Bayesian Data Analysis - Assignment 7

November 13, 2017

Model Type	**PSIS-LOO**	$\mathbf{p_{eff}}$	**k-value**$(\mathbf{k > 0.7})$)
Pooled	-131.0168	2.042215	0 (0.0%)
Separate	-132.8809	10.13113	4 (13.3%)
Hierarchical	-126.6512	5.536214	0(0.0%)

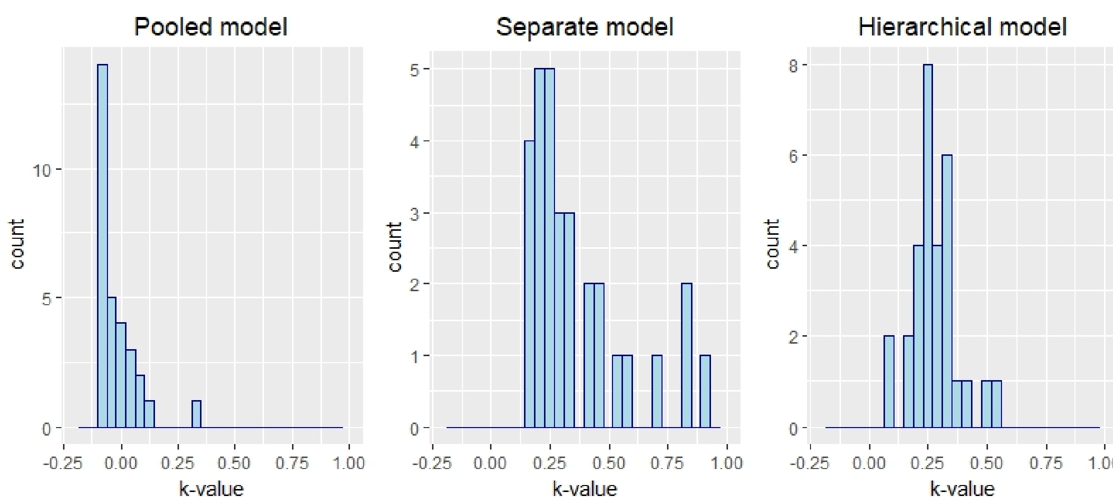

Figure 1: histogram of the k-values

From the above table and figure, we can get the following conclusion:

1. All the k-values of the pooled model and those of the hierarchical model are less than or equal to 0.7. So the PSIS-LOO estimate of the pooled model and hierarchical model can be considered to be reliable.

2. There are 13.3 of the k-values are more than 0.7 using separate model. So the PSIS-LOO estimate of the separate model can not be considered to be reliable.

3. Considering the differences of the PSIS-LOO value between the three model as well, hierachical model should be selected. Because it has the highest PSIS-LOO value, meaning that it fits the data best, and it is a reliable model as we discuss before.

Appendix

A Stan

A.1 Pooled model

```
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; // measurements
}
parameters {
  real mu;    // common mean
  real sigma; // common std
}
model {
  y ~ normal(mu, sigma);
}
generated quantities {
  real ypred;
  vector[N] log_lik;
  ypred = normal_rng(mu,sigma);
  for (i in 1:N)
    log_lik[i] = normal_lpdf(y[i] | mu, sigma);
}
```

A.2 Separate model

```
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; // measurements
}
parameters {
  vector[K] mu;             // group means
  vector<lower=0>[K] sigma; // group stds
}
model {
  y ~ normal(mu[x], sigma[x]);
}
generated quantities {
  real ypred;
  vector[N] log_lik;
  ypred = normal_rng(mu[6],sigma[6]);
  for (i in 1:N)
    log_lik[i] = normal_lpdf(y[i] | mu[x[i]], sigma[x[i]]);
}
```

A.3 Hierachical model

```
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; // measurements
}
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 ypred;
```

```
  real mu7_pred;
  vector[N] log_lik;
  ypred = normal_rng(mu[6],sigma);
  mu7_pred = normal_rng(mu0,sigma0);
  for (i in 1:N)
    log_lik[i] = normal_lpdf(y[i] | mu[x[i]], sigma);
}
```

B R code

```
library("rstan")
library("ggplot2")
library("loo")
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
# import and organize data
raw_data<-read.table("factory.txt")
factory_data<-list(N = ncol(raw_data)*nrow(raw_data),
    K = ncol(raw_data),
    x = rep(1:ncol(raw_data),nrow(raw_data)),
    y = c(t(raw_data[,1:ncol(raw_data)]))
)

# pooled model
pf_fit<-stan(file="pooled_factory.stan",data=factory_data)
pf_result<-extract(pf_fit,permuted=TRUE)
log_lik_pf <- extract_log_lik(pf_fit, parameter_name = 'log_lik')
loo_pf <- loo(log_lik_pf)
print(loo_pf)
k1<-loo_pf$pareto_k
k1_df<-data.frame(x=k1)
loo_pf$elpd_loo
loo_pf$p_loo

# separate model
sf_fit<-stan(file="separate_factory.stan",data=factory_data)
log_lik_sf <- extract_log_lik(sf_fit, parameter_name = 'log_lik')
sf_result<-extract(sf_fit,permuted=TRUE)
loo_sf <- loo(log_lik_sf)
```

```
print(loo_sf)
k2<-loo_sf$pareto_k
k2_df<-data.frame(x=k2)
loo_sf$elpd_loo
loo_sf$p_loo

# hierarchical model
hf_fit<-stan(file="hierarchical_factory.stan",data=factory_data)
hf_result<-extract(hf_fit,permuted=TRUE)
log_lik_hf <- extract_log_lik(hf_fit, parameter_name = 'log_lik')
loo_hf <- loo(log_lik_hf)
print(loo_hf)
k3<-loo_hf$pareto_k
k3_df<-data.frame(x=k3)
loo_hf$elpd_loo
loo_hf$p_loo

# plot k-values
p1<-ggplot(data=k1_df,aes(x=k1_df$x))+
  geom_histogram(color="darkblue",,fill="lightblue")+
  labs(title="Pooled model")+
  labs(x="k-value") +
  theme(plot.title = element_text(hjust = 0.5))+
  xlim(-0.2,1)

p2<-ggplot(data=k2_df,aes(x=k2_df$x))+
  geom_histogram(color="darkblue",,fill="lightblue")+
  labs(title="Separate model")+
  labs(x="k-value") +
  theme(plot.title = element_text(hjust = 0.5))+
  xlim(-0.2,1)

p3<-ggplot(data=k3_df,aes(x=k3_df$x))+
  geom_histogram(color="darkblue",,fill="lightblue")+
  labs(title="Hierarchical model")+
  labs(x="k-value") +
  theme(plot.title = element_text(hjust = 0.5))+
  xlim(-0.2,1)
require(gridExtra)
grid.arrange(p1,p2,p3,ncol=3)
```