BDA - Assignment 9

Anonymous

Contents


```
library(aaltobda)
library(cmdstanr)
library(posterior)
library("loo")
data("factory")
```

```
data <- list(
  J = ncol(factory),
  N = nrow(factory),
  y = factory
)
model = cmdstan_model(stan_file="hierarchical.stan")
fit = model$sample(data = data, refresh=0)
draws = as_draws_df(fit$draws())
```

Exercise 1 - Utility function

```
utility = function(draws) {

  n = length(draws)

  good_enough = length(draws[draws >= 85])

  win = good_enough * 200

  costs = n * 106

  profit = win - costs

  profit_per_product = profit / n

  return(profit_per_product)
```

```
}
```

Exercise 2 - Expected utility from machine 1-6

```
all_utility = c()
for (j in 1:ncol(factory)) {

  key = sprintf("ypred[%i]", j)
  single_utility = utility(draws=draws[[key]])

  all_utility = append(all_utility, single_utility)

}
print("The uitlities from machine 1 - 6:")
```

```
## [1] "The uitlities from machine 1 - 6:"
```

```
print(all_utility)
```

```
## [1] -13.20  84.60  40.75  88.40  49.95  31.55
```

```
print("The machines ordered by their uitlity from worst to best:")
```

```
## [1] "The machines ordered by their uitlity from worst to best:"
```

```
print(order(all_utility))
```

```
## [1] 1 6 3 5 2 4
```

A positive utility value means that the machine is profitable –> the earned money from the sold products exceed the costs. A negative utility value instead means the contrary, that the machine is not profitable. Only machine 1 is not profitable. Since the value is normalized for one product, we could compare machines with different number of products to each other.

Exercise 3 - Expected utility from machine 7

```
utility_7 = utility(draws=draws$ypred_7)

print("The expected utility of the 7th machine")
```

```
## [1] "The expected utility of the 7th machine"
```

```
print(utility_7)
```

```
## [1] 55
```

Exercise 4 - Analysis

Since the expected utility from the 7th machine is positive, the company owener should buy a new machine in order to increase their utility function which is more money.

Exercise 5 - Stan code

```r
writeLines(readLines("hierarchical.stan"))
```

```
## // Hierarchical model
## data {
##  int < lower = 0 > J;
##  int < lower = 0 > N;
##  matrix[N,J] y;
## }
##
## parameters {
##    real mu_theta;
##    real < lower = 0 > sigma_theta;
##    vector[J] mu;
##    real < lower = 0 > sigma;
## }
##
## model {
##    // hyper prior
##    mu_theta ~ normal(0,100);
##    sigma_theta ~ normal(0,1);
##
##    // priors
##    sigma ~ normal(0,1);
##    for (j in 1: J) {
##          mu[j] ~ normal(mu_theta, sigma_theta);
##    }
##
##    // likelihood
##    for (j in 1: J) {
##          y[, j] ~ normal(mu[j], sigma);
##    }
##   }
##
## generated quantities {
##
##      vector[J] ypred;
##      real mu_7;
##      real ypred_7;
##
##  for (j in 1: J) {
##      ypred[j] = normal_rng(mu[j], sigma);
##  }
##  mu_7 = normal_rng(mu_theta, sigma_theta);
##  ypred_7 = normal_rng(mu_7, sigma);
##
## }
```