

Bayesian Study Group Test

Novri Suhermi, GradStat

PhD candidate in Applied Social Statistics, Lancaster University

Introduction

Saya adalah kandidat doktor Applied Social Statistics di Lancaster University. Topik tesis saya berhubungan dengan Functional Data Analysis dan High-Dimensional Statistics untuk pemodelan konsumsi energi gedung. Saya juga memiliki pengalaman riset di bidang time series forecasting, reliability analysis, dan machine learning. Publikasi saya bisa dilihat di scholar.google.com/citations?user=f13VxpsAAAAJ.

Saya menyelesaikan studi S1 Statistika di Institut Teknologi Sepuluh Nopember (ITS) dengan topik skripsi tentang model hybrid Vector Autoregressive (VAR) dan Neural Network untuk peramalan gerak kapal. Saya melanjutkan studi S2 double degree Statistika ITS dan Applied Mathematics Université Paris Diderot. Pengalaman saya selama S2 yaitu bekerja sebagai research intern di Laboratoire Jean Kuntzmann Grenoble-INP dengan project implementasi Monte Carlo exact goodness-of-fit test untuk Imperfect Maintenance Model menggunakan Rcpp. Saya juga menjadi research intern di Université Paris Descartes dengan project analisis time series pada data Near-infrared spectroscopy (NIRS) untuk kasus respon bayi dalam mengenali pola suara.

Saya juga merupakan dosen tetap di Departemen Statistika ITS sejak tahun 2017 sampai saat ini. Sebelumnya saya pernah bekerja sebagai Data Analyst di GDP Labs (full-time) dan KASKUS (part-time) dari 2016 hingga 2019.

Research interest saya lainnya juga meliputi Gaussian process, Nonparametric regression, dan Nonparametric Bayesian models. Saat studi S2 saya pernah mengambil mata kuliah Analisis Bayesian. Saya tertarik mempelajari Bayesian karena estimasi Bayesian sangat efektif terutama untuk model-model yang sangat kompleks. Saya senang jika bisa bergabung dalam study group ini karena saya belum punya banyak teman diskusi khususnya tentang Bayesian.

Tes kemampuan statistika: P-value hacking

P-value hacking adalah suatu teknik manipulasi data untuk menghasilkan nilai p-value yang signifikan. P-value hacking dapat dilakukan dengan berbagai cara. Beberapa di antaranya adalah memonitor data, menambah data, dan menghentikan pengumpulan data sampai diperoleh p-value yang diinginkan, memilih-milih jenis uji, memilih-milih parameter, memilih-milih sampel, dan sebagainya. Ini menjadi buruk untuk sains karena hal ini dapat membuat kesimpulan-kesimpulan yang tidak valid seolah-olah menjadi valid. Para peneliti terkadang menjadi seolah-olah terpaksa untuk melakukan p-value hacking demi publikasi karena publikasi yang layak terbit haruslah menunjukkan temuan baru yang signifikan. Sementara itu, peneliti lain terkadang enggan untuk mengkonfirmasi kebenaran dari suatu publikasi karena penelitian yang hanya berupa replikasi dari penelitian yang sudah ada dan tidak ada kontribusi kebaruan yang diberikan, tidak layak untuk dipublikasikan.

Kita coba lakukan simulasi p-value hacking pada uji mean. Hacking yang dilakukan adalah memonitor dan menambah data hingga mencapai p-value signifikan dengan $\alpha=5\%$. Skenario yang dilakukan adalah membangkitkan data sebanyak 100 observasi berdistribusi normal dengan mean 0 dan standar deviasi 100 (sengaja dipilih besar). Dimulai dengan mengambil 2 dari 100 data dan dihitung p-value nya. Kemudian ditambahkan data satu per satu hingga semua data terambil dan dihitung p-value untuk tiap pengambilannya. Kita gunakan R untuk implementasi simulasi data.

Hipotesis

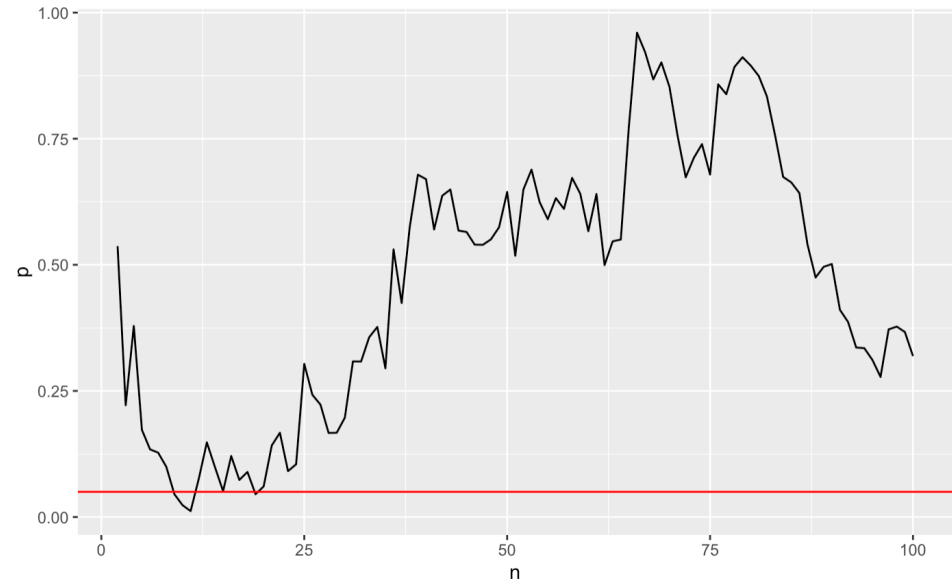
$H_0: \mu = 0$

$H_1: \mu \neq 0$

```
N <- 100
x <- rnorm(N, mean = 0, sd = 100)
M <- sample(1:N, N)
pval <- tibble(n = 2:N, p = rep(0,N-1), mean = rep(0,N-1))
for (i in 1:(N-1)) {
  pval$p[i] <- t.test(x[M[1:(i+1)]], mu=0)$p.value
  pval$mean[i] <- mean(x[M[1:(i+1)]])
}
ggplot(pval, aes(x=n, y=p)) +
  geom_line() +
  geom_hline(yintercept=0.05, color='red')
```

Tes kemampuan statistika: P-value hacking

Plot di samping menunjukkan nilai p-value untuk tiap jumlah data. Karena data yang dibangkitkan berasal dari distribusi normal dengan mean = 0, sd = 100 dan diuji mean dengan null hypothesis $\mu = 0$, maka seharusnya pengujian menghasilkan p-value yang tidak signifikan. Berdasarkan hasil simulasi, terdapat beberapa p-value yang ternyata signifikan (kurang dari 5%). P-value hacking dalam hal ini adalah menambah jumlah data sampai diperoleh p-value signifikan. Padahal, jika datanya terus ditambahkan, maka p-value yang diperoleh tidak akan signifikan.



Tes kemampuan statistika & coding: Koin FIFA

Solusi 1

Pendekatan Bayesian mengasumsikan bahwa parameter adalah variabel random. Informasi penggunaan koin pada 1925-1975 dapat kita jadikan sebagai suatu “prior”, dengan informasi $\theta = 5255/10000 = 0.5255$, yang dapat dikatakan bahwa koin ini seimbang (θ mendekati 0.5). Untuk mengestimasi parameter θ , kita akan menghitung posterior dari θ berdasarkan prior dan likelihood (data pada 2010-2020), yang diberikan oleh

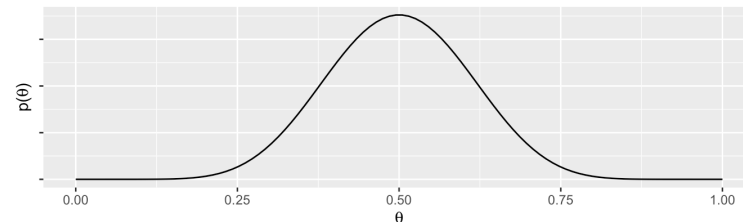
Posterior \propto Prior \times Likelihood

$$p(\theta|X) = \frac{p(\theta)p(X|\theta)}{\int_{\Theta} p(\theta)p(X|\theta)d\theta} \quad (1)$$

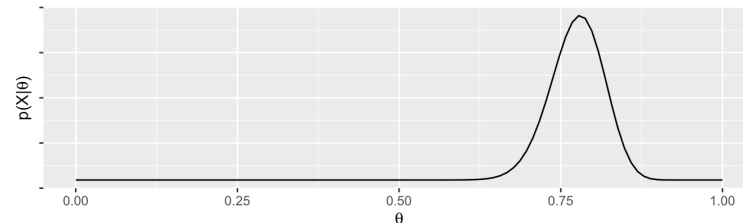
Kejadian pelemparan koin merupakan sebuah kejadian binomial. Maka, kita bisa menggunakan conjugate prior dari distribusi binomial yaitu distribusi beta. Karena diketahui pada tahun 1925-1975 muncul 5255 head dari 10000 pelemparan koin ($\theta = 0.5255$), pada kasus ini kita pilih prior $\theta \sim \text{Beta}(10,10)$ yang berbentuk simetris dengan puncak pada titik 0.5. Diketahui bahwa pada tahun 2010-2020, dari 100 pelemparan koin, 78 kali muncul head. Untuk kasus ini, perhitungan posterior $\theta|\text{data}$ dapat diperoleh secara **analitis**. Maka, dengan menggunakan (1), diperoleh $\theta|\text{data} \sim \text{Beta}(88, 32)$. Estimasi titik dapat diperoleh dengan menghitung ekspektasi $E(\theta|\text{data}) = (10+78)/(10+10+100) = 0.733$. Berikut implementasinya menggunakan R.

```
p_grid <- seq(0, 1, length.out = 100)
a <- b <- 10; n <- 100; y <- 78
# prior density
prior <- tibble(x = p_grid, y = dbeta(p_grid, a, b))
p1 <- ggplot(prior, aes(x=x,y=y)) +
  geom_line() +
  labs(x = TeX("$\\theta$"), y = TeX("$p(\\theta)$"), title = "Prior density") +
  theme(axis.text.y = element_blank())
# likelihood
likelihood <- tibble(x = p_grid, y = sapply(p_grid, function(p) dbinom(y,n,p)))
p2 <- ggplot(likelihood, aes(x=x,y=y)) +
  geom_line() +
  labs(x = TeX("$\\theta$"), y = TeX("$p(X | \\theta)$"), title = "Likelihood") +
  theme(axis.text.y = element_blank())
# posterior density
posterior <- tibble(x = p_grid, y = dbeta(p_grid, a+y, b+n-y))
lower_lim <- qbeta(0.05, a+y, b+n-y)
upper_lim <- qbeta(0.95, a+y, b+n-y)
p3 <- ggplot(posterior, aes(x=x,y=y)) +
  geom_line() +
  geom_area(data = filter(posterior, between(x, lower_lim, upper_lim)), fill = 'grey') +
  labs(x = TeX("$\\theta$"), y = TeX("$p(\\theta | X)$"),
       title = "Posterior density with 90% credible interval") +
  theme(axis.text.y = element_blank())
grid.arrange(p1, p2, p3, ncol = 1)
```

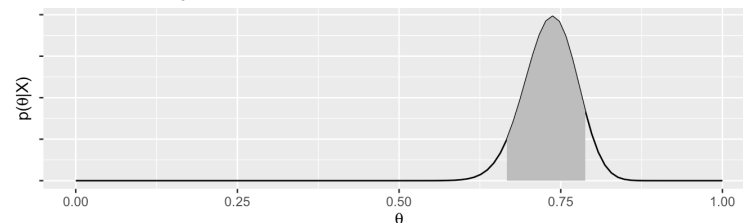
Prior density



Likelihood



Posterior density with 90% credible interval



Tes kemampuan statistika & coding: Koin FIFA

Solusi 2

Estimasi Bayesian sangat bergantung pada prior yang kita gunakan. Kita coba menggunakan prior yang berbeda, yaitu distribusi Uniform(0.3, 0.8), yang artinya kita mengasumsikan bahwa nilai peluang munculnya head tersebar secara seragam di antara 0.3 dan 0.8. Perhitungan posterior dilakukan secara sederhana dengan menggunakan Grid Approximation. Implementasinya dengan R adalah sebagai berikut.

```
p_grid <- seq(0, 1, length.out = 100)
prior <- dunif(p_grid, min = 0.3, max = 0.8)
likelihood <- dbinom(y, size=n, prob=p_grid)
posterior <- prior * likelihood / sum(prior * likelihood)
par(mfrow = c(3,1))
plot(p_grid, prior, xlab=TeX("$\\theta$"),
     ylab=TeX("$p(\\theta)$"), main = "Prior", type="l")
plot(p_grid, likelihood, xlab=TeX("$\\theta$"),
     ylab=TeX("$p(X | \\theta)$"), main = "Likelihood", type="l")
plot(p_grid, posterior, xlab=TeX("$\\theta$"),
     ylab=TeX("$p(\\theta | X)$"), main = "Posterior", type="l")
```

```
```{r}
Estimasi titik parameter (ekspektasi)
sum(p_grid * posterior)
```
```

```
[1] 0.7574179
```

