STA365 HW2

Stephen Shannon

February 3, 2020

```
library(rstan)

library(ggplot2)
library(dplyr)
```

Question 1: Generating 3 data-sets of n=10, 100, 1000. Beta1 = 0.2 beta2 = 0.8

```
n1 = 10
beta1 = 0.2
beta2 = 0.8
sigma = 1
x1 = rnorm(n1)
y1 = rnorm(n1, beta1*x1 + beta2*x1, sigma)


n2 = 100
x2 = rnorm(n2)
y2 = rnorm(n2, beta1*x2 + beta2*x2, sigma)


n3 = 1000
x3 = rnorm(n3)
y3 = rnorm(n3, beta1*x3 + beta2*x3, sigma)
```

(1-1)

Question 2: The stan model and fitting p(Beta)

```
data {
  int<lower = 0> n; // number of observations
  int<lower = 0> p; // number of covariates
  matrix[n,p] x; // covariates are the rows!
  vector[n] y;
  vector[p] x_pred; // where to predict
}

parameters {
  vector[p] beta;
  real<lower = 0> sigma;
}

model {
```

```
  y ~ normal(x*beta, sigma);
  sigma ~ normal(0,1);
}

generated quantities {
  real y_pred;
  y_pred = normal_rng(x_pred'*beta, sigma);
}
```

(2-1)

```
p = 2;
x1_matrix <- matrix(c(x1, x1), nrow = n1, ncol = p)
x_pred = c(1,1)
stan_data1 <- list(n=n1, p=p, x=x1_matrix, y=y1, x_pred=x_pred)

fit <- sampling(homework2, data = stan_data1)

## Warning: There were 3339 transitions after warmup that exceeded the
maximum treedepth. Increase max_treedepth above 10. See
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

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

## Warning: The largest R-hat is 1.89, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating
posterior means and medians may be unreliable.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating
posterior variances and tail quantiles may be unreliable.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess

print(fit)

## Inference for Stan model: 7cf73e1bdbd00751f5406209e53e5ba2.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##              mean se_mean      sd     2.5%      25%     50%     75%   97.5%
n_eff
## beta[1]  473.28 1704.88 2647.77 -3091.94 -1428.63 -338.46 1871.76 5585.10
2
## beta[2] -472.23 1704.88 2647.77 -5584.11 -1870.98  339.53 1429.68 3092.73
2
## sigma      1.11    0.04    0.24     0.75     0.93    1.09    1.27    1.61
29
```

```
## y_pred     1.06    0.02    1.23    -1.41     0.29    1.05    1.84    3.59
4050
## lp__      -6.15    0.09    0.94    -8.63    -6.55   -5.91   -5.46   -5.15
109
##         Rhat
## beta[1] 2.91
## beta[2] 2.91
## sigma   1.11
## y_pred  1.00
## lp__    1.01
##
## Samples were drawn using NUTS(diag_e) at Tue Feb 04 03:55:22 2020.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

(2-2)

```
p = 2;
x2_matrix <- matrix(c(x2, x2), nrow = n2, ncol = p)
x_pred = c(1,1)
stan_data2 <- list(n=n2, p=p, x=x2_matrix, y=y2, x_pred=x_pred)

fit <- sampling(homework2, data = stan_data2)

## Warning: There were 3317 transitions after warmup that exceeded the
maximum treedepth. Increase max_treedepth above 10. See
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

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

## Warning: The largest R-hat is 1.63, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating
posterior means and medians may be unreliable.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating
posterior variances and tail quantiles may be unreliable.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess

print(fit)

## Inference for Stan model: 7cf73e1bdbd00751f5406209e53e5ba2.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##             mean se_mean    sd     2.5%     25%     50%     75%   97.5%
```

```
n_eff
## beta[1]  997.39  231.85 454.96  -141.72   783.47  1091.90 1313.23 1645.89
4
## beta[2] -996.47  231.85 454.96 -1644.96 -1312.42 -1090.91 -782.64  142.70
4
## sigma      1.02    0.02   0.07     0.89     0.97     1.01    1.06    1.19
9
## y_pred     0.92    0.02   1.03    -1.13     0.25     0.93    1.59    2.93
3927
## lp__     -49.00    0.23   1.11   -52.08   -49.39   -48.65  -48.22  -47.91
23
##          Rhat
## beta[1] 1.74
## beta[2] 1.74
## sigma   1.49
## y_pred  1.00
## lp__    1.23
##
## Samples were drawn using NUTS(diag_e) at Tue Feb 04 03:55:54 2020.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

(2-3)

```
p = 2;
x3_matrix <- matrix(c(x3, x3), nrow = n3, ncol = p)
x_pred = c(1,1)
stan_data3 <- list(n=n3, p=p, x=x3_matrix, y=y3, x_pred=x_pred)

fit <- sampling(homework2, data = stan_data2)

## Warning: There were 3270 transitions after warmup that exceeded the
maximum treedepth. Increase max_treedepth above 10. See
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

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

## Warning: The largest R-hat is 2.24, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating
posterior means and medians may be unreliable.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating
posterior variances and tail quantiles may be unreliable.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
print(fit)

## Inference for Stan model: 7cf73e1bdbd00751f5406209e53e5ba2.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##            mean se_mean      sd     2.5%     25%     50%    75%   97.5%
n_eff
## beta[1] -157.76  684.31 1052.06 -1485.51 -871.20 -610.73 328.81 2074.93
2
## beta[2]  158.68  684.31 1052.06 -2074.08 -327.87  611.72 872.14 1486.42
2
## sigma      0.99    0.02    0.07     0.86    0.94    1.00   1.04    1.13
9
## y_pred     0.91    0.02    1.02    -1.10    0.25    0.91   1.59    2.90
4007
## lp__     -48.94    0.08    0.93   -51.25  -49.37  -48.67 -48.24  -47.91
124
##         Rhat
## beta[1] 3.74
## beta[2] 3.74
## sigma   1.35
## y_pred  1.00
## lp__    1.04
##
## Samples were drawn using NUTS(diag_e) at Tue Feb 04 03:56:26 2020.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

(2-4)

Examining the fits, we find values of Rhat for Beta1 and 2 is consistently over 1.05, which indicates that the Markov chains did not converge to a single estimate. Because there is no unique MLE, the chains rapidly diverge. We also find that despite 3 different sample sizes of 10, 100 and 1000, the effective sample sizes for each fit was 3, 4, 2 respectively. As the chains have not converged, the estimate of the variance for a parameter increases greatly, in turn, samples become more greatly correlated. n_eff measures indepedently drawn samples, and a low number indicates many correlated drawn samples, a consequence of an incredibly large variance for the value of Beta.

As a result of divergent chains, the fits for all 3 data sets vary greatly from each other, despite being generated from the same model. Our estimates are poor and should not be used.

Question 3: Constructing a prior of N(0,1) for Beta.

We create a new stan model, assigning a prior to all Beta

```
data {
  int<lower = 0> n; // number of observations
  int<lower = 0> p; // number of covariates
  matrix[n,p] x; // covariates are the rows!
  vector[n] y;
  vector[p] x_pred; // where to predict
}

parameters {
  vector[p] beta;
  real<lower = 0> sigma;
}

model {
  y ~ normal(x*beta, sigma);
  sigma ~ normal(0,1);
  for (i in 1:p) {
  beta[i] ~ normal(0,1);
  }
}

generated quantities {
  real y_pred;
  y_pred = normal_rng(x_pred'*beta, sigma);

}
```

(3-1)

```
p = 2;
x1_matrix <- matrix(c(x1, x1), nrow = n1, ncol = p)
x_pred = c(1,1)
stan_data1 <- list(n=n1, p=p, x=x1_matrix, y=y1, x_pred=x_pred)

fit <- sampling(homework2gang, data = stan_data1)

print(fit)

## Inference for Stan model: c45f6c5061188b2f3239c3a5c27936b6.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd   2.5%   25%   50%   75% 97.5% n_eff Rhat
## beta[1]  0.49    0.02 0.73  -0.96 -0.01  0.49  0.99  1.89  1474    1
## beta[2]  0.48    0.02 0.74  -1.00 -0.01  0.48  0.97  1.95  1403    1
## sigma    1.08    0.01 0.25   0.71  0.90  1.04  1.21  1.66  2083    1
## y_pred   0.95    0.02 1.19  -1.44  0.20  0.96  1.68  3.31  3967    1
## lp__    -6.93    0.03 1.27 -10.30 -7.52 -6.60 -6.00 -5.49  1475    1
##
## Samples were drawn using NUTS(diag_e) at Tue Feb 04 03:57:05 2020.
```

```
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

(3-2)

```
p = 2;
x2_matrix <- matrix(c(x2, x2), nrow = n2, ncol = p)
x_pred = c(1,1)
stan_data2 <- list(n=n2, p=p, x=x2_matrix, y=y2, x_pred=x_pred)

fit <- sampling(homework2gang, data = stan_data2)

print(fit)

## Inference for Stan model: c45f6c5061188b2f3239c3a5c27936b6.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##             mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## beta[1]     0.47    0.02 0.72  -0.91  -0.03   0.46   0.95   1.89  1194    1
## beta[2]     0.45    0.02 0.72  -0.99  -0.04   0.45   0.95   1.82  1200    1
## sigma       0.99    0.00 0.07   0.86   0.94   0.98   1.03   1.13  1569    1
## y_pred      0.93    0.02 1.01  -1.09   0.26   0.93   1.62   2.90  4059    1
## lp__      -49.57    0.03 1.18 -52.62 -50.14 -49.27 -48.71 -48.23  1231    1
##
## Samples were drawn using NUTS(diag_e) at Tue Feb 04 03:57:06 2020.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

(3-3)

```
p = 2;
x3_matrix <- matrix(c(x3, x3), nrow = n3, ncol = p)
x_pred = c(1,1)
stan_data3 <- list(n=n3, p=p, x=x3_matrix, y=y3, x_pred=x_pred)

fit <- sampling(homework2gang, data = stan_data2)

print(fit)

## Inference for Stan model: c45f6c5061188b2f3239c3a5c27936b6.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##             mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## beta[1]     0.48    0.02 0.71  -0.95   0.02   0.49   0.94   1.89  1392    1
## beta[2]     0.44    0.02 0.71  -0.96  -0.02   0.42   0.90   1.86  1381    1
## sigma       0.99    0.00 0.07   0.86   0.94   0.98   1.03   1.13  1931    1
## y_pred      0.91    0.02 0.98  -0.95   0.24   0.93   1.59   2.76  3848    1
```

```
## lp__    -49.58    0.03 1.18 -52.57 -50.12 -49.28 -48.71 -48.21  1225    1
##
## Samples were drawn using NUTS(diag_e) at Tue Feb 04 03:57:08 2020.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

(3-4)

Examining all model fits, the markov chains converged for each data set as Rhat is equal to 1 for all estimates of beta. As the sample size of the data-set increased, the estimated mean for beta1 and beta2 approaches to 0.5. Furthermore, the n_eff for each beta estimator is consistently above 1100, indicating that there was a low correlation between the samples. This is a result of setting a prior on Beta 1 and 2 of being normal distributions with mu 0 and sigma 1. The prior enforces the model to search within a small, fixed range of values, thus arbitrarily increasing n_eff and ensuring convergence of the chains. Regardless, these estimates do not match the true value of 0.2 and 0.8 for Beta1 and Beta2. Despite having converging chains and low variability for each estimate, the prior only made it possible for chains to converge on a degenerate model. However, the model did accurately estimate the value of sigma, which was 1. This is because sigma was not a degenerate term in the model.

Appendix

```
n1 = 10
beta1 = 0.2
beta2 = 0.8
sigma = 1
x1 = rnorm(n1)
y1 = rnorm(n1, beta1*x1 + beta2*x1, sigma)


n2 = 100
x2 = rnorm(n2)
y2 = rnorm(n2, beta1*x2 + beta2*x2, sigma)


n3 = 1000
x3 = rnorm(n3)
y3 = rnorm(n3, beta1*x3 + beta2*x3, sigma)
```

(1-1)

```
data {
  int<lower = 0> n; // number of observations
  int<lower = 0> p; // number of covariates
  matrix[n,p] x; // covariates are the rows!
  vector[n] y;
  vector[p] x_pred; // where to predict
}
```

```
parameters {
  vector[p] beta;
  real<lower = 0> sigma;
}

model {
  y ~ normal(x*beta, sigma);
  sigma ~ normal(0,1);
}

generated quantities {
  real y_pred;
  y_pred = normal_rng(x_pred'*beta, sigma);
}
```

(2-1)

```
p = 2;
x1_matrix <- matrix(c(x1, x1), nrow = n1, ncol = p)
x_pred = c(1,1)
stan_data1 <- list(n=n1, p=p, x=x1_matrix, y=y1, x_pred=x_pred)

fit <- sampling(homework2, data = stan_data1)
print(fit)
```

(2-2)

```
p = 2;
x2_matrix <- matrix(c(x2, x2), nrow = n2, ncol = p)
x_pred = c(1,1)
stan_data2 <- list(n=n2, p=p, x=x2_matrix, y=y2, x_pred=x_pred)

fit <- sampling(homework2, data = stan_data2)
print(fit)
```

(2-3)

```
p = 2;
x3_matrix <- matrix(c(x3, x3), nrow = n3, ncol = p)
x_pred = c(1,1)
stan_data3 <- list(n=n3, p=p, x=x3_matrix, y=y3, x_pred=x_pred)

fit <- sampling(homework2, data = stan_data2)
print(fit)
```

(2-4)

```
data {
  int<lower = 0> n; // number of observations
  int<lower = 0> p; // number of covariates
  matrix[n,p] x; // covariates are the rows!
```

```
  vector[n] y;
  vector[p] x_pred; // where to predict
}

parameters {
  vector[p] beta;
  real<lower = 0> sigma;
}

model {
  y ~ normal(x*beta, sigma);
  sigma ~ normal(0,1);
  for (i in 1:p) {
  beta[i] ~ normal(0,1);
  }
}

generated quantities {
  real y_pred;
  y_pred = normal_rng(x_pred'*beta, sigma);

}
```

(3-1)

```
p = 2;
x1_matrix <- matrix(c(x1, x1), nrow = n1, ncol = p)
x_pred = c(1,1)
stan_data1 <- list(n=n1, p=p, x=x1_matrix, y=y1, x_pred=x_pred)

fit <- sampling(homework2gang, data = stan_data1)
print(fit)
```

(3-2)

```
p = 2;
x2_matrix <- matrix(c(x2, x2), nrow = n2, ncol = p)
x_pred = c(1,1)
stan_data2 <- list(n=n2, p=p, x=x2_matrix, y=y2, x_pred=x_pred)

fit <- sampling(homework2gang, data = stan_data2)
print(fit)
```

(3-3)

```
p = 2;
x3_matrix <- matrix(c(x3, x3), nrow = n3, ncol = p)
x_pred = c(1,1)
stan_data3 <- list(n=n3, p=p, x=x3_matrix, y=y3, x_pred=x_pred)
```

```
fit <- sampling(homework2gang, data = stan_data2)
print(fit)
```

(3-4)