Probability: Analytical vs Simulation Approaches

Tonnar Castellano

This week we will be discussing how to calculate the chance of something happening from two different perspectives. The first is analytically and the second is by simulation. Before getting started we should discuss what is the definition of both of these terms. The first is analytically. When we compute something analytically we figure out the actual chance of it happening. Another way to think of this is we are able to set up a mathematical formula for calculating the probability. In other words this gives us the true probability of an event occurring QED. Simulation on the other hand tries to figure the outcome through repeated trials.

The best way to try and understand would be through an example. Lets take dice even though that is somewhat of a boring case. There would be two ways to figure out the probability of getting an outcome from the dice. The first would be analytically. We make some assumptions about the outcomes being far then we divide one outcome by the rest. The other way would be to roll the dice repeatedly then tally up the chance of getting a desired outcome divided by the total outcomes.

You now might be wondering why we would ever use simulation to calculate the outcome of something when it has the potential for error. Well the answer is for certain events its easier to compute by simulation and for some its impossible to compute analytically. I will give an example of where it is easier to compute via simulation though not impossible analytically.

So lets say you wanted to analytically compute the chance of a team to win the world series. You would have to find all the out comes where they win four in four, four in five, four in six, and four in seven. This ends up coming out to 70 different outcomes and if you get one wrong or forget one the whole calculation is off. However, it is possible just difficult. If you decide to go through the trouble of the calculation you will find it comes out to roughly 60.4% chance of winning if the team has a 55% chance of winning each game with advantage, without advantage you have a 60.8%.

```
# Get all possible outcomes
apo <- fread("https://raw.githubusercontent.com/thomasgstewart/data-science-5620-fall-2020/master/deliv

# Home field indicator
hfi <- c(0,0,1,1,1,0,0) #{NYC, NYC, ATL, ATL, ATL, NYC, NYC}

# P_B
pb <- 0.55
advantage_multiplier <- 1 # Set = 1 for no advantage
pbh <- 0.55*advantage_multiplier
pba <- 1 - (1 - 0.55)*advantage_multiplier

# Calculate the probability of each possible outcome
apo[, p := NA_real_] # Initialize new column in apo to store prob
for(i in 1:nrow(apo)){
  prob_game <- rep(1, 7)
  for(j in 1:7){
    p_win <- ifelse(hfi[j], pbh, pba)
    prob_game[j] <- case_when(
        apo[i,j,with=FALSE] == "W" ~ p_win
      , apo[i,j,with=FALSE] == "L" ~ 1 - p_win
```

```
      , TRUE ~ 1
    )
  }
  apo[i, p := prod(prob_game)] # Data.table syntax
}

# Sanity check: does sum(p) == 1?
apo[, sum(p)] # This is data.table notation
```

```
## [1] 1
```

```
# Probability of overall World Series outcomes
apo[, sum(p), overall_outcome]
```

```
##    overall_outcome        V1
## 1:               W 0.6082878
## 2:               L 0.3917122
```

However, lets take the simulation approach. Here instead of having to come up with 70 different outcomes we simply find the chance of them winning one game simulate it happening then divide the total series wins by total series played. When we do this we get a chance of roughly 60.3% with advantage and without you have a 60.78%. When we find the absolute error is for no advantage and advantage is .001 and 0.0005 roughly. The relative error we find they are roughly .00182 and and .00089 for advantage and no advantage respectively. So we are very close for much less work. Hopefully this helps you to see the power of simulation.

```
simulation <- function(n, p, ...) {
  series_wins = 0
  series_lost = 0
  counter = 0

  while (counter < n) {
    wins = 0
    loses = 0

    while (loses != 4 & wins != 4) {
      result <- rbinom(1, 1, p)
      ifelse(result == 1, wins <- wins + 1, loses <- loses + 1)
    }

    ifelse(wins == 4,
           series_wins <- series_wins + 1,
           series_lost <- series_lost + 1)

    counter = counter + 1
  }
  total_prob <- series_wins / (series_wins + series_lost)
}
```

```
simulation_with_adv <- function(n, p, adv) {
  series_wins = 0
  series_lost = 0
  counter = 0

  while (counter < n) {
    wins = 0
    loses = 0
```

```
    game_played <- 1
    while (loses != 4 & wins != 4) {
      ifelse(
        game_played == 3 | game_played == 4 | game_played == 5,
        result <- rbinom(1, 1, p * adv),
        result <- rbinom(1, 1, 1 - (1 - p) * adv)
      )
      game_played <- game_played + 1
      ifelse(result == 1, wins <- wins + 1, loses <- loses + 1)
    }
    ifelse(wins == 4,
           series_wins <- series_wins + 1,
           series_lost <- series_lost + 1)

    counter = counter + 1
  }
  total_prob <- series_wins / (series_wins + series_lost)
}
print(simulation_with_adv(10,.55,1.1))
```

```
## [1] 0.3
```

```
amount_of_games <- seq(10,5000,10)
prob_winning <- rep(NA,length(amount_of_games))

i = 1
for(games in amount_of_games){
  prob_winning[i] <- simulation(games,.55,1.1)
  i = i + 1
}
no_adv <- prob_winning[length(prob_winning)]
no_adv
```

```
## [1] 0.6132
```

```
amount_of_games <- seq(10,5000,10)
prob_winning <- rep(NA,length(amount_of_games))

i = 1
for(games in amount_of_games){
  prob_winning[i] <- simulation_with_adv(games,.55,1.1)
  i = i + 1
}
with_adv <- prob_winning[length(prob_winning)]
with_adv
```

```
## [1] 0.6182
```

```
ae_no_adv <- abs(.6082878   - .6078)
re_no_adv <- ae_no_adv/.55
print(re_no_adv*100)
```

```
## [1] 0.08869091
```

```
print(ae_no_adv*100)
```

```
## [1] 0.04878
```