TP3 Statistiques : Estimation du maximum de vraisemblance et l'intervalle de confiance

21 mars 2025

Le modèle d'un paramètre: La loi Bernoulli

Soit $X_1, \ldots, X_n$ un échantillon de Bernoulli $\mathcal{B}(p)$ avec $p = 0.6$. Nous voulons estimer p à partir d'une réalisation de cet échantillon $x = (x_1, \ldots, x_n)$. de X

Ex1. Simuler un échantillon i.i.d de taille $n = 10$ en utilisant `rbinom`.

```
n <- ...
p <- ...
x <- rbinom(n=...,size = ..., prob =... )
```

Quelle est une estimation simple de p ?

Ex2. Créer une fonction de vraisemblance, nommée `L_bern`, en fonction de (p, x), qui donne la vraisemblance d'un échantillon $x = (x_1, \ldots, x_n)$ pour une valeur donnée de p.

```
L_bern <- function(p, x){## vraisemblance
  ...
  return(...)
}
```

Ex3.. Pour l'échantillon généré dans l'**Ex1**, calculer la vraisemblance de cet échantillon des lois Bernoulli de paramètres p allant de 0 à 1. Tracer la courbe des valeurs calculées. Que remarquez-vous?

```
pvec = seq(0,1,by=...)
Lp = sapply(pvec, L_bern, x=...)
Lp
plot(pvec, Lp)
```

Ex4.. En utilisant la fonction `optim` de R, trouvez la valeur de p la plus probable d'avoir généré cet échantillon.

```
?optim  ### la description de l'optim
```

```
mL_bern <- function(p,x){ -L_bern(p,x) }
## optimization standard (minimization)
p0 = ... # valueur initiale pour l'algorithme
res = optim(p0, mL_bern, x=x, method = "...")
res
```

Attention : `optim` est par défaut une routine de **minimization**. Avec la méthode de *L-BFGS-B* dans la fonction `optim`, vous pouvez traiter des contraintes sur le(s) paramètre(s), lorsque qu'il est nécessaire.

```
## optimization avec contraintes
pmin = ...; pmax = ...
res1 = optim(p0, mL_bern, x=x, method="L-BFGS-B", lower=pmin, upper=pmax)
res1
```

Ex5. Tester avec des échantillons de taille n allant de $n = 10$ à $n = 2000$ et comparer l'écart entre la valeur théorique attendue et la valeur obtenue. Que remarquez-vous? Comment combattre l'instabilité numérique due aux multiplications de probabilités?

```
size <- seq(10,2000, by=...)
para <- ...

for(k in ...)
{
  x<-rbinom(k,1,p)
  pmin<-...
  pmax<-...
  res2 <- optim(...)
  para[k]<-res2$par
}
```

Utiliser la log-vraisemblance au lieu de la vraisemblance:

```
log_L <- function(p,x){ ... } #log vraisemblance
size <- seq(10,2000, by=...)
...
for(k in size)
{
  ...
}
plot(...)
```

Ex6. Donner les intervalles de confiance de niveau 80% pour le paramètre p, à partir de la loi théorique asympotique.

Alternativement, donner les intervalles de confiance d'après l'inégalité de Bienaymé-Chebycheff et l'inégalité de Hoeffding.

Ex7. Pour n fixé, simuler 500 échantillons et donner des intervalles de confiance correspondants. Vous pouvez faire varier la taille de l'échantillon et le paramètre réel. Comptez le nombre de fois où l'intervalle de confiance contient le vrai paramètre. Quelle est votre conclusion?

Un modèle à deux paramètres: La loi Beta

Soit $X_1, \ldots, X_n$ un échantillon de n variables indépendantes de loi de Beta ($\theta = (\alpha, \beta)$). Simuler un échantillon i.i.d de taille $n = 50$ avec $\theta_0 = c(2, 3)$.

```
n<-...
theta<-c(...)
X <- rbeta(...)
head(X)
```

Ex8. Présenter l'histogramme des données simulées. Choisir trois paramètres candidats, disons, θ_0 (vrai) θ_1, θ_2. Comparer l'histogramme avec les densités candidates. Que remarquez-vous?

```
theta_0 <- theta
X<- rbeta(n=...,...)
theta_1 <- ...
theta_2 <- ...
xx<- seq(..., by=...)
hist(X,..., main ="...")
lines(xx,dbeta(...),col="red")
lines(xx,dbeta(...),col="blue")
legend("topright", legend=c("Beta(...)","Beta(...)"),col = c("red","blue"),lty = c(1,1))
```

Ex9. Générer une fonction de la log-vraisemblance avec les arguments (θ, x), qui donne la log-vraisemblance d'un échantillon $x = (x_1, \ldots, x_n)$ pour une valeur donnée de $\theta = (\alpha, \beta)$. Pour votre échantillon, évaluer la log-vraisemblance de paramètre $\theta = (\alpha, \beta)$, en faisant varier un paramètre à la fois, et visualizer la fonction sur $a_0 \leq \alpha \leq a_1, b_0 \leq \beta \leq b_1$ pour votre choix appropriés de limites. Que remarquez-vous?

Indication : Pour plus de simplicité, il suffit de tracer la tranche des courbes unidimensionnelle en supposant que l'autre est fixe: $\ell(\alpha|\beta = \beta_0)$ pour plusierus valeurs de β_0 et $\ell(\beta|\alpha = \alpha_0)$ pour plusieurs valeurs de α_0.

Nous pouvons également utiliser des visualisations 2-dim ou 3-dim avec `contour` ou `image`. Une visualisation plus avancée peut être réalisée à l'aide du mudule (`plot3D`). Voici un exemple de code.

```
aa = seq(0.5, 5, by=...)
bb = seq(0.5, 6, by=0.2)
vtheta = expand.grid(a=aa, b=bb); vtheta

# evaluer la (negative) vraisemblance
mlogL_beta = ...
log_lik = apply(vtheta, 1, mlogL_beta, x=X)
cbind(vtheta, log_lik)
# chercher le minimum
iopt = which.min(log_lik)
a1 = vtheta[iopt,1]
b1 = vtheta[iopt,2]
## transformer en matrice
zmat = matrix(log_lik, nrow=length(aa), ncol=length(bb))
rownames(zmat) = aa
colnames(zmat) = bb
## contour plots
?contour
contour(aa, bb, zmat)
points(a1, b1, pch=8, col=2) ## ajouter les points minimum
```

Ex10. Donner l'expression mathématique du vecteur Score (les dérivées premières) à laquelle l'EMV répond et trouver l'information de Fisher. En utilisant la fonction `optim`, trouver l'estimateur du maximum de vraisemblance. Quelle est la loi asymptotique d'estimateur?

Ex11. Construire des intervalles de confiance asymptotique de niveau 0.90. Incluent-ils les vrais paramètres ? Tester avec des échantillons de taille $n = 20$ et $n = 100$. Quel est l'effet de la taille de l'échantillon ?

Ex12. Répéter l'estimation 100 fois avec de nouveaux ensembles de données et tracer les valeurs estimées $\hat{\alpha}$ vs $\hat{\beta}$. Sont-elles indépendantes ? Visualisez vos résultats à l'aide d'un histogramme. Ajoutez la densité de la loi asymptotique. Que remarquez-vous?

Ex13. Réalisez une étude de simulation pour évaluer la performance des intervalles de confiance et résumez vos conclusions.