Plan d'expérience adaptatif et sélection de modèle dans une base multi-résolution.

S. Gadat, S. Déjean, S. Cohen

Institut de Mathématiques de Toulouse
Séminaire INRA BIA

Avril 2011

Plan de l'exposé

Introduction

- Dans cet exposé, on s'intéresse à l'estimation d'une fonction η définie sur un hypercube Ω.
- On considère le modèle de bruit blanc : on mesurera une fonction f définie par

$$\forall x \in \Omega \qquad f(x) = \eta(x) + \sigma\xi(x).$$

- Le bruit ξ est supposé gaussien et la variance σ^2 est constante sur Ω et inconnue.
- On observe le signal en différents points $(x_1, \ldots x_n) \in \Omega^n$.

Problématique plan d'expérience

- On adopte en plus le point de vue plan d'expérience : on choisit les points x_i où f est mesurée.
- Adaptation du modèle de bruit blanc : pour deux mesures de f distinctes, les bruits de mesure sont supposés indépendants.
- Objectif : Obtenir la "meilleure" approximation de η avec le minimum de points de mesures.
- Nombreuses applications
  - Données cliniques
  - Crash tests
  - simulation de gros codes numériques

Base multi-résolution

- On va utiliser une base multi-résolution notée $(\Lambda_{r,t})_{r,t}$:

$$\forall r \in \mathbb{N} \quad \forall t \in \{0 \dots 2^r - 1\} \quad \forall x \in \Omega \qquad \Lambda_{r,t} = 2^{r/2} \Lambda_{0,0}(2^r x - t).$$

- On suppose que η se décompose dans cette base :

$$\eta = \sum_{r,t} \theta_{r,t} \Lambda_{r,t}.$$

- La base $(\Lambda_{r,t})_{r,t}$ n'est pas nécessairement orthonormée. On peut ainsi considérer des bases de Schauder du type les fonctions triangles (primitives de la base de Haar).
- On omet volontairement les problématiques théoriques décrivant les espaces de fonctions atteignables par de telles décompositions (Besov, Besov homogènes, . . .).

Objectifs

- On cherchera conjointement un sous-ensemble $\Lambda_{(r,t)\in I}$ de la base multi-résolution initiale ainsi qu'un plan d'expérience $\mathbf{x}_n = \{x_1, \ldots x_n\}$.
- La question qu'on cherche à résoudre : peut-on avec un "simple" modèle linéaire et peu de données développer une stratégie d'estimation de η dans la base $(\Lambda_{r,t})_{r,t}$.
- Extension naturelle : faire la même chose avec des modèles de prédiction plus sophistiqués (Ridge régression, pénalisations ℓ^1, Elasstic Net, ...).
- Approche : On n'utilisera pas de statistiques non paramétriques mais plutôt des techniques de Machine Learning (Boosting et MARS de Friedman, Sélection de variables à la Vapnik).

Plan de l'exposé

Notations

Étant donnés $\mathbf{x} = \{x_1, \ldots x_n\}$ un plan d'expérience et I un sous-ensemble de couples (r, t) résolution + translation, on note :

- $\hat{\eta}_{\mathbf{x},I}$ une estimation de η dépendant uniquement de $\mathbf{x}$, I et des mesures effectuées $(f(x_i))_{i=1\ldots n}$.
- $\bar{\Lambda}_I(\mathbf{x})$ la matrice de régression de taille $(|I| + 1) \times |\mathbf{x}|$.
- $M_{\mathbf{x},I}$ est la matrice d'information du plan d'expérience :

$$M_{\mathbf{x},I} = \bar{\Lambda}_I(\mathbf{x})\bar{\Lambda}_I(\mathbf{x})'.$$

- On désigne par $\mu_{1,1}(I)$ la matrice des moments d'ordre 1

$$\mu_{1,1}(I) = \int_\Omega \bar{\Lambda}_I(u)\bar{\Lambda}_I(u)'du$$

- On mesure l'efficacité d'estimation par modèle linéaire sur $(\mathbf{x}, I)$ par le biais de l'IMSE :

$$J(\mathbf{x}, I) = \int_\Omega \mathbb{E}\left[\hat{\eta}_{\mathbf{x},I} - \eta\right]^2.$$

Décomposition du risque

Proposition

J se décompose en

$$J(\mathbf{x}, I) = \int_{\Omega} \left[\mathbb{E}\hat{\eta}_{\mathbf{x},I} - \eta\right]^2 + \int_{\Omega} Var(\hat{\eta}_{\mathbf{x},I}(u))du,$$

avec

$$\int_{\Omega} Var(\hat{\eta}_{\mathbf{x},I}(u))du = \sigma^2 Tr\left(\mu_{1,1}(I)M_{\mathbf{x},I}^{-1}\right).$$

- Le terme de variance est d'autant plus grand que la matrice $M_{\mathbf{x},I}$ est difficile à inverser.
- Dans la théorie des plans d'expériences, les bons designs sont souvent ceux qui rendent $M_{\mathbf{x},I}$ le plus inversible possible en négligeant le terme de biais engendré par le premier terme.
- Dans l'approche sélection de modèle, on ne dispose pas du levier de choisir le lieu des observations.

Algorithme

- On adopte un point de vue séquentiel : on va construire récursivement des designs $\mathbf{x}_n$ et des sous-ensembles I_n.
- On choisi une structure de designs telle que

$$\mathbf{x}_{n+1} = \mathbf{x}_n \cup x_{n+1} \qquad \text{on ne "jette" pas une mesure.}$$

- Les sous-ensembles I_n suivront une structure "pyramidale"

$$I_n \Delta I_{n+1} = (r_{n+1}, t_{n+1}).$$

- Description sommaire de l'algorithme :
  1. Initialisation de $I_0 = \{(0,0),(1,0),(1,1)\}$ et $(\mathbf{x}_0)$ optimal pour le terme de variance (voir plus loin).
  2. À l'étape n :

     $\mathcal{MS}$ Choix d'une mise à jour de I_{n+1}.

     $\mathcal{OD}$ Calcul d'un nouveau point x_{n+1} "optimal" pour l'expérience $(\mathbf{x}_n \cup x, I_{n+1})$.

Commentaires sur l'algorithme

Quelques idées générales sur l'algorithme dans l'optique $\mathcal{MS} + \mathcal{OD}$.

- Contrôler la variance : c'est le rôle du choix du design étant donné I et donc de $\mathcal{OD}$.
- Plusieurs critères découlent naturellement de la formule de la variance dans l'IMSE.
- Contrôler le biais : c'est le rôle naturel de $\mathcal{MS}$ puisqu'on a jamais vu un terme de variance réduire un biais (du moins je crois...)
- Il est nécessaire de donner un critère d'efficacité de réduction du biais pour la phase $\mathcal{MS}$.

Plan de l'exposé

$\mathcal{OD}$: Présentation générale

Rappelons que l'on connait I_{n+1} lors de cette étape. On doit fixer x_{n+1} sans connaître au préalable $f(x_{n+1})$. Ainsi, nous ne pouvons que jouer sur le terme de variance

$$\int_\Omega Var(\hat{\eta}_{\mathbf{x},I}(u))du = \sigma^2 Tr\left(\mu_{1,1}(I)M_{\mathbf{x},I}^{-1}\right).$$

- Dans notre stratégie séquentielle, il s'agira donc de trouver x_{n+1} selon une formule du type

$$x_{n+1} = \arg\min_{\xi\in\Omega} F(M_{\mathbf{x}_n\cup\xi,I_{n+1}}, I_{n+1}), \tag{1}$$

  où F quantifie la variance de l'estimation.
- On prend classiquement $F(M,I) = Tr(\mu_{1,1}(I)M^{-1})$ (designs A-optimaux). On peut également opter pour $F(M,I) = \det M^{-1}$ (designs D-optimaux), ...
- La résolution de (1) est parfois explicite (T-systèmes, base de Haar, système de Schauder), et parfois non...

$\mathcal{OD}$: Factorisation de Schur

Tous les algorithmes d'optimisation de (1) sont basés sur les idées suivantes :

- $$M_{\mathbf{x}_n \cup \xi, I} = M_{\mathbf{x}, I} + M_{\xi, I} = M_{\mathbf{x}, I} + \bar{\Lambda}_I(\mathbf{x}) \bar{\Lambda}_I(\mathbf{x})'.$$

- **Proposition (Factorisation de Schur)**
  On peut déduire facilement l'inverse de $M_{\mathbf{x}_n \cup \xi, I}$ à partir de $M^{-1}_{\mathbf{x}_n, I}$:

$$M^{-1}_{\mathbf{x}_n \cup \xi, I} = M^{-1}_{\mathbf{x}_n} \left[Id - \frac{\bar{\Lambda}_I(\xi)' \bar{\Lambda}_I(\xi) M^{-1}_{\mathbf{x}_n}}{1 + \bar{\Lambda}_I(\xi)' M^{-1}_{\mathbf{x}_n} \bar{\Lambda}_I(\xi)} \right]$$

  - Ainsi, on a pour toute matrice carrée μ :

$$Tr\left(\mu M^{-1}_{\mathbf{x}_n \cup \xi, I}\right) = Tr\left(\mu M^{-1}_{\mathbf{x}_n, I}\right) - \frac{\bar{\Lambda}_I(\xi)' M^{-1}_{\mathbf{x}_n, I} \mu M^{-1}_{\mathbf{x}_n, I} \bar{\Lambda}_I(\xi)}{1 + \bar{\Lambda}_I(\xi)' M^{-1}_{\mathbf{x}_n} \bar{\Lambda}_I(\xi)}$$

$\mathcal{OD}$: Base de Haar/ Base de Schauder

- En exploitant l'aspect constant par morceaux de la base de Haar, il est facile de concevoir une localisation numérique rapide des solutions de (1).
- Pour la base de Schauder (base des triangles), c'est plus délicat. On note I le sous-ensemble courant de fonctions et on note $\mathcal{S}(I)$ les singularités des $\Lambda_{(r,t)}, (r,t) \in I$, alors on a le théorème suivant :

Théorème (Plans D/A-optimaux - base de Schauder)

Les solutions de (1) pour la base de Schauder avec

- $F(M,I) = Tr(\mu_{1,1}(I)M^{-1}(I))$,
- $F(M,I) = \det M^{-1}$,
- $F(M,I) = \det(M + \alpha Id)^{-1}, \quad \forall \alpha > 0.$

sont incluses dans $\mathcal{S}(I)$.

- C'est un résultat très pratique numériquement pour attraper les designs optimaux !
- À noter que ce résultat s'étent à la situation où les designs ne sont plus séquentiels...

$\mathcal{OD}$: Bases générales

- Il est difficile de donner un résultat de localisation explicite pour les bases de Meyer, Daubechies, . . .
- On est obligé de passer par un algorithme d'optimisation glouton (parcours exhaustif de points dyadiques, algorithme de gradient, . . .).
- Une localisation plus exlicite est pour le moment un problème ouvert sur de telles bases...
- Un premier pas serait de comprendre la maximisation de la fonctionnelle

$$\xi \mapsto \bar{\Lambda}_I(\xi)' \Sigma \bar{\Lambda}_I(\xi),$$

où Σ est n'importe quelle matrice symétrique positive.

$\mathcal{MS}$: Présentation générale

- Étant donné $(\mathbf{x}_n, I_n)$, on veut trouver I_{n+1} qui va améliorer le biais.
- Problème : on ne peut pas tester toutes les fonctions possibles et imaginables (il y en a trop !) et f n'est pas mesurée une nouvelle fois dans cette étape.
- Il faut donc une "heuristique".
- On utilise trois idées :
  - Boosting sur les zones de Ω qui possèdent un fort biais d'estimation.
  - Localisation des fonctions de la base multi-résolution : chaque $\Lambda_{r,t}$ possède une zone d'influence privilégiée $[2^{-r}t; 2^{-r}t + 1]$.
  - Structure arborescente déjà étudiée dans le MARS de Friedman : on ajoutera un descendant direct à I_n ou soustraira une fonction de I_n.

$\mathcal{MS}$: Évaluation du biais (1)

Il s'agit ici de traduire l'importance de chaque fonction de $(\Lambda_{r,t})_{(r,t)\in I_n}$ pour la prédiction de η. On rappelle que le biais est donné par :

$$B_{\mathbf{x}_n,I_n} = \int_\Omega (\mathbb{E}\hat{\eta}_{\mathbf{x}_n,I_n} - \eta)^2$$

- On a choisi d'opter pour une idée s'inspirant des idées de Vapnik pour sélectionner des variables dans les SVM.
- $\mathbb{E}\hat{\eta}_{\mathbf{x}_n,I_n}$ s'écrit :

$$\mathbb{E}\hat{\eta}_{\mathbf{x}_n,I_n} = \sum_{(r,t)\in I_n} \theta_{r,t,I_n}\Lambda_{r,t} + \beta_{I_n}$$

- On quantifie l'importance de chaque $\Lambda_{r,t}$ par son influence sur le biais B. Comment évolue B lorsque je modifie légèrement θ_{r,t,I_n} ?
- Plus $|\partial_{r,t,I_n}B|$ est grand et plus $\Lambda_{r,t}$ influence le biais de régression. Inversement : plus il est petit et plus la fonction elle-même n'influence pas la régression.

$\mathcal{MS}$: Évaluation du biais (2)

On peut formellement écrire la dérivée :

$$\partial_{r,t,I_n} B = 2 \int_{\Omega} \Lambda_{r,t}(u)[\mathbb{E}\hat{\eta}_{\mathbf{x}_n,I_n} - \eta](u)du.$$

Bien sûr, $\partial_{r,t,I_n} B$ n'est pas calculable en pratique mais peut être approché par simulation sur les données :

- On estime par validation croisée bootstrapée le biais $[\mathbb{E}\hat{\eta}_{\mathbf{x}_n,I_n} - \eta](x_i), \forall x_i \in \mathbf{x}_n$.
- On estime le biais $[\hat{\eta}_{\mathbf{x}_n,I_n} - \eta](x), x \in \Omega$ par le biais d'un noyau.
- On conclut par intégration pour obtenir $|\widehat{\partial_{r,t,I_n} B}|$.

$\mathcal{MS}$: Mise à jour stochastique de I_n (1)

- Si $I \mapsto E(I)$ désigne une fonction qu'on souhaite minimiser sur un espace dénombrable $I \in D$, une approche consiste à se munir d'une règle de parcours Markovienne dans D (probabilité de transition $Q(I, \tilde{I})$) et à fabriquer la chaîne de Markov
  - $I_0 \in D$.
  - Pour tout $n \in \mathbb{N}$, on tire $\tilde{I}_{n+1}$ selon $P(I_n, .)$ et on accepte $I_{n+1} = \tilde{I}_n$ avec la probabilité

$$Q(I_n, \tilde{I}_n) = 1 \wedge e^{-\frac{\Delta E(I_n \to I_{n+1})}{T_n}} \frac{P(I_n, \tilde{I}_n)}{P(\tilde{I}_n, I_n)}.$$

    Sinon $I_{n+1} = I_n$.
- On se sert des quantités $|\widehat{\partial_{r,t,I_n} B}|$ pour une proposition dans un algorithme d'acceptation/rejet avec température décroissante.

$\mathcal{MS}$: Mise à jour stochastique de I_n (2)

- On favorise les fils des éléments ayant un fort coefficient $|\widehat{\partial_{r,t,I_n}B}|$.
- Étant donné une distribution de probabilité $(p_{birth}, p_{deletion})$, l'algorithme est alors décrit par
  1. Sélection d'un type de mouvement $\alpha_n = (+1 p_{birth}; -1 p_{deletion})$.
  2. Calcul des $|\widehat{\partial_{r,t,I_n}B}|$ et on fixe $P(I_n, \tilde{I}_n) \propto |\widehat{\partial_{r,t,I_n}B}|^{\alpha_n}$.
  3. On estime la variance σ par maximum de vraisemblance dans le modèle linéaire.
  4. On estime le différentiel énergétique

$$\Delta E((\mathbf{x}_n, I_n) \to (\mathbf{x}_n, I_{n+1}) = \Delta B + \hat{\sigma}_n \Delta V.$$

  5. On accepte la transition I_{n+1} avec une probabilité $1 \wedge e^{-\frac{\Delta E((\mathbf{x}_n, I_n) \to (\mathbf{x}_n, I_{n+1}))}{T_n}} \frac{P(I_n, \tilde{I}_n)}{P(\tilde{I}_n, I_n)}$.

Plan de l'exposé

Protocole expérimental

- Algorithme présenté avec les bases de Haar, Schauder et Meyer.
- Comparaison avec Adaptive Lasso (Zhou, 2006) et seuillage d'ondelettes sur designs non réguliers (Amato *et al.*, 2006).
- Les IMSE ont été calculés après de très nombreux runs.
- $\Omega = [0; 1]$ ou $\Omega = [0; 1]^2$ mais ce n'est pas réellement restrictif...

Mélange de Gaussiennes : Données

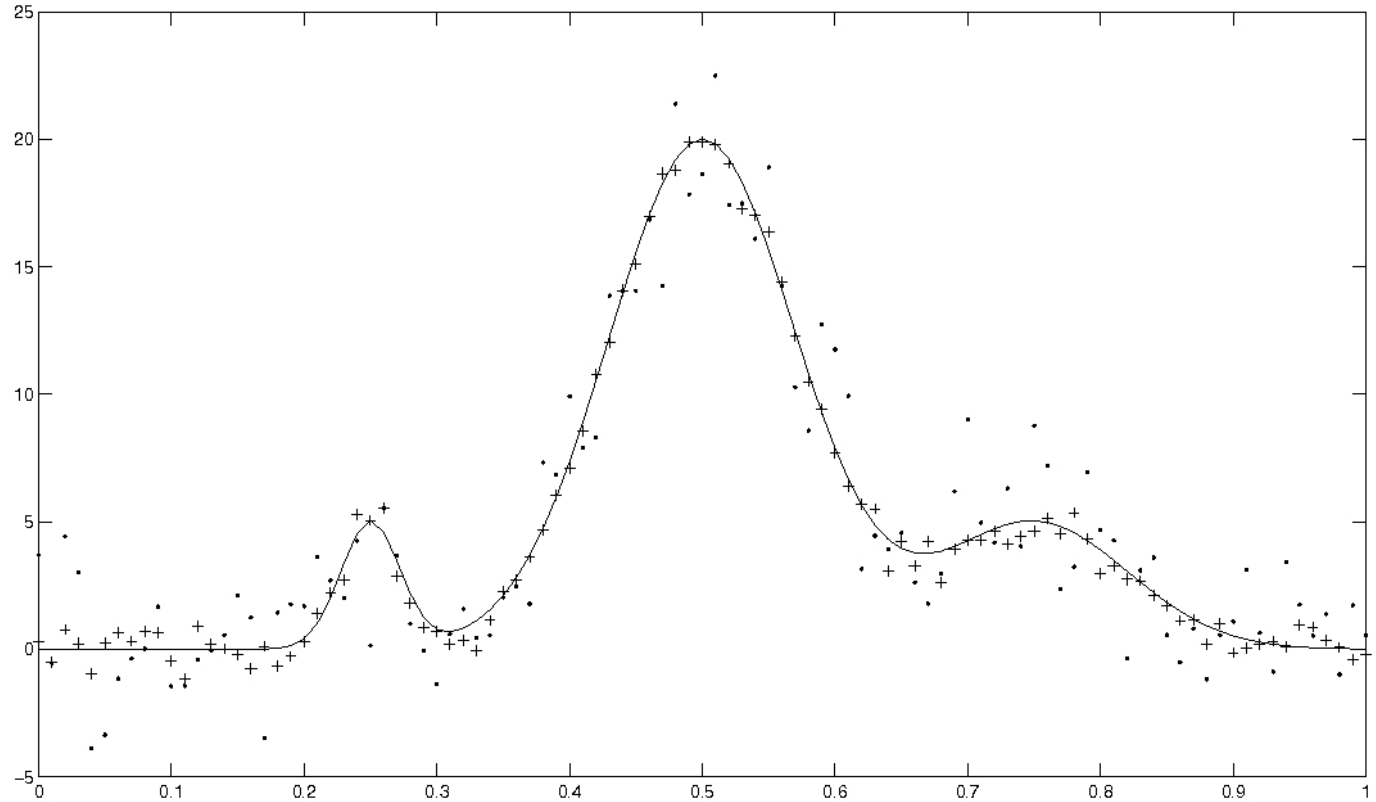

FIGURE: η et quelques réalisations de $f(x)$ avec $\sigma = 0.5$ (croix) ou $\sigma = 2$ (points).

Mélange de Gaussiennes

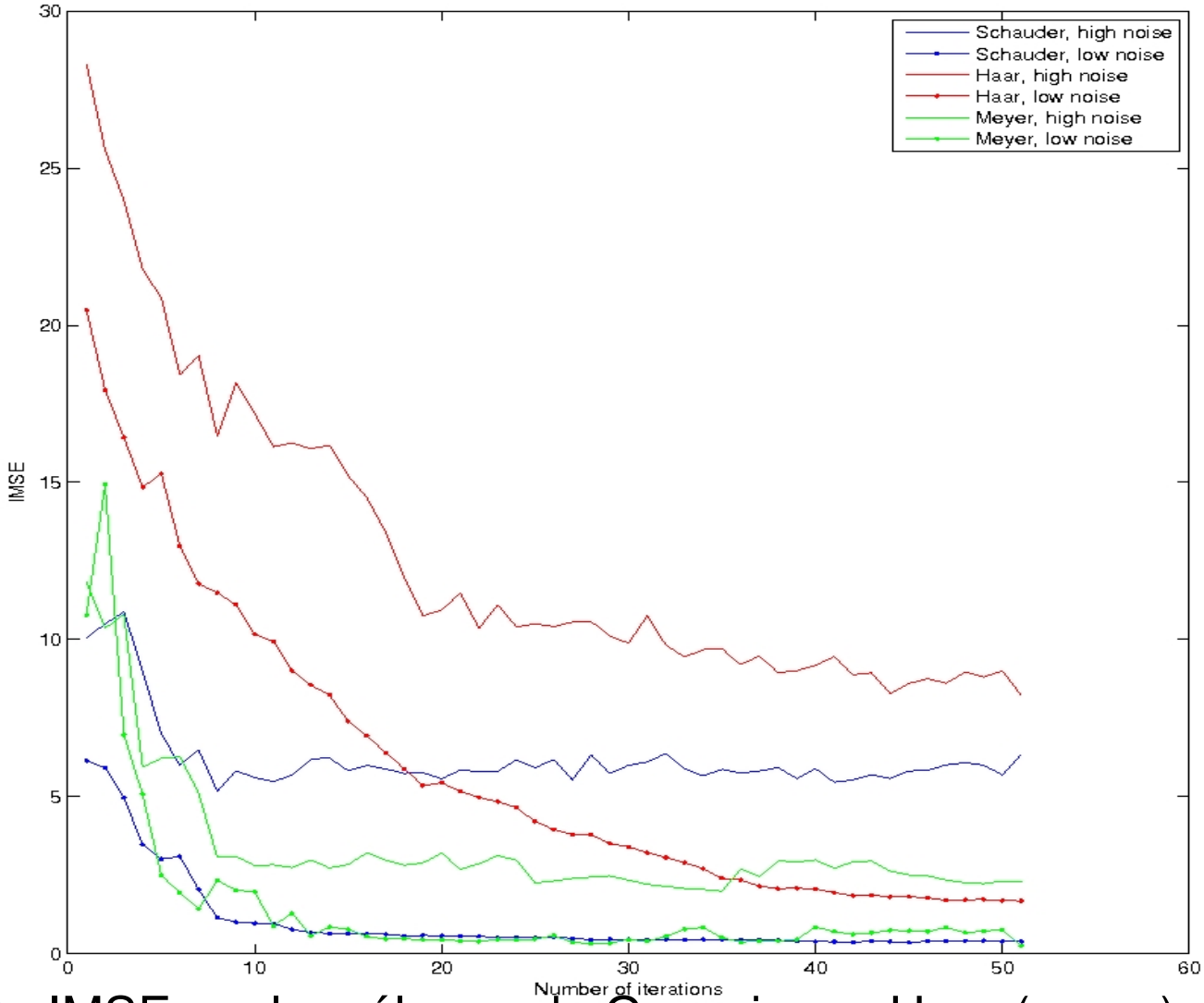

FIGURE: IMSE sur le mélange de Gaussienne Haar (rouge), Schauder (bleu) and Meyer (vert).

Mélange de Gaussiennes : IMSE

Method	IMSE (n=10)	IMSE (n=30)	IMSE (n=50)
Sequential Haar	10.1	3.4	1.7
Sequential Schauder	1.0	0.45	0.4
Sequential Meyer	0.9	0.4	0.38
Ada-Lasso Haar Random	70.8	75.7	56.2
Ada-Lasso Haar Regular	69	42.9	31
Ada-Lasso Schauder Random	50.2	20.8	14.3
Ada-Lasso Schauder Regular	13.6	13.9	12.3
Ada-Lasso Meyer Random	116.4	66.8	72.6
Ada-Lasso Meyer Regular	290	47.8	45.2
Wavelet Kernel Penalized D6 Random	8.2	10.3	1.8
Wavelet Kernel Penalized D6 Regular	4.9	1.0	0.9
Wavelet Kernel Penalized S6 Random	5.2	2.1	0.4
Wavelet Kernel Penalized S6 Regular	83.5	27.7	0.4

FIGURE: IMSE sur les données synthétiques avec bruit faible.

L'algorithme se montre relativement performant avec un très petit nombre de points... C'est moins vrai pour $n = 50$.

Mélange de Gaussiennes : IMSE

Method	IMSE (n=10)	IMSE (n=30)	IMSE (n=50)
Sequential Haar	17.2	9.9	9.0
Sequential Schauder	5.6	5.9	5.6
Sequential Meyer	2.8	2.3	2.3
Ada-Lasso Haar Random	85	71.6	71.5
Ada-Lasso Haar Regular	71.1	50.6	43.1
Ada-Lasso Schauder Random	24.3	37.3	24.1
Ada-Lasso Schauder Regular	16.9	17.1	12.2
Ada-Lasso Meyer Random	155	195	301
Ada-Lasso Meyer Regular	282	49	43
Wavelet Kernel Penalized D6 Random	21.4	2.5	22.9
Wavelet Kernel Penalized D6 Regular	15.5	11.9	2.7
Wavelet Kernel Penalized S6 Random	8.5	4.1	2.4
Wavelet Kernel Penalized S6 Regular	4.0	3.9	2.2

FIGURE: IMSE sur les données synthétiques avec bruit fort.

L'algorithme se montre toujours relativement performant avec un très petit nombre de points... et c'est toujours moins vrai pour $n = 50$.

Crash Test de Motos (Silvermann 1985) IMSE

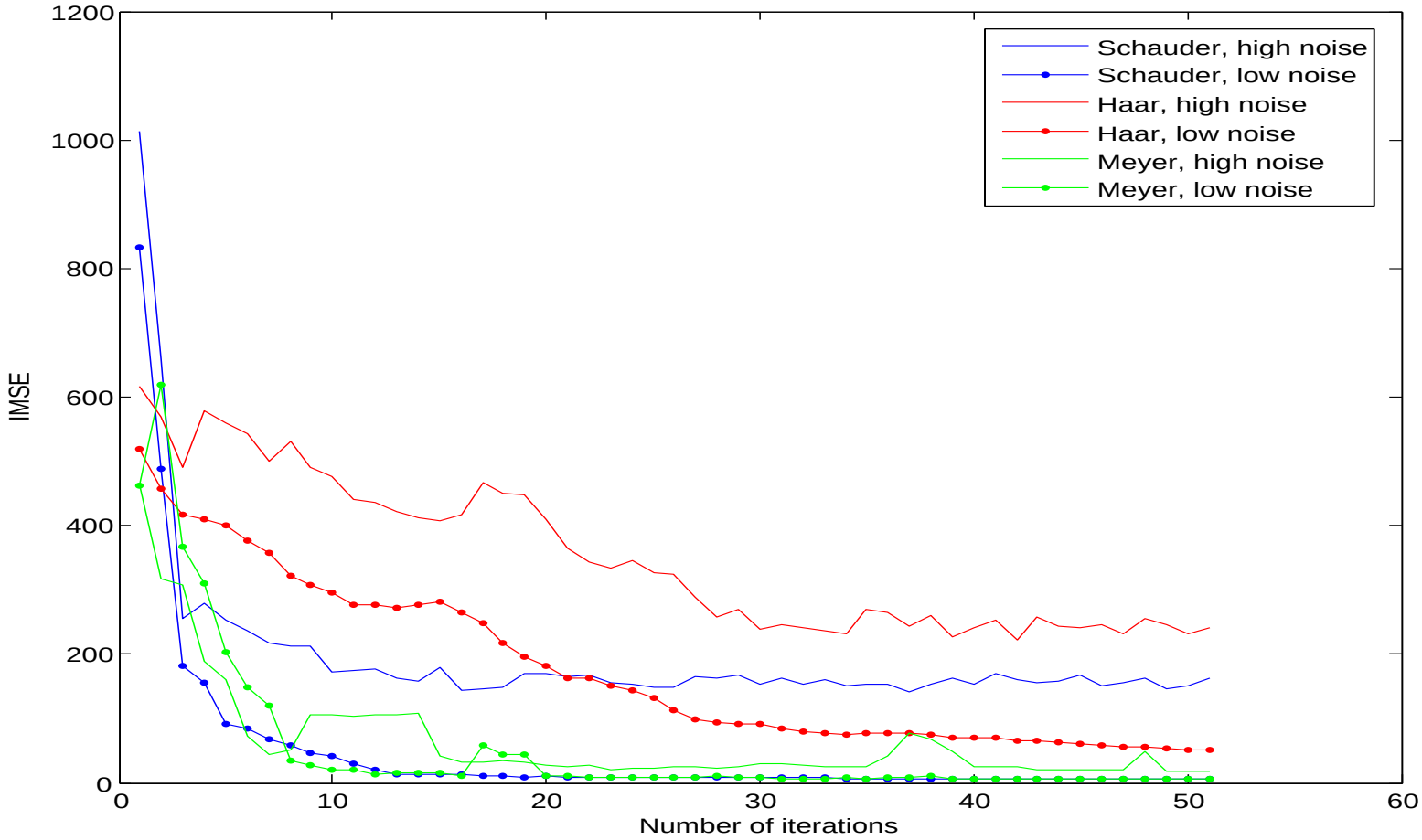

FIGURE: IMSE sur les données de Crashtest Haar (rouge), Schauder (bleu) and Meyer (vert).

Crash Test de Motos (Silvermann 1985) Interpolation

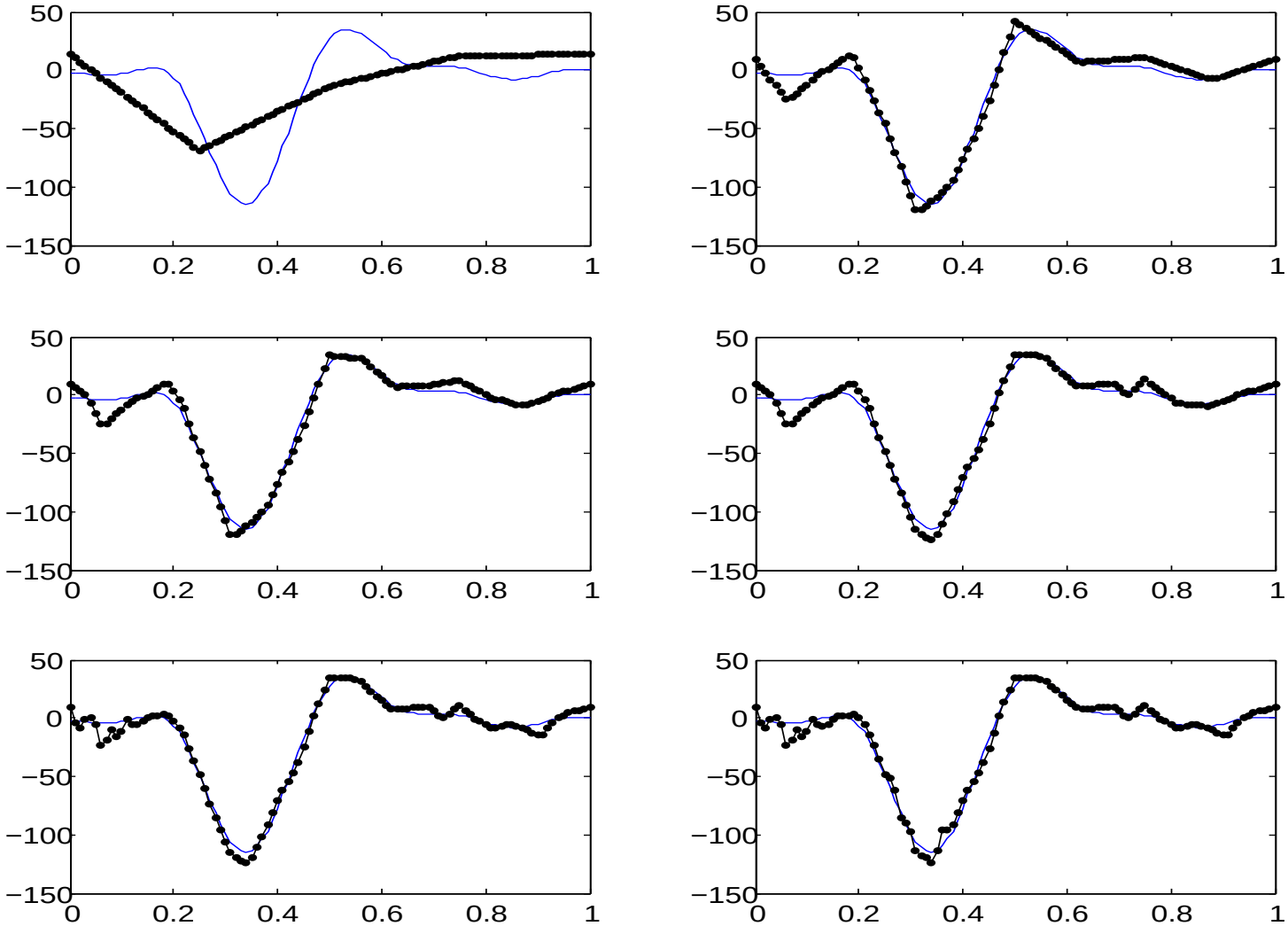

FIGURE: Interpolation avec bruit fort ($\sigma = 10$) iterations 0 (a), 10 (b), 20 (c), 30 (d), 40 (e), 50 (f). Bases de Schauder.

Crash Test de Motos (Silvermann 1985) Interpolation

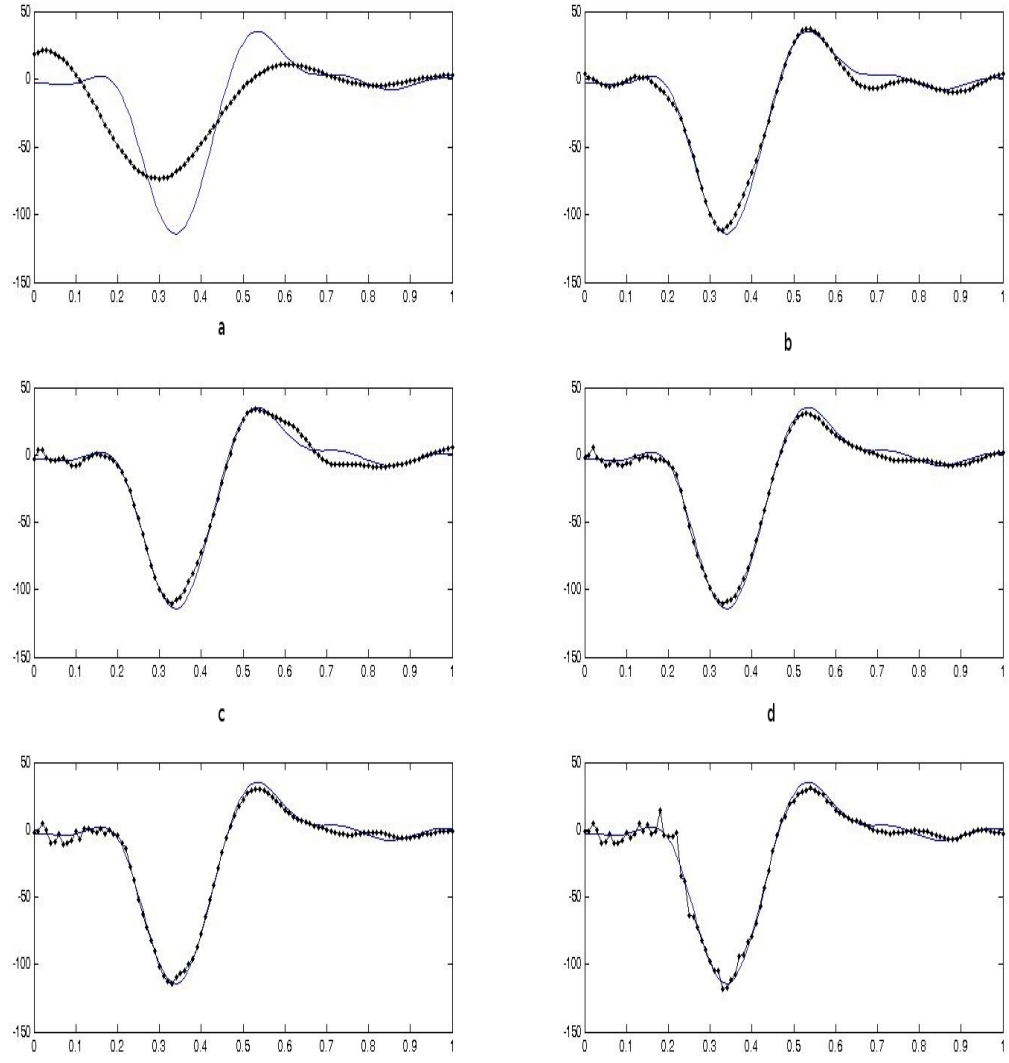

FIGURE: Interpolation avec bruit fort ($\sigma = 10$) iterations 0 (a), 10 (b), 20 (c), 30 (d), 40 (e), 50 (f). Bases de Meyer.

Crash Test de Motos (Silvermann 1985) IMSE

Method	IMSE (n=10)	IMSE (n=30)	IMSE (n=50)
Sequential Haar	296	91	50
Sequential Schauder	41.9	7.5	6.5
Sequential Meyer	19.7	7.4	6.0
Wavelet Kernel Penalized D6 Random	1549	26.4	9
Wavelet Kernel Penalized D6 Regular	458	15	12
Wavelet Kernel Penalized S6 Random	188	154	8.9
Wavelet Kernel Penalized S6 Regular	28.4	11.3	9.5

FIGURE: IMSE - bruit faible.

Method	IMSE (n=10)	IMSE (n=30)	IMSE (n=50)
Sequential Haar	477	239	232
Sequential Schauder	171	153	152
Sequential Meyer	104	28.5	18.3
Wavelet Kernel Penalized D6 Random	1074	158	93
Wavelet Kernel Penalized D6 Regular	556	115.7	135
Wavelet Kernel Penalized S6 Random	180	129	30
Wavelet Kernel Penalized S6 Regular	122	59	18

FIGURE: IMSE - bruit fort.

Plan de l'exposé

Conclusion

Différents points à améliorer théoriquement

- Convergence de l'algorithme de sélection de modèle couplé ?
- Localisation pour des bases plus générales que des ondelettes de Haar ou Schauder ?
- Prendre en compte des termes de discrépance pour "étaler" le design ?
- Développer un algorithme séquentiel de construction de design optimal pour des méthodes pénalisées.

Et en pratique :

- Développer un package propre.
- Assainir dans le cas de bases générales les algorithmes 'gloutons'. Il y a des disparités de temps de calcul impportant : pour n=50, moins de 30 secondes pour la base de Schauder, plus de 5 minutes pour la base de Meyer...

Plan de l'exposé

Extension au cas ℓ^1

- Est-ce utile de mentionner l'intérêt d'une stratégie explicite pour construire des plans d'expérience apte à faire tourner les modèles linéaires pénalisés ℓ^1 ?
- Pourquoi c'est difficile :
  - Les critères théoriques sont bien décrits dans van de Geer et Bühlmann, EJS, 2010.
  - Ils sont vérifiés avec grande probabilité quand les vecteurs servant à former les matrices d'information (ou de Gram) suivent certaines lois (gaussiennes, . . .).
  - En général, les critères théoriques sont des critères de variance (cf lemme 10.1 de van de Geer et Bühlmann, EJS, 2010) et sont des conditions sur la plus petite valeur propre de $\Sigma = X'X$.
- Peut-on envisager une construction récursive de tels designs d'un point de vue 'computationnel' ?

Extension au cas ℓ^1

- On se place dans un modèle

$$f(x) = \sum_i \theta_i \Lambda_i(x) + \sigma\xi(x), \qquad \text{où} \qquad \xi \sim \mathcal{N}(0,1).$$

- On connait approximativement la variance des méthodes pénalisées :
  - H désigne l'opposée de la Hessienne de la vraisemblance
  - si ϕ est la densité de la gaussienne centrée réduite et G la matrice diagonale

$$G(\theta,\sigma) = \frac{2}{\sigma} diag\left(\phi(\theta_1/\sigma), \ldots, \phi(\theta_p/\sigma)\right)$$

- Pour λ le coefficient de pénalisation utilisé

$$Var(\hat{\theta}) = (H(\hat{\theta}) + \lambda G(\hat{\theta},\sigma))^{-1} \Sigma(\hat{\theta}) (H(\hat{\theta}) + \lambda G(\hat{\theta},\sigma))$$

- L'idée serait donc d'optimiser récursivement le déterminant de cette matrice (le maximiser) afin de se prémunir de la présence de 'petites' valeurs propres.

Un théorème de convergence

- Dans le cas d'un dictionnaire de taille fixe, la stratégie de design adaptative est consistante : on peut démontrer la convergence :

Théorème

Si η_Λ est la projection de η dans la famille Λ_I, alors il existe une constante C telle que

$$\|\hat{\theta}_n - \theta\|_\infty \leq C\sqrt{\frac{\log n}{n}}$$

- Qu'en est-il du cas d'un dictionnaire finie avec une norme ℓ^1 ?