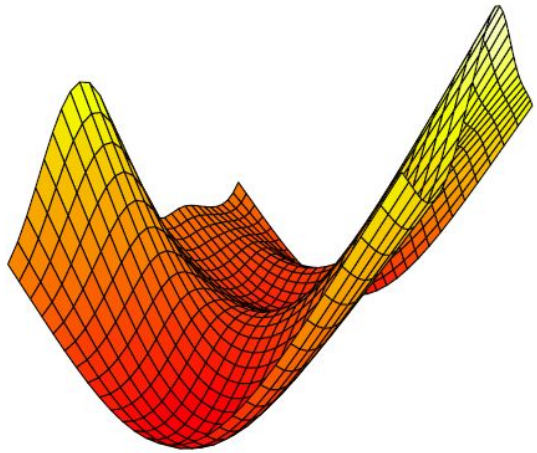
Modèles Additifs Semi-paramétriques



Yannig Goude

EDF R&D yannig.goude@edf.Fr

- Considérons une variable réelle y et les variables explicatives $x_1, ..., x_p$
- Un modèle additif lisse a la structure suivante:

$$y_i = X_i\beta + f_1(x_{1,i}) + f_2(x_{2,i}) + f_3(x_{3,i}, x_{4,i}) + ... + \varepsilon_i$$

- $X_i\beta$ partie linéaire du modèle
- functions f_j sont des fonctions supposées **lisses**
- ε_i:
  - iid, (possiblement AR(1))
  - $E(\varepsilon_i) = 0, V(\varepsilon_i) = \sigma^2$
  - normalité si besoin (test...)

Plus précisément, nous cherchons à résoudre le problème d'optimisation suivant:

$$min_{\beta, f_j} ||y - X\beta - f_1(x_1) - f_2(x_2) + ...||^2 + \lambda_1 \int f_1^{''}(x)^2 dx + \lambda_2 \int f_2^{''}(x)^2 dx + ...$$

- les f_j sont estimés par régression sur une base de spline

$$f_j(x) = \sum_{q=1}^{k_j} a_{j,q}(x)\beta_{j,q}$$

- alors le modèle additif s'écrit

$$y_i = X_i\beta + \sum_{q=1}^{k_1} a_{1,q}(x)\beta_{1,q} + \sum_{q=1}^{k_2} a_{2,q}(x)\beta_{2,q} + ... + \varepsilon_i$$

Inconnues du problème:

- choix de la base de spline, nombre et position des noeuds k_j
- β et $a_{j,q}$

Idée $\Rightarrow$ prendre k_j grand et procéder par régression spline pénalisée (ridge regression sur la base de spline)

le problème d'optimisation s'écrit:

$$min_{\beta,f_j}||y - X\beta - f_1(x_1) - f_2(x_2) + ...||^2 + \lambda_1 \int f_1^{''}(x)^2 dx + \lambda_2 \int f_2^{''}(x)^2 dx + ...$$

ce qui devient alors un problème de régression linéaire pénalisée classique

$$min_\beta ||y - X\beta||^2 + \sum \lambda_j \beta^T S_j \beta$$

- avec $\int f_j^{''}(x)^2 dx$ pouvant s'écrire $\beta^T S_j \beta$
- incorporant $a_{j,q}(x_i)$ dans X_i (ie en incorporant dans X et β les bases de splines et les coefficients de régression associés)

Solution:

$$\widehat{\beta}_\lambda = (X^T X + \sum \lambda_j S_j)^{-1} X^T y$$

les fonctions splines sont des fonctions polynomiales par morceau. Elles sont généralement définies ainsi:

- un ensemble de noeuds
- leurs degrés

Soit $[a, b]$ un intervalle, des noeuds $x_1, ..., x_k$ définissant une partition Δ de $[a, b]$: $a = x_0 < x_1 < .. < x_{k+1} = b$, et m un entier, l'ensemble des fonction splines d'ordre m et de noeuds $x_1, ..., x_k$ est $P_m(\Delta) \cap \mathcal{C}^{m-2}([a, b])$

- $P_m(\Delta)$ l'ensemble des polynômes par morceau d'ordre m sur la partition Δ
- $\mathcal{C}^{m-2}$ l'ensemble des fonctions continue, dérivée $m - 2$e continue

ex: les splines cubiques, $m = 4$

De nombreuses bases de splines sont disponibles dans la littérature: B-spline (implémentation puissante), cyclic cubic splines, thinplate regression splines...

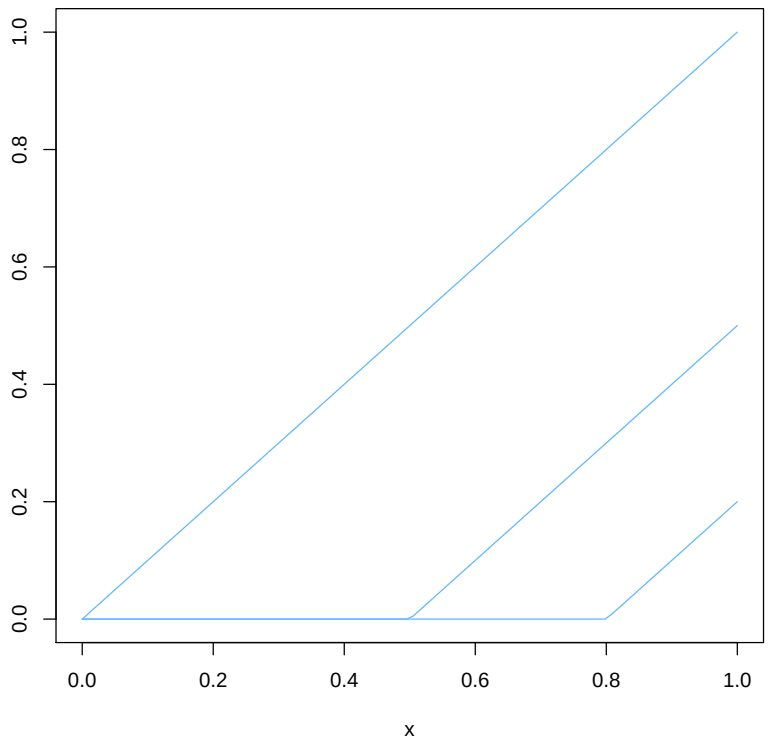
Truncated Power Function

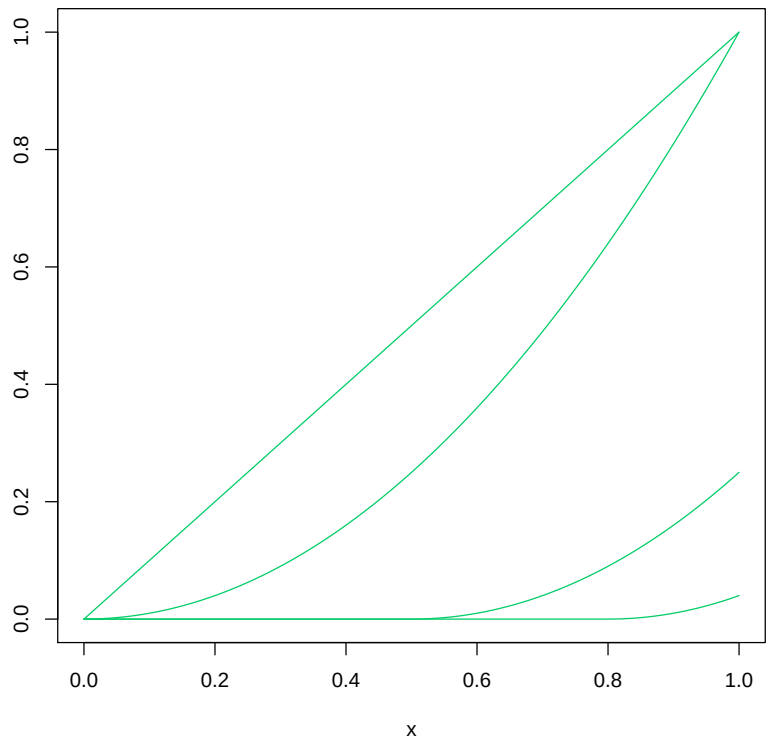
Pour un degré p fixé, un vecteur de noeuds $(k_1, ..., k_q)$:

$$(1, x, x^2, ..., x^p, (x - k_1)_+^p, (x - k_2)_+^p, ..., (x - k_q)_+^p)$$

splines tronquées de degré p.

Notons que $(x - k_1)_+^p$ possède $p - 1$ dérivées continues.





Le pb s'écrit:

$$\min ||Y - X\beta|| + \lambda \beta' D \beta$$

avec $D = diag(0_{p+1}, 1_q)$, on ne pénalise par la partie non-locale de la base. et

$$\widehat{y} = X(X'X + \lambda D)^{-1} X'y$$

Avantages et inconvénients des polynômes tronqués:

- aisément compréhensibles
- facilement interprétables et implémentables
- à utiliser si les positions des noeuds sont connues
- la non-orthogonalité de la base peut générer des pb d'instabilité pour λ faible, d'ou l'introduction des B-splines

B-Splines

Soit $q+1$ noeuds $(k_0 \leqslant k_1 \leqslant ... \leqslant k_q)$, p le degré des splines

Les fonctions B-splines de degré p sont définies par récurrence sur le degré inférieur:

$$b_{j,0}(x) = \begin{cases} 1 & \text{si} \quad k_j \leqslant x \leqslant k_{j+1} \\ 0 & \text{sinon} \end{cases}$$

$$b_{j,p}(x) = \frac{x - k_j}{k_{j+p} - k_j} b_{j,p-1}(x) + \frac{k_{j+p+1} - x}{k_{j+p+1} - k_{j+1}} b_{j+1,p-1}(x)$$

Pour un même degré et les mêmes noeuds, les B-splines s'exprime comme combinaison linéaire des polynômes tronquées correspondant. Si on note X_B la matrice des B-splines, X_T celle des polynômes tronquées, alors $X_B = X_T L_p$ avec L_p une matrice de carré inversible.

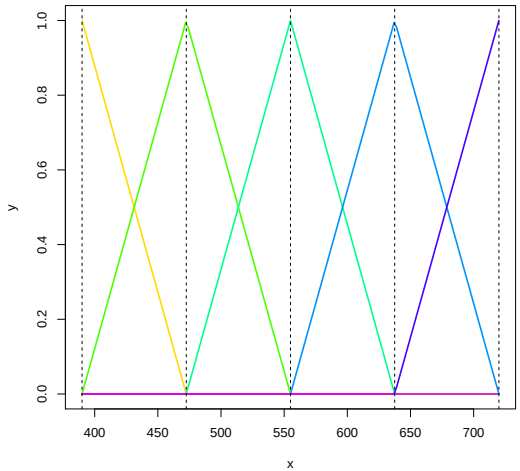
Le pb s'écrit:

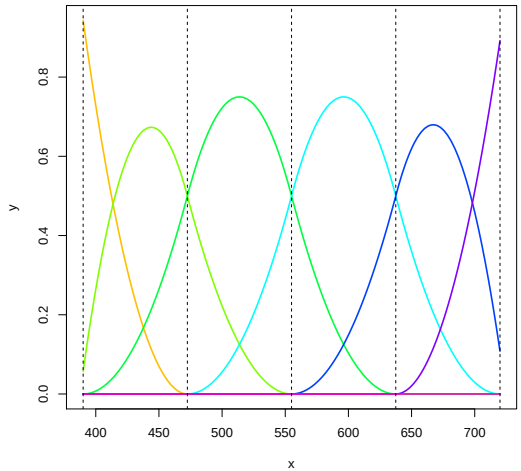
$$\min ||Y - X\beta|| + \lambda \beta' L_p' L_p \beta$$

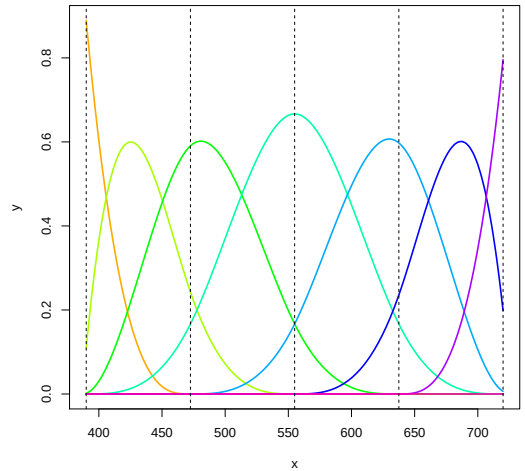
et la solution dans la base de B-spline:

$$\widehat{y} = X_B (X_B' X_B + \lambda L_p' D L_p)^{-1} X_B' y$$

- implémentation efficace
- splines "locales": évite les problèmes de corrélations entre variables de la régression







B-splines de degrés 1, 2, et 3.

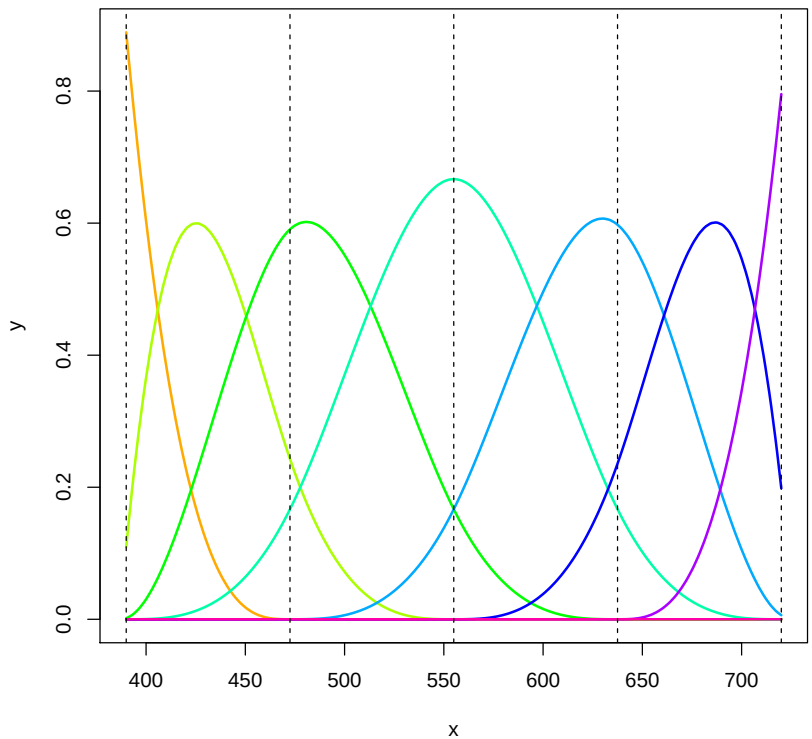
B-splines bidimensionnelles

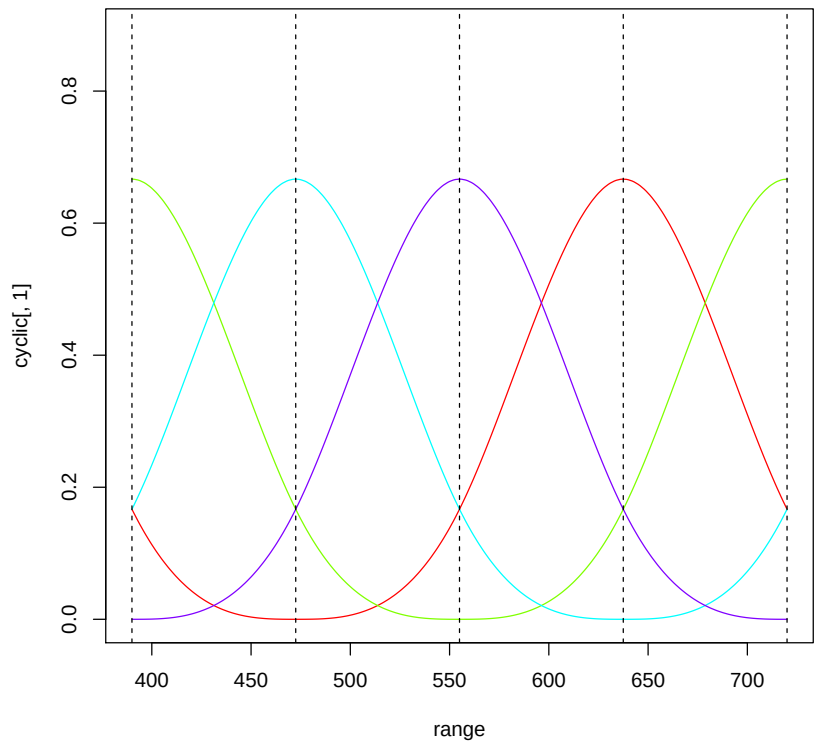
Soit $b_{j,p}(x)$ une base de B-spline de degrés p évalué sur l'axe x et $b_{i,q}(z)$ une base de spline de degré q évaluée sur l'axe y.

La famille constituée des fonctions $b_{j,p}(x)b_{i,q}(z)$ forme une base de l'ensemble des fonction bivariées de carré intégrable $f(x, z)$.

Splines Cycliques

Pour modéliser des phénomènes cycliques, on peut contraindre $f(x)$ à avoir les mêmes valeurs et les mêmes valeurs de dérivées aux bornes de son espace de définition.





Comment choisir le paramètre de pénalisation λ?

- sans pénalisation: $\widehat{\beta}_0 = (X^T X)^{-1} X^T y$
- avec pénalisation: $\widehat{\beta}_\lambda = (X^T X + \sum \lambda_j S_j)^{-1} X^T y$
- $\widehat{\beta}_\lambda = F_\lambda \widehat{\beta}_0$

Où

$$F_\lambda = (X^T X + \sum \lambda_j S_j)^{-1} X^T X$$

$tr(F_\lambda)$: degrés de liberté estimés

- Validation croisée OCV (Ordinary Cross Validation)
  - enlever une observation y_i
  - estimer le modèle $\widehat{\mu}^{-i}$ sur les nouvelles données ainsi formées
  - prévoir y_i par $\widehat{\mu}_i^{-i}$
  - faire ça pour tout i
  - choisir le λ qui minimise le OCV score:

$$V_0(\lambda) = \sum_{i=1}^{n} (y_i - \widehat{\mu}_i^{-i})^2 / n$$

  ↬ Pb: **temps de calcul!**

- GCV: Generalized Cross Validation [Craven and Wahba (1979)]

$$V_g(\lambda) = n\|y - X\widehat{\beta}_\lambda|^2 / (n - tr(F_\lambda))^2$$

Advantages of GCV:

- λ s'obtient par minimisation numérique de V_g (rapide, peu couteux en calcul)
- $V_g(\lambda)$ est invariant pour des transformations utiles des données (on-line update, big data)

$\Rightarrow$ Software: R, package mgcv (see[Wood (2001)] and [Wood (2006)])

Notons que critère de VC s'écrit également:

$$\widehat{f}^{-i} = \sum_{j \neq i} \frac{H_{i,j}(\lambda)}{1 - H_{i,i}} y_j$$

et

$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \widehat{f}_\lambda)^2}{(1 - H_{i,i})^2}$$

$$GCV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \widehat{f}_\lambda)^2}{(1 - tr(H)/n)^2}$$

le critère de VC est donc une moyenne de l'erreur d'estimation pondérée par "l'importance" de chaque observation. Le GCV est une erreur de VC dans laquelle chaque observation à le même "poids".

Autre alternative: Cp On considère le modèle (simplifié pour plus de commodité): $y = f + \varepsilon$, on a $\widehat{f} = Hy$
$||f - \widehat{f}||^2 = ||f - Hy||^2 = ||y - Hy - \varepsilon||^2$
$= ||y - Hy||^2 + ||\varepsilon||^2 - 2\varepsilon'(y - Hy)$
$= ||y - Hy||^2 + ||\varepsilon||^2 - 2\varepsilon'(f + \varepsilon) + 2\varepsilon'(Hf + h\varepsilon)$
D'ou:
$E||f - \widehat{f}||^2 = E||y - Hy||^2 + n\sigma^2 - 2n\sigma^2 + 2E(\varepsilon' H\varepsilon)$

$$E||f - \widehat{f}||^2 = E||y - Hy||^2 - n\sigma^2 + 2tr(H)\sigma^2$$

L'heuristique de Mallows vue en régression linéaire se généralise ici et on peut choisir le modèle qui minimise:

$$||y - Hy||^2 - n\sigma^2 + 2tr(H)\sigma^2$$

ie qui minimise $C_p(\lambda) = ||y - Hy||^2/n + 2tr(H)\sigma^2/n$ Si on ne connait pas σ^2, on l'estime par $\widehat{\sigma} = ||Y - H_{\lambda^*} Y||^2/(n - tr(H^*_\lambda))$ avec λ^* relativement "petit".

Comparaison du GCV et du Cp:

- $C_p = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f})^2 + \frac{2\sigma^2 tr(H_\lambda)}{n}$
- $GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f})^2/(1 - tr(H)/n)^2$

En utilisant l'approximation: $1/(1 - tr(H)/n) \sim 1 + 2tr(H)/n$ on obtient:

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f})^2 + 2\frac{tr(H)}{n^2}\sum_{i=1}^{n}(y_i - \widehat{f})^2$$

On remarque que le GCV est proche du critère de Mallows pour lequel on prendrait comme estimateur de la variance $\widehat{\sigma^2} = \sum_{i=1}^{n}(y_i - \widehat{f})^2/n$

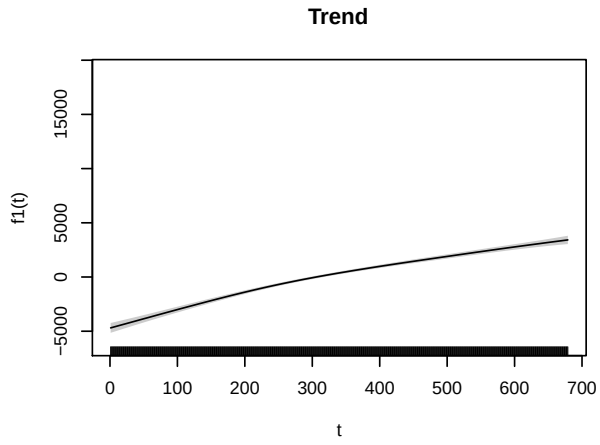
Statistiques intéressantes pour la pratique

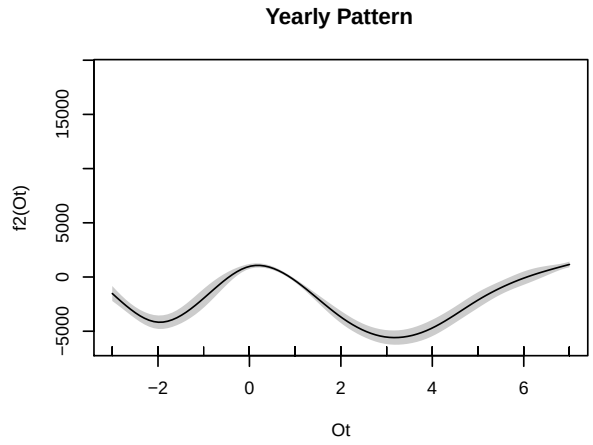
- degrés de liberté estimés: $tr(F_\lambda)$
- $R^2 = 1 - \sum(y_i - x_i\widehat{\beta})^2 / \sum(y_i - \bar{y})^2$,
  ajusted-$R^2 = 1 - \frac{1}{n-p}\sum(y_i - x_i\widehat{\beta})^2 / \sum(y_i - \bar{y})^2$
- GCV score
$$V_g(\lambda) = n\|y - X\widehat{\beta}_\lambda|^2/(n - tr(F_\lambda))^2$$
- tests (Fisher, Student) (hyp. de normalité)

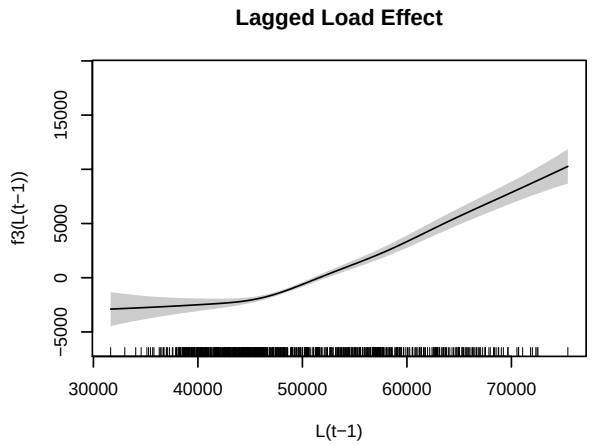
Outil informatique: R, package mgcv (Simon Wood, voir [Wood (2001)] et [Wood (2006)])

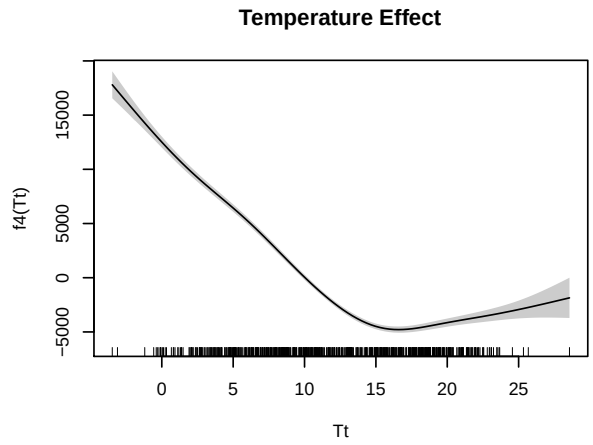
En utilisant les propriétés vues en régression linéaire on peut obtenir un intervalle de confiance pour β:

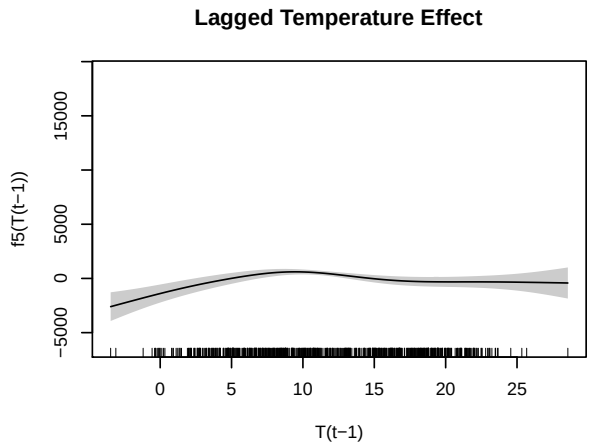
$$\beta|\lambda \sim \mathcal{N}(\widehat{\beta}, (X'X + \lambda S)^{-1}\sigma^2)$$

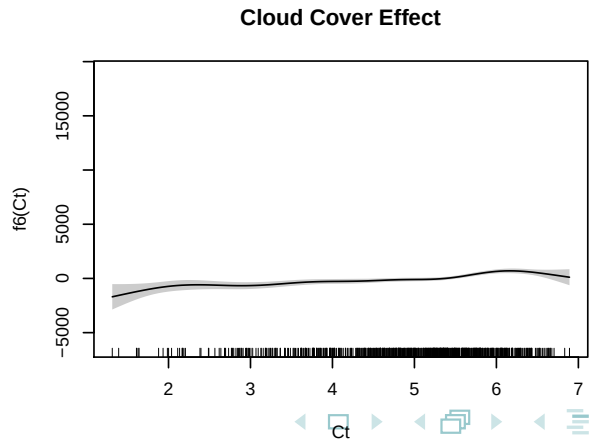

Trend
f1(t)
15000
5000
0
−5000
0
100
200
300
400
500
600
700
t


Yearly Pattern
f2(Ot)
15000
5000
0
−5000
−2
0
2
4
6
Ot


Lagged Load Effect
f3(L(t−1))
15000
5000
0
−5000
30000
40000
50000
60000
70000
L(t−1)


Temperature Effect
f4(Tt)
15000
5000
0
−5000
0
5
10
15
20
25
Tt


Lagged Temperature Effect
f5(T(t−1))
15000
5000
0
−5000
0
5
10
15
20
25
T(t−1)


Cloud Cover Effect
f6(Ct)
15000
5000
0
−5000
2
3
4
5
6
7
Ct

Generalized

En résumé, on a vu:

1. comment un modèle $y_i = f(x_i) + \varepsilon_i$ peut s'écrire comme un problème de régression ridge pénalisée $\min_{\lambda,\beta} ||Y_X\beta||^2 + \lambda\beta' S\beta$
2. comment estimer λ par CV, GCV, Cp...
3. comment obtenir un intervalle de confiance pour les fonctions f_j

On peut étendre ces résultats au cas non Gaussien, c'est le sens du terme "Generalized" dans **G**eneralized **A**dditive **M**odels.

On suppose que l'espérance conditionnelle $E(Y/X) = \mu$ est t.q.:

$$g(\mu) = AX + \sum_j f_j(x_j) \quad y \sim EF(\mu_i, \phi)$$

où g est une fonction de lien connue a-priori. la loi de Y appartient à une famille exponentielle (ex: Normale, Gamma, Poisson, Binomiale).
$g : x \to x$ dans le cas Gaussien, d'autres fct classiques sont $\sqrt{x}$, $log(x)$, $logit(x) = log(x/(1-x))$
Dans ce cas on estime β par maximum de vraisemblance pénalisé, par la méthode PIRLS:

$$\widehat{\beta} = \operatorname{argmin} D(\beta) + \lambda \beta' S \beta$$

avec $D(\beta) = l_{max} - l(\beta)$ est la déviance du modèle, où $l(\beta)$ est la log vraisemblance, $l_{max} = l(y)$ la log vraisemblance saturée.
La déviance est l'équivalent de la somme des résidus au carré dans le cas Gaussien.

Penalized Iterative Relative Least Squares

- initialiser à $\widehat{\eta}_i = g(y_i)$
- constituer les "pseudo data": $z_i = g'(\widehat{\mu}_i)(y_i - \widehat{\mu}_i)/\alpha_i + \widehat{\eta}_i$ et les poids itératifs $w_i = \alpha_i / V(\widehat{\mu}_i) g'(\widehat{\mu}_i)^2$
- minimiser $\sum w_i (z_i - X_i\beta)^2 + \lambda\beta' S\beta$ et obtenir des nouveaux $\widehat{\beta}$,$\widehat{\eta}$, $\widehat{\mu}$

- $\alpha_i = 1$: fisher scoring
- $\alpha_i = 1 + (y_i - \widehat{\mu}_i)(V_i'/V_i + g_i''/g_i')$: Newton method

avec $Var(y) = V(\mu)\phi$

l'algorithme converge vers l'estimateur du maximum de vraisemblance

- la matrice des degrés de liberté estimé devient:
  $F = (X'WX + \lambda S)X'WX$, $W = diag(w_1, ..., w_n)$
- $edf = tr(F)$

2 stratégies possibles pour estimer λ:

1. à chaque étape de PIRLS minimiser le GCV ou un autre critère de pénalisation
2. calculer un critère dépendant de λ à partir de la déviance du modèle et optimiser. chaque évaluation du critère implique d'utiliser

1: rapide, 2: plus lourd mais efficace
Généralisation des critères de pénalisation:

- Mallows Cp: $D(\widehat{\beta}_\lambda) + tr(F)\phi$
- GCV: $nD(\widehat{\beta}_\lambda)/(n - tr(F))^2$

BAM: **B**ig **A**dditive **M**odels

$\Rightarrow$ Pour les "gros" jeux de données (plus de 10 000 observations), basé sur la décomposition QR des données.

- $X = QR$, $f = Q^T y$ et notons $||r||^2 = ||y||^2 - ||f||^2$
  - Q matrice orth., R triang. sup.
- Alors:

$$V_g(\lambda) = \frac{n||f - R\widehat{\beta}_\lambda||^2 + ||r||^2}{(n - tr(F_\lambda))^2}$$

- où F_λ vaut maintenant $(R^T R + \sum \lambda_j S_j)^{-1} R^T R$

$\Rightarrow$ connaissant R, f and $||r||^2$, X ne joue plus aucun rôle

$\Rightarrow$ Application aux "gros" jeux de données

- X est une matrice importante, déco. par blocs: $\begin{pmatrix} X_0 \\ X_1 \end{pmatrix}$, similarly $y = \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}$
- déc. QR $X0 = Q_0 X_0$ et $\begin{pmatrix} R_0 \\ X_1 \end{pmatrix} = Q_1 R$ voir section 12.5 of [Golub and Van Loan (1996)]
- alors $X = QR$ avec $Q = \begin{pmatrix} Q_0 & 0 \\ 0 & I \end{pmatrix} Q_1$ et $Q^T y = Q_1^T \begin{pmatrix} Q_0^T y_0 \\ y_1 \end{pmatrix}$

$\Rightarrow$ Adaptation "on-line"

- X_0, y_0 données passée, X_1, y_1 dernières observations
- Utilisation de X_1, y_1 pour mettre à jour R, f et $||r||^2$
- Réestimation de λ et β_λ (utilisation des dernières valeurs optimales de λ pour initialiser l'optimisation du critère GCV)

Un autre travail réalisé récemment dans [Ba, Goude, Sinn and Pompey (2012)] consiste à mettre à jour les paramètres du modèle en "oubliant" le passé.
On considère le pb suivant:
Soit un échantillon de taille K de données $(x_1, y_1), ..., (x_K, y_K)$, on chercher:

$$\hat{\beta}_K = \min_{\beta} \left\{ (\mathbf{y}_K - \mathbf{B}_K \beta)^T \, \mathbf{\Omega}_K \, (\mathbf{y}_K - \mathbf{B}_K \beta) + \beta^T \mathbf{S}_K \beta \right\},$$

ou $\mathbf{\Omega}_K$ représente les pondérations des données, S_K est la matrice de pénalisation.
on a:

$$\hat{\beta}_K \quad = \quad (\mathbf{B}_K^T \mathbf{\Omega}_K \mathbf{B}_K + \mathbf{S}_K)^{-1} \mathbf{B}_K^T \mathbf{\Omega}_K \mathbf{y}_K$$

Pb: temps de calcul, poids optimaux

Nous cherchons à estimer ces coefficients de façon adaptative en oubliant le passé. On utilise pour cela les moindres carrés pénalisés récursifs pondérés.

- $b_K = b(x_k)$, l'erreur est noté $\hat{\varepsilon}_{k+1} \in \mathbb{R}$
- gain de kalman: $g_{k+1} \in \mathbb{R}^{J \times 1}$
- $P_{k+1} \in \mathbb{R}^{J \times J}$ l'inverse de la matrice de corrélation

alors l'équation de récurrence est donnée par:

$$
\begin{aligned}
g_{k+1} &= \frac{\mathbf{P}_k b_{k+1}}{w + b_{k+1}^T \mathbf{P}_k b_{k+1}}, \\
\mathbf{P}_{k+1} &= w^{-1} \left[\mathbf{P}_k - g_{k+1} b_{k+1}^T \mathbf{P}_k \right], \\
\hat{\varepsilon}_{k+1} &= y_{k+1} - b_{k+1}^T \hat{\beta}_k, \\
\hat{\beta}_{k+1} &= \hat{\beta}_k + g_{k+1} \hat{\varepsilon}_{k+1},
\end{aligned}
$$

ou $y_{k+1} \in \mathbb{R}$ la réponse et $b_{k+1}^T \in \mathbb{R}^{1 \times J}$ est le vecteur des coefficients mis à jour.

ω est le facteur d'oubli du modèle, de sorte que dans l'équation

$$\hat{\beta}_K = (\mathbf{B}_K^T \mathbf{\Omega}_K \mathbf{B}_K + \mathbf{S}_K)^{-1} \mathbf{B}_K^T \mathbf{\Omega}_K \mathbf{y}_K$$

$$\mathbf{\Omega}_k = \begin{cases} 0 & \text{if } k \leqslant K - \tau \\ w^{K-k} & \text{if } K - \tau < k \leqslant K \end{cases}$$

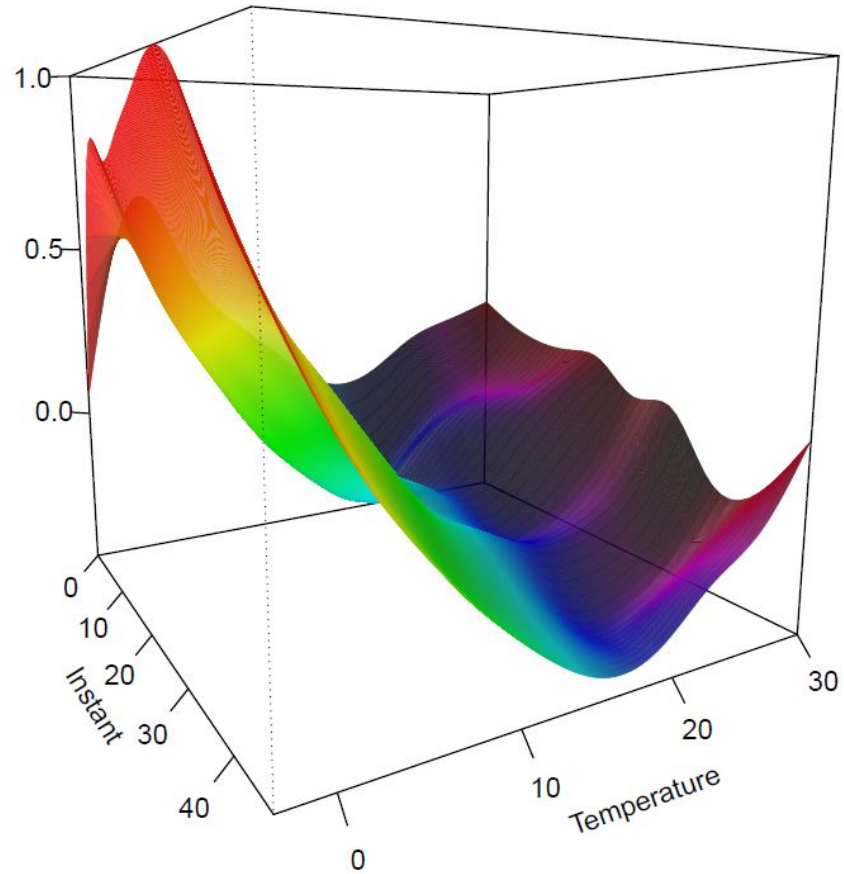
Application à la consommation électrique

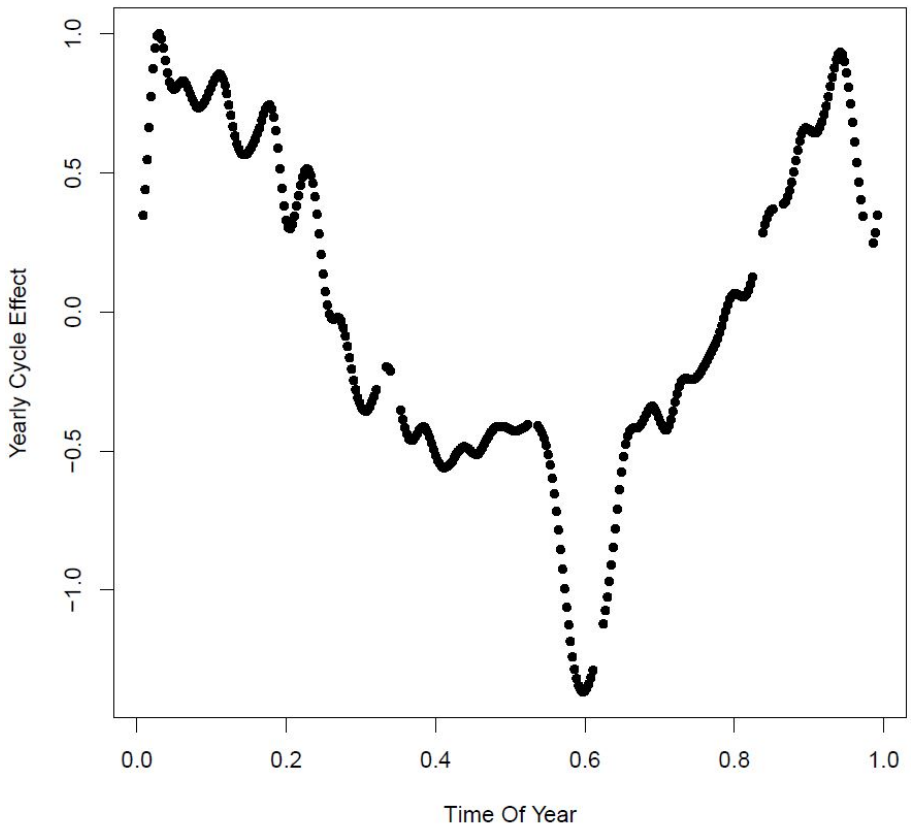
$$\begin{array}{rcll} y_k & = & c & + \sum_{l=0}^{7} m_l I_{DayType_k=l} + \sum_{l=0}^{7} f_l(in_k) I_{DayType_k=l} \\ & & + & g_0(\theta_k, in_k) + g_1(\theta_{k-48}) + g_2(\nu_k) + h(toy_k) + l(y_{k-48}) \\ & & + & s(t) + N_k e(in_k) + \varepsilon_k \end{array}$$

where:

- y_k conso. inst. k
- c constante, $DayType_k$ type de jour obs. k: 0 dimanche, 1 lundi, 2 mardi-mercredi-jeudi, 5 vendredi, 6 samedi et 7 jours fériés
- in_k instant de 0 à 47
- θ_k température inst. k
- ν_k nébulosité
- toy_k position d'un jour dans l'année
- $l(y_{k-48})$ effet conso. de la veille
- $s(k)$ tendance
- $N_k e(in_k)$ EJP

- les fonctions sont estimées via des B-splines cubiques simple et bidimensionnelles.
- les noeuds sont positionnés sur une partition régulière, leur nombre est fixé à 10 dans le cas univarié, 30 dans le cas bivarié




Yearly Cycle Effect
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
Time Of Year

Pour mesurer les performances on compare nos résultats à différentes procédures de prévision:

- ofl-forecaster: paramètres fixé
- fff-forecaster: fix forgetting factor forecaster
- affg-forecaster: adaptive forgetting factor with an on-line optimisation on a grid
- aff-forecaster: adaptive forgetting factor forecaster

Forecaster	ofl	fff	affg	aff	post-fff
MAPE (%)	1.83	2.28	1.7	1.63	1.64
RMSE (MW)	1185	1869	1124	1071	1073

Table: Performances of the different forecasters

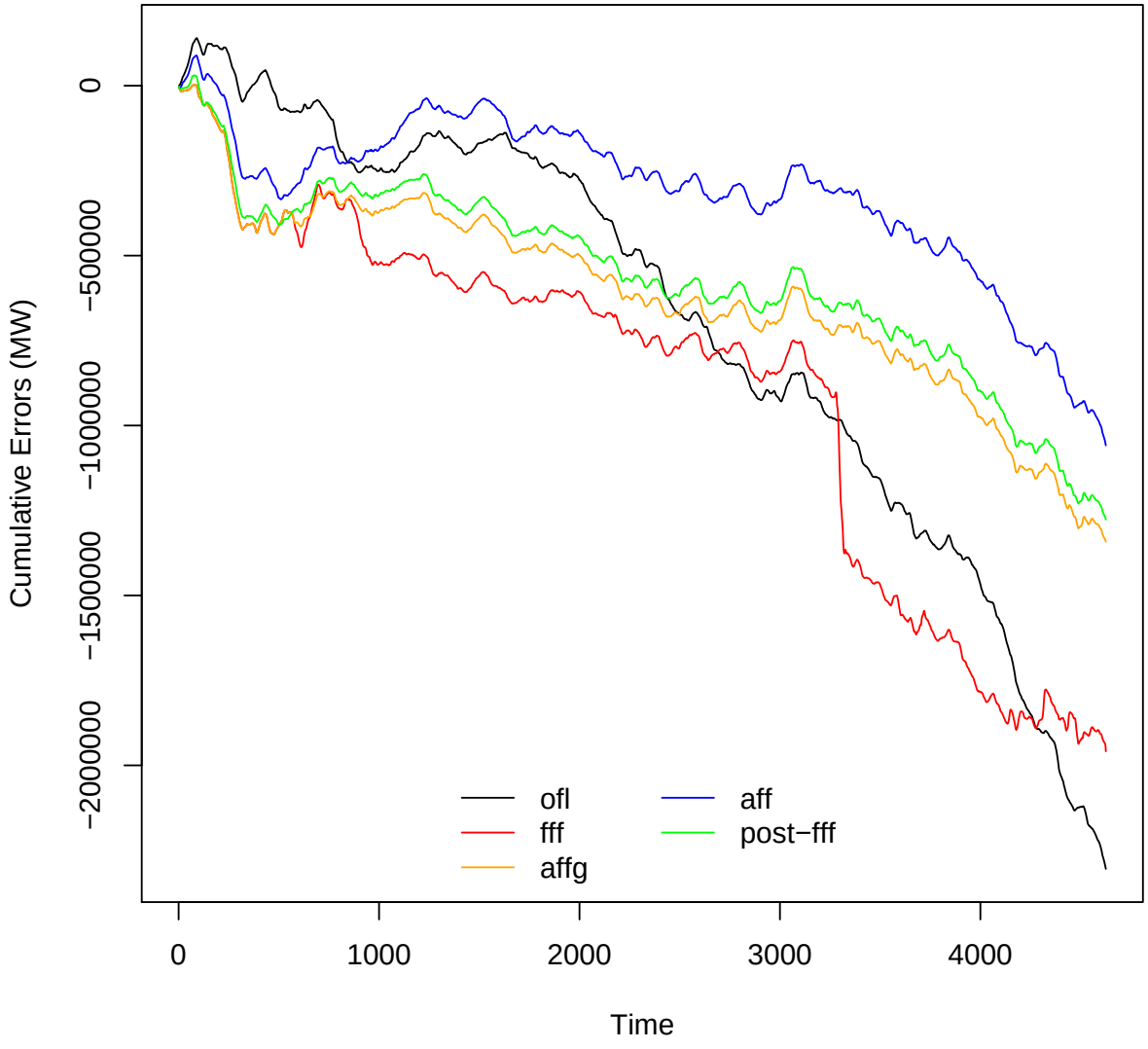


Figure: cumulative sum of the errors


Amadou Ba, Yannig Goude, Matthieu Sinn and Pascal Pompey, Adaptive Learning of Smoothing Functions: Application to Electricity Load Forecasting, accepted for NIPS 2012.


Craven and Wahba (1979) "Smoothing noisy data with spline functions: estimated the correct degree of smoothing by the method of general cross validation". Numerische Mathematik 31, 377-403.


Golub and Van Loan (1996) "Matrix Computations, 3rd edition". John Hopkins Studies in the Mathematical Sciences.


Green and Silverman (1994) "Nonparametric Regression and Generalized Linear Models". Chapman and Hall.


Hastie and Tibshirani (1990) " Generalized Additive Models". Chapman and Hall.


Pierrot and Goude (2011) "Short-Term Electricity Load Forecasting With Generalized Additive Models", Proceedings of ISAP power 2011.


Wahba (1990) "Spline Models of Observational Data". SIAM


Wood (2001) mgcv:GAMs and Generalized Ridge Regression for R. R News 1(2):20-25


Wood and Augustin (2002) "GAMs with integrated model selection using penalized regression splines and applications to environmental modelling". Ecological Modelling 157:157-177


Wood (2006)Generalized Additive Models, An Introduction with R (Chapman and Hall, 2006)


Wood, Goude and Shaw (2012), Generalized additive models for large datasets, Preprint, submitted to JRSS.