BDA - Assignment 9

Contents

```r
library("rstan")
```

```
## Loading required package: StanHeaders

## Loading required package: ggplot2

## rstan (Version 2.19.2, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)

## For improved execution time, we recommend calling
## Sys.setenv(LOCAL_CPPFLAGS = '-march=native')
## although this causes Stan to throw an error on a few processors.
```

```r
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
library(aaltobda)
data("factory")
```

1) Model Implementation

The implementation here is very similiar to the previous assignment, which is showed below. We can found the test result is correct (-26).

```r
stan_code_hierarchical = '
data {
int<lower=0> N; // number of data points
int<lower=0> K; // number of groups
int<lower=1,upper=K> x[N]; // group indicator
vector[N] y;
}
parameters {
real mu0; // prior mean
real<lower=0> sigma0; // prior std
vector[K] mu; // group means
real<lower=0> sigma; // common std
}
model {
mu ~ normal(mu0, sigma0);
y ~ normal(mu[x], sigma);
}
generated quantities {
real ypred1;
real ypred2;
real ypred3;
real ypred4;
real ypred5;
real ypred6;
real ypred7;
real mu7;
ypred1 = normal_rng(mu[1], sigma);
ypred2 = normal_rng(mu[2], sigma);
ypred3 = normal_rng(mu[3], sigma);
ypred4 = normal_rng(mu[4], sigma);
ypred5 = normal_rng(mu[5], sigma);
ypred6 = normal_rng(mu[6], sigma);
mu7 = normal_rng(mu0, sigma0);
ypred7 = normal_rng(mu7, sigma);
}'
```

Then we need to fit it and use it to generate desired values.

```
data_hierarchical <- list(
N=length(c(as.matrix(factory))),
K=6,
x=c(
1, 1, 1, 1, 1,
2, 2, 2, 2, 2,
3, 3, 3, 3, 3,
4, 4, 4, 4, 4,
5, 5, 5, 5, 5,
6, 6, 6, 6, 6
),
y=c(as.matrix(factory))
)
fit_hierarchical <- stan(model_code = stan_code_hierarchical, data = data_hierarchical)
```

```
## Warning: There were 16 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
extract_hierarchical <- rstan::extract(fit_hierarchical, permuted = T)
```

And here's the function to compute utility of a given machine.

```
compute_utility <- function(draws){
sum <- -106*length(draws)
for (i in draws){
if (i >= 85){
sum <- sum+200
}
}
sum/length(draws)
}
```

```
y_pred <- c(123.80, 85.23, 70.16, 80.57, 84.91)
compute_utility(y_pred)
```

```
## [1] -26
```

2) Ranking

Let's rank the machnes based on the expected utilities.

```
cat('expected utility of 1st machine:',compute_utility(draws = extract_hierarchical$ypred1),'\n')
```

```
## expected utility of 1st machine: -32.45
```

```
cat('expected utility of 2nd machine:',compute_utility(draws = extract_hierarchical$ypred2),'\n')
```

```
## expected utility of 2nd machine: 66.7
```

```
cat('expected utility of 3rd machine:',compute_utility(draws = extract_hierarchical$ypred3),'\n')
```

```
## expected utility of 3rd machine: 15.1
```

```
cat('expected utility of 4th machine:',compute_utility(draws = extract_hierarchical$ypred4),'\n')
```

```
## expected utility of 4th machine: 74.35
```

```
cat('expected utility of 5th machine:',compute_utility(draws = extract_hierarchical$ypred5),'\n')
```

```
## expected utility of 5th machine: 20.05
```

```
cat('expected utility of 6th machine:',compute_utility(draws = extract_hierarchical$ypred6),'\n')
```

```
## expected utility of 6th machine: 4.55
```

From the results we can see that the rank (from worst to best) should be 1,6,3,5,2,4

3) The expected utility of a new machine

```
cat('The expected utilities of a new machine:',compute_utility(draws = extract_hierarchical$ypred7),'\n
```

```
## The expected utilities of a new machine: 22.75
```

4) Result

According to the above result, the company owner should buy a new machine because its expected utility is larger than 0.