encontrar o vetor inputs que maximiza o output, e qual o valor desse máximo.
2. Váriaveis de input e output: Uma descrição completa é essencial
    1. para cada variável a deve se fazer um "CV completo"

Alguns pontos importantes para planejar um experimentos computacional

3 Um modelo inicial

1. O que nós já sabemos da relação input-output?
2. Existe alguma ordem de importância dos inputs?
3. Existe alguma região do espaço de inputs que a output varia bastante?

4 Custo do experimento

1. Tempo de execução
2. Custo

Alguns pontos importantes para planejar um experimentos computacional

3 Um modelo inicial

1. O que nós já sabemos da relação input-output?
2. Existe alguma ordem de importância dos inputs?
3. Existe alguma região do espaço de inputs que a output varia bastante?

4 Custo do experimento

1. Tempo de execução
2. Custo

Alguns pontos importantes para planejar um experimentos computacional

3. Um modelo inicial
    1. O que nós já sabemos da relação input-output?
    2. Existe alguma ordem de importância dos inputs?
    3. Existe alguma região do espaço de inputs que a output varia bastante?
4. Custo do experimento
    1. Tempo de execução
    2. Custo

Alguns pontos importantes para planejar um experimentos computacional

3 Um modelo inicial

1. O que nós já sabemos da relação input-output?
2. Existe alguma ordem de importância dos inputs?
3. Existe alguma região do espaço de inputs que a output varia bastante?

4 Custo do experimento

1. Tempo de execução
2. Custo

Alguns pontos importantes para planejar um experimentos computacional

3. Um modelo inicial
   1. O que nós já sabemos da relação input-output?
   2. Existe alguma ordem de importância dos inputs?
   3. Existe alguma região do espaço de inputs que a output varia bastante?
4. Custo do experimento
   1. Tempo de execução
   2. Custo

Alguns pontos importantes para planejar um experimentos computacional

3. Um modelo inicial
   1. O que nós já sabemos da relação input-output?
   2. Existe alguma ordem de importância dos inputs?
   3. Existe alguma região do espaço de inputs que a output varia bastante?
4. Custo do experimento
   1. Tempo de execução
   2. Custo

Alguns pontos importantes para planejar um experimentos computacional

3. Um modelo inicial
   1. O que nós já sabemos da relação input-output?
   2. Existe alguma ordem de importância dos inputs?
   3. Existe alguma região do espaço de inputs que a output varia bastante?
4. Custo do experimento
   1. Tempo de execução
   2. Custo

Um protocolo de planejamento em quatro estágios

1. Experimento nominal
2. Experimento para fazer redução de variáveis (screening)
3. **Experimento principal**, que vai usar
   - conhecimento a priori dos possíveis modelos
   - inputs relevantes?
4. Experimento confirmatório (Experimento para validação)

Um protocolo de planejamento em quatro estágios

1. Experimento nominal
2. Experimento para fazer redução de variáveis (screening)
3. **Experimento principal**, que vai usar
   - conhecimento a priori dos possíveis modelos
   - inputs relevantes?
4. Experimento confirmatório (Experimento para validação)

Um protocolo de planejamento em quatro estágios

1. Experimento nominal
2. Experimento para fazer redução de variáveis (screening)
3. **Experimento principal**, que vai usar
    - conhecimento a priori dos possíveis modelos
    - inputs relevantes*
4. Experimento confirmatório (Experimento para validação)

Um protocolo de planejamento em quatro estágios

1. Experimento nominal
2. Experimento para fazer redução de variáveis (screening)
3. **Experimento principal**, que vai usar
    - conhecimento a priori dos possíveis modelos
    - inputs relevantes*
4. Experimento confirmatório (Experimento para validação)

Um protocolo de planejamento em quatro estágios

1. Experimento nominal
2. Experimento para fazer redução de variáveis (screening)
3. **Experimento principal**, que vai usar
    - conhecimento a priori dos possíveis modelos
    - inputs relevantes*
4. Experimento confirmatório (Experimento para validação)

Um protocolo de planejamento em quatro estágios

1. Experimento nominal
2. Experimento para fazer redução de variáveis (screening)
3. **Experimento principal**, que vai usar
   - conhecimento a priori dos possíveis modelos
   - inputs relevantes*
4. Experimento confirmatório (Experimento para validação)

Experimento principal

- Vamos considerar o experimento principal para construir um emulador
- Chamamos de dados de treinamento os inputs selecionados, e seus respectivos outputs, para construir o emulador.
- O planejamento depende do objetivo do uso do emulador
- Um bom planejamento vai nos permitir construir um bom emulador, devemos portanto levar em consideração:
  - Incerteza a respeito dos hiperparâmetros $(\beta, \sigma^2, \delta)$
  - Incerteza a respeito do output do simulador (expressada por um processo gaussiano)
- Focaremos em planejamentos que lidam basicamente com a segunda fonte de incerteza.

Experimento principal

- Vamos considerar o experimento principal para construir um emulador
- Chamamos de dados de treinamento os inputs selecionados, e seus respectivos outputs, para construir o emulador.
- O planejamento depende do objetivo do uso do emulador
- Um bom planejamento vai nos permitir construir um bom emulador, devemos portanto levar em consideração:
  1. Incerteza a respeito dos hiperparâmetros $(\beta, \sigma^2, \delta)$
  2. Incerteza a respeito do output do simulador (expressada por um processo gaussiano)
- Focaremos em planejamentos que lidam basicamente com a segunda fonte de incerteza.

Experimento principal

- Vamos considerar o experimento principal para construir um emulador
- Chamamos de dados de treinamento os inputs selecionados, e seus respectivos outputs, para construir o emulador.
- O planejamento depende do objetivo do uso do emulador
- Um bom planejamento vai nos permitir construir um bom emulador, devemos portanto levar em consideração:
    1. Incerteza a respeito dos hiperparâmetros $(\beta, \sigma^2, \delta)$
    2. Incerteza a respeito do output do simulador (expressada por um processo gaussiano)
- Focaremos em planejamentos que lidam basicamente com a segunda fonte de incerteza.

Experimento principal

- Vamos considerar o experimento principal para construir um emulador
- Chamamos de dados de treinamento os inputs selecionados, e seus respectivos outputs, para construir o emulador.
- O planejamento depende do objetivo do uso do emulador
- Um bom planejamento vai nos permitir construir um bom emulador, devemos portanto levar em consideração:
    1. Incerteza a respeito dos hiperparâmetros $(\beta, \sigma^2, \delta)$
    2. Incerteza a respeito do output do simulador (expressada por um processo gaussiano)
- Focaremos em planejamentos que lidam basicamente com a segunda fonte de incerteza.

Experimento principal

- Vamos considerar o experimento principal para construir um emulador
- Chamamos de dados de treinamento os inputs selecionados, e seus respectivos outputs, para construir o emulador.
- O planejamento depende do objetivo do uso do emulador
- Um bom planejamento vai nos permitir construir um bom emulador, devemos portanto levar em consideração:
  1. Incerteza a respeito dos hiperparâmetros $(\beta, \sigma^2, \delta)$
  2. Incerteza a respeito do output do simulador (expressada por um processo gaussiano)
- Focaremos em planejamentos que lidam basicamente com a segunda fonte de incerteza.

Experimento principal

- Vamos considerar o experimento principal para construir um emulador
- Chamamos de dados de treinamento os inputs selecionados, e seus respectivos outputs, para construir o emulador.
- O planejamento depende do objetivo do uso do emulador
- Um bom planejamento vai nos permitir construir um bom emulador, devemos portanto levar em consideração:
  1. Incerteza a respeito dos hiperparâmetros $(\beta, \sigma^2, \delta)$
  2. Incerteza a respeito do output do simulador (expressada por um processo gaussiano)
- Focaremos em planejamentos que lidam basicamente com a segunda fonte de incerteza.

Experimento principal

- Vamos considerar o experimento principal para construir um emulador
- Chamamos de dados de treinamento os inputs selecionados, e seus respectivos outputs, para construir o emulador.
- O planejamento depende do objetivo do uso do emulador
- Um bom planejamento vai nos permitir construir um bom emulador, devemos portanto levar em consideração:
  1. Incerteza a respeito dos hiperparâmetros $(\beta, \sigma^2, \delta)$
  2. Incerteza a respeito do output do simulador (expressada por um processo gaussiano)
- Focaremos em planejamentos que lidam basicamente com a segunda fonte de incerteza.

Space filling designs

- Observações de inputs próximas tendem a ter outputs próximos
- Portanto, temos interesse em observar inputs bem distantes uns dos outros. Chamamos essa propriedade de *Space-filling*
- Alguns *space filling designs*
  - Hipercubos latinos (McKay et al., 1979)
  - Planejamentos não aleatórios (usados em integração númerica)
  - Planejamentos ótimos, e.g. ASCM (*Adaptive Sampler for Complex Models*) que faz uso da expansão de Karhunen-Loeve como aproximação de um gaussian process. (Youssef, 2010)

Space filling designs

- Observações de inputs próximas tendem a ter outputs próximos
- Portanto, temos interesse em observar inputs bem distantes uns dos outros. Chamamos essa propriedade de *Space-filling*
- Alguns *space filling designs*
  - **Hipercubos latinos** (McKay et al., 1979)

Space filling designs

- Observações de inputs próximas tendem a ter outputs próximos
- Portanto, temos interesse em observar inputs bem distantes uns dos outros. Chamamos essa propriedade de *Space-filling*
- Alguns *space filling designs*
  - **Hipercubos latinos** (McKay et al., 1979)
  - **Planejamentos não aleatórios** (usados em integração númerica)
  - **Planejamentos ótimos**, e.g. ASCM (*Adaptive Sampler for Complex Models)* que faz uso da expansão de Karhunen-Loeve como aproximação de um gaussian process. (Youssef, 2010)

Space filling designs

- Observações de inputs próximas tendem a ter outputs próximos
- Portanto, temos interesse em observar inputs bem distantes uns dos outros. Chamamos essa propriedade de *Space-filling*
- Alguns *space filling designs*
  - **Hipercubos latinos** (McKay et al., 1979)
  - **Planejamentos não aleatórios** (usados em integração númerica)
    - Lattice designs (Bates et al. 1996, 1998; em exp. comp.)
    - Sequencias de Wey, Halton e Sobol (o pacote "fOptions" do R gera sequencias de Sobol (Neiderreiter, 1992)
  - **Planejamentos ótimos**, e.g. ASCM (*Adaptive Sampler for Complex Models)* que faz uso da expansão de Karhunen-Loeve como aproximação de um gaussian process. (Youssef, 2010)

Space filling designs

- Observações de inputs próximas tendem a ter outputs próximos
- Portanto, temos interesse em observar inputs bem distantes uns dos outros. Chamamos essa propriedade de *Space-filling*
- Alguns *space filling designs*
  - **Hipercubos latinos** (McKay et al., 1979)
  - **Planejamentos não aleatórios** (usados em integração númerica)
    - Lattice designs (Bates et al. 1996, 1998; em exp. comp.)
    - Sequencias de Wey, Halton e **Sobol** (a pacote '*fOptions*' do R gera sequencias de Sobol (Neiderreiter, 1992)
  - **Planejamentos ótimos**, e.g. ASCM (*Adaptive Sampler for Complex Models)* que faz uso da expansão de Karhunen-Loeve como aproximação de um gaussian process. (Youssef, 2010)

Space filling designs

- Observações de inputs próximas tendem a ter outputs próximos
- Portanto, temos interesse em observar inputs bem distantes uns dos outros. Chamamos essa propriedade de *Space-filling*
- Alguns *space filling designs*
  - **Hipercubos latinos** (McKay et al., 1979)
  - **Planejamentos não aleatórios** (usados em integração númerica)
    - Lattice designs (Bates et al. 1996, 1998; em exp. comp.)
    - Sequencias de Wey, Halton e **Sobol** (a pacote '*fOptions*' do R gera sequencias de Sobol (Neiderreiter, 1992)
  - **Planejamentos ótimos**, e.g. ASCM (*Adaptive Sampler for Complex Models)* que faz uso da expansão de Karhunen-Loeve como aproximação de um gaussian process. (Youssef, 2010)

Space filling designs

- Observações de inputs próximas tendem a ter outputs próximos
- Portanto, temos interesse em observar inputs bem distantes uns dos outros. Chamamos essa propriedade de *Space-filling*
- Alguns *space filling designs*
  - **Hipercubos latinos** (McKay et al., 1979)
  - **Planejamentos não aleatórios** (usados em integração númerica)
    - Lattice designs (Bates et al. 1996, 1998; em exp. comp.)
    - Sequencias de Wey, Halton e **Sobol** (a pacote '*fOptions*' do R gera sequencias de Sobol (Neiderreiter, 1992)
  - **Planejamentos ótimos**, e.g. ASCM (*Adaptive Sampler for Complex Models)* que faz uso da expansão de Karhunen-Loeve como aproximação de um gaussian process. (Youssef, 2010)

Space filling designs

- Observações de inputs próximas tendem a ter outputs próximos
- Portanto, temos interesse em observar inputs bem distantes uns dos outros. Chamamos essa propriedade de *Space-filling*
- Alguns *space filling designs*
  - **Hipercubos latinos** (McKay et al., 1979)
  - **Planejamentos não aleatórios** (usados em integração númerica)
    - Lattice designs (Bates et al. 1996, 1998; em exp. comp.)
    - Sequencias de Wey, Halton e **Sobol** (a pacote '*fOptions*' do R gera sequencias de Sobol (Neiderreiter, 1992)
  - **Planejamentos ótimos**, e.g. ASCM (*Adaptive Sampler for Complex Models)* que faz uso da expansão de Karhunen-Loeve como aproximação de um gaussian process. (Youssef, 2010)

Hipercubos Latinos

- Os hipercubos latinos têm a propriedade de representar bem todos os inputs marginalmente.
- ...mas os hipercubos latinos não têm boas propriedades de preenchimento de espaço
- Solução hipercubos latinos ótimos, e.g. *Maximin Latin Hypercube* (Morris and Mitchell 1997)

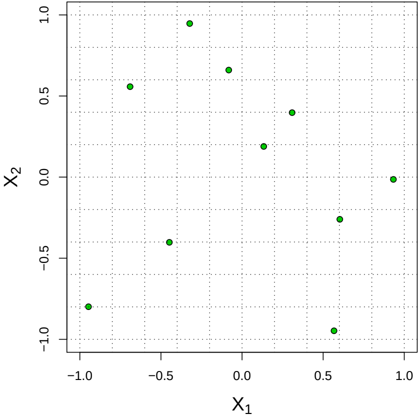
Hipercubos Latinos

- Os hipercubos latinos têm a propriedade de representar bem todos os inputs marginalmente.
- ...mas os hipercubos latinos não têm boas propriedades de preenchimento de espaço
- Solução hipercubos latinos ótimos, e.g. *Maximin Latin Hypercube* (Morris and Mitchell 1997)

Hipercubos Latinos

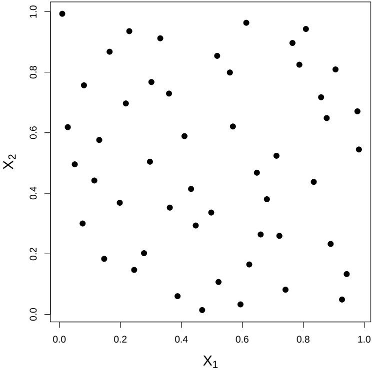
- Os hipercubos latinos têm a propriedade de representar bem todos os inputs marginalmente.
- ...mas os hipercubos latinos não têm boas propriedades de preenchimento de espaço
- Solução hipercubos latinos ótimos, e.g. *Maximin Latin Hypercube* (Morris and Mitchell 1997)

Hipercubos Latinos

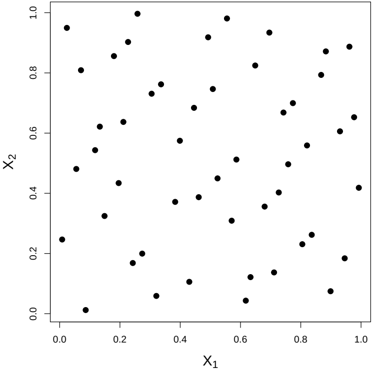


Hipercubo latino $n = 10$

Hipercubos Latinos



Maximin hipercubo latino $n = 50$



Sequência de Sobol $n = 50$

Outros tópicos em planejamento

- Planejamento para estimar os parâmetros de correlação δ
- Planejamentos sequênciais, onde devemos rodar o simulador para melhorar nossas previsões?
- Qual o tamanho da amostra? Receita de bolo $n = 10p$.

Outros tópicos em planejamento

- Planejamento para estimar os parâmetros de correlação δ
- Planejamentos sequênciais, onde devemos rodar o simulador para melhorar nossas previsões?
- Qual o tamanho da amostra? Receita de bolo $n = 10p$.

Outros tópicos em planejamento

- Planejamento para estimar os parâmetros de correlação δ
- Planejamentos sequênciais, onde devemos rodar o simulador para melhorar nossas previsões?
- Qual o tamanho da amostra? Receita de bolo $n = 10p$.

Outline

1. Planejamento para experimentos computacionais
   - Introdução
   - Protocolo de planejamento
   - Space filling designs
   - Outros tópicos

2. Emuladores com múltiplos outputs
   - Introdução
   - Emuladores separáveis
   - Emuladores independentes
   - Exemplo

Introdução

- É muito comum que os simuladores tenha não apenas um output, mas vários outputs.

$$(x_1, x_2, \ldots, x_p) \longrightarrow \underset{\{\text{SIMULADOR}\}}{\eta(\cdot)} \longrightarrow (y_1, y_2, \ldots, y_r)$$

- Ainda assim podemos emula-los, mas a complexidade aumenta.
- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot,\cdot))$$

onde $m(\mathbf{x})$ é um vetor de tamanho r e $V(\mathbf{x}, \mathbf{x}')$ é uma matrix $r \times r$.

- Estruturas para a função de variância e covariâncias:
  - Estrutura Separável: $V(\cdot,\cdot) = \Sigma C(\cdot,\cdot)$
  - Estrutura não separável (Fricker, 2010)

Introdução

- É muito comum que os simuladores tenha não apenas um output, mas vários outputs.

$$(x_1, x_2, \ldots, x_p) \longrightarrow \underset{\{\text{SIMULADOR}\}}{\eta(\cdot)} \longrightarrow (y_1, y_2, \ldots, y_r)$$

- Ainda assim podemos emula-los, mas a complexidade aumenta.
- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot, \cdot))$$

  onde $m(\mathbf{x})$ é um vetor de tamanho r e $V(\mathbf{x}, \mathbf{x}')$ é uma matrix $r \times r$.
- Estruturas para a função de variância e covariâncias:
  - Estrutura Separável: $V(\cdot, \cdot) = \Sigma C(\cdot, \cdot)$
  - Estrutura não separável (Fricker, 2010)

Introdução

- É muito comum que os simuladores tenha não apenas um output, mas vários outputs.

$$(x_1, x_2, \ldots, x_p) \longrightarrow \underset{\{\text{SIMULADOR}\}}{\eta(\cdot)} \longrightarrow (y_1, y_2, \ldots, y_r)$$

- Ainda assim podemos emula-los, mas a complexidade aumenta.
- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot, \cdot))$$

onde $m(\mathbf{x})$ é um vetor de tamanho r e $V(\mathbf{x}, \mathbf{x}')$ é uma matrix $r \times r$.

- Estruturas para a função de variância e covariâncias:
    - Estrutura Separável: $V(\cdot, \cdot) = \Sigma C(\cdot, \cdot)$
    - Estrutura não separável (Fricker, 2010)

- É muito comum que os simuladores tenha não apenas um output, mas vários outputs.

$$(x_1, x_2, \ldots, x_p) \longrightarrow \underset{\{\text{SIMULADOR}\}}{\eta(\cdot)} \longrightarrow (y_1, y_2, \ldots, y_r)$$

- Ainda assim podemos emula-los, mas a complexidade aumenta.
- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot, \cdot))$$

onde $m(\mathbf{x})$ é um vetor de tamanho r e $V(\mathbf{x}, \mathbf{x}')$ é uma matrix $r \times r$.

- Estruturas para a função de variância e covariâncias:
  - Estrutura Separável: $V(\cdot, \cdot) = \Sigma C(\cdot, \cdot)$
  - Estrutura não separável (Fricker, 2010)
    - CONV: $V_{ij}(x, x') = \sigma_{ij} \int_{\mathbb{R}^p} \kappa_i(u - x)\kappa_j(u - x')\, du.$
    - LMC: $V(\cdot, \cdot) = \sum_{l=1} \Sigma_l \kappa_l(\cdot, \cdot)$

Introdução

- É muito comum que os simuladores tenha não apenas um output, mas vários outputs.

$$(x_1, x_2, \ldots, x_p) \longrightarrow \underset{\text{\{SIMULADOR\}}}{\eta(\cdot)} \longrightarrow (y_1, y_2, \ldots, y_r)$$

- Ainda assim podemos emula-los, mas a complexidade aumenta.
- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot, \cdot))$$

  onde $m(\mathbf{x})$ é um vetor de tamanho r e $V(\mathbf{x}, \mathbf{x}')$ é uma matrix $r \times r$.
- Estruturas para a função de variância e covariâncias:
  - Estrutura Separável: $V(\cdot, \cdot) = \Sigma C(\cdot, \cdot)$
  - Estrutura não separável (Fricker, 2010)
    - CONV: $V_{ij}(\mathbf{x}, \mathbf{x}') = \tilde{\sigma}_{ij} \int_{\mathbb{R}^p} \kappa_i(u - \mathbf{x})\kappa_j(u - \mathbf{x}')\, du$.
    - LMC: $V(\cdot, \cdot) = \sum_{i=1}^{r} \Sigma_i \rho_i(\cdot, \cdot)$.

Introdução

- É muito comum que os simuladores tenha não apenas um output, mas vários outputs.

$$(x_1, x_2, \ldots, x_p) \longrightarrow \underset{\{\text{SIMULADOR}\}}{\eta(\cdot)} \longrightarrow (y_1, y_2, \ldots, y_r)$$

- Ainda assim podemos emula-los, mas a complexidade aumenta.
- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot, \cdot))$$

  onde $m(\mathbf{x})$ é um vetor de tamanho r e $V(\mathbf{x}, \mathbf{x}')$ é uma matrix $r \times r$.
- Estruturas para a função de variância e covariâncias:
  - Estrutura Separável: $V(\cdot, \cdot) = \Sigma C(\cdot, \cdot)$
  - Estrutura não separável (Fricker, 2010)
    - CONV: $V_{ij}(\mathbf{x}, \mathbf{x}') = \tilde{\sigma}_{ij} \int_{\mathbb{R}^p} \kappa_i(u - \mathbf{x})\kappa_j(u - \mathbf{x}')\, du$.
    - LMC: $V(\cdot, \cdot) = \sum_{i=1}^{l} \Sigma_i \kappa_i(\cdot, \cdot)$.

Introdução

- É muito comum que os simuladores tenha não apenas um output, mas vários outputs.

$$(x_1, x_2, \ldots, x_p) \longrightarrow \underset{\{\text{SIMULADOR}\}}{\eta(\cdot)} \longrightarrow (y_1, y_2, \ldots, y_r)$$

- Ainda assim podemos emula-los, mas a complexidade aumenta.
- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot, \cdot))$$

  onde $m(\mathbf{x})$ é um vetor de tamanho r e $V(\mathbf{x}, \mathbf{x}')$ é uma matrix $r \times r$.
- Estruturas para a função de variância e covariâncias:
  - Estrutura Separável: $V(\cdot, \cdot) = \Sigma C(\cdot, \cdot)$
  - Estrutura não separável (Fricker, 2010)
    - CONV: $V_{ij}(\mathbf{x}, \mathbf{x}') = \tilde{\sigma}_{ij} \int_{\mathbb{R}^p} \kappa_i(u - \mathbf{x})\kappa_j(u - \mathbf{x}')\, du$.
    - LMC: $V(\cdot, \cdot) = \sum_{i=1}^{l} \Sigma_i \kappa_i(\cdot, \cdot)$.

Introdução

- É muito comum que os simuladores tenha não apenas um output, mas vários outputs.

$$(x_1, x_2, \ldots, x_p) \longrightarrow \underset{\{\text{SIMULADOR}\}}{\eta(\cdot)} \longrightarrow (y_1, y_2, \ldots, y_r)$$

- Ainda assim podemos emula-los, mas a complexidade aumenta.
- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot,\cdot))$$

  onde $m(\mathbf{x})$ é um vetor de tamanho r e $V(\mathbf{x}, \mathbf{x}')$ é uma matrix $r \times r$.
- Estruturas para a função de variância e covariâncias:
  - Estrutura Separável: $V(\cdot,\cdot) = \Sigma C(\cdot,\cdot)$
  - Estrutura não separável (Fricker, 2010)
    - CONV: $V_{ij}(\mathbf{x}, \mathbf{x}') = \tilde{\sigma}_{ij} \int_{\mathbb{R}^p} \kappa_i(u - \mathbf{x})\kappa_j(u - \mathbf{x}')\, du$.
    - LMC: $V(\cdot,\cdot) = \sum_{i=1}^{l} \Sigma_i \kappa_i(\cdot,\cdot)$.

Emuladores com covariância separável

- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot,\cdot))$$

- A função de covariância

$$V(\cdot,\cdot) = \Sigma C_\delta(\cdot,\cdot),$$

  onde Σ é uma matriz de covariância $r \times r$ entre outputs e $C_\delta(\cdot,\cdot)$ é uma função de correlação entre inputs.
  Problema: $C_\delta(\cdot,\cdot)$ é a mesma para todos outputs!
- Os hiperpâmetros são (Σ, δ), e uma priori tipica para (Σ, δ) é

$$\pi(\Sigma, \delta) \propto \pi(\delta)|\Sigma|^{\frac{k+1}{2}}.$$

- A atualização é análoga ao caso univariado

Emuladores com covariância separável

- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot, \cdot))$$

- A função de covariância

$$V(\cdot, \cdot) = \Sigma C_\delta(\cdot, \cdot),$$

  onde Σ é uma matriz de covariância $r \times r$ entre outputs e $C_\delta(\cdot, \cdot)$ é uma função de correlação entre inputs.
  Problema: $C_\delta(\cdot, \cdot)$ é a mesma para todos outputs!
- Os hiperpâmetros são (Σ, δ), e uma priori tipica para (Σ, δ) é

$$\pi(\Sigma, \delta) \propto \pi(\delta)|\Sigma|^{\frac{k+1}{2}}.$$

- A atualização é análoga ao caso univariado

Emuladores com covariância separável

- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot, \cdot))$$

- A função de covariância

$$V(\cdot, \cdot) = \Sigma C_{\delta}(\cdot, \cdot),$$

  onde Σ é uma matriz de covariância $r \times r$ entre outputs e $C_{\delta}(\cdot, \cdot)$ é uma função de correlação entre inputs.
  Problema: $C_{\delta}(\cdot, \cdot)$ é a mesma para todos outputs!
- Os hiperpâmetros são (Σ, δ), e uma priori tipica para (Σ, δ) é

$$\pi(\Sigma, \delta) \propto \pi(\delta)|\Sigma|^{\frac{k+1}{2}}.$$

- A atualização é análoga ao caso univariado

Emuladores com covariância separável

- O emulador gaussiano a priori para $\eta(\cdot)$ é dado por

$$\eta(\cdot) \sim PG(m(\cdot), V(\cdot, \cdot))$$

- A função de covariância

$$V(\cdot, \cdot) = \Sigma C_\delta(\cdot, \cdot),$$

  onde Σ é uma matriz de covariância $r \times r$ entre outputs e $C_\delta(\cdot, \cdot)$ é uma função de correlação entre inputs.
  Problema: $C_\delta(\cdot, \cdot)$ é a mesma para todos outputs!
- Os hiperpâmetros são (Σ, δ), e uma priori tipica para (Σ, δ) é

$$\pi(\Sigma, \delta) \propto \pi(\delta)|\Sigma|^{\frac{k+1}{2}}.$$

- A atualização é análoga ao caso univariado

Atualizando um processo Gaussiano

- Seja $\eta(\cdot) \sim PG\,(m(\cdot), V(\cdot,\cdot))$
- $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ e $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$ inputs e outputs de um experimento computacional.
- Usando propriedades da Normal multivariada é fácil mostrar que

$$\eta(\cdot)|\mathbf{Y}, \mathbf{X} \sim PG(m^*(\cdot), V^*(\cdot,\cdot))$$

onde

$$\begin{aligned} m^*(\mathbf{x}) &= m(\mathbf{x}) + V(\mathbf{x}, \mathbf{X})V(\mathbf{X}, \mathbf{X})^{-1}(\mathbf{Y} - m(\mathbf{X})) \\ V^*(\mathbf{x}, \mathbf{x}') &= V(\mathbf{x}, \mathbf{x}') - V(\mathbf{x}, \mathbf{X})V(\mathbf{X}, \mathbf{X})^{-1}V(\mathbf{X}, \mathbf{x}') \end{aligned}$$

Atualizando um processo Gaussiano

- Seja $\eta(\cdot) \sim PG\,(m(\cdot), V(\cdot, \cdot))$
- $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ e $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$ inputs e outputs de um experimento computacional.
- Usando propriedades da Normal multivariada é fácil mostrar que

$$\eta(\cdot)|\mathbf{Y}, \mathbf{X} \sim PG(m^*(\cdot), V^*(\cdot, \cdot))$$

onde

$$\begin{aligned} m^*(\mathbf{x}) &= m(\mathbf{x}) + V(\mathbf{x}, \mathbf{X})V(\mathbf{X}, \mathbf{X})^{-1}(\mathbf{Y} - m(\mathbf{X})) \\ V^*(\mathbf{x}, \mathbf{x}') &= V(\mathbf{x}, \mathbf{x}') - V(\mathbf{x}, \mathbf{X})V(\mathbf{X}, \mathbf{X})^{-1}V(\mathbf{X}, \mathbf{x}') \end{aligned}$$

Atualizando um processo Gaussiano

- Seja $\eta(\cdot) \sim PG\,(m(\cdot), V(\cdot,\cdot))$
- $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ e $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$ inputs e outputs de um experimento computacional.
- Usando propriedades da Normal multivariada é fácil mostrar que

$$\eta(\cdot)|\mathbf{Y}, \mathbf{X} \sim PG(m^*(\cdot), V^*(\cdot,\cdot))$$

onde

$$\begin{aligned} m^*(\mathbf{x}) &= m(\mathbf{x}) + V(\mathbf{x}, \mathbf{X})V(\mathbf{X}, \mathbf{X})^{-1}(\mathbf{Y} - m(\mathbf{X})) \\ V^*(\mathbf{x}, \mathbf{x}') &= V(\mathbf{x}, \mathbf{x}') - V(\mathbf{x}, \mathbf{X})V(\mathbf{X}, \mathbf{X})^{-1}V(\mathbf{X}, \mathbf{x}') \end{aligned}$$

Emuladores independentes

- Para cada output $\eta_i(\cdot)$, teremos um emulador diferente e independente, ou seja

$$\eta_i(\cdot) \sim PG(m_i(\cdot), \sigma_i^2 C_{\delta_i}(\cdot, \cdot))$$

- Vantagem: A estrutura de correlação entre os inputs será diferente para ouput.
- Desvantagem: Estamos ignorando a correlação entre os outputs, que geralmente não devem ser descartadas.
- Os hiperpâmetros são $\omega = (\sigma_1^2, \ldots, \sigma_r^2, \delta_1^2, \ldots, \delta_r^2)$, e uma priori tipica para ω é dada por

$$\pi(\omega) \propto \prod \pi(\delta_i)\sigma_i^{-2}.$$

Emuladores independentes

- Para cada output $\eta_i(\cdot)$, teremos um emulador diferente e independente, ou seja

$$\eta_i(\cdot) \sim PG(m_i(\cdot), \sigma_i^2 C_{\delta_i}(\cdot, \cdot))$$

- Vantagem: A estrutura de correlação entre os inputs será diferente para ouput.
- Desvantagem: Estamos ignorando a correlação entre os outputs, que geralmente não devem ser descartadas.
- Os hiperpâmetros são $\omega = (\sigma_1^2, \ldots, \sigma_r^2, \delta_1^2, \ldots, \delta_r^2)$, e uma priori tipica para ω é dada por

$$\pi(\omega) \propto \prod \pi(\delta_i)\sigma_i^{-2}.$$

Emuladores independentes

- Para cada output $\eta_i(\cdot)$, teremos um emulador diferente e independente, ou seja

$$\eta_i(\cdot) \sim PG(m_i(\cdot), \sigma_i^2 C_{\delta_i}(\cdot,\cdot))$$

- Vantagem: A estrutura de correlação entre os inputs será diferente para ouput.
- Desvantagem: Estamos ignorando a correlação entre os outputs, que geralmente não devem ser descartadas.
- Os hiperpâmetros são $\omega = (\sigma_1^2, \ldots, \sigma_r^2, \delta_1^2, \ldots, \delta_r^2)$, e uma priori tipica para ω é dada por

$$\pi(\omega) \propto \prod \pi(\delta_i)\sigma_i^{-2}.$$

Emuladores independentes

- Para cada output $\eta_i(\cdot)$, teremos um emulador diferente e independente, ou seja

$$\eta_i(\cdot) \sim PG(m_i(\cdot), \sigma_i^2 C_{\delta_i}(\cdot, \cdot))$$

- Vantagem: A estrutura de correlação entre os inputs será diferente para ouput.
- Desvantagem: Estamos ignorando a correlação entre os outputs, que geralmente não devem ser descartadas.
- Os hiperpâmetros são $\omega = (\sigma_1^2, \ldots, \sigma_r^2, \delta_1^2, \ldots, \delta_r^2)$, e uma priori tipica para ω é dada por

$$\pi(\omega) \propto \prod \pi(\delta_i)\sigma_i^{-2}.$$

Emuladores não-separáveis
Covariância baseada em modelos de convolução

- A idéia: uma convoluçao de um processo gaussiano ruido branco com uma função *kernel* arbitrária.
- Precisamos de uma função *kernel* κ_i para cada output i e uma matriz $r \times r$ positiva definida $\tilde{\Sigma}$. O elemento (i, j) de $V(\cdot, \cdot)$ é

$$V_{ij}(\mathbf{x}, \mathbf{x}') = \tilde{\Sigma}_{ij} \int_{\mathbb{R}^p} \kappa_i(u - \mathbf{x}) \kappa_j(u - \mathbf{x}')\, du \,.$$

- Uma escolha conveniente para a função *kernel*

$$\kappa_i(x) = \left[\left(\frac{4}{\pi} \right)^p \prod_{\ell=1}^{p} \phi_i^{(\ell)} \right]^{\frac{1}{4}} \exp\{-2x^T \Phi_i x\}.$$

- Detalhes em Fricker (2010)

Emuladores não-separáveis
Covariância baseada em modelos de convolução

- A idéia: uma convoluçao de um processo gaussiano ruido branco com uma função *kernel* arbitrária.
- Precisamos de uma função *kernel* κ_i para cada output i e uma matriz $r \times r$ positiva definida $\tilde{\Sigma}$. O elemento (i,j) de $V(\cdot,\cdot)$ é

$$V_{ij}(\mathbf{x},\mathbf{x}') = \tilde{\Sigma}_{ij} \int_{\mathbb{R}^p} \kappa_i(u-\mathbf{x})\kappa_j(u-\mathbf{x}')\,du\,.$$

- Uma escolha conveniente para a função *kernel*

$$\kappa_i(x) = \left[\left(\frac{4}{\pi}\right)^p \prod_{\ell=1}^{p} \phi_i^{(\ell)}\right]^{\frac{1}{4}} \exp\{-2x^T\Phi_i x\}.$$

- Detalhes em Fricker (2010)

Emuladores não-separáveis
Covariância baseada em modelos de convolução

- A idéia: uma convoluçao de um processo gaussiano ruido branco com uma função *kernel* arbitrária.
- Precisamos de uma função *kernel* κ_i para cada output i e uma matriz $r \times r$ positiva definida $\tilde{\Sigma}$. O elemento (i, j) de $V(\cdot, \cdot)$ é

$$V_{ij}(\mathbf{x}, \mathbf{x}') = \tilde{\Sigma}_{ij} \int_{\mathbb{R}^p} \kappa_i(u - \mathbf{x}) \kappa_j(u - \mathbf{x}') \, du \, .$$

- Uma escolha conveniente para a função *kernel*

$$\kappa_i(x) = \left[\left(\frac{4}{\pi} \right)^p \prod_{\ell=1}^{p} \phi_i^{(\ell)} \right]^{\frac{1}{4}} \exp\{-2x^T \Phi_i x\}.$$

- Detalhes em Fricker (2010)

- A idéia: uma convoluçao de um processo gaussiano ruido branco com uma função *kernel* arbitrária.
- Precisamos de uma função *kernel* κ_i para cada output i e uma matriz $r \times r$ positiva definida $\tilde{\Sigma}$. O elemento (i,j) de $V(\cdot,\cdot)$ é

$$V_{ij}(\mathbf{x},\mathbf{x}') = \tilde{\Sigma}_{ij} \int_{\mathbb{R}^p} \kappa_i(u-\mathbf{x})\kappa_j(u-\mathbf{x}')\,du\,.$$

- Uma escolha conveniente para a função *kernel*

$$\kappa_i(x) = \left[\left(\frac{4}{\pi}\right)^p \prod_{\ell=1}^{p} \phi_i^{(\ell)}\right]^{\frac{1}{4}} \exp\{-2x^T\Phi_i x\}.$$

- Detalhes em Fricker (2010)

Emuladores não-separáveis
Covariância baseada em LMC

- A idéia: Representar os outputs originais como funções lineares de outputs latentes que são modelados usando PG independentes.
- Em Geoestatística isso é conhecido como o modelo linear de corregionalização (LMC)
- A função de covariância é

$$V(\cdot,\cdot) = \sum_{i=1}^{r} \Sigma_i \tilde{\kappa}_i(\cdot,\cdot),$$

  onde $\tilde{\kappa}_i$ é uma função de correlação base, e $\Sigma_i = r_i r_i^T$, com r_i a i-ésima coluna de R, $\Sigma = RR^T$.
- Σ é a matriz de covariância entre outputs.
- Mais detalhes em Fricker (2010)

Emuladores não-separáveis
Covariância baseada em LMC

- A idéia: Representar os outputs originais como funções lineares de outputs latentes que são modelados usando PG independentes.
- Em Geoestatística isso é conhecido como o modelo linear de corregionalização (LMC)
- A função de covariância é

$$V(\cdot,\cdot) = \sum_{i=1}^{r} \Sigma_i \tilde{\kappa}_i(\cdot,\cdot)\,,$$

  onde $\tilde{\kappa}_i$ é uma função de correlação base, e $\Sigma_i = r_i r_i^T$, com r_i a i-ésima coluna de R, $\Sigma = RR^T$.
- Σ é a matriz de covariância entre outputs.
- Mais detalhes em Fricker (2010)

Emuladores não-separáveis
Covariância baseada em LMC

- A idéia: Representar os outputs originais como funções lineares de outputs latentes que são modelados usando PG independentes.
- Em Geoestatística isso é conhecido como o modelo linear de corregionalização (LMC)
- A função de covariância é

$$V(\cdot,\cdot) = \sum_{i=1}^{r} \Sigma_i \tilde{\kappa}_i(\cdot,\cdot)\,,$$

onde $\tilde{\kappa}_i$ é uma função de correlação base, e $\Sigma_i = r_i r_i^T$, com r_i a i-ésima coluna de R, $\Sigma = RR^T$.

- Σ é a matriz de covariância entre outputs.
- Mais detalhes em Fricker (2010)

Emuladores não-separáveis
Covariância baseada em LMC

- A idéia: Representar os outputs originais como funções lineares de outputs latentes que são modelados usando PG independentes.
- Em Geoestatística isso é conhecido como o modelo linear de corregionalização (LMC)
- A função de covariância é

$$V(\cdot,\cdot) = \sum_{i=1}^{r} \Sigma_i \tilde{\kappa}_i(\cdot,\cdot),$$

  onde $\tilde{\kappa}_i$ é uma função de correlação base, e $\Sigma_i = r_i r_i^T$, com r_i a i-ésima coluna de R, $\Sigma = RR^T$.
- Σ é a matriz de covariância entre outputs.
- Mais detalhes em Fricker (2010)

Emuladores não-separáveis
Covariância baseada em LMC

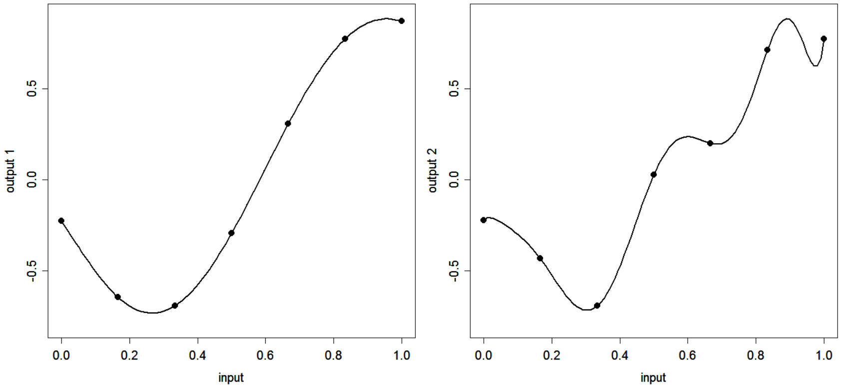
- A idéia: Representar os outputs originais como funções lineares de outputs latentes que são modelados usando PG independentes.
- Em Geoestatística isso é conhecido como o modelo linear de corregionalização (LMC)
- A função de covariância é

$$V(\cdot,\cdot) = \sum_{i=1}^{r} \Sigma_i \tilde{\kappa}_i(\cdot,\cdot)\,,$$

  onde $\tilde{\kappa}_i$ é uma função de correlação base, e $\Sigma_i = r_i r_i^T$, com r_i a i-ésima coluna de R, $\Sigma = RR^T$.
- Σ é a matriz de covariância entre outputs.
- Mais detalhes em Fricker (2010)

Exemplo

Seja o seguinte simulador com 1 input e 2 ouputs:

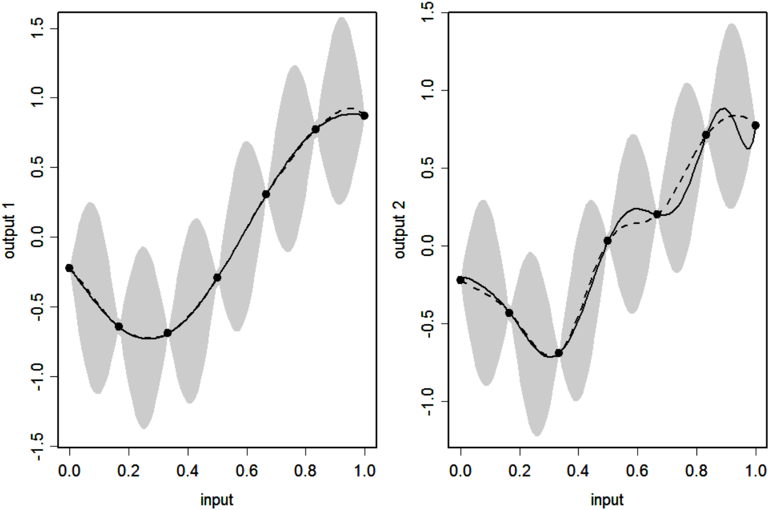


Exemplo

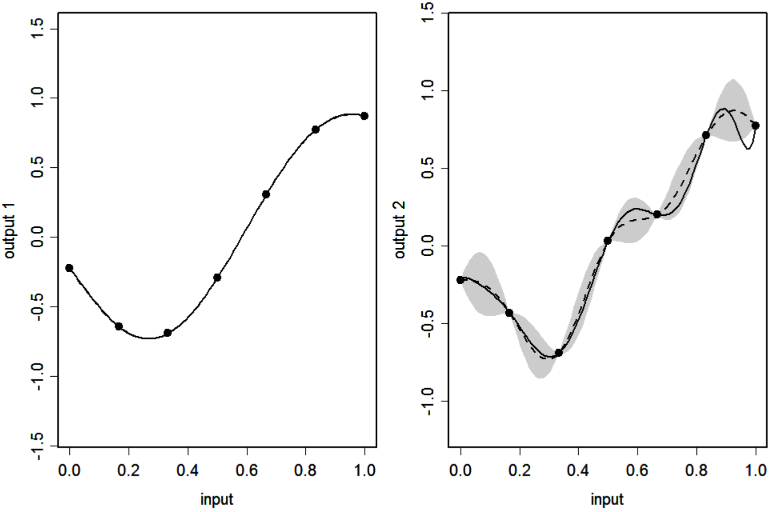
Vamos ajustar 4 emuladores gaussianos multivariados com 4 diferentes estruturas de correlação:

- Separável
- Independente
- Não-separável via convolução
- Não-separável via LMC

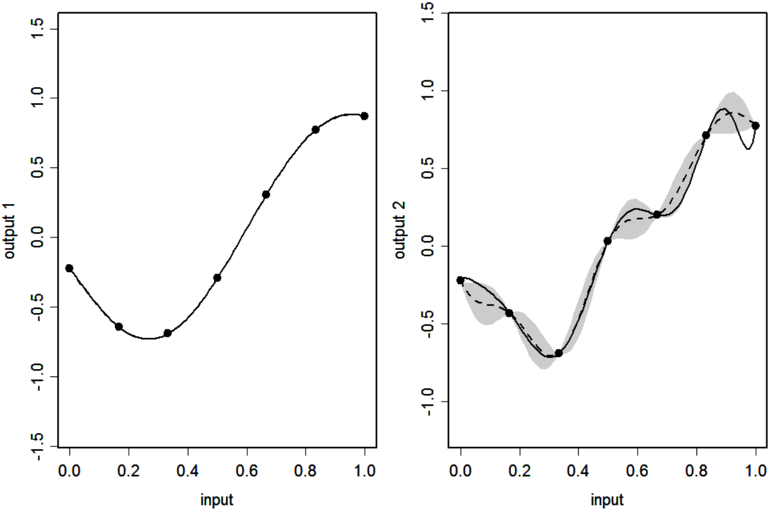
Exemplo - Emulador Separável



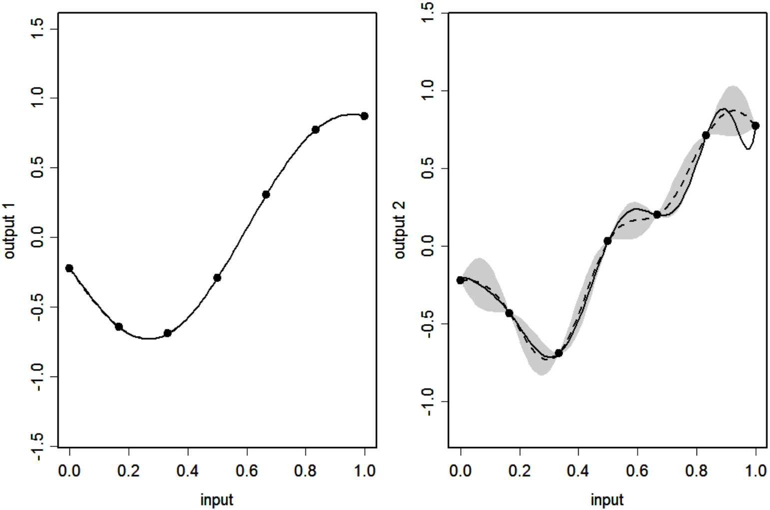
Exemplo - Emulador Independente



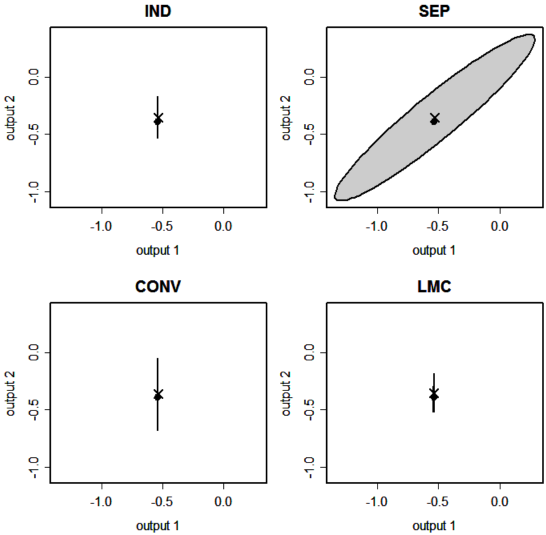
Exemplo - Emulador Não-separável via convolução



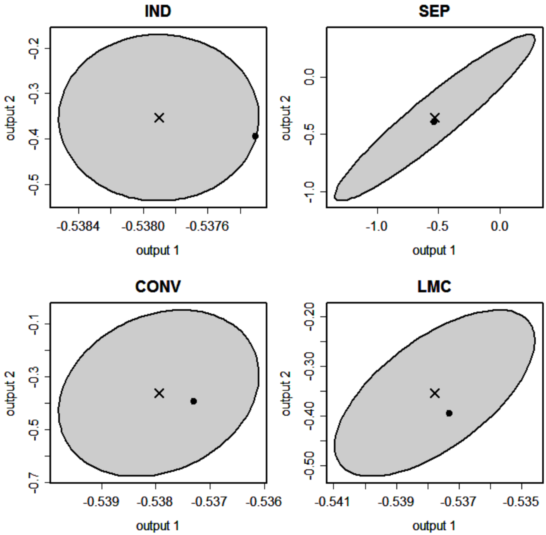
Exemplo - Emulador Não-separável via LMC



Exemplo - Prevendo $\eta(0.75)$



Exemplo - Prevendo $\eta(0.75)$



Cenas dos próximos capítulos...

- Calibração
  - O que fazer quando temos além do simulador dados reais?
  - Modelos de calibração.
- Emuladores via componentes principais
- Exemplo de modelagem climática.
- Outras análises
  - Análises de Incerteza e Sensibilidade (UA/SA)
  - Diagnósticos e Validação
  - ABC (Approximate Bayesian computation)

Cenas dos próximos capítulos...

- Calibração
  - O que fazer quando temos além do simulador dados reais?
  - Modelos de calibração.
- Emuladores via componentes principais
- Exemplo de modelagem climática.
- Outras análises
  - Análises de Incerteza e Sensibilidade (UA/SA)
  - Diagnósticos e Validação
  - ABC (Approximate Bayesian computation)

Cenas dos próximos capítulos...

- Calibração
  - O que fazer quando temos além do simulador dados reais?
  - Modelos de calibração.
- Emuladores via componentes principais
- Exemplo de modelagem climática.
- Outras análises
  - Análises de Incerteza e Sensibilidade (UA/SA)
  - Diagnósticos e Validação
  - ABC (Approximate Bayesian computation)

Cenas dos próximos capítulos...

- Calibração
  - O que fazer quando temos além do simulador dados reais?
  - Modelos de calibração.
- Emuladores via componentes principais
- Exemplo de modelagem climática.
- Outras análises
  - Análises de Incerteza e Sensibilidade (UA/SA)
  - Diagnósticos e Validação
  - ABC (Approximate Bayesian computation)

Cenas dos próximos capítulos...

- Calibração
  - O que fazer quando temos além do simulador dados reais?
  - Modelos de calibração.
- Emuladores via componentes principais
- Exemplo de modelagem climática.
- Outras análises
  - Análises de Incerteza e Sensibilidade (UA/SA)
  - Diagnósticos e Validação
  - ABC (Approximate Bayesian computation)

Cenas dos próximos capítulos...

- Calibração
  - O que fazer quando temos além do simulador dados reais?
  - Modelos de calibração.
- Emuladores via componentes principais
- Exemplo de modelagem climática.
- Outras análises
  - Análises de Incerteza e Sensibilidade (UA/SA)
  - Diagnósticos e Validação
  - ABC (Approximate Bayesian computation)

Cenas dos próximos capítulos...

- Calibração
  - O que fazer quando temos além do simulador dados reais?
  - Modelos de calibração.
- Emuladores via componentes principais
- Exemplo de modelagem climática.
- Outras análises
  - Análises de Incerteza e Sensibilidade (UA/SA)
  - Diagnósticos e Validação
  - ABC (Approximate Bayesian computation)

Cenas dos próximos capítulos...

- Calibração
  - O que fazer quando temos além do simulador dados reais?
  - Modelos de calibração.
- Emuladores via componentes principais
- Exemplo de modelagem climática.
- Outras análises
  - Análises de Incerteza e Sensibilidade (UA/SA)
  - Diagnósticos e Validação
  - ABC (Approximate Bayesian computation)

Cenas dos próximos capítulos...

- Calibração
  - O que fazer quando temos além do simulador dados reais?
  - Modelos de calibração.
- Emuladores via componentes principais
- Exemplo de modelagem climática.
- Outras análises
  - Análises de Incerteza e Sensibilidade (UA/SA)
  - Diagnósticos e Validação
  - ABC (Approximate Bayesian computation)