732A73: Bayesian Learning

Computer Lab 4

Oriol Garrobé, Dávid Hrabovszki

19 May 2020

Question 1. Time series models in Stan

(a)

We are asked to develop a simulator from the AR(1)-process,

$$x_t = \mu + \phi(x_{t-1} - \mu) + \epsilon_t, \quad \epsilon_t \overset{iid}{\sim} N(0, \sigma^2),$$

for the given values: $\mu = 10$, $\sigma^2 = 2$ and $\phi \in (-1, 1)$. The code of the simulator can be seen in the Appendix.

From this point we are asked to generate 200 values of x_t. In Figure 1 two simulations can be seen for $\phi = -0.95, \phi = 0.95$. The parameter ϕ determines how the current value of x_t depends on the previous value. If $\phi > 0$ then they are positively correlated, if $\phi < 0$ they are negatively correlated and if $\phi = 0$ they are not correlated at all. With $\phi = -0.95$, the process oscillates intensely, as the new value has a negative relationship with the old value. When $\phi = 0.95$, the process doesn't change as rapidly, because the new values will be very close to the old ones. The amount of change from one point to the next also largely depends on σ^2.

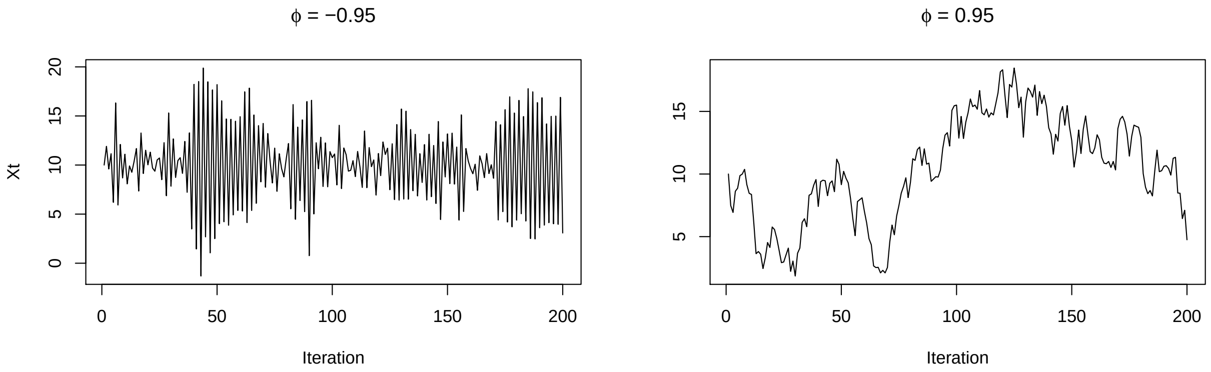

Figure 1: AR(1)-process simulations for two different values of ϕ

(b)

Now we use the simulator from a) to simulate two AR(1)-processes with $\phi = 0.3$ and $\phi = 0.95$. Using this two simulations as synthetic data, we estimate μ, ϕ and σ^2 using MCMC. The priors used for the parameters are:

- $\mu \sim N(10, 10)$. This is a non-informative prior as it has a large standard deviation.
- $\phi \sim N(0, \sqrt{10})$. $\phi \in (-1, 1)$, therefore this is a weakly informative prior.
- $\sigma^2 \sim Scale - inv - \chi^2(1, 1.4)$. Degrees of freedom are very small, therefore the prior is non-informative.

The implemented Stan-code can be found in the appendix.

i.

For each of the simulated AR(1)-processes we get the following values for the three inferred parameters.

Parameter	Mean	95% CI	Effective Samples
mu	7.95	$(-43.72, 43.74)$	760
phi	0.97	$(0.92, 1.00)$	435
sigma2	1.67	$(1.35, 2.07)$	594

Table 1: *Posterior values for* y_t*.*

Parameter	Mean	95% CI	Effective Samples
mu	10.25	$(9.88, 10.61)$	3848
phi	0.39	$(0.26, 0.53)$	3461
sigma2	2.23	$(1.84, 2.71)$	3377

Table 2: *Posterior values for* x_t*.*

We are able to estimate the true values for the synthetic data x_t but not for y_t. The standard deviation of μ is very large in the case of y_t, and not very accurate anyway. The number of effective samples is low, so the MCMCs are inefficient, since we obtained 4000 post-warmup samples for each parameter. This is because ϕ is large for process y_t, therefore it is close to being unstable, so the sampling cannot be perfect.

ii.

In Figure 2 it can be seen that there is bad mixing in the chains that use y_t as data, especially for μ. The chains do not oscillate too much and they don't cover the same areas. The reason for this is that ϕ is limited to being smaller than 1 to keep it stable.

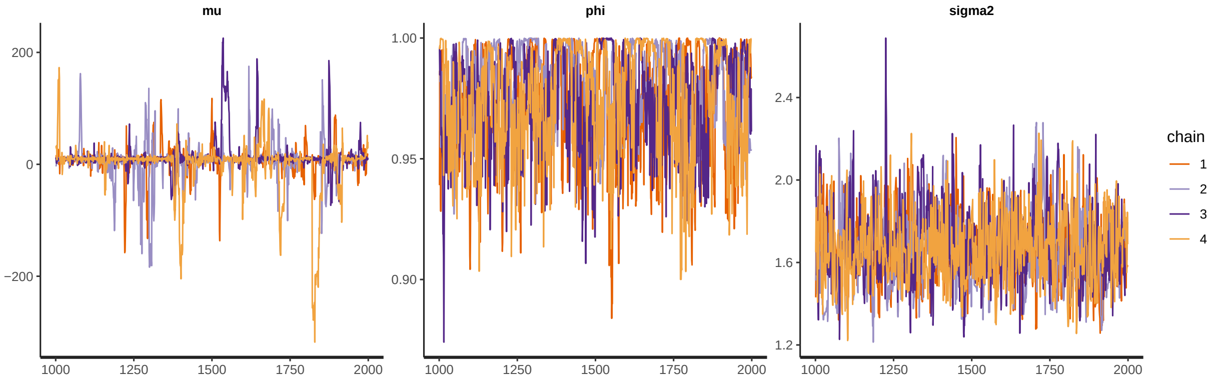

Figure 2: Traceplots of parameters for y_t

The joint posterior of μ and ϕ can be observed in Figure 3. Here it seems obvious that the large standard deviation of μ comes from that ϕ is limited and that the sample doesn't work well with nearly unstable processes.

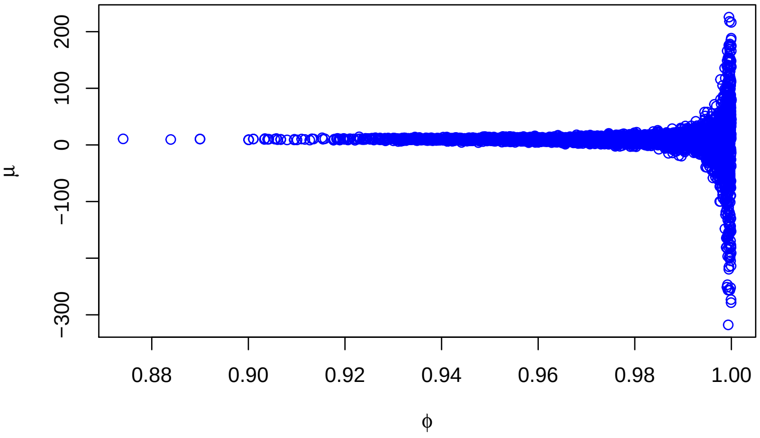

Figure 3: Joint posterior of μ and ϕ for y_t

On the other hand Figure 4 confirms that there is good convergence in x_t, because the chains oscillate a lot, and they cover the same areas, therefore they arrived at the same conclusion.

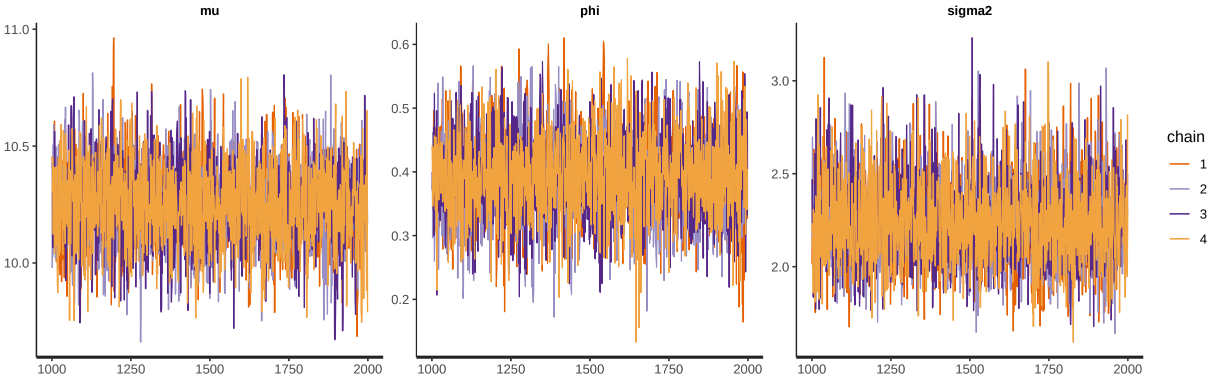

Figure 4: Traceplots of parameters for x_t.

This time the scatter plot of the μ and ϕ parameters looks much better in Figure 5, like they follow normal distributions.

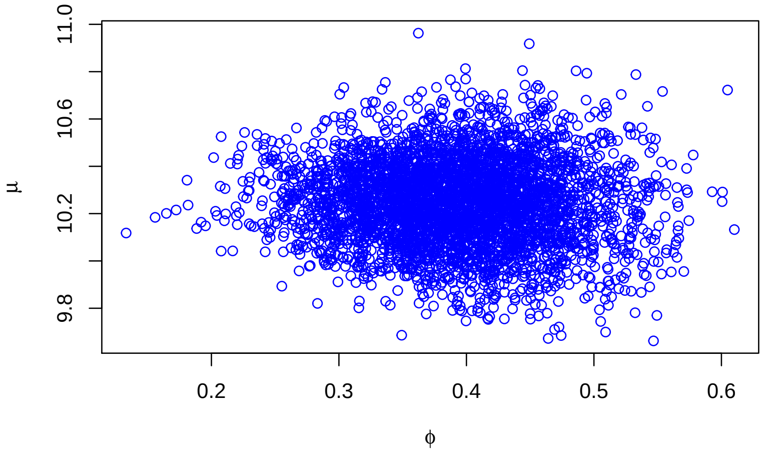

Figure 5: Joint posterior of μ and ϕ for x_t

(c)

In this question we use the dataset `campy.dat` that contains the number of cases of campylobacter infections in Quebec in four weeks intervals from January 1990 to October 2000. We assume that the number of infections c_t at each point follows an independent Poisson distribution when conditioned on a latent AR(1)-process x_t, such as

$$c_t|x_t \sim Poisson(exp(x_t)),$$

We implement and estimate the model using suitable priors. As we know nothing about the parameters, the priors chosen are the following:

- $\mu \sim N(10, 10)$. This is a non-informative prior as it has a large standard deviation.
- $\phi \sim N(0, \sqrt{10})$. $\phi \in (-1, 1)$, therefore this is a weakly informative prior.
- $\sigma^2 \sim Scale - inv - \chi^2(1, 2)$. Degrees of freedom is very small, therefore the prior is non-informative.

The R-Stan code can be found in the Appendix.

In Figure 6 we can observe the data, and the posterior mean and 95% Credible Intervals for the latent intensity $\theta_t = exp(x_t)$ parameter over time. The posterior mean fits the data quite well, and the intervals don't seem very wide.

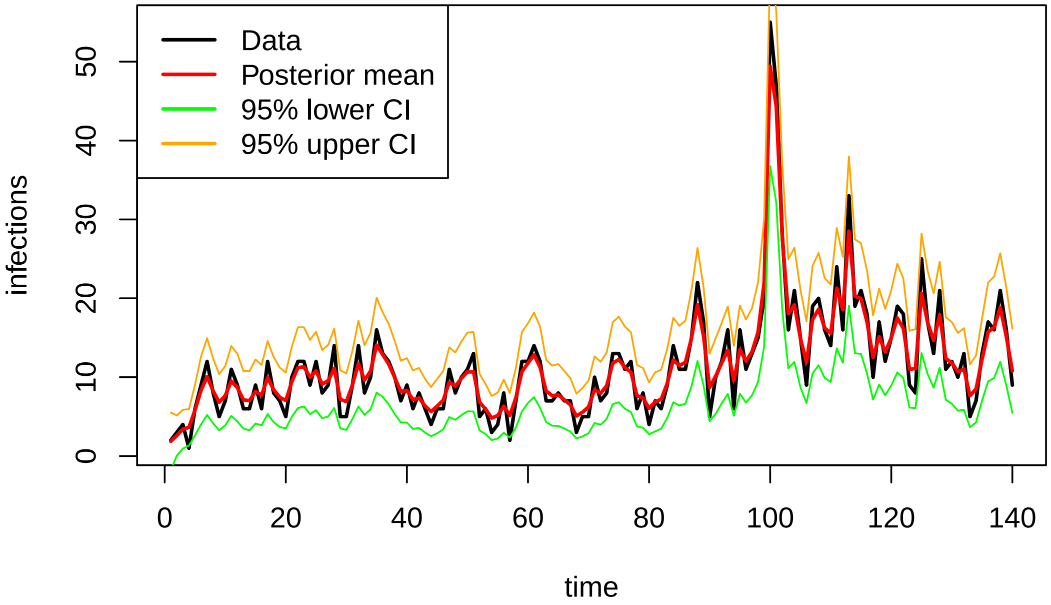

Figure 6: Posterior latent intensity over time with non-informative prior.

(d)

Finally, having the prior belief that the true underlying intensity θ_t varies more smoothly than the data suggests, we change the prior for σ^2 so it is more informative about that the AR(1)-process increments ϵ_t are small. We re-estimate the model with the new prior. The priors for the parameters are:

- $\mu \sim N(10, 10)$. This is a non-informative prior as it has a large standard deviation.
- $\phi \sim N(0, \sqrt{10})$. $\phi \in (-1, 1)$, therefore this is a weakly informative prior.
- $\sigma^2 \sim Scale - inv - \chi^2(100, 0.2)$. Due to the large value of degrees of freedom, this prior is very informative. It provides the model with the information that ϵ is around 0.2 (small).

The R-stan code can be found in the Appendix.

In Figure 7 it can be seen the posterior mean and the sample mean along with the latent intensity $\theta_t = exp(x_t)$ 95% Credible Intervals parameters over time.

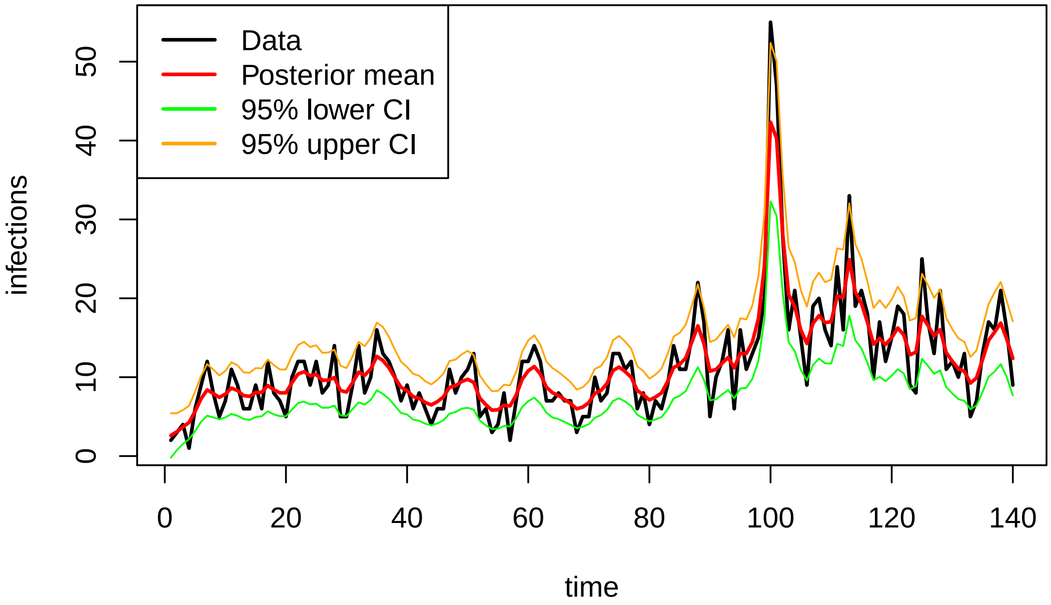

Figure 7: Posterior latent intensity over time with informative prior.

Changing the prior for σ^2 changes the posterior in the following way. As the increments ϵ_t are smaller, the posterior mean is smoother than the data, and the shape of the Credible Intervals for the latent intensity $\theta_t = exp(x_t)$ is very similar to the posterior mean, only this time they are closer to it.

Appendix

```
knitr::opts_chunk$set(echo=FALSE, eval=TRUE)
# R version
RNGversion('3.5.1')

# libraries
library(rstan)
library(car)

# Seed
set.seed(1234567890)
################# QUESTION 1 ###################

#### (a)

# Given values
mu = 10
sigma2  = 2
t = 200

# AR simulator
sim_AR_1 = function(start, mu, sigma2, phi, t){
  x = numeric(t)
  x[1] = start
  for (i in 2:t) {
    eps = rnorm(1, mean = 0, sd = sqrt(sigma2))
    x[i] = mu + phi*(x[i-1] - mu) + eps
  }
  return(x)
}
par(mfrow = c(1,2))
# Plot of one simulation for phi=1
set.seed(1234567890)
x1 = sim_AR_1(mu, mu, sigma2, -0.95, t)
x2 = sim_AR_1(mu, mu, sigma2, 0.95, t)
plot(x1, type = 'l', main = expression(paste(phi, ' = -0.95')), xlab="Iteration", ylab="Xt")
plot(x2, type = 'l', main = expression(paste(phi, ' = 0.95')), xlab="Iteration", ylab="")

### (b)

# Synthetic data
set.seed(1234567890)
x = sim_AR_1(mu, mu, sigma2, 0.3, t)
y = sim_AR_1(mu, mu, sigma2, 0.95, t)

# Stan model
StanModel = '
data {
  int<lower=0> N;
  vector[N] y;
}
parameters {
  real mu;
  real<lower=-1, upper=1> phi;
  real<lower=0> sigma2;
}
model {
  mu ~ normal(10,100); //non-informative, because of large sd
  phi ~ normal(0,10); //weakly informative, because we know it has to be between -1 and 1
  sigma2 ~ scaled_inv_chi_square(1,1.4); //non-informative, because of small df
  for (n in 2:N)
    y[n] ~ normal(mu+phi*(y[n-1]-mu),sqrt(sigma2));
}'

# i.

# Simulation for y_t
N = length(y)
data_y = list(N=N, y=y)
burnin = 1000
niter = 2000

# Seed
set.seed(1234567890)
fit_y = stan(model_code = StanModel, data = data_y, warmup = burnin, iter = niter, chains = 4)

# Simulation for x_t
data_x = list(N=N, y=x)
fit_x = stan(model_code = StanModel, data = data_x, warmup = burnin, iter = niter, chains = 4)
# Print the fitted model
#print(fit_y)
#print(fit_x)

# ii.
traceplot(fit_y)

scatterplot(mu ~ phi, data = fit_y, grid = FALSE, boxplots = F, regLine=F, smooth=F,
            xlab = expression(phi), ylab = expression(mu))

traceplot(fit_x)
scatterplot(mu ~ phi, data = fit_x, grid = FALSE, boxplots = F, regLine=F, smooth=F,
            xlab = expression(phi), ylab = expression(mu))

### (c)

# read data
data_campy = read.table('campy.dat', header = T)
plot(x=c(1:140), y=data_campy$c, type = 'l')

# Poisson Stan Model
StanModel_poisson = '
data {
  int<lower=0> N;
  int<lower=0> c[N];
}
```

```
parameters {
  real mu;
  real<lower=-1, upper=1> phi;
  real<lower=0> sigma2;
  real x[N];
}
model {
  mu ~ normal(0,100); // non-informative prior, we know nothing about mu
  phi ~ normal(0,10); //weakly informative, because we know it has to be between -1 and 1
  sigma2 ~ scaled_inv_chi_square(1,2); //non-informative, because of small df
  for (n in 2:N){
    x[n] ~ normal(mu+phi*(x[n-1]-mu),sqrt(sigma2));
    c[n] ~ poisson(exp(x[n]));
  }
}'

N = length(data_campy$c)
data = list(N=N, c=data_campy$c)
burnin = 1000
niter = 2000
# Seed
set.seed(1234567890)
fit_poisson = stan(model_code = StanModel_poisson, data = data, warmup = burnin,
                   iter = niter, chains = 4)

# Extract posterior samples
postDraws <- extract(fit_poisson)
#plot
intensity_posterior = data.frame(exp(postDraws[["x"]]))
intensity_posterior_means = colMeans(intensity_posterior)
intensity_posterior_sd = apply(intensity_posterior, 2, sd)

#first row = lower bound of 95% interval
#second row = upper bound of 95% interval
intensity_95_intervals = rbind(intensity_posterior_means - 1.96*intensity_posterior_sd,
                               intensity_posterior_means + 1.96*intensity_posterior_sd)

plot(x=c(1:140), y=data_campy$c, type = 'l', lwd = 2, xlab = 'time', ylab = 'infections')
lines(x=c(1:140), y=intensity_posterior_means, col = 'red', lwd = 2)
lines(x=c(1:140), y=intensity_95_intervals[1,], col = 'green')
lines(x=c(1:140), y=intensity_95_intervals[2,], col = 'orange')
legend(x='topleft', legend=c('Data', 'Posterior mean', '95% lower CI', '95% upper CI'),
       col = c('black', 'red', 'green', 'orange'), lwd = 2)

### (d)

# Poisson Stan Model with informative prior
StanModel_poisson2 = '
data {
  int<lower=0> N;
  int<lower=0> c[N];
}
parameters {
  real mu;
  real<lower=-1, upper=1> phi;
  real<lower=0> sigma2;
  real x[N];
}
model {
  mu ~ normal(0,100); // non-informative prior, we know nothing about mu
  phi ~ normal(0,10); //weakly informative, because we know it has to be between -1 and 1
  sigma2 ~ scaled_inv_chi_square(100,0.2); // informative, because of big df
  for (n in 2:N){
    x[n] ~ normal(mu+phi*(x[n-1]-mu),sqrt(sigma2));
    c[n] ~ poisson(exp(x[n]));
  }
}'

N = length(data_campy$c)
data = list(N=N, c=data_campy$c)
burnin = 1000
niter = 2000
# Seed
set.seed(1234567890)
fit_poisson2 = stan(model_code = StanModel_poisson2, data = data, warmup = burnin,
                    iter = niter, chains = 4)

# Extract posterior samples
postDraws2 <- extract(fit_poisson2)

#plot
intensity_posterior2 = data.frame(exp(postDraws2[["x"]]))
intensity_posterior_means2 = colMeans(intensity_posterior2)
intensity_posterior_sd2 = apply(intensity_posterior2, 2, sd)

#first row = lower bound of 95% interval
#second row = upper bound of 95% interval
intensity_95_intervals2 = rbind(intensity_posterior_means2 - 1.96*intensity_posterior_sd2,
                               intensity_posterior_means2 + 1.96*intensity_posterior_sd2)

plot(x=c(1:140), y=data_campy$c, type = 'l', lwd = 2, xlab = 'time', ylab = 'infections')
lines(x=c(1:140), y=intensity_posterior_means2, col = 'red', lwd = 2)
lines(x=c(1:140), y=intensity_95_intervals2[1,], col = 'green')
lines(x=c(1:140), y=intensity_95_intervals2[2,], col = 'orange')
legend(x='topleft', legend=c('Data', 'Posterior mean', '95% lower CI', '95% upper CI'),
       col = c('black', 'red', 'green', 'orange'), lwd = 2)
```