Red ball Green Ball

2024-08-29

Abstract

A dive into renewal processes. Find the web-page here.

Probability is a fascinating yet often perplexing field because it deals with uncertainty and randomness, concepts that can be challenging to intuitively grasp. Many probability problems appear simple on the surface but reveal surprising and counterintuitive results upon closer inspection. For example, the Monty Hall problem, where switching doors in a game show leads to better odds of winning, defies most people's gut instincts. This highlights the importance of rigorous mathematical analysis in probability. Without careful calculation and logical reasoning, our intuitive judgments can lead us astray, underscoring the need for a systematic approach to understanding and applying probability in real-world situations. Today, we will tackle such a problem.

1 Problem Statement

Quoting from Quanta magazine, [1]

" Imagine, that you have an urn filled with 100 balls, some red and some green. You can't see inside; all you know is that someone determined the number of red balls by picking a number between zero and 100 from a hat. You reach into the urn and pull out a ball. It's red. If you now pull out a second ball, is it more likely to be red or green (or are the two colors equally likely)? "

— Daniel Litt

Let us define the number of balls in the urn as N. Originally, there are $N+1$ different possible configurations for the urns with the number of red balls from 0 to N. Let is label the urns as u_i, with $i \in \{0, 1, 2, \cdots, N\}$. Since the number of red balls is pulled from a uniform distribution, the probability of getting any u_i is simply $\frac{1}{N+1}$.

First, we will shut-up and apply the rigorous machinery of probability theory. Second, we will do a simulation to double check the results.

2 Shut up and calculate

Given the first ball is red, the probability that it came from uln u_i (i.e., an uln with i red balls) is

$$P(u_i|x_1 = R) = \frac{P(x_1 = R|u_i)P(u_i)}{P(x_1 = R)}, \tag{1}$$

which is the Bayesian formula. We can compute the denominator as

$$P(x_1 = R) = \sum_{i=0}^{N} P(x_1 = R|u_i)P(u_i) = \sum_{i=0}^{N} \frac{i}{N}\frac{1}{N+1} = \frac{1}{2}. \tag{2}$$

And the numerator as

$$P(x_1 = R|u_i)P(u_i) = \frac{i}{N}\frac{1}{N+1} = \frac{i}{N(N+1)}. \tag{3}$$

Putting Eqs. (2) and (3) back into Eq. (1) gives

$$P(u_i|x_1 = R) = \frac{2i}{N(N+1)}. \tag{4}$$

Note that this is very important. Originally getting an urn with i red balls was flat $\frac{1}{N+1}$. However, the fact that first ball is red implies that it is more likely that it came from an urn with more red balls, see Fig. 4.

Now that we have a probability distribution for having u_i, we can compute the probability of getting a red ball in the second draw:

$$\begin{aligned} P(x_2 = R) &= \sum_{i=0}^{N} P(x_2 = R|u_i)P(u_i|x_1 = R) = \sum_{i=0}^{N} \frac{i-1}{N-1}\frac{2i}{N(N+1)} = \frac{2}{(N-1)N(N+1)}\sum_{i=0}^{N}\left(i^2 - i\right) \\ &= \frac{2}{(N-1)N(N+1)}\left[\frac{N(N+1)(2N+1)}{6} - \frac{N(N+1)}{2}\right] = \frac{2N+1}{3(N-1)} - \frac{1}{N-1} \\ &= \frac{2}{3} \end{aligned} \tag{5}$$

Therefore, getting a red ball is twice as likely as getting a green ball, independent of how many balls there are in the urn.

3 A tweak

The statement *"All you know is that someone determined the number of red balls by picking a number between zero and 100 from a hat"* is so critical. In order to show why that is, let's revise the preparation of the urn as follows: *"The urn is prepared by adding balls to the urn one by one by flipping a coin. If the coin is heads, you add a red ball, a green one otherwise."* This will completely transform the question. The number of red balls in the urn will have the the binomial probability density:

$$f_b(i, N) = \binom{N}{i} p^i (1-p)^{N-i}. \tag{6}$$

Now, we just need to redo the math. Given the first ball is red, the probability that it came from urn u_i (i.e., an urn with i red balls) is

$$P(u_i|x_1 = R) = \frac{P(x_1 = R|u_i)P(u_i)}{P(x_1 = R)}, \tag{7}$$

which is still the Bayesian formula. We can compute the denominator as

$$P(x_1 = R) = \sum_{i=0}^{N} P(x_1 = R|u_i)P(u_i) = \sum_{i=0}^{N} f_b(i, N)\frac{i}{N} = \frac{1}{N}\langle i \rangle = p. \tag{8}$$

For a fair coin used in the preparation $p = 1/2$. The numerator reads

$$P(x_1 = R|u_i)P(u_i) = \frac{i}{N} f_b(i, N). \tag{9}$$

Putting Eqs. (8) and (9) back into Eq. (7) gives

$$P(u_i|x_1 = R) = \frac{1}{Np} i f_b(i, N). \tag{10}$$

Now that we have a probability distribution for having u_i, we can compute the probability of getting a red ball in the second draw:

$$\begin{aligned}
P(x_2 = R) &= \sum_{i=0}^{N} P(x_2 = R|u_i)P(u_i|x_1 = R) = \frac{1}{pN(N-1)}\sum_{i=0}^{N}(i-1)if_b(i,N) = \frac{1}{pN(N-1)}\left(\langle i^2\rangle - \langle i\rangle\right) \\
&= \frac{1}{pN(N-1)}\left(\langle i\rangle^2 + \sigma^2 - Np\right) = \frac{1}{pN(N-1)}\left(N^2p^2 + Np(1-p) - Np\right) \\
&= \frac{1}{pN(N-1)}N(N-1)p^2 \\
&= p, \qquad (11)
\end{aligned}$$

which is the probability of the coin that was used to build the urn! Therefore, getting a red ball is as probable as getting a green one provided that the coin was unbiased.

4 Simulation

Here is a simulation code written in R. If you are interested in playing with the simulation, you can copy it below.

Show the simulation code (R)

Hide

```
# A code to simulate the problem https://www.quantamagazine.org/perplexing-the-web-one-probability-puzz
# Find the math at https://tetraquark.netlify.app/post/redballgreenball/redballgreenball/index.html?src
library(plotly)
colorize <- function(x) { if (x<0){"R"}else{"G"}} # will use this to map +1,-1 to colors
probBalls=0.5# Set this as a global variable.
Nballs=100;Nsim=20000; #set the number of balls in an urn, and the number of simulations
t2 <- list(size = 20)
wbinomProb<- function(x) {  x/Nballs* dbinom(x, size=Nballs, prob=probBalls) } # will use this to map +
binomProb<- function(x) {   dbinom(x, size=Nballs, prob=probBalls) } # will use this to map +1,-1 to co

firstBall=c();secondBall=c();gBalls=c();rBalls=c();BallsOneTwo=c() # initialize arrays to store various

simulator <- function(drawMode) {

  for(s in c(1:Nsim)){
    if(drawMode=="uniform"){
      randomRedCount=sample.int(Nballs, 1)  # this is how the urn is set up: number of red balls is pul
      balls=c(rep(-1,randomRedCount ),rep(1,Nballs-randomRedCount )) # repeat -1 randomRedCount times t
    }
    if(drawMode=="binomial"){
      balls=2*rbinom(Nballs, 1, probBalls)-1
      }
    ballsC=sapply(balls,colorize) # map numbers to colors
    NumOfGreens=round(Nballs/2+sum(balls)/2) #compute the number of Green balls
    gBalls=c(gBalls,NumOfGreens)# log the value for the green balls in this urn
    rBalls=c(rBalls,Nballs-NumOfGreens)# log the value for the red balls in this urn

    randomIndex=sample.int(length(balls), 1) # draw a random index, this will select a ball from the ur
    thisfirstBall=ballsC[randomIndex] # color of the first ball
    firstBall=c(firstBall,thisfirstBall) # log the first ball color
    ballsC=ballsC[-randomIndex]#remove this ball from the urn.

    randomIndex=sample.int(length(ballsC), 1) # draw another random index for the second ball
    thissecondBall=ballsC[randomIndex] # color of the second ball
    secondBall=c(secondBall,thissecondBall)# log the second ball
    BallsOneTwo=c(BallsOneTwo,paste0(thisfirstBall,"_",thissecondBall)) # Create a pair for later use
  }
  dtBall=data.frame("gBallsC"=gBalls,"rBallsC"=rBalls,"firstBall"=firstBall,"secondBall"=secondBall,"Bal
  dtBallS=dtBall[dtBall$firstBall=="R",] # we are told that the first ball is red, so, just keep these
  redRatio=nrow(dtBallS[dtBallS$secondBall=="R",])/nrow(dtBallS) # simply compute the ratio of counts o
  xv=c(1:Nballs);
  if(drawMode=="uniform"){
    titleText="Uniform Preperation"
    yv=2*xv/(Nballs*(Nballs+1))
    rAVGtheory=2/3
    yv0=rep(1/(1+Nballs),length(xv ))
    } # theoretical formula; https://tetraquark.netlify.app/post/redballgreenball/redballgreenball/inde
  if(drawMode=="binomial"){
    titleText="Binomial Preperation"
   yv=sapply(xv ,wbinomProb)
   yv0=sapply(xv ,binomProb);yv0=yv0/sum(yv0)

    yv=yv/sum(yv)
   rAVGtheory=probBalls

    } # theoretical formula; https://tetraquark.netlify.app/post/redballgreenball/redballgreenball/inde

    m <- list(l = 20,r = 10,b = 20, t = 20,pad = 8)
  figOut <- plot_ly(alpha = 0.5) %>%
    add_histogram(x = dtBallS$rBalls, name="Simulation", histnorm = "probability", nbinsx =Nballs)%>%
    add_trace(x = xv,y=yv, name="Revised", type='scatter', mode="markers+lines")%>%
    layout(xaxis=list(title="TeX($$\\Large{\\text{Number of Red Balls}}$$)"),yaxis=list( title="TeX($$\
    layout(title=list(y=0.95,x=0.2,text=paste0( titleText),font = t2))%>%
    layout(legend = list(x = 0.1, y = 0.85,orientation = 'h',font = t2),margin=m) #,plot_bgcolor  = "rg
  #figOut%>%config(mathjax = "cdn"). # enable this to render MathJax
  ;
    figOut<-figOut%>%  add_trace(x = xv,y=yv0, name="Original", type='scatter', mode="markers+lines")

  output<-list(figOut,redRatio, rAVGtheory)
  return(output)
}

if(FALSE){
  drawMode="uniform";
  #drawMode="binomial";
  returns=simulator(drawMode)
  redRatioFormatted=formatC(signif(returns[[2]], digits=3), digits=3, format="fg", flag="#")
  redRatioThFormatted=formatC(signif(returns[[3]], digits=3), digits=3, format="fg", flag="#")
  print(paste0("Nballs:",Nballs,";",drawMode, "-->probability of second ball being red is:",redRatioFor
  returns[[1]]
}
```

The code simulates the original problem with 100 balls and repeats it 20000 times.

Green: original density, Orange:The probability distribution of having an urn with i red balls given that the first ball was red, Blue: simulation results.

We can now simply count the cases of second ball being red, and the simulation result is 0.667 ± 0.007 with 95% confidence level which is very close to the 0.667 value we predicted in Eq. (5).

Below is the tweaked problem where the balls are decided with coin flip.

Green: original density, Orange:The probability distribution of having an urn with i red balls inside that the first ball was red, Blue: simulation results.

We can now simply count the cases of second ball being red, and the simulation result is 0.494 ± 0.007 with 95% confidence level which is very close to the 0.5 value we predicted in Eq. (11). The color of the first ball isn't really a useful information, it barely moves the distribution. In fact, the shift towards the higher values of red ball count exactly cancels the fact that we threw away a red ball in the first draw. It is a perfect cancellation!

[1] "Perplexing the web, one probability puzzle at a time," *Quanta Magazine*, 2024 [Online]. Available: https://www.quantamagazine.org/perplexing-the-web-one-probability-puzzle-at-a-time-20240829/. [Accessed: 30-Aug-2024]