Coin Flip Guess Simulation

Justin Burruss

9/20/2021

Background

In this variation of the Levine Hat Puzzle using coin flips, players A and B are each given a sequence of as many fair random coin flips as necessary but can see only their own coin flips. Each player names an index in the unseen coin flip sequence of the other player, e.g., 2 for the other player's second coin flip. The players collectively win if and only if both name an index having matching outcomes, i.e., both are heads or both are tails. As with the basic Levine Hat Puzzle, the two players may agree upon a strategy ahead of time, but no communication is permitted after the strategy session.

At first glance it may be tempting to quickly answer that one strategy is as good as any other since fair coin tosses have random 50/50 outcomes. However, as with the basic Levine Hat Puzzle, there exist strategies which allow the players to win more than half the time.

Simulating the triplet-based strategy

Stack Exchange Puzzling community member Jaap Scherphuis posted the following strategy (adapted from the Levine Hat Puzzle). In theory, the probability of winning is 70% when following this triplet-based strategy.

> Split the coin tosses into triplets. Both players skip over any triplets that have three heads or three tails to get to the first triplet containing both heads and tails. For player A, if there is a single tail then pick that index. If there is a single head, then pick the index to the right of that one (wrapping around to the first of the triplet if the head is already at the right end). For player B, if there is a single tail then pick that index. If there is a single head, then pick the index to the left of that one (wrapping around to the last of the triplet if the head is already at the left end).

To validate, we repeatedly simulate games, outputting the result of two strategies: the triplet-based strategy vs. simply picking the first index.

```
library(data.table)

fGuess <- function(returnBitStrings = TRUE) {
    Ai <- 0L
    Bi <- 0L
    An <- sample.int(8, 1) - 1L
    Bn <- sample.int(8, 1) - 1L

    # A and B obtain trios of random bits (i.e., octal
    # digits) until their trio is not all 0s or all 1s
```

```r
while (An[Ai/3 + 1] == 0 || An[Ai/3 + 1] == 7) {
    Ai <- Ai + 3L
    An <- append(An, sample.int(8, 1) - 1L)
}

while (Bn[Bi/3 + 1] == 0 || Bn[Bi/3 + 1] == 7) {
    Bi <- Bi + 3L
    Bn <- append(Bn, sample.int(8, 1) - 1L)
}

# A and B apply their strategy for selecting an index

Ab <- switch(An[Ai/3 + 1], 0L, 2L, 0L, 1L, 1L, 2L)
Bb <- switch(Bn[Bi/3 + 1], 1L, 0L, 0L, 2L, 1L, 2L)

if (Ai <= Bi)
    A <- bitwAnd(Bn[Ai/3 + 1], 2^(2 - Ab))

if (Bi <= Ai)
    B <- bitwAnd(An[Bi/3 + 1], 2^(2 - Bb))

if (Ai < Bi) {
    while (Ai < Bi) {
        Ai <- Ai + 3L
        An <- append(An, sample(8, 1) - 1L)
    }
    B <- bitwAnd(An[Bi/3 + 1], 2^(2 - Bb))
}

if (Bi < Ai) {
    while (Bi < Ai) {
        Bi <- Bi + 3L
        Bn <- append(Bn, sample(8, 1) - 1L)
    }
    A <- bitwAnd(Bn[Ai/3 + 1], 2^(2 - Ab))
}

# If specified, also output a string of '0's and '1's;
# otherwise leave blank.

if (returnBitStrings == TRUE) {
    As <- ""
    Bs <- ""
    for (i in 1:length(An)) As <- paste(As, sign(bitwAnd(An[i],
        4)), sign(bitwAnd(An[i], 2)), sign(bitwAnd(An[i],
        1)), sep = "")
    for (i in 1:length(Bn)) Bs <- paste(Bs, sign(bitwAnd(Bn[i],
        4)), sign(bitwAnd(Bn[i], 2)), sign(bitwAnd(Bn[i],
        1)), sep = "")

    return(data.table(A = as.logical(A), B = as.logical(B),
        Match = (as.logical(A) == as.logical(B)), As = As,
        Bs = Bs))
```

```r
    }

    return(data.table(A = as.logical(A), B = as.logical(B), Match = (as.logical(A) ==
        as.logical(B))))
}

t0 <- Sys.time()

trials <- data.table(RunId = seq(1, 10^6, 1))
trials <- trials[, fGuess(), by = RunId]

t1 <- Sys.time()

# Display our wall clock run time
t1 - t0
```

```
## Time difference of 2.787701 mins
```

```r
# How successful was the triplet strategy? In theory this
# should be 70%.
sum(trials$Match)/nrow(trials)
```

```
## [1] 0.699638
```

```r
# Did A and B outcomes guess true/false equally? In theory
# these should be 50% true.
sum(trials$A)/nrow(trials)
```

```
## [1] 0.500775
```

```r
sum(trials$B)/nrow(trials)
```

```
## [1] 0.499651
```

```r
# What if we simply took the first flips? In theory these
# should match 50% of the time.
trials[, `:=`(FirstMatch, as.logical(substr(As, 1, 1) == substr(Bs,
    1, 1))), by = RunId]
sum(trials$FirstMatch)/nrow(trials)
```

```
## [1] 0.500205
```

```r
# How many flips per player were necessary?
trials[, .(TrialCnt = .N, TrialPct = .N/nrow(trials)), keyby = nchar(As)]
```

```
##     nchar TrialCnt TrialPct
##  1:     3   562530 0.562530
##  2:     6   316510 0.316510
##  3:     9    90097 0.090097
##  4:    12    23071 0.023071
```

```
##  5:    15      5793 0.005793
##  6:    18      1533 0.001533
##  7:    21       336 0.000336
##  8:    24       103 0.000103
##  9:    27        17 0.000017
## 10:    30         8 0.000008
## 11:    33         2 0.000002
```

Other strategies

Stack Exchange Mathematics community member Mike Earnest posted a simple first heads strategy: both players pick the index corresponding with the first occurrence of heads in their sequence. Theoretically this strategy yields a win 2/3 of the time.

Stack Exchange Mathematics community member Teo Miklethun posted a 2-flip strategy: players pick index 1 if their own first flip is tails, index 2 otherwise. This strategy should win 5/8 of the time and never requires more than two flips per player.

```r
# Run the first heads strategy. This is expected to yield
# 2/3.
trials[, `:=`(FirstHeadsMatch, as.logical(substr(As, regexpr("1",
    Bs), regexpr("1", Bs)) == substr(Bs, regexpr("1", As), regexpr("1",
    As)))), by = RunId]
sum(trials$FirstHeadsMatch)/nrow(trials)
```

```
## [1] 0.666319
```

```r
# Run the two-flip strategy. This is expected to yield 5/8
# (62.5%)
trials[, `:=`(TwoFlipMatch, as.logical(substr(As, 2 - (substr(Bs,
    1, 1) == "0"), 2 - (substr(Bs, 1, 1) == "0")) == substr(Bs,
    2 - (substr(As, 1, 1) == "0"), 2 - (substr(As, 1, 1) == "0")))),
    by = RunId]
sum(trials$TwoFlipMatch)/nrow(trials)
```

```
## [1] 0.62581
```

Conclusion

Analysis of 1,000,000 simulations finds that observation is consistent with theory: the triplet strategy described leads to success 70% of the time with the other strategies matching their theoretical 50% (first guess), 2/3 (first heads), and 5/8 (two-flip) success rates.