Métamodèles pour variables d'entrée mixtes

Claire Cannamela

CEA, DAM, DIF, F-91297 Arpajon, France

Atelier MASCOT-Num

16 mai 2014

Contexte de l'application (1/5)

Stockage et entreposage des déchets nucléaires

- Les déchets nucléaires sont classés suivant deux propriétés :
  - Le temps de vie (2 catégories)
  - L'activité du déchet (4 catégories)
- Les sites de stockage et d'entreposage n'acceptent pas tous les déchets
- Le producteur de déchets doit garantir la conformité du déchet vis à vis des exigences des sites

  $\rightarrow$ Caractérisation du déchet par des techniques de mesures nucléaires :
  - active
  - passive
  - neutronique
  - spectrométrie gamma

Principe de la spectrométrie gamma

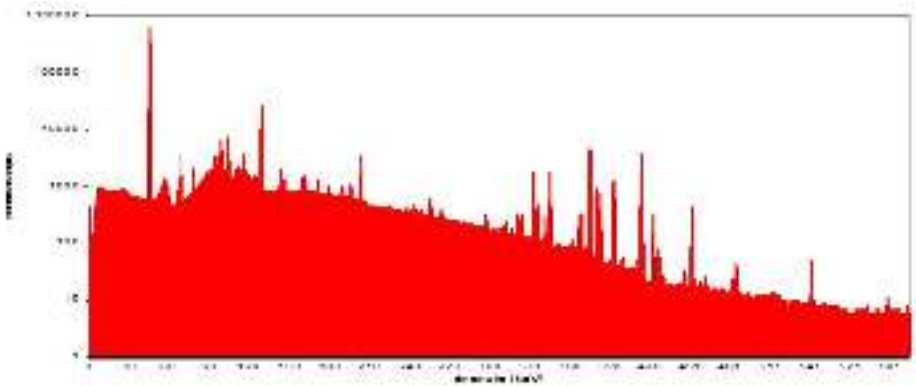
- Objectif : identification et quantification des radionucléides d'un déchet par contrôle non destructif.

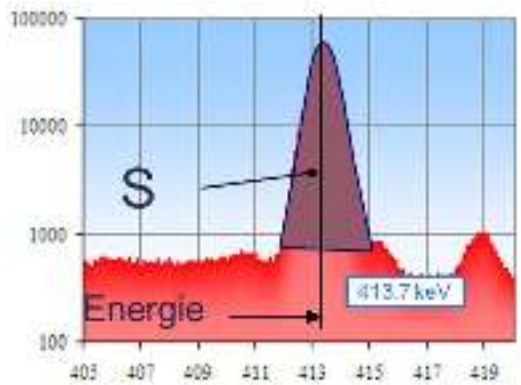
- Détection des rayonnements gamma émis spontanément de l'objet par le retour des noyaux excités à leur état fondamental.

  → Spectre d'acquisition

  → Extraction des énergies et des surfaces nettes des pics du spectre

  → Identification des radionucléides (comparaison bases de données nucléaires)





Contexte de l'application (3/5)

Mesure de l'activité du déchet

- L'activité du déchet mesuré est donnée par :

$$A(E) = \frac{S(E)}{TI(E)\varepsilon(E)}$$

avec :

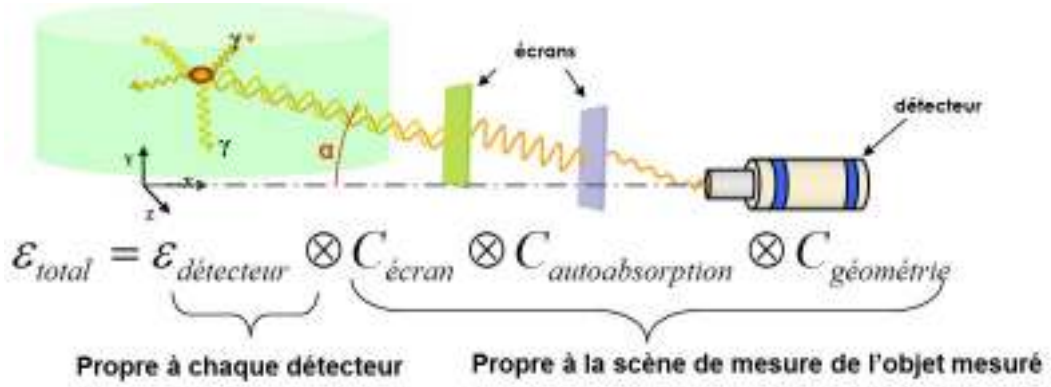
  - E la ou les énergies des photons émis par le radionucléide considéré,
  - $S(E)$ la surface des pics,
  - $I(E)$ le taux d'emmission du rayonnement,
  - T le temps de mesure,
  - $\varepsilon(E)$ le rendement de détection.

- $\varepsilon(E)$ est la seule fonction inconnue pour déterminer $A(E)$.
- $\varepsilon(E)$ est défini comme le rapport du nombre de photons détectés à l'énergie E sur le nombre de photons émis par les radionucléides à cette même énergie E.
- $\varepsilon(E)$ dépend d'un nombre important de paramètres propres à l'objet à caractériser

Contexte de l'application (4/5)

Rendement de détection $\varepsilon(E)$ dépend de nombreux paramètres inconnus *a priori* :

- les constituants de l'objet : formes géométriques, matériaux source et de la matrice, densités, épaisseurs, ...
- l'agencement de l'objet : position spatiale, localisation et répartition des radionucléides, ...
- les constituants des écrans : géométrie, position, matériaux,...



$\rightarrow$ Modélisation de ε via un code de transport de radiation coûteux $G(.)$

$\rightarrow$ Plusieurs itérations nécessaires pour modéliser une géométrie approchée de la vraie géométrie de mesure.

(Comparaison sur la base de l'homogéneité du calcul final d'activité sur plusieurs raies d'un radionucléide.)

Contexte de l'application (5/5)

Métamodélisation du rendement de détection

- On cherche à remplacer la réponse ε du code de calcul $G(.)$ par un métamodèle construit à partir d'un nombre limité d'appels au code :

$$X \in \mathcal{X} \longrightarrow \varepsilon = G(X) \in \mathbb{R}$$

- Pourquoi un métamodèle ?
  1. Procéder à une analyse de sensibilité (ex : aide opérateur pour la recherche de la vraie géométrie)
  2. Propager les incertitudes des paramètres d'entrées sur la sortie
  3. Problème inverse (trouver une configuration géométrique)

  $\longrightarrow$ Trop coûteux avec le code

Simplification du problème physique $\longrightarrow$ 5 paramètres d'entrée incertains :

- l'énergie $E \in [100, 600]$
- la densité de l'objet $\rho \in]0, 8]$
- dimension de l'objet $e \in]0, 10]$
- le matériau $M \in \{eau, fer, plomb\}$
- la géométrie de l'objet $V \in \{sphere, cylindre, cube\}$

Notations et hypothèses

- On s'intéresse à la réponse d'un code de calcul $y(w) \in \mathbb{R}$
- $w = (x^t, z^t)$ vecteur des paramètres d'entrées indépendants de dimension d
- $x = (x_1, \ldots, x_p)^t$ paramètres continus
- $z = (z_1, \ldots, z_q)^t$ paramètres catégoriels (discrets et non ordonnés)
- Chaque $(z_i)_{i=1,\cdots,q}$ est représenté par m_i modalités.
- Pour simplifier les notations, on suppose $z_i \in \{1, \cdots m_i\}$ (ordre arbitraire)
- $\mathbf{D} = \{w_1, \ldots, w_n\}$ le plan d'expériences utilisé pour construire le métamodèle
- et les sorties du code $\mathbf{y}^n = (y(w_1), \ldots, y(w_n))$ aux points de $\mathbf{D}$

Différents types de métamodèles

1. Méthodes basées sur la régression linéaire
2. Méthodes basées sur des arbres de régression
3. Méthodes basées sur les splines de lissage
4. Méthodes basées sur des processus Gaussien

Outline

Principe de la régression linéaire

- La réponse est modélisée par une combinaison linéaire de fonctions de base :

$$\hat{y}(x,z) = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \sum_{i<j} \beta_{ij} x_i x_j + \sum_{i=1}^{q} \sum_{k=1}^{m_i - 1} \gamma_{ik} \mathbf{1}_{\{z_i = k\}} + \sum_{i=1}^{p} \sum_{j=1}^{q} \sum_{k=1}^{m_j - 1} \delta_{ijk} x_i \mathbf{1}_{\{z_j = k\}}$$

- On peut aussi tenir compte des interactions entre les interactions du premier ordre des variables continus et les modalités des variables catégorielles.

  On ajoute alors le terme suivant :

$$\sum_{i<j}^{p} \sum_{l=1}^{q} \sum_{k=1}^{m_l - 1} \eta_{ijlk} x_i x_j \mathbf{1}_{\{z_l = k\}}.$$

- On a les mêmes résultats que pour la RL avec des variables continues :

  Ecriture matricielle : $\mathbf{y}^n = \beta \mathbf{X} + \varepsilon$

  Estimation des coefficients par la méthode des moindres carrés $\hat{\beta} = (X^t X)^{-1} X \mathbf{y}^n$ avec $\mathbf{X}$ la matrice du modèle.

Régression linéaire pour variables catégorielles

- Exemple : $Y = \beta_0 + \beta_1 X_1 + \beta_2 \mathbf{1}_{\{Z_1 = A\}}$

  $\rightarrow Y = \beta_0 + \beta_1 X_1 + \beta_2$ si $Z_1 = A$

  et $Y = \beta_0 + \beta_1 X_1$ si $Z_1 = B$

- Un modèle de regression linéaire pour chaque combinaison de variables catégorielles.

- Il faut suffisamment d'observations pour chaque combinaison de variables catégorielles pour obtenir un métamodèle précis.

  $\rightarrow$ Trop coûteux dès qu'il y a un grand nombre de variables catégorielles et/ou un grand nombre de modalités.

  (5 variables catégorielles à 6 modalités : 30 combinaisons de modalités différentes !)

Principe de la régression régularisée/pénalisée

Le critère moindres carrés est modifié pour pénaliser certaines caractéristiques des coefficienst estimés (ridge regression, lasso, elastic net, ...)

Régression lasso

- Régularisation L_1 : $\hat{\beta}_{lasso} = \underset{\beta}{\text{argmin}}\{\frac{1}{2} \parallel \mathbf{y}^n - \mathbf{X}\beta \parallel_2^2 + \lambda \parallel \beta \parallel_1\}$
- Sélection de variables naturel (coefficients $\beta_i \neq 0$) dans le cas continu
- Pour des variables catégorielles : sélection des variables $(X_i)_{i=1,\dots,p}$ et $(\mathbf{1}_{\{z_j=k\}})_{(j=1,\dots,q),(k=1,\dots,m_j-1)}$ $\rightarrow$ dépend donc des catégories choisies
- Group lasso : on regroupe les coefficients d'une même variable

$$\hat{\beta}_{GL} = \underset{\beta}{\text{argmin}}\{\frac{1}{2} \parallel \mathbf{y}^n - \mathbf{X}\beta \parallel_2^2 + \lambda \sum_{j=1}^{d} \parallel \beta^{(j)} \parallel_2$$

  $\rightarrow$ Sélection des variables $(X_i)_{i=1,\dots,p}$ et $(Z_i)_{i=1,\dots,q}$

- Fused lasso : on considère les différences des coefficients d'une même variable

$$\hat{\beta}_{GL} = \underset{\beta}{\text{argmin}}\{\frac{1}{2} \parallel \mathbf{y}^n - \mathbf{X}\beta \parallel_2^2 + \lambda \sum_{j=1}^{d} w_{il}^{(j)} \sum_{i>l} |\beta_{ji} - \beta_{jl}|$$

  $\rightarrow$ Regroupement des catégories en fonction de leur influence sur la réponse

Implémentation sous R

- Méthodes basées sur la régression implémentées sous `R`
- Bien définir les variables catégorielles : commande `factor`

```
modal1 =  c('bleue','rouge','blanc')
x = factor(as.integer(runif(10)*length(modal1)),label=modal1)

[1] bleue rouge bleue blanc rouge rouge bleue rouge blanc blanc
Levels: bleue rouge blanc
```

- Régression linéaire : commande classique `lm`
- Régression lasso : plusieurs packages `lasso2` `lars` `penalized`
- Group lasso : package `grplasso`
- Fused lasso : package `genlasso`
- ...

Application

Utilisation d'un cas jouet de notre application (analytique)

$$\varepsilon = G(E, \rho, e, M, V)$$

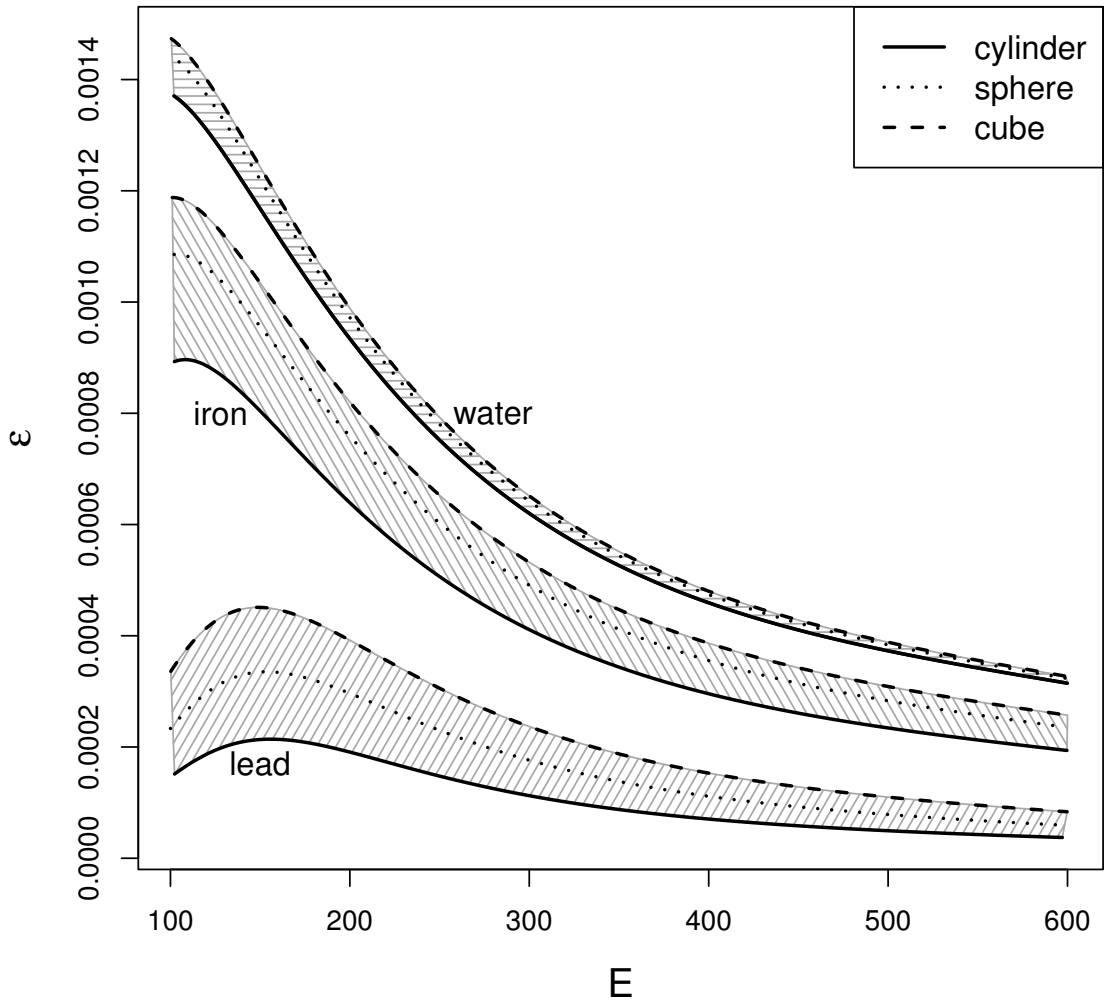
l'énergie $E \in [100, 600]$

la densité de l'objet $\rho \in]0, 8]$

dimension de l'objet $e \in]0, 10]$

le matériau $M \in \{eau, fer, plomb\}$

la géométrie de l'objet
$V \in \{sphere, cylindre, cube\}$



Application - régression linéaire

```
> Head(Xapp)
        X1       X2       X3    X4  X5
1 296.8814 3.507764 4.424005 water box
2 103.2795 4.072445 9.676911 water cyl
3 193.9784 1.951701 2.022876 water cyl

> model = Y ~ (X1+X2+X3+X4+X5)
> fit_lm = lm(model,data=data.frame(Xapp,Y=Yapp))
> summary(fit_lm)

Call:
lm(formula = model, data = data.frame(Xapp, Y = Yapp))

Residuals:
       Min         1Q     Median         3Q        Max
-2.095e-04 -8.471e-05 -2.912e-05  5.803e-05  9.045e-04

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.374e-04  2.322e-05  31.758  < 2e-16 ***
X1          -5.016e-07  4.052e-08 -12.379  < 2e-16 ***
X2          -3.029e-05  2.542e-06 -11.916  < 2e-16 ***
X3          -2.602e-05  1.999e-06 -13.022  < 2e-16 ***
X4iron      -1.999e-04  1.431e-05 -13.968  < 2e-16 ***
X4lead      -2.942e-04  1.406e-05 -20.926  < 2e-16 ***
X5box        2.545e-05  1.383e-05   1.840   0.0664 .
X5sph        7.028e-05  1.424e-05   4.936 1.09e-06 ***

Residual standard error: 0.0001287 on 492 degrees of freedom
Multiple R-squared:  0.69,    Adjusted R-squared:  0.683
F-statistic: 145.2 on 7 and 492 DF,  p-value: < 2.2e-16
```

Application - régression linéaire

```
> model = Y ~ (X1+X2+X3)^2+X4+X5 + I(X1)^2 +I(X2)^2 + I(X3)^2
> fit_lm = lm(model,data=data.frame(Xapp,Y=Yapp))
> summary(fit_lm)

Call:
lm(formula = model, data = data.frame(Xapp, Y = Yapp))

Residuals:
       Min          1Q      Median          3Q         Max
-2.765e-04 -5.876e-05 -3.930e-06  4.943e-05  5.717e-04

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.409e-03  4.771e-05  29.531  < 2e-16 ***
X1          -2.373e-06  1.962e-07 -12.094  < 2e-16 ***
X2          -1.420e-04  9.933e-06 -14.300  < 2e-16 ***
X3          -1.130e-04  8.326e-06 -13.572  < 2e-16 ***
X4iron      -2.020e-04  1.135e-05 -17.789  < 2e-16 ***
X4lead      -2.876e-04  1.111e-05 -25.883  < 2e-16 ***
X5box        3.215e-05  1.095e-05   2.936  0.00349 **
X5sph        6.606e-05  1.131e-05   5.841 9.52e-09 ***
I(X1^2)      1.315e-09  2.464e-10   5.337 1.45e-07 ***
I(X2^2)      7.932e-06  9.502e-07   8.347 7.30e-16 ***
I(X3^2)      4.675e-06  6.669e-07   7.009 8.04e-12 ***
X1:X2        1.100e-07  1.414e-08   7.780 4.37e-14 ***
X1:X3        9.938e-08  1.109e-08   8.960  < 2e-16 ***
X2:X3        1.157e-06  6.826e-07   1.695  0.09074 .

Residual standard error: 0.0001016 on 486 degrees of freedom
Multiple R-squared:  0.7994,    Adjusted R-squared:  0.794
F-statistic: 148.9 on 13 and 486 DF,  p-value: < 2.2e-16
```

Application - régression linéaire

```
> model0 = Y ~ (X1+X2+X3+X4+X5)
> model1 = Y ~ (X1+X2+X3)^2 + X4 + X5
> model2 = Y ~ (X1+X2+X3)^2 + X4 + X5 + I(X1)^2 +I(X2)^2 + I(X3)^2
> model3 = Y ~ (X1+X2+X3+X4+X5)^2 + I(X1)^2 +I(X2)^2 + I(X3)^2

model0 :
> R2 = 0.69
> Q2 = 0.6

model1 :
> R2 = 0.73
> Q2 = 0.67

model2 :
> R2 = 0.79
> Q2 = 0.76

model3 :
> R2 = 0.88
> Q2 = 0.85
```

Outline

Principe de arbres de regresison binaires

- Méthode itérative de division de l'espace d'entrée des paramètres $\mathcal{X}$ en partitions disjointes

$\rightarrow$ On divise le DoE en 2 partitions, lesquelles sont à nouveaux divisées en 2, ...

- Pour chaque partition, le modèle est une constante calculée à partir des réponses aux points de la partition.
- Elagage : pour éviter un surapprentissage, l'arbre est élagué.

  Il existe de nombreuses procédures, la plus simple est de fixer un nombre de données minimale pour chaque partition.
- Le métamodèle s'écrit alors :

$$\hat{y}(w) = \sum_{k=1}^{K} c_k \mathbf{1}_{\{w \in P_k\}}$$

avec K représente le nombre de partitions de l'arbre élagué , $(P_k)_{k=1,\dots,K}$ les partitions dijointes et $(c_k)_{k=1,\dots,K}$ la constante ajusté à partir des données de la régions P_k.

Construction des arbres de regression binaires

- Choix des partitions P_k et de la constante c_k : le critère est la minimisation de l'erreur quadratique

$$\min \sum_{i=1}^{n} \left(y(w_i) - \sum_{k=1}^{K} c_k \mathbf{1}_{\{w \in P_k\}} \right)^2$$

- c_k : moyenne des $y(w_i)$ avec $w_i \in P_k$
- P_k et variables continues : division à partir d'un seuil

$$P_1(j,s) = \{X | X_j \leq s\} \text{ et } P_2(j,s) = \{X | X_j \geq s\}$$

- P_k et variables catégorielles : division à partir d'un groupe de catégories

$$P_1(j,s) = \{Z | Z_j = s_j\} \text{ et } P_2(j,s) = \{Z | Z_j \neq s_j\} \text{ avec } s_j \subset \{1, ..., m_j\}$$

- Pour déterminer les partitions P_k, on cherche itérativement le meilleur couple (j,s)
- Adapté à un grand nombre de variables et beaucoup de données
- Principal inconvénient : le métamodèle est une fonction constante par morceaux.

Quelques extensions/variantes

PRIM : Patient rule Induction Method

- La division de l'espace des entrée des paramètres $\mathcal{X}$ se fait en tenant compte de plusieurs variables à chaque fois
- Exemple : $\{a_1 \leq X_1 \leq b_1\} \cap \{a_3 \leq X_3 \leq b_3\}$
- $\mathcal{X}$ sera alors composé de petites "boites" disjointes dans lesquelles la réponse sera constante.

HME : Hierarchical Mixture of Experts

- La décision d'une branche de l'arbre ou de l'autre est probabiliste : à chaque noeud, une observation peut aller "gauche" ou à "droite" avec une probabilité dépendant de sa valeur
- Pour chaque partition, le modèle est un modèle linéaire plutôt qu'une constante

BCART : Bayesian Classification and Regression Method

- Distribution *a priori* pour le choix des variables à chaque division
- Distribution *a priori* pour le seuil ou le nombre de catégories
- Distribution *a priori* pour le nombre de données minimal dans chaque partition.

Random Forest

Idée

- Construction d'une forêt d'arbres de regression
- Combinaison du bagging et des arbres de régression
- Bagging : aggregation aléatoire d'un grand nombre de modèle

Principe

- Création de $\mathcal{T}$ échantillons bootstrap du plan d'expériences initial
- Pour chaque échantillon bootstrap, construction d'un arbre de régression non élagué. Chaque partition est construite à partir d'un nombre réduit de variables (choisies aléatoirement)
- Le métamodèle est alors défini par :

$$\hat{y}(w) = \frac{1}{\mathcal{T}} \sum_{\tau=1}^{\mathcal{T}} \sum_{k=1}^{K_\tau} c_{k,\tau} \mathbf{1}_{\{w \in P_{k,\tau}\}}$$

  où $\mathcal{T}$ est le nombre d'arbres ; pour le τème arbre, $(K_\tau)_{\tau=1,\dots,\mathcal{T}}$ est le nombre de partitions, $(P_{k,\tau})_{k=1,\dots,K}$ sont les partitions et $(c_{k,\tau})_{k=1,\dots,K}$ la constante de la partition $P_{k,\tau}$.
- Le nombre d'arbre $\mathcal{T}$ doit être grand.
- Méthode efficace en grande dimension mais interprétation délicate

Boosting et arbres de regression

Idée

- Combinaison du boosting et des arbres de régression
- boosting : aggregation d'un grand nombre de modèle où chaque modèle est une version adaptative du précédent (en donnant plus de poinds aux observations mal ajustées ou mal prédites)

Principe

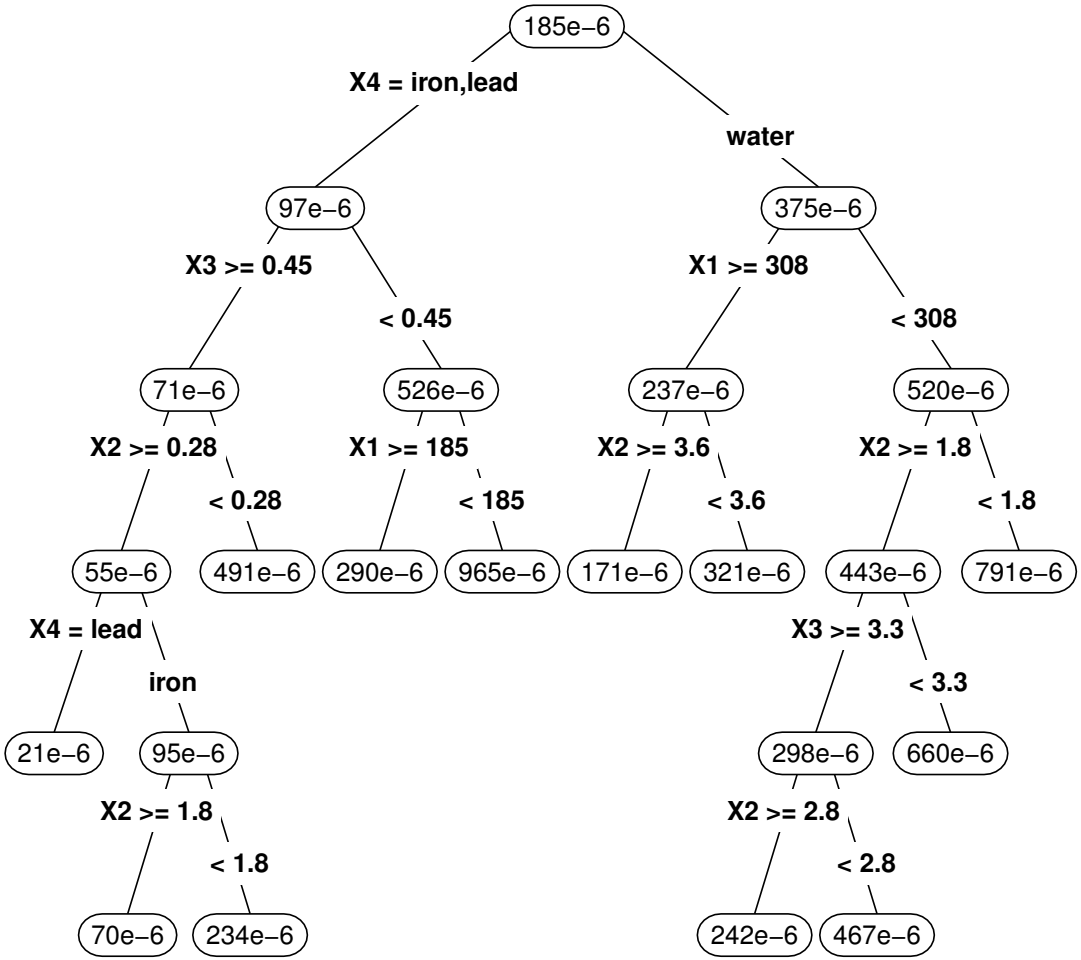
- Construction du premier arbre de régression élagué $\hat{y}_0(w) = \sum_{k_0=1}^{K_0} c_{k_0} \mathbf{1}_{\{w \in P_{k_0}\}}$
- A chaque étape k construction d'un arbre de régression élagué $\hat{r}_k(w)$ sur les résidus, *i.e.* à partir des données (r_{ik}, w_i) où $r_{ik} = y(w_i) - \hat{y}_{k-1}(w_i)$.
- Ajout de cet arbre au précédent : $\hat{y}_k(w) = \hat{y}_{ku-1}(w) + \hat{r}_k(w)$
- Le métamodèle est alors défini par : $\hat{y}(w) = \hat{y}_K(w)$
- Méthode (et amélioration) sous les noms de MART (Multiple additive Regression Trees) et Gradient Tree Boosting
- Méthode efficace en grande dimension

Implémentation sous R

- Méthodes basées sur les arbres de regression bien implémentées sous `R`
- Arbres binaires CART : package `rpart`
- Patient rule Induction Method : package `prim`
- Gradient boosting : package `gbm`
- Random Forest : package `randomforest`
- Bayesian additive Regression Trees BART : `BayesTree` et `bartMachine`
- Multiple adaptive regression splines MARS : `earth` , `polyspline`
  (sorte de généralisation des arbres de régression)
- ...

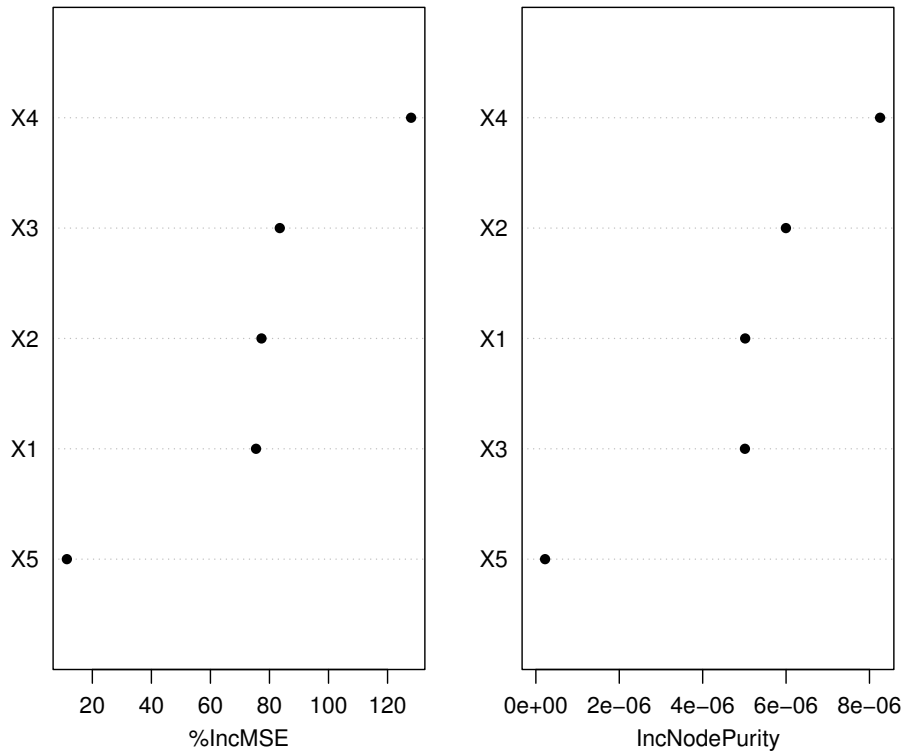
Application - CART

```
rpart(formula, data, weights, subset, na.action = na.rpart, method,
      model = FALSE, x = FALSE, y = TRUE, parms, control, cost, ...)
> library(rpart)
> library(rpart.plot)
> tree = rpart(Y~X1+X2+X3+X4+X5, data = data.frame(Xapp,Y=Yapp))
> rpart.plot(tree,type=4, faclen=0)
> R2 = 0.75
> Q2 = 0.7
```



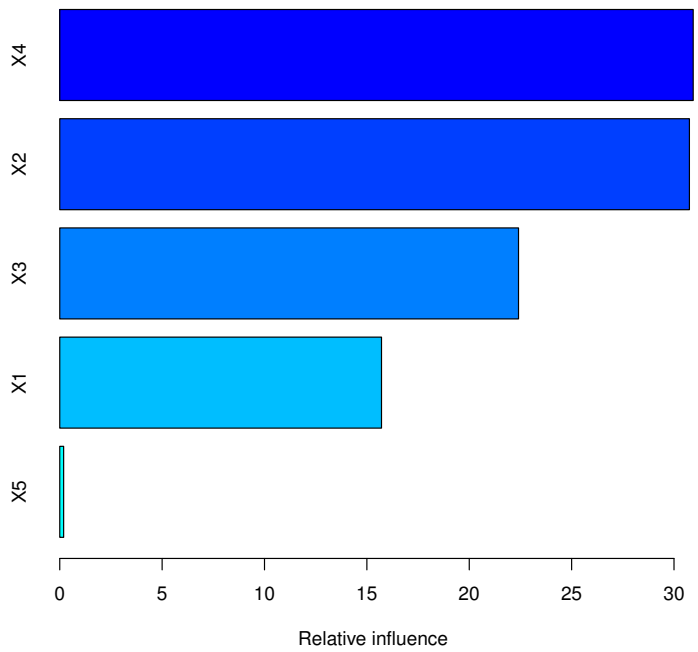
Application - RandomForest

```
> library(randomForest)
> foret = randomForest(Y~  X1+X2+X3+X4+X5, data = data.frame(Xapp,Y=Yapp),mtry=5,ntree=600,nodesize=5)

mty : Number of variables randomly sampled as candidates at each split.
ntree : Number of trees grown.
nodesize : Minimum size of terminal nodes.

> R2 = 0.97
> Q2 = 0.87
```



Application - gbm

```
> library(gbm)
> boost = gbm(Y~  (X1+X2+X3+X4+X5),  n.trees=20000, data = data.frame(Xapp,Y=Yapp),cv.folds=10)
> best.iter <- gbm.perf(boost,method="cv")

> R2 = 0.81
> Q2 = 0.75
```



Outline

Méthodes basées sur les splines de lissage (1/3)

Introduction

- Décomposition ANOVA d'une fonction :

$$y(w) = y_0 + \sum_{i=1}^{d} y_i(w_i) + \sum_{i<j}^{d} y_{ij}(w_i, w_j) + \cdots + y_{1\ldots d}(w_1, \ldots, w_d)$$

- Le métamodèle est souvent une version tronquée de cette décomposition :

$$\hat{y}(w) = y_0 + \sum_{i=1}^{d} y_i(w_i) + \sum_{i<j}^{d} y_{ij}(w_i, w_j)$$

- On suppose que $\hat{y}(w) \in \mathcal{H}$. $\mathcal{H}$ RKHS (Reproducing Kernel Hilbert Space).
- $\mathcal{H}$ admet la décomposition orthogonale suivante : $\mathcal{H} = \{1\} \oplus \left\{\oplus_{j=1}^{K} \mathcal{H}_j\right\}$ où les $\mathcal{H}_j$ sont aussi des RKHS et $K+1$ le nombre de composantes dans la décomposition.
  On a alors : $y_0 \in \{1\}$, $y_i(.) \in \mathcal{H}_i$ et $y_{ij}(.) \in \mathcal{H}_i \otimes \mathcal{H}_j$
- Le métamodèle a la forme suivante : $\hat{y}(w) = b + \sum_{i=1}^{n} c_i \sum_{j=1}^{K} f(\theta_j)\mathbf{k}_j(w_i, w)$

  où $b \in \mathbb{R}$ et $c_i \in \mathbb{R}$

  $f(\theta_j)$ dépend de la méthode utilisée (SS ANOVA, COSSO, ...)

  $\mathbf{k}_j$ le noyau reproduisant de $\mathcal{H}_j$

Méthodes basées sur les splines de lissage (2/3)

Estimation des composantes

- Minimisation de la somme des erreurs quadratiques en pénalisant chaque composante :

$$\min_{\hat{y}(x)\in\mathcal{H}} \left\{ \sum_{i=1}^{n} \left(y(x_i) - \hat{y}(x_i)\right)^2 + \lambda J\left(\hat{y}\right) \right\}$$

- Dans SS-ANOVA : $J\left(\hat{y}\right) = \sum_{j=1}^{K} \frac{1}{\theta_j} \parallel P^j \hat{y} \parallel_{\mathcal{H}}^2$

  avec $\parallel . \parallel_{\mathcal{H}}$ norme dans $\mathcal{H}$ et $P^j\hat{y}$ projection orthogonale de $\hat{y}(w)$ sur $\mathcal{H}_j$

  Réglage du "degré de lissage" $(\lambda/\theta_1, \ldots, \lambda/\theta_K)$ par validation croisée.
  En pratique quand K est grand, on fixe $\theta_i = 1$

- Dans COSSO : $J\left(\hat{y}\right) = \sum_{j=1}^{K} ||P^j\hat{y}||_{\mathcal{H}}$
  Pénalité de type LASSO $\rightarrow$ plusieurs composantes nulles

  Réglage de l'unique paramètre λ par validation croisée

- Dans ACOSSO : $J\left(\hat{y}\right) = \sum_{j=1}^{K} w_j ||P^j\hat{y}||_{\mathcal{H}}$

  Les poids sont estimés à partir des données

Méthodes basées sur les splines de lissage (3/3)

Prise en compte des variables catégorielles

- On peut écrire la décomposition ANOVA tronquée :

$$\hat{y}(w) = y_0 + \sum_{i=1}^{p} h_i(x_i) + + \sum_{i=1}^{q} g_j(z_i) + \sum_{\substack{j=1,\dots,q \\ i=1,\dots,p}} y_{ij}(x_i, z_j)$$

- Choix des RKHS :
  - Pour les variables continues, c'est le plus souvent un espace de Sobolov du second ordre $\mathcal{S}^2 = \{f : f, f' \text{absolument continues et } f'' \in \mathcal{L}^2[0,1]\}$.
    Ainsi $\mathcal{H}_i^{cont} = \mathcal{S}^2, \;\; i = 1, \dots, p$
  - Pour les variables catégorielles, c'est l'ensemble des fonctions de carrés intégrables sur le domaine de z_j (i. e. $\{1, \dots, m_j\}$).
    On le note $\mathcal{H}_j^{cat}, \;\; i = 1, \dots, p$
- SS-ANOVA : pas de pénalisation sur les variables catégorielles
- COSSO : pénalisation sur les variables catégorielles

Implémentation sous R

- SS-ANOVA : package `gss`
- COSSO : package `cosso`
- ACOSSO : fonction `acosso.R` disponible sur le site de Curtis Storlie.
- BSS-ANOVA : fonction `bssanova.R` disponible sur le site de Curtis Storlie.
- ...

Application - acosso

```
> source('acosso.R')

> # les données sont des matrices.
> # Les variables catégorielles ne sont plus définies par factor

           X1        X2       X3 X4 X5
[1,] 135.6388 3.6889888 7.098801  3  3
[2,] 549.7489 4.3109315 9.588638  1  1
[3,] 144.7599 6.3880998 9.714214  3  3

> fit_acosso = acosso(Xapp,Yapp,categorical=c(4,5),order=2,wt.pow=1.6, cv='gcv')

# categorical : l'indice des colonnes des variables catégorielles
# order=2 : ordre de la troncature
# wt.pow :  les poids utilisés dans la pénalité (COSSO : wt.pow=0)
# cv : la méthode pour "choisir" lambda

> R2 = 0.97
> Q2 = 0.93
```

Outline

Principe (dans le cas continu)

- La fonction a approchée $y(x)$ est supposée être la réalisation d'un processus Gaussien :

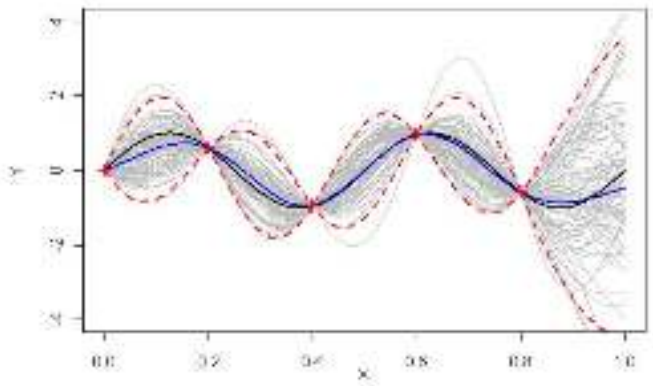
$$Y(x) = f^t(x)\beta + \varepsilon(x)$$

avec $f^t(x)\beta$ tendance déterministe (identique à celle des moindres carrés classiques) et $\varepsilon(x)$ un processus Gaussien de moyenne nulle, de variance σ^2 et de fonction de corrélation K.

- On s'intéresse à la loi de $Y(x)$ conditionnellement aux observations :

$$[Y(x)|\mathbf{y}^n] \sim \mathcal{N}(\mu(x), s^2(x))$$

avec $\mu(x) = m(x) + k^t(x)K^{-1}k(x)(\mathbf{y}^n - F\beta)$ représentant le métamodèle et $s^2(x) = \sigma^2(1 - k^t(x)K^{-1}k(x))$ l'erreur quadratique moyenne



TGP : Treed Gaussian Process

Principe

- Combinaison des arbres de régression et des processus Gaussiens
- Construction d'un arbre de régression sur les variables catégorielles
- Construction d'un métamodèle processus Gaussien sur chaque partition de l'espace déterminé par les branches de l'arbre.
- Le métamodèle s'écrit alors : $\hat{y}(x) = \sum_{k=1}^{K} m_k(x) \mathbf{1}_{\{x \in P_k\}}$ avec K le nombre de partitions, $(\mu_k(x))_{k=1,\dots,K}$ les métamodèles processus Gaussien et $(P_k)_{k=1,\dots,K}$ les partitions de l'espace des entrées.
- Les auteurs de TGP utilisent BCART et une approche full bayésienne pour estimer les paramètres du modèle de krigeage.
- Les variables catégorielles sont codées en binaire (stationnarité dans les partitions)
- $+$ pour les fonctions stationnaires seulement localement.
  - quand les variables continues ne sont pas prédictives pour certaines combinaison

Idée

- Définir une structure de corrélation entre les différents modalités :

$$\text{cor}(\varepsilon(w_1), \varepsilon(w_2)) = cor(\varepsilon_{z_1}(x_1), \varepsilon_{z_2}(x_2))$$

- Pour une variable catégorielle : $\text{cor}(\varepsilon(w_1), \varepsilon(w_2)) = \tau_{z_{11}, z_{12}} K_\phi(x_1, x_2)$
- Pour q variables catégorielles : $\text{cor}(\varepsilon(w_1), \varepsilon(w_2)) = \prod_{j=1}^{q} (\tau_{j, z_{11}, z_{12}} K_\phi(x_1, x_2))$
- La matrice $\mathcal{T}_j = (\tau_{j,r,s})$ est définie positive et constituée de 1 sur la diagonale.
- Fonction de corrélations isotropique (1 variable catégorielle) :

$$\tau_{r,s} = (1 - c)\mathbf{1}_{\{r=s\}} + c, \quad c \in]0, 1[$$

- Fonction de corrélations multiplicative (1 variable catégorielle) :

$$\tau_{r,s} = (1 - c_r c_s)\mathbf{1}_{\{r=s\}} + c_r c_s, \quad c_r \text{ et } c_s \in]0, 1[$$

le nombre d'hyperparamètres augmente avec le nombre de modalités.

Autres exemples

- Structure de covariance adaptée aux variables ordonnées
- Covariance isotropique sur des groupes de modalités d'une variable catégorielle.
- Méthode basée sur des hypersphères :

  Toute matrice de corrélation $\mathcal{T}_j = (\tau_{j,r,s})$ (définie positive et constituée de 1 sur la diagonale) peut se décomposer sur une hypersphère en deux étapes

  - Décomposition de Cholesky
  - Paramétrisation sur une hypersphère de la matrice triangulaire supérieure

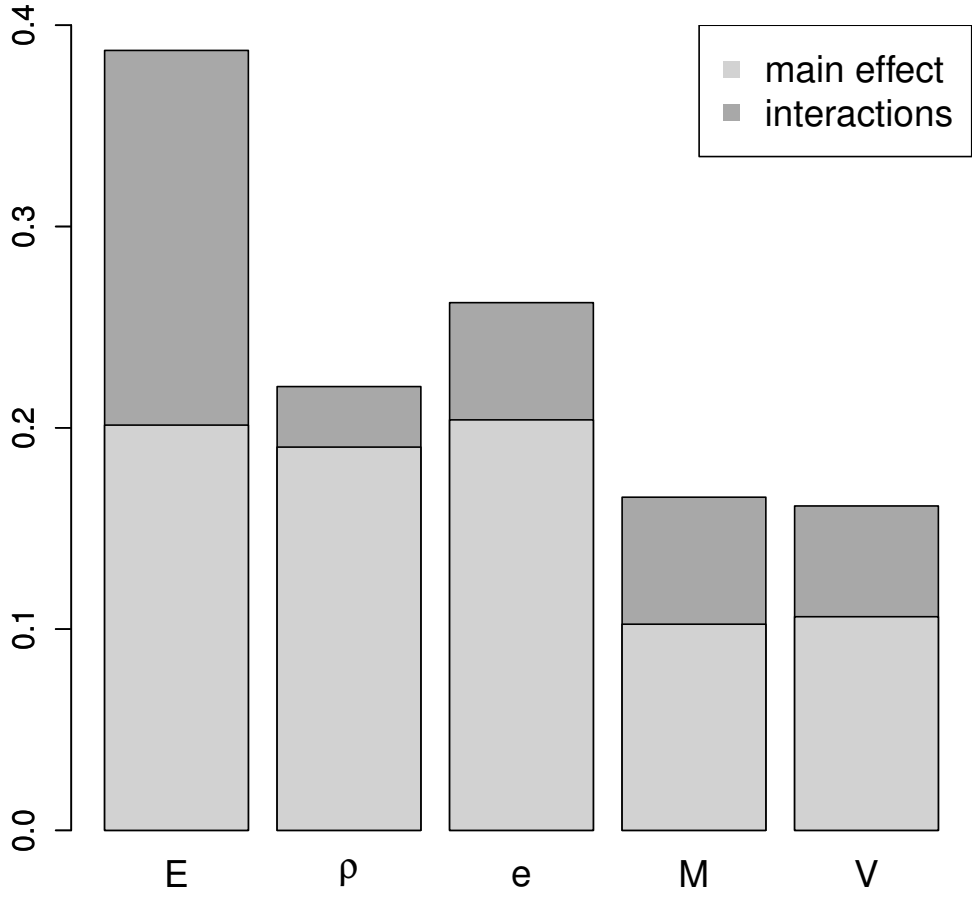
Implémentation sous R

- Treed Gaussian Process : package `tgp`
- Processus Gaussien (variables continues) : package `DiceKriging`
- Définition de structures de covariance : `DKlab2` (consortium ReDice)
  (pas encore disponible sur le CRAN)

Application - Comparaison des résultats

Physical model	Analytical model		Numerical model	
Coefficient	R^2	Q^2	R^2	Q^2
Linear 1	0.73	0.67	0.7	0.67
Linear 2	0.79	0.76	0.76	0.72
Linear 3	0.88	0.85	0.89	0.86
CART	0.75	0.7	0.76	0.69
Random Forest	0.98	0.87	0.98	0.86
GBM	0.81	0.75		
SS-ANOVA	0.91	0.87	0.9	0.84
COSSO	0.96	0.91	0.92	0.89
ACOSSO	0.97	0.93	0.95	0.92
TGP	0.78	0.7	0.84	0.79
GP ind		0.83		
GP iso		0.86		

Utilisation de ACOSSO



Conclusion et perspectives

- Il existe de nombreux types de métamodèles acceptant un mélange de paramètres continus et discrets
- ACOSSO donne de très bons résultats pour notre application
- Perspectives
  - Regression PLS
  - Autres noyaux pour le krigeage
  - Méthodes de cokrigeage
  - Influence du choix du plan d'expériences
  - Taille critique du plan d'expériences en fonction du nombre de modalités et de la régularité de la fonction
  - Influence du choix du nombre de variables catégorielles : 1 avec toutes les combinaisons ou plusieurs avec seulement 2 combinaisons ?

Références (1/4)

✓ Sur la régression linéaire avec variables catégorielles et pénalisation type lasso

J. Gertheiss and G. Tutz.

Sparse modeling of categorial explanatory variables.

The Annals of Applied Statistics, 4 :2150–2180, 2010.

N. Simon and R. Tibshirani.

Standardization and the group lasso penalty.

Technical report, 2011.

S. Tunali and I. Batmaz.

A metamodeling methodology involving both qualitative and quantitative input factors.

European Journal of Operational Research, 150 :437–450, 2003.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight.

Sparsity and smoothness via the fused lasso.

Journal of the Royal Statistical Society Series B, pages 91–108, 2005.

H. Wang and C. Leng.

A note on adaptive group lasso.

Computational Statistics and Data Analysis, 52 :5277–5286, 2008.

M. Yuan and Y. Lin.

Model selection and estimation in regression with grouped variables.

Journal of the Royal Statistical Society, Series B, 68 :49–67, 2006.

Références (2/4)

✓ Sur les arbres de régression et méthodes dérivées

L. Breiman, J. Friedman, R. Olshen, and C. Stone.

Classification and Regression Trees.

Wadsworth and Brooks, 1984.

L. Breiman.

Random forests.

Machine Learning, 45 :5–32, 2001.

H. Chipman, E. George, and R. McCulloch.

Bayesian cart model search (with discussion).

Journal of American Statistical Association, 93 :935–960, 1998.

J. H. Friedman and N. I. Fisher.

Bump hunting in high-dimensional data.

Statistics and Computing, 9 :123–143, 1999.

J. H. Friedman.

Stochastic gradient boosting.

Computational Statistics and Data Analysis, 38 :367–378, 2002.

M. I. Jordan.

Hierarchical mixtures of experts and the EM algorithm.

Neural Computation, 6 :181–214, 1994.

Références (3/4)

✓ Autour des splines de lissage et méthodes dérivées

A. Berlinet and C. Thomas-Agnan.

Reproducing kernel Hilbert spaces in probability and statistics.

Kluwer Academic, 2004.

C. Gu.

Smoothing spline ANOVA models.

Springer, Series in statistics. 2002.

Y. Lin and H. Zhang.

Component selection and smoothing in smoothing spline analysis of variance models.

Annals of Statistics, 34(5) :2272–2297, 2006.

B.J. Reich, C.B. Storlie, and H.D. Bondell.

Variable selection in bayesian smoothing spline anova models : Application to deterministic computer codes.

Technometrics, 51 :110–120, 2009.

C.B. Storlie, H.D. Bondell, B.J. Reich, and H.H. Zhang

Surface Estimation, Variable Selection, and the Nonparametric Oracle Property.

Statistica Sinica, 21 :679–705, 2011.

G. Wahba.

Spline Models for Observational Data.

Society for Industrial and Applied Mathematicsl, USA, 1990.

Références (4/4)

✓ Méthodes basées sur des processus Gaussiens

R. B. Gramacy and H K. H. Lee.

Bayesian treed gaussian process models with an application to computer modeling.

Journal of the American Statistical Association, 2007.

R. B. Gramacy and M. Taddy.

Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an r package for treed gaussian process models.

Technical report, R manual, http ://cran.r-project.org, 2009.

P. Z. G. Qian, H. Wu, and C. F. J. Wu.

Gaussian process models for computer experiments with qualitative and quantitative factors.

Technometrics, 50(3) :383–396, 2008.

P.Z.G. Qian, Q. Zhou and S. Zhou.

A simple approach to emulation for computer models with qualitative and quantitative factors.

Technometrics, 53 :266–273, 2011.

✓ Comparaisons de méthodes pour variables catégorielles

L.P. Swiler, P. D.Hough, P. Qian, Peter,X. Xu and C. Storlies and H. Lee.

Surrogate Models for Mixed Discrete-Continuous Variables.

Studies in Computational Intelligence, Springer International Publishin, 181–202, 2014.

C. B. Storlie, B. J. Reich, J. C. Helton,L. P. Swiler and C. J.Sallaberry

Analysis of computationally demanding models with continuous and categorical inputs.

Reliability Engineering and System Safety, 113 :30–41, 2013.