TP R 2: statistiques bayésiennes

BINET Thomas

Statistiques biomédicales

Table des matières

Important : Ceci est un document R notebook. En cliquant sur knit vous aurez le choix entre un document html, pdf ou word. Le TP doit être rendu sous la forme d'un fichier NOM_prenom_Bayes.html. Il est à remettre dans moodle. CTRL+Alt+i permet d'ouvrir une cellule de code compilable.

Tests Bayésiens

Echantillon de Bernoulli

Cette partie a pour but d'étudier les erreurs de type I de plusieurs tests, dans le cas standard d'une comparaison d'efficacité d'un nouvel agent avec un médicament de référence. On suppose que l'efficacité du médicament de référence est $p_0 = 0.7$ et les hypothèses antagonistes sont $H_0 : p \leq p_0$ et $H_1 : p > p_0$.

1. La fonction suivante renvoit la moyenne, la p-valeur, et la borne supérieur de l'intervalle de fluctuation sous l'hypothèse H_0, ci-dessus.

```
test.p = function(p0, vector, conf.level) {
     if (length(vector)==0) { cat("Erreur ! Le vecteur ",substitute(vector),"est vide.\n")}
      else {
      n = length(vector)-sum(is.na(vector))
      moyenne = mean(vector,na.rm=T)
      p_val = 1- pnorm(moyenne, p0, sqrt(p0*(1-p0)/n))
      borne_sup = qnorm(conf.level, p0, sqrt(p0*(1-p0)/n))
      return(list(moyenne=moyenne, p_val=p_val, borne_sup=borne_sup)) }}

x = rbinom(100,1,0.7) ; test.p(p0=0.7,vector=x,conf.level = 0.95)
```

```
## $moyenne
## [1] 0.72
##
## $p_val
```

```
## [1] 0.3312603
##
## $borne_sup
## [1] 0.7753767
```

Créér une fonction permettant d'évaluer par simulation numérique le pourcentage d'erreur de type 1 commises. Cette fonction prendra en paramètre d'entrée : `N`, `n p0`, `conf.level` où `N` est le nombre de tests simulés, `n` la taille de l'échantillon et `p` la vraie probabilité de succès du nouvel agent.

2. Nous allons maintenant effectuer des tests bayésiens. On choisi un a priori conjugué avec les données (loi Beta) et la fonction de perte considérée est la fonction "0-1" :

$$l(\theta, T) = \alpha_0 \mathbb{1}_{\{T=1, \theta \in \Theta_0\}} + \alpha_1 \mathbb{1}_{\{T=0, \theta \in \Theta_1\}}.$$

Le tests de Bayes T pour cette fonction de perte et un a priori Π est :

$$T(X_1^n) = \mathbb{1}_{\left\{B_n \leq \frac{\alpha_1 \Pi(\Theta_1)}{\alpha_0 \Pi \Theta_0}\right\}}, \text{ avec } \quad B_n = \frac{\Pi_n(\Theta_0)}{\Pi_n(\Theta_1)} \frac{\Pi(\Theta_1)}{\Pi(\Theta_0)}.$$

Créer une fonction permettant d'effectuer un test bayésien. Cette fonction prendra en paramètre d'entrée : `p0`, `vector`, `a`, `b` (les paramètres de l'apriori), `alpha0`, `alpha1`.

3. Créer une fonction, `err_I_b`, permettant d'étudier l'erreur de type 1 dans le sens fréquentiste (données générées sous une unique loi de paramètre `p`) .

4. Tester la fonction pour les différentes circonstances suivantes :`n=10, 50, 100` et $\alpha_0/\alpha_1 = \Pi([0, p_0])/\Pi(]p_0, 1])$ ou $\alpha_0/\alpha_1 = 19/1$. On choisira un a prioiri uniforme.

Echantillon de loi normale

Les médecins souhaitent qu'un traitement amènent le biomarqueur b_0 à se situer en moyenne dans l'intervalle $I = [45, 65]$. Chacune des 5 doses testées ont été soumises à 25 patients ; les patients ne sont traités qu'à une dose (125 volontaires). Pour chacune des doses, on souhaite calculer la probabilité que le biomarqueur soit dans l'intervalle. Le modèle bayésien suivant est utilisé :

$$X \mid \theta, \sigma^2 \sim \mathcal{N}(\theta, \sigma^2),$$

avec une loi conjuguée pour les paramètres θ et σ (conjugate prior).

```
data=read.csv("biomarker_dose.csv")
```

1. Dans un premier temps, on suppose la variance connue est identique dans chaque groupe : $\sigma^2 = 6$. Quelle est la dose la plus adaptée

2. Maintenant, la variance n'est plus connue. Les probabilités sont à évaluer par méthode de Monte Carlo en utilisant comme a priori la loi normale inverse-gamma : $\theta, \sigma^2 \sim \mathcal{N}\Gamma^{-1}(\mu, \lambda, \alpha, \beta)$. Pour générer sous l'a posteriori, on a :

$$x \mid y, \mu, \lambda \sim \mathcal{N}(\mu, \sigma^2/\lambda) \quad \text{et} \quad y \mid \alpha, \beta \sim \Gamma^{-1}(\alpha, \beta) \;\Rightarrow\; (x, y) \sim \mathcal{N}\Gamma^{-1}(\mu, \lambda, \alpha, \beta)$$

3. Cette question nécessite d'avoir installer le logiciel `JAGS` et les packages `rjags`, et `r2jags`. Il s'agit d'utiliser ce logiciel pour effectuer la même analyse qu'à la question précédente mais sans forcément utiliser un a priori conjugué.

Pour vous aider, l'exemple suivant est détaillé dans la section qui suit : $\theta \sim \mathcal{N}(0, 1)$ et $X_i \sim \mathcal{N}(\theta, 1)$.

JAGS, un premier exemple

Le code ci-dessous permet de construire un model bayésien pour JAGS. L'exemple est très simple pour que vous puissiez vous concentrer sur la structure à donner aux codes : un modèle, des données, des paramètres et la fonction `jags` pour obtenir l'a posteriori.

```
#install.packages('R2jags')
library(R2jags)
N <- 50
x <-  rnorm(N,2,1) # data

#Le modèle pour JAGS
model1 <- "
model{
for (i in 1:N) {
x[i] ~ dnorm(theta,inv_sigma)
}
theta ~ dnorm(0,1)
inv_sigma = 1/10
}
"

# Les données
datum <- list(N=N,x=x)

# Les paramètres à étudier
parameters <- c("theta","inv_sigma")

# Compile et estime le modèle conditionnellement aux données
Mrun1 <- jags(
  data = datum,
  parameters.to.save = parameters,
  model.file = textConnection(model1),
  n.chains = 2, n.iter = 10000,
  n.burnin = 2000
)
```

Un résumé des résultats obtenus :

```
Mrun1; mean(x)
```

Un histogramme des simulations sous "l'a posteriori" pour le paramètre θ et la valeur moyenne du paramètre. On notera l'utilisation de BUGSoutput, une sous-liste très utile de notre modèle (taper manuellement `Mrun1$BUGSoutput$` dans la console pour obtenir des propositions).

```
hist(Mrun1$BUGSoutput$sims.matrix[,"theta"], xlim=c(-5,5))
Mrun1$BUGSoutput$mean$theta
```

Et un visuel d'une des chaînes de Markov. On suppose que la chaîne a convergée si la distribution semble non corrélée et si `Rhat` est proche de 1 (inférieur à 1.05).

```
traceplot(Mrun1)
```

Question 1 : Changer le paramètre θ dans l'a priori et commenter le résultat obtenu. on utilisera par exemple : $\theta \sim \mathcal{N}(0, 1/100)$ (attention : le paramètre dans dnorm correspond à l'inverse de l'écart-type).

Question 2 : Changer le paramètre de variance afin que celui-ci suive une loi exponentielle : $\sigma^2 \sim \mathcal{E}(1)$ (attention : le paramètre dans dnorm correspond à l'inverse de l'écart-type).

Quelques distributions potentiellement utiles avec JAGS :
`dpois`, `dnorm`, `dt`,`dexp`,`dchisqr`,`dbin`,`ddexp`,`dbeta`

Estimateur du maximum de vraisemblance

Exercice 1 (EMV) : Importer le data set `Survie.Rdata` contenant la dataframe `df3`. Ce sont des données de survie en nombre de jours pour des souris infectés par un virus et ayant subit un traitement antiviral.

```
load(file='Survie.Rdata')
```

On modélise ces données par une loi $\Gamma(a, b)$ (a et $b > 0$) de densité :

$$f_{(a,b)}(x) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}, \quad x > 0.$$

Dans ce modèle la fonction de log-vraisemblance est :

$$\ell(a, b) = \sum_{i=1}^{n} \log f_{(a,b)}(x_i) = na \log(b) - n \log(\Gamma(a)) + (a-1) \sum_{i=1}^{n} \log(x_i) - b \sum_{i=1}^{n} x_i$$

Dans ce modèle, L'EMV n'a pas de forme explicite. On utilise l'algorithme de Newton-Raphson pour l'approcher. Soit $(a^{(t)}, b^{(t)})$, les valeurs des paramètres au temps t. On a alors :

$$(a^{(t+1)}, b^{(t+1)})^T = (a^{(t)}, b^{(t)})^T - [H(a^{(t)}, b^{(t)})]^{-1} \nabla \ell(a^{(t)}, b^{(t)})$$

Le gradient est :

$$\nabla \ell(a, b) = \left(\frac{\partial}{\partial a} \ell(a, b), \frac{\partial}{\partial b} \ell(a, b) \right)^T = \left(n \log b - n(\log \Gamma(a))' + \sum_{i=1}^{n} \log x_i, \quad \frac{na}{b} - \sum_{i=1}^{n} x_i \right)^T$$

et le laplacien est :

$$[H(a, b)]^{-1} = \begin{pmatrix} \frac{\partial^2}{\partial^2 a} \ell(a, b) & \frac{\partial^2}{\partial a \partial b} \ell(a, b) \\ \frac{\partial^2}{\partial a \partial b} \ell(a, b) & \frac{\partial^2}{\partial^2 b} \ell(a, b) \end{pmatrix}^{-1} = \frac{1}{n\,(1 - a(\log \Gamma(a))'')} \begin{pmatrix} a & b \\ b & b^2 (\log \Gamma(a))'' \end{pmatrix}$$

1. Ecrire une fonction qui met à jour les paramètres. Pour les dérivées de $\log \Gamma(a)$, on utilise les fonctions `digamma` et `trigamma`.

2. A l'aide d'une boucle while calculer un estimateur de l'EMV. Critère d'arrêt :

$$\left| \frac{\ell(a^{(t+1)}, b^{(t+1)}) - \ell(a^{(t)}, b^{(t)})}{\ell(a^{(t+1)}, b^{(t+1)})} \right| < \varepsilon,$$

avec $\varepsilon = 1e-2$. Initialiser les paramètres à l'aide des estimateurs de la méthode des moments (ou par une méthode simple de votre choix).

3. Estimer les paramètres à l'aide d'un modèle bayésien et du logiciel JAGS.

4. D'après vos résultats à la question précédente quelle est la probabilité que la survie moyenne soit inférieur à 20 jours ?