# Bayesian và bài toán đồng xu (fairness of coin)

# Objective

* Nắm bắt khái niệm cơ bản Bayesian-MCMC
* So sánh cách tiếp cận bài toán trên quan điểm frequentis và bayesian
* Xây dựng mô hình stan trên R cho bài toán bằng package bayvl

# Problem

Bài toán tung đồng xu là bài toán kinh điển trong xác suất thống kê. Khi tung đồng xu, ta có xác xuất đồng xu ra mặt hình người (head) hoặc ngược lại.

Nếu đồng xu là cân bằng (fair) ta có xác xuất tung được mặt head là 50%. Nếu gọi khả năng ra mặt head là θ, với đồng xu cân bằng ta sẽ có θ = 0.5

Bây giờ nếu chúng ta giả định đồng xu của chúng ta bị méo (bias coin). Khả năng khi tung ra 2 mặt head và tail không đều nhau. Như thế ta sẽ có θ của các đồng xu méo khác nhau từ θ = 0.0 đến θ = 1.0. Có thể coi giá trị θ này như mức độ cân bằng hai mặt (fairness) của đồng xu.

Nếu chúng ta có 1 túi xu, nhặt 1 đồng xu bất kỳ, ta sẽ có θ của đồng xu i là θi. Nếu gọi xác xuất chúng ta nhặt được đồng xu là p, ta sẽ có xác xuất nhặt được đồng xu cân bằng là p(θ = 0.5), hay nói cách khác, p(θ = 0.5) là khả năng nhặt được đồng xu cân bằng.

Bài toán: giả sử ta có 1 đồng xu, liệu ta có thể xác định được θ của đồng xu không? Hay nói cách khác, khi có 1 đồng xu, khả năng tung đồng xu để được mặt head là bao nhiêu?

# Solve

Để giải quyết bài toán này, chúng ta sẽ thử đi theo 2 cách tiếp cận bằng frequentis và bằng baysian.

**Thực nghiệm:**

Thu thập dữ liệu thực nghiệm được thực hiện bằng cách tung thử đồng xu, khi được mặt head ta đánh dấu 1, khi được mặt tail ta đánh dấu 0, sau n lần thực nghiệm ta sẽ được một chuỗi kết quả. Ví dụ:

Flip trial sequences = 011010000

**Hướng tiếp cận frequentist:**

Sau n lần thực nghiệm:

θ = tổng số lần ra mặt head / n số lần gieo

Giả sử khi chúng ta tung 10 lần cho kết quả các lần tung liên tiếp như ví dụ phần thực nghiệm (1 là mặt head, 0 là tail)



0.666

0.75

0.4

Ta có thể thấy:

* Nếu gieo thử 3 lần kết quả 101 ta có θ = 2/3 = 0.666
* Nếu gieo thử 4 lần kết quả 1011 ta có θ = 3/4 = 0.75
* ….
* Nếu gieo thử 10 lần kết quả 1011010000 ta có θ = 0.4

Như vậy mặc dù đồng xu không thay đổi, θ của đồng xu tính được lại thay đổi sau mỗi lần thực nghiệm.

Ta có thể viết 1 script ngắn trên R để mô phỏng quá trình gieo thử Ntoss lần đồng xu có xác xuất mặt head là θ như sau:

|  |
| --- |
| theta <- 0.5 # this is a fairness of the coin  Ntoss <- 10 # number of trials  # flip the coin Ntoss times  flips <- rbinom(n = Ntoss,  size = 1,  prob = theta) |

Ta xử dụng hàm ngẫu nhiên rbinom trên R để tạo ra loạt Ntoss mẫu thử cho đồng xu với fairness bằng theta. Như vậy ta có thể mô phỏng quá trình gieo thử nhiều lần với các đồng xu giả định ở các θ khác nhau tuỳ ý.

Ta có thể thử thực hiện thực nghiệm này nhiều lần bằng chạy lệnh:

|  |
| --- |
| > flips <- rbinom(n = Ntoss, size = 1, prob = theta)  > flips  [1] 0 0 0 1 1 0 0 0 1 1 |

Dễ nhận thấy mặc dù giá trị theta giữ nguyên nhưng mỗi lần thực nghiệm ta lại có chuỗi kết quả khác nhau. Sử dụng lệnh length(flips[flips==1]) để lấy số lần thử ra mặt head, ta có thể tính lại giá trị θ từ thực nghiệm bằng lệnh:

|  |
| --- |
| Nheads = length(flips[flips==1])  thetaTrial = Nheads / Ntoss  thetaTrial |

Đoạn script sau đây sẽ thử gieo đồng xu 500 lần và tính lại θ của đồng xu sau mỗi lần gieo và lưu lại vào vector trialTheta với giả định đồng xu là cân bằng (θ = 0.5).

|  |
| --- |
| theta <- 0.5 # this is a fair coin  Ntoss <- 500  # Simulate 500 toss  flips <- rbinom(n = Ntoss, size = 1, prob = theta)  trialTheta = c()  nHead = 0 # number of heads  for (i in 1: Ntoss) {  # Cumulative number of heads at step i  nHead = nHead + flips[i]  # Compute the running proportion of heads  trialTheta = c(trialTheta, nHead / i)  # Print proportion of heads at step i  print(paste0("Theta after ", i, " trials is ", trialTheta[i]))  } |

Nếu vẽ lại θ thực nghiệm sau mỗi lần gieo ta bằng script có đồ thị 500 lần gieo như sau:

|  |
| --- |
| # Graph the running proportion  plot( 1:Ntoss, trialTheta , type="o" , log="x" , col="skyblue" , xlim=c(1, Ntoss) , ylim=c(0.0,1.0) , cex.axis=1.5 ,  xlab="Flip Number" , ylab="Proportion Heads" , cex.lab=1.5 , main="Running Proportion of Heads" , cex.main=1.5 )  # Plot a dotted horizontal reference line at theta  abline( h=theta , lty="dotted" )  text( 1 , theta, theta, adj=c(1,1.2) , cex=1.3 ) |



Như vậy ta thấy giá trị θ tính toán bằng thực nghiệm của đồng xu sẽ thay đổi và có xu hướng hội tụ dần về giá trị θ thực sự của đồng xu. Số lần thử càng nhiều, có vẻ như giá trị sẽ càng sát, nhưng vấn đề là bao nhiêu mẫu thử nghiệm thì là “đạt”? Có cách nào để đánh giá độ tin cậy của kết quả không? Kết quả θ = 0.4989 ở lần gieo thứ 495 có đáng tin cậy hơn kết quả θ = 0.513 ở lần gieo thứ 487 không? Có thể thấy ở đồ thị trên, kết quả ở lần gieo thứ 5 chính xác hơn rất nhiều ở lần gieo thứ 15, tuy nhiên chúng ta lại không thể xác định được, ở chuỗi thực nghiệm khác có thể có đồ thị hoàn toàn khác.

Tổng kết lại xử lý bài toán đồng xu bằng phương pháp frequentist ta gặp phải 1 số vấn đề:

* Độ chính xác của nghiên cứu phụ thuộc rất nhiều vào số lượng mẫu thực nghiệm
* Kết quả chỉ chính xác khi có đủ số mẫu nhưng không thể đánh giá bao nhiêu là đủ
* Không có cách nào đánh giá được độ tin cậy của kết quả sau mỗi lần thực nhiệm

**Hướng tiếp cận bayesian:**

Trở lại với đồng xu với thực nghiệm 10 lần gieo cho chuỗi kết quả D=1011010000 ở trên.

Theo định lý bayes áp dụng với chuỗi dữ liệu D và đồng xu θ ta có:

Với hướng tiếp cận bayesian:

Bước 1: để xác định tham số θ, ta bắt đầu từ giả định nếu ta có 1 đồng xu với độ lệch θ bất kỳ, xác suất gieo khi 10 lần cho kết quả giống giống như chuỗi D là bao nhiêu?

Hàm tính xác suất ở bài toán này chính là likelihood của mô hình.

Với mỗi giá trị thực nghiệm yi trong chuỗi D, nếu yi = 1 (đồng xu ra mặt head) khi đó:

P(Y=1 | θ)=θ

Ngược lại nếu yi = 0:

P(Y=0 | θ)=1−θ

Như vậy xác xuất ra chuỗi kết quả D của đồng xu θ là:

Đây chính là công thức của hàm Probability Mass Function (pmf) của phân phối Bernoulli.

Bước 2: ta đưa ra dự đoán khả năng đồng xu của ta có θ bằng bao nhiêu. Dự đoán này sẽ là hàm prior

Ví dụ: ta nghiêng về khả năng đồng xu của chúng ta là đồng xu cân bằng θ = 0.5. Hàm prior ta chọn là hàm phân phối chuẩn