Thesis Simulation Results Analysis for Chapter 4

Dasha Asienga

2024-04-01

Contents

This file is intended to synthesize and analyze the results from the simulation.

Reading in the Result Data Sets

Logistic Regression

The results data set has 200 observations for each of the simulation trials, 50 from each sample size: $n = 500, 1000, 2500, 5000$.

```
lr_500 <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/R/Simulation/LogisticRegression/Res
lr_1000 <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/R/Simulation/LogisticRegression/Re
lr_2500 <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/R/Simulation/LogisticRegression/Re
lr_5000 <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/R/Simulation/LogisticRegression/Re

lr_500 <- lr_500 |>
  mutate(sample_size = 500) |>
  dplyr::select(-X)

lr_1000 <- lr_1000 |>
  mutate(sample_size = 1000) |>
  dplyr::select(-X)

lr_2500 <- lr_2500 |>
  mutate(sample_size = 2500) |>
  dplyr::select(-X)
```

```
lr_5000 <- lr_5000 |>
  mutate(sample_size = 5000) |>
  dplyr::select(-X)

logistic_results <- rbind(lr_500, lr_1000, lr_2500, lr_5000)
glimpse(logistic_results)
```

```
## Rows: 200
## Columns: 5
## $ dataset_id        <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ lr_convergence    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ lr_accuracy       <dbl> 0.790, 0.738, 0.778, 0.758, 0.788, 0.822, 0.776, 0.7~
## $ lr_discrimination <dbl> 0.2020, 0.2402, 0.2730, 0.3123, 0.2663, 0.2171, 0.31~
## $ sample_size       <dbl> 500, 500, 500, 500, 500, 500, 500, 500, 500, 500, 50~
```

Seldonian Solutions

The results data set has 200 observations for each of the simulation trials, 50 from each sample size: $n = 500, 1000, 2500, 5000$.

```
seldonian_results <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/Python/COMPAS Simulatio
seldonian_results <- distinct(seldonian_results) #remove duplicate rows
glimpse(seldonian_results)
```

```
## Rows: 200
## Columns: 14
## $ sample_size       <int> 1000, 1000, 1000, 2500, 1000, 1000, 1000, 2500, 500,~
## $ dataset_id        <int> 25, 10, 22, 17, 34, 38, 36, 2, 49, 17, 7, 41, 8, 33,~
## $ passed_safety_02  <chr> "True", "True", "True", "True", "True", "True", "Tru~
## $ passed_safety_01  <chr> "True", "True", "True", "True", "True", "True", "Tru~
## $ passed_safety_005 <chr> "True", "True", "True", "True", "True", "True", "Tru~
## $ passed_safety_001 <chr> "True", "True", "True", "False", "True", "True", "Tr~
## $ sa_02_accuracy    <dbl> 0.6420, 0.6410, 0.6370, 0.7832, 0.5520, 0.6180, 0.73~
## $ sa_01_accuracy    <dbl> 0.5560, 0.5030, 0.5200, 0.4844, 0.4930, 0.5190, 0.49~
## $ sa_005_accuracy   <dbl> 0.5330, 0.5030, 0.5460, 0.4844, 0.4930, 0.5200, 0.49~
## $ sa_001_accuracy   <dbl> 0.5430, 0.5030, 0.5830, 0.4844, 0.4930, 0.5190, 0.49~
## $ sa_02_disc_stat   <dbl> 0.1791, 0.0995, 0.0948, 0.1428, 0.0787, 0.1081, 0.22~
## $ sa_01_disc_stat   <dbl> 0.0345, 0.0000, 0.0355, 0.0000, 0.0000, 0.0000, NA, ~
## $ sa_005_disc_stat  <dbl> 0.0184, 0.0000, 0.0496, 0.0000, 0.0000, NA, 0.0000, ~
## $ sa_001_disc_stat  <dbl> 0.0252, 0.0000, 0.0347, 0.0000, 0.0000, 0.0000, 0.00~
```

Combine the Data Sets from Both Simulations

```
sim_results <- inner_join(logistic_results, seldonian_results,
                          by = c("sample_size", "dataset_id"))

glimpse(sim_results)
```

```
## Rows: 200
## Columns: 17
## $ dataset_id        <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ lr_convergence    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ lr_accuracy       <dbl> 0.790, 0.738, 0.778, 0.758, 0.788, 0.822, 0.776, 0.7~
## $ lr_discrimination <dbl> 0.2020, 0.2402, 0.2730, 0.3123, 0.2663, 0.2171, 0.31~
## $ sample_size       <dbl> 500, 500, 500, 500, 500, 500, 500, 500, 500, 500, 50~
## $ passed_safety_02  <chr> "True", "True", "True", "True", "True", "True", "Tru~
## $ passed_safety_01  <chr> "True", "True", "False", "True", "True", "False", "T~
## $ passed_safety_005 <chr> "True", "True", "False", "True", "True", "False", "T~
## $ passed_safety_001 <chr> "True", "True", "False", "True", "True", "False", "T~
## $ sa_02_accuracy    <dbl> 0.526, 0.510, 0.528, 0.520, 0.512, 0.508, 0.608, 0.5~
## $ sa_01_accuracy    <dbl> 0.522, 0.510, 0.528, 0.520, 0.496, 0.508, 0.482, 0.5~
## $ sa_005_accuracy   <dbl> 0.522, 0.510, 0.528, 0.520, 0.496, 0.508, 0.756, 0.5~
## $ sa_001_accuracy   <dbl> 0.522, 0.510, 0.528, 0.520, 0.496, 0.508, 0.482, 0.5~
## $ sa_02_disc_stat   <dbl> NA, 0.0000, 0.0000, 0.0000, 0.0429, 0.0000, 0.0106, ~
## $ sa_01_disc_stat   <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.00~
## $ sa_005_disc_stat  <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.13~
## $ sa_001_disc_stat  <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.00~
```

Probability of a Solution

This section assesses what proportion of the trials returned a solution. It is expected that all logistic regression trials will return a solution. However, for the Seldonian algorithms, while all trials will return a candidate solution (based on the logistic regression as a starting point), it is expected that not all candidate solutions will pass the safety test. The table below records the number of Seldonian solutions that passed the safety test in each sample size.

```r
reps <- nrow(sim_results)/4

sim_results |>
  group_by(sample_size) |>
  summarise(LR = 100*count(lr_convergence == 1)/reps,
            `SA (0.2)` = 100*count(passed_safety_02 == "True")/reps,
            `SA (0.1)` = 100*count(passed_safety_01 == "True")/reps,
            `SA (0.05)` = 100*count(passed_safety_005 == "True")/reps,
            `SA (0.01)` = 100*count(passed_safety_001 == "True")/reps) |>
  rename("Sample Size" = sample_size) |>
  kable()
```

Sample Size	LR	SA (0.2)	SA (0.1)	SA (0.05)	SA (0.01)
500	100	100	90	86	70
1000	100	100	100	100	88
2500	100	100	100	100	52
5000	100	100	100	96	28

Accuracy

```r
sim_results_converged <- sim_results
```

only seldonian solutions – need to break down for each then combine back together avg, se, table, visuals

Discrimination

only seldonian solutions – need to break down for each then combine back together avg, se, table, visuals

number of times constraint was satisfied by both (table)

NSF

compare these solutions with logistic regression