TP R sur méthodes de Monte Carlo

Emmanuel Rachelson and Matthieu Vignes

22 mars 2013, SupAero - ISAE

1 Un exemple simple: une régression linéaire simple *bootstrappée*

L'outil statistique "bootstrap" est basé sur le fait que la distribution empirique d'un échantillon $X_1 \dots X_n$ converge vers la vraie distribution quand n tend vers l'infini. La procédure de bootstrap consiste à utiliser (pour un n fixé assez grand) la distribution empirique comme un substitut de la distribution réelle et à construire des estimations de la variance et des intervalles de confiance. Le support est fini mais on a quand même n^n combinaisons possibles. Ainsi, le plus souvent, on aura recours à des approximation de Monte Carlo pour évaluer ces quantités. Attention, une des difficultés majeures dans l'utilisation de cette technique concerne la quantité à bootstrapper ?!

(a) Pour commencer, générez un échantillon e de taille 8 de loi $\gamma(4, 1)$, calculez-en sa moyenne. Puis créez un échantillon boostrappé e^* et calculez en sa moyenne avec le code :

```
estar<-sample(e,replace=T)
mean(estar).
```

C'est un échantillon de la même taille où on tire parmi e avec remise. Tracez l'histogramme de 2 500 moyennes bootstrappées avec une approximation normale adaptée (pourquoi on vous demande ça ?). Comparez l'écart-type de e à celui des 2 500 bootstraps.

(b) On va faire une régression sur le modèle

$$Y_i = \alpha + x\beta + \epsilon_i,$$

où α et β sont les inconnus de la régression : ordonnée à l'origine (*intercept*) et pente (*slope*) et ϵ_{ij} sont des erreurs iid distribuées normalement.

La régression linéaire se fait sous R selon:

```
x <- seq(-3,3,le=5) # regresseur equidisperse
y <- 2+4*x+rnorm(5) # simulation de la reponse
```

```
reg <- lm(y∼x)
summary(reg)
```

Au passage, pourquoi si vous effectuez plusieurs fois ces lignes obtenez vous une sortie différente ? D'où vient réellement l'*alea* ?

Les résidus de l'estimation des moindres carrés sont donnés par:

$$\hat{\epsilon_i} = y_i - \hat{\alpha} - x\hat{\beta}.$$

C'est à partir de ces réalisation de la variable aléatoire ϵ que nous créerons des échantillons *bootstrappés*: c'est un échantillon de la même taille où on tire parmi les $(\hat{\epsilon_i})_i$ avec remise: $(\hat{\epsilon_i}^*)_i$. Les données bootstrappées sont alors les $y_i^* = y_i + \hat{\epsilon_i}^*$.

Effectuez cette manipulation pour 2 000 bootstraps et tracez l'histogramme des réplicats des coefficients de la régression. Expliquez comment on peut trouver des intervalles de confiance pour ces coefficients et pour les valeurs prédites (des nouveaux x). Comparez ces intervalles avec les intervalles de confiance issus de t-test.

2 Calcul d'intégrales

(a) Soit la fonction

$$f(x) = [\cos(50x) + \sin(20x)]^2.$$

Représentez là graphiquement et intégrez là sur $[0, 1]$ en voyant l'intégrale comme une espérance uniforme. Tracez la moyenne empirique courante qui sert à approcher l'espérance (je vous aide !) et un encadrement par $+/-2$ fois une estimation de l'écart-type en fonction du nombre de simulation.

Note 1 : attention, l'évaluation simultannée des erreurs de l'estimation par une méthode de Monte Carlo est un bonus indéniable. Mais cela est valable uniquement quand l'estimée de l'écart-type $\frac{\sum_i (f(x_i - \bar{f_n})}{n^2}$ est une bonne estimation de la variance de $\bar{f_n}$. Il y a des situations critiques où cette estimée ne converge pas assez vite pour appliquer un TCL ou pire, ne converge pas du tout !

Note 2 : une pinte à celui qui calcule l'intégrale à la main.

(b) On va approcher la fonction de répartition d'une distribution $\mathcal{N}(0, 1)$ à l'aide d'un échantillon $n_1 \dots x_n$ grâce à $\hat{\phi}(t) = \frac{1}{n} \sum_i \mathbb{1}_{x_i \leq t}$ dont la variance exacte est $\phi(t)(1 - \phi(t))$ (pourquoi ?). Produisez une table qui donne la fonction de répartition en $0, 0.67, 0.84, 1.28, 1.65, 2.32, 2.58, 3.09$ et 3.72 en fonction de n (variant de 10^2 à 10^8). Quelle est la variance près de 0 ? Combien d'itérations pour avoir une précision à 4 chiffres après la virgule ? Notez la plus grande précision (absolue) dans la queue de la loi normale.

Probablement, des méthodes de simulations plus efficaces pourraient être utilisées...

(b') Aussi, les méthodes de Monte Carlo standards s'appliquent-elles si on veut par exemple calculer $P(Z > 4.5)$ où $Z \sim \mathcal{N}(0,1)$? Pour une idée de l'ordre de grandeur, `pnorm(-4.5,log=T) =` ? Et si on faisait de l'*importance sampling* (ou échantillonage pondéré): la loi exponentielle $\mathcal{E}(1)$ tronquée à $[4.5, \infty[$ de densité $\exp(-y)/\int_{4.5}^{\infty} \exp(-x)dx$ vous semble-t-elle un bon candidat ? La précision est-elle bonne rapidement (par rapport à une simulation à partir d'une loi uniforme) ? Quel est l'impact de l'utilisation d'une loi $\mathcal{E}$ sur la variance de la probabilité estimée ? Un changement de variable dans l'intégrale initiale aurait-il fait l'affaire ?

3 Optimisation

(a) Pour la fonction $f(x) = [\cos(50x) + \sin(20x)]^2$ ci-avant, un appel à la fonction `optimise()` donne la maximum en $x_m = 0.379$ avec une valeur $h(x_m) = 3.8325$. Essayons d'évaluer la variabilité d'un échantilloneur uniforme...

(c) Mettons maintenant qu'on veuille minimiser la fonction sur $\mathbb{R}^2$: $h(x,y) = (x\sin(20y)+y\sin(20x))^2\cosh(\sin(10x)x)+(x\cos(10y)-y\sin(10x))^2\cosh(\cos(20y)y)$.

Les quelques lignes suivantes vous aident à visualiser cette fonction:

```
h=function(x,y)(x*sin(20*y)+y*sin(20*x))^2*cosh(sin(10*x)*x)
+(x*cos(10*y)-y*sin(10*x))^2*cosh(cos(20*y)*y)
x=y=seq(-3,3,le=435) # on definit une grille pour persp()
z=outer(x,y,h)
par(bg="wheat",mar=c(1,1,1,1))
persp(x,y,z,theta=155,phi=30,col="green4",ltheta=-120,shade=.75,
border=NA,box=FALSE)
```

Et voir que les conditions pour utiliser les méthodes de minismisation traditionnelles ne vont pas être remplies ?! Vous sentez vous de calculer ce minimum (c'est 0 !!) par MCMC ?