Estimation du coefficient de queue en présence de covariables

En collaboration avec Armelle Guillou et Laurent Gardes

Antoine Schorgen

Université de Strasbourg - IRMA

3 Juillet 2012
1ères Rencontres R - Bordeaux

Plan

Modèle

Notre observation Y est mesurée conjointement avec une covariable fixe (non aléatoire) $x \in \mathbb{R}$. On se place dans le domaine de Fréchet, où les distributions sont à queues lourdes:

$$\overline{F}(y, x) = y^{-1/\gamma(x)} L(y, x),$$

où $L(., x)$ est une fonction à variations lentes telle que $\forall \lambda > 0$,

$$\lim_{y \to \infty} \frac{L(\lambda y, x)}{L(y, x)} = 1.$$

De façon équivalente, $\forall \alpha \in (0, 1)$,

$$q(\alpha, x) = \overline{F}^{\leftarrow}(\alpha, x) = \alpha^{-\gamma(x)} \ell(\alpha^{-1}, x), \tag{1}$$

où $\overline{F}^{\leftarrow}(\alpha, x) = \inf\{t,\ \overline{F}(t, x) \leq \alpha\}$ est l' inverse généralisée de la fonction de survie conditionnelle, et $\ell(\cdot, x)$ une fonction à variations lentes.

Objectif: Estimation de $\gamma(x)$

Idée de l'estimation

si $\ell(\cdot, x) = C$, alors $q(\alpha, x) = C\alpha^{-\gamma(x)}$. Ou encore,

$$\log q(\alpha, x) = \gamma(x) \log(1/\alpha) + \log C.$$

On peut construire un estimateur sur la base de

$$\gamma(x) = \int_0^1 \Psi(\alpha) \log q(\alpha, x) d\alpha,$$

avec $\int_0^1 \Psi(\alpha) d\alpha = 0$ et $\int_0^1 \Psi(\alpha) \log(1/\alpha) d\alpha = 1$.

Estimateur à noyau

Utilisation d'un estimateur à noyau pour estimer la fonction de survie conditionnelle:

$$\widehat{\overline{F}}_n(y,x) = \sum_{i=1}^{n} \mathbb{1}\left(\frac{d(x,x_i)}{h_{1,n}}\right) K\left(\frac{y-Y_i}{h_{2,n}}\right) \Bigg/ \sum_{i=1}^{n} \mathbb{1}\left(\frac{d(x,x_i)}{h_{1,n}}\right),$$

où $h_{1,n}$ et $h_{2,n}$ sont deux suites positives non aléatoires. (voir p. ex. Ferraty et Vieu, 2006).

- $d(\cdot,\cdot)$ est une distance. $|\cdot|$ pour faire simple
- K est un noyau intégré: $K(v) = \int_v^\infty g(s)ds$, où g est une densité de support $[-1,1]$
- $m_x := \sum_{i=1}^{n} \mathbb{1}\left(\frac{d(x,x_i)}{h_{1,n}}\right)$, nombre de covariables dans la boule $B(x, h_{1,n})$.

Estimateur de $\gamma(x)$

Notre classe d'estimateurs

$$\widehat{\gamma}_\theta(x) = \int_0^{k_x/m_x} \Psi_\theta(k_x/m_x, x) \log \widehat{q}_n(\alpha, x) d\alpha$$

- $\widehat{q}_n(\alpha, x) := \widehat{\widehat{F}}_n^{\leftarrow}(\alpha, x)$
- Ψ_θ est une fonction dans $L_1(0,1)$ vérifiant qq hypothèses. Pour $\alpha < u$:

$$\Psi_\theta(\alpha, u, x) := \begin{cases} \frac{(\theta+1)^2}{\theta u^{\theta+1}} \left(\frac{u^\theta}{\theta+1} - \alpha^\theta \right) & \text{if } \theta > 0, \\ \frac{1}{u} \left(\log \left(\frac{u}{\alpha} \right) - 1 \right) & \text{if } \theta = 0, \\ \frac{1}{u} & \text{if } \theta = \infty. \end{cases}$$

- k_x correspond au nombre de valeurs extrêmes utilisées dans le voisinage, avec $1 < k_x < m_x$

Convergence de l'estimateur

Condition du second ordre

$$\forall v > 0, \log \frac{\ell(vy, x)}{\ell(y, x)} = \Delta(y, x) \frac{v^{\rho(x)} - 1}{\rho(x)} (1 + o(1)) \text{ qd } y \to \infty,$$

avec $\rho(x) < 0$ et $\Delta(y, x) \to 0$ quand $y \to \infty$.

Normalité Asymptotique

Sous certaines hypothèses, on obtient la normalité asymptotique de notre estimateur $\widehat{\gamma}_\theta(x)$:

$$k_x^{1/2} \left(\widehat{\gamma}_\theta(x) - \gamma(x) - \Delta \left(\frac{m_x}{k_x}, x \right) \frac{\theta + 1}{(1 - \rho(x))(1 + \theta - \rho(x))} \right)$$

converge vers une v.a. $\mathcal{N}(0, 2\gamma^2(x)(\theta + 1)/(1 + 2\theta))$.

Choix des paramètres

Paramètres inconnus:

- les fenêtres $h_{1,n}$ et $h_{2,n}$
- m_x et k_x
- θ et $\rho(x)$

Simplification:

- Pour x fixé, on se place dans le voisinage $B(x, h_{1,n})$: $h_{1,n} \longleftrightarrow m_x$
- Choix canonique pour θ et $\rho(x)$: $\theta_\pi = 1$ et $\forall x, \rho(x) = -1$.

Finalement, il reste seulement à choisir m_x, k_x et $h_{2,n}$.

Choix de (m_x, k_x)

En utilisant la normalité asymptotique, on obtient :

$$AMSE(m_x, k_x) = \frac{4}{3}\frac{\gamma^2(x)}{k_x} + \left[\frac{1}{3}\Delta\left(\frac{m_x}{k_x}, x\right)\right]^2$$

On prendra donc:

$$(\widehat{m}_x, \widehat{k}_x) = \underset{m_x, k_x}{\arg\min}\, \widehat{AMSE}(m_x, k_x).$$

Remarque: pour estimer l'AMSE, il faut encore estimer 2 paramètres: $\gamma(x)$ et $\Delta(k_x/m_x, x)$. On utilise un modèle de régression exponentielle et les moindres carrés. (Beirlant et al., 2002)

Choix de $h_{2,n}$

$h_{2,n}$ n'apparait pas dans l'AMSE, il faut donc un autre critère!

Algorithme de sélection (critère inspiré de Ferraty & Vieu, 2006):

- on fixe un point x_0.
- on divise en 2 sous-échantillons I_1 et I_2 et on note $x_0^{(1)}$ et $x_0^{(2)}$ les valeurs les plus proche de x_0 dans chaque sous-échantillon.
- $\widehat{h}_{2,n} = \arg\min_{h_{2,n}} |\widehat{\gamma}_{\theta_\pi}^{(1)}(x_0^{(1)}) - \widehat{\gamma}_{\theta_\pi}^{(2)}(x_0^{(2)})|$.

Simulations

- $n = 5\,000$ couples $(Y_i, x_i), i = 1, ..., n$
- les $x_i = i/n$ et les Y_i suivent une loi de type Burr: $\overline{F}(y, x) = \left(1 + y^{1/\gamma(x)}\right)^{-1}$
- $\gamma(x) = \frac{1}{2}\left(\frac{1}{10} + \sin(\pi x)\right)\left(\frac{11}{10} - \frac{1}{2}\exp\left(-64\left(x - \frac{1}{2}\right)^2\right)\right)$.
- $R = 100$ répétitions
- $M = 100$ points pour estimer la courbe $\gamma(x)$
- On choisit le noyau biquadratique intégré: $K(x) = \int_x^\infty \frac{15}{16}(1 - s^2)^2 \mathbb{1}_{\{s \in [-1,1]\}} ds$.

Code en R: temps de calcul

Temps écoulé pour une estimation de $\gamma(x)$:

```
 system.time(gam.cond.vois.R(xi,y.vois,x.vois,mx,
 kx,theta,method="sum",type="kernel",h2))
utilisateur     système      écoulé
     48.486       0.000      48.492
```

Temps écoulé

Pour une estimation en 1 seul point, pour 1 seul échantillon et pour 1 seule valeur de $h_{1,n}$ et $h_{2,n}$: 48 sec!

Temps de calcul à améliorer

Rappel: notre estimateur de $\gamma(x)$

$$\widehat{\gamma}_\theta(x) = \int_0^{k_x/m_x} \Psi_\theta(k_x/m_x, x) \log \widehat{\overline{F}}_n^{\leftarrow}(\alpha, x) d\alpha$$

Le point essentiel de l'estimateur consiste en l'inversion de la fonction de survie (inversion par dichotomie) pour obtenir l'inverse généralisée de la fonction de survie.

Intégration de code C

Solution numéro 1

Coder la fonction essentielle (inversion) en C, puis la rappeler dans R.

Fonction en R:

```
invgen<-function(alpha,eps,Yobs,h2)
```

Fonction en C:

```
void invgen(double *a,double *eps,int *nobs,
                  double *obs, double *h, double *min,
                  double *max,double *sortie)
```

Remarques:

- toujours `void` en sortie,
- les variables d'entrée sont des pointeurs.

Utilisation du code C depuis R

1. Compilation(depuis un terminal) :

   ```
   R CMD SHLIB fonction.c
   ```

2. Charger le code C dans R:

   ```
   dyn.load("fonction.so")
   ```

3. Appeler le code C depuis R:

   ```
   .C("invgen") ou .Call("invgen")
   ```

Exemple

```
.C("invgen",as.double(a),as.double(eps),as.integer(nobs),
as.double(obs),as.double(h),as.double(min),
as.double(max),sortie=double(1))
```

Nouveau temps de calcul

Temps écoulé pour une estimation de $\gamma(x)$ avec du code C:

```
 system.time(gam.cond.vois(xi,y.vois,x.vois,mx,
 kx,theta,method="sum",type="kernel",h2))
utilisateur     système      écoulé
      2.157       0.000       2.157
```

GAIN: x25 !

Vraiment considérable dans mon cas vu le nombre de répétitions, de points d'estimations...

Plusieurs processus R

Solution numéro 2

Lancer plusieurs processus R en simultané (nécessité de plusieurs processeurs).

```
nohup R --no-save <script.R> output.txt &
```

Estimation en 100 points: découpage en 4 scripts de 25 points chacun.

GAIN: x4

Modulable selon le nombre de processeurs dont on dispose.

Graphs finaux

Finalement, après 1 jour de calcul,

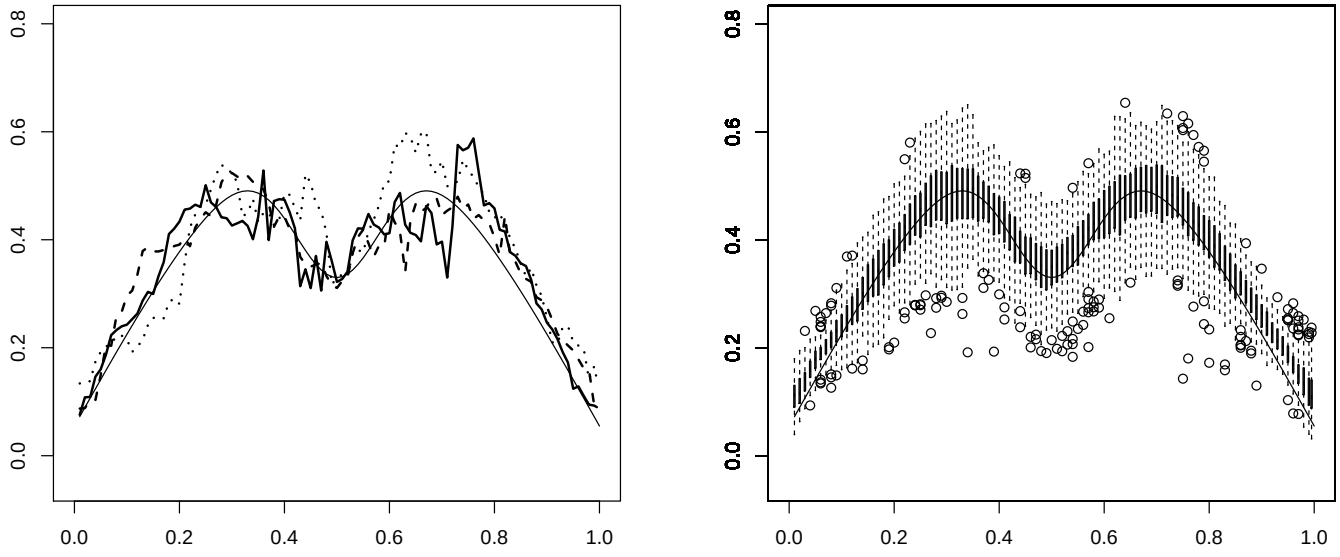

Figure: Estimation de $\gamma(.)$: Médiane (plein), Quantiles de niveaux 10% (tirets) et 90% (pointillés) du carré de l'erreur L^2 et vraie function (—) pour 100 échantillons de taille 5000.

Références

1. Beirlant, J., Dierckx, G., Guillou, A., Stărică, C. 2002. On exponential representations of log-spacings of extreme order statistics, *Extremes*, **5**, 157–180.
2. Daouia, A., Gardes, L., Girard, S., Lekina, A. 2010. Kernel estimators of extreme level curves, *Test*, **20**(2).
3. Ferraty, F., Vieu, P. 2006. Nonparametric Functional Data Analysis: Theory and Practice, Springer Series in Statistics, Springer.
4. Gardes, L., Guillou, A., Schorgen, A. 2012. Estimating the conditional tail index by integrating a kernel conditional quantile estimator, *JSPI*, à paraitre.