Practical 2.7

CHEN

2024-03-26

Lie detector problem

In a big store, around 10% of employees are stealing. Everybody has to take a lie detector test that is correct in 80% of cases (and mistakes are equally likely in either direction). Everybody says that they are not a thief.

If the lie detector says that 50 people are lying, how many of them are probably thieves? Work out the answer in two ways: 1. By setting up and running a simulation

```
p_real_thief <- 0.1
p_test_correct <- 0.8
simulation_test <- function(employee_size, prob_real_thief, test_accuracy) {
  employee <- sample(c(TRUE, FALSE), employee_size, replace = TRUE, prob = c(prob_real_thief, 1 - prob_
  results <- sapply(employee, function(test_to_be_thief) {
    if (test_to_be_thief) {
      return(sample(c(TRUE, FALSE), 1, prob = c(test_accuracy, 1 - test_accuracy)))
    } else {
      return(sample(c(TRUE, FALSE), 1, prob = c(1 - test_accuracy, test_accuracy)))
    }
  })

  real_thief <- sum(employee[results])
  tested_thief <- sum(employee)
  return(real_thief/tested_thief)
}

real_ratio <- replicate(1000, simulation_test(1000, p_real_thief, p_test_correct))
avg_ratio <- mean(real_ratio)

real_thief <- 50*avg_ratio
```

2. By using Bayes' theorem

Coin toss

Somebody tosses a fair coin repeatedly, and records the sequence of outcomes (e.g. "H-T-T-T-T-H-H-T-. . ."). How long would it take until the sequence "H-T-T-H" first appears? 1. Complete the Markov chain diagram we started in the lecture

2. Using the Markov chain, create a simulation and run it many times to answer your question

```r
# Define the Markov chain transitions
transitions <- list(
  'HHHH' = c('H' = 'HHHT', 'T' = 'THHH'),
  'HHHT' = c('H' = 'HHTH', 'T' = 'THHH'),
  'HHTH' = c('H' = 'HTTH', 'T' = 'THHH'),
  'HTHH' = c('H' = 'THHT', 'T' = 'THHH'),
  'HTHT' = c('H' = 'THTH', 'T' = 'THHH'),
  'HTTH' = c('H' = 'THHH', 'T' = 'THTH'),
  'THHH' = c('H' = 'HHHT', 'T' = 'THHT'),
  'THHT' = c('H' = 'HHTH', 'T' = 'THHH'),
  'THTH' = c('H' = 'HTTH', 'T' = 'THHH'),
  'TTTH' = c('H' = 'TTHH', 'T' = 'THHH'),
  'TTHT' = c('H' = 'THTH', 'T' = 'THHH'),
  'TTHH' = c('H' = 'THHH', 'T' = 'TTHH')
)

# Function to simulate until the target sequence appears
simulate_until_target <- function() {
  current_state <- 'HHHH'
  steps <- 0
  while (TRUE) {
    next_toss <- sample(c('H', 'T'), 1, replace = TRUE)
    current_state <- transitions[[current_state]][[next_toss]]
    steps <- steps + 1
    if (current_state == 'HTTH') {
      return(steps)
    }
  }
}

# Run simulation many times
num_simulations <- 10000
total_steps <- 0
for (i in 1:num_simulations) {
  total_steps <- total_steps + simulate_until_target()
}

# Calculate average steps
average_steps <- total_steps / num_simulations
print(paste("Average steps until 'H-T-T-H' appears:", average_steps))
```

```
## [1] "Average steps until 'H-T-T-H' appears: 10.5219"
```