Computational aspects of some simple statistical models on the Bayesian approach using STAN: basic concepts

https://github.com/clobos/Seminario_STAN_UFBA

Cristian Villegas, ESALQ/USP

UFBA (26/08/2020)

1. Intoduction to Stan
2. Introduction to Bayes Theorem
3. Beta prior + Binomial Likelihood: two cases
4. Bayesian Logistic Regression

Section 1

Intoduction to Stan

What is Stan?

Stan is a state-of-the-art platform for statistical modeling and high-performance statistical computation. Thousands of users rely on Stan for statistical modeling, data analysis, and prediction in the social, biological, and physical sciences, engineering, and business.

Brief intoduction to Stan

Users specify log density functions in Stan's probabilistic programming language and get:

- full Bayesian statistical inference with MCMC sampling (NUTS, HMC)
- approximate Bayesian inference with variational inference (ADVI)
- penalized maximum likelihood estimation with optimization (L-BFGS)

Brief intoduction to Stan

Stan's math library provides differentiable probability functions & linear algebra (C++ autodiff). Additional R packages provide expression-based linear modeling, posterior visualization, and leave-one-out cross-validation.

Section 2

Introduction to Bayes Theorem

Introduction to Bayes Theorem

$$f(\theta|\text{Data}) = \frac{f(\text{Data}|\theta)f(\theta)}{f(\text{Data})} \tag{1}$$

where

- $f(\theta|\text{Data})$ Posterior distribution
- $f(\text{Data}|\theta)$ Likelihood function
- $f(\theta)$ **Prior distribution**
- $f(\text{Data})$ Normalized constant
- Problems? $f(\text{Data})$ not easy to calculate?
- Solutions? MCMC methods (Metropolis-Hasting, Gibbs Sampling, Hamiltonian Monte Carlo)

Section 3

Beta prior + Binomial Likelihood: two cases

Posterior distribution based on : Beta prior + Binomial Likelihood

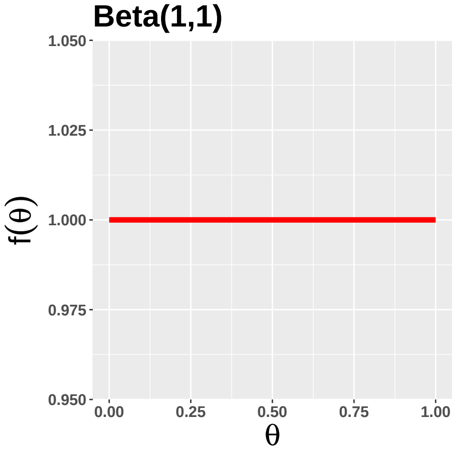
$$f(\theta|\text{Data}) = \frac{f(\text{Data}|\theta)f(\theta)}{f(\text{Data})} \tag{2}$$

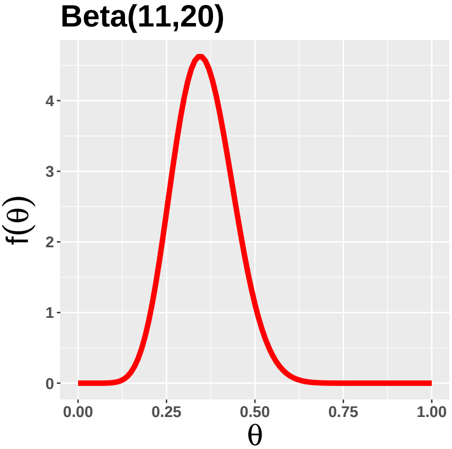
where

- $f(\text{Data}|\theta)$ Binomial(N,θ) distribution (Likelihood)
- $f(\theta)$ Beta(a,b) distribution (Prior)

Posterior? Beta distribution (Conjugate families)

Beta distribution





Beta(1,1) + Binomial(10, θ)

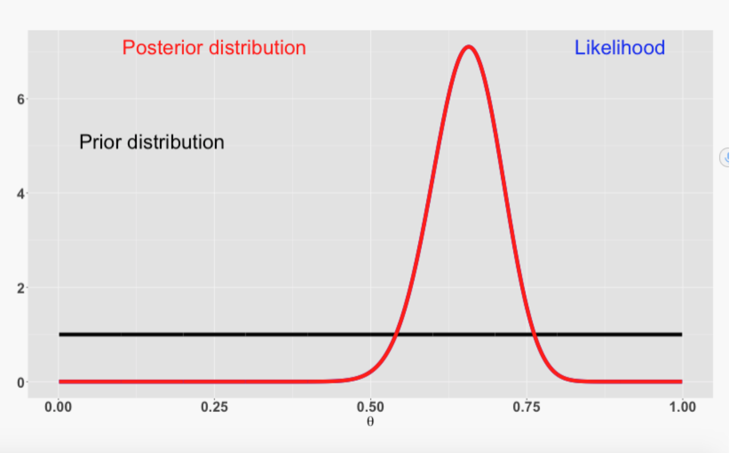


Figure 1: Beta(1,1) non-informative prior

Beta(11,20) + Binomial(N=10, θ)

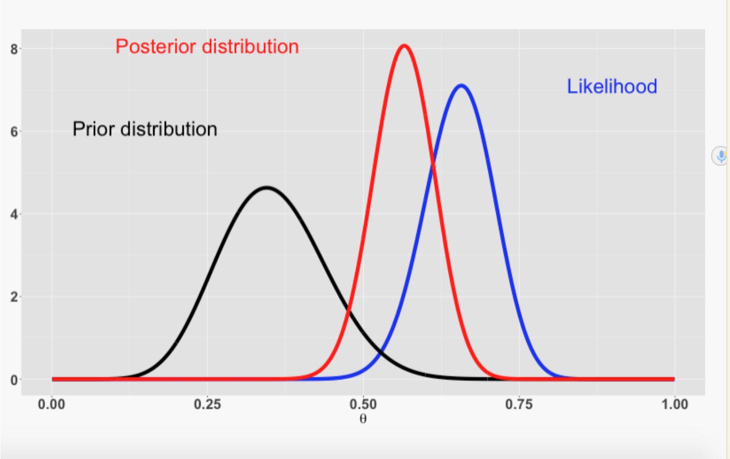


Figure 2: Beta(11,20) informative prior

Beta(1,1)+Binomial(10,θ): Stan Code

```
beta_binomial1<-
'data {
  int<lower=0> N;
  int<lower=0> y;
}
parameters {
  real<lower=0,upper=1> theta;
}
model {
  theta ~ beta(1,1);
  y ~ binomial(N,theta);
}
'
```

Beta(1,1)+Binomial(10,θ): Stan Code

```
fit_beta_binomial1 <- stan(model_code = beta_binomial1,
               data = list(N = 10,y = 7),
               chain = 3,
               iter = 11000,
               warmup = 1000,
               thin = 10,
               refresh=0)
#save.image("fit_beta_binomial1_beta_1_1.Rdata")
#fit_beta_binomial
```

Summary from the posterior distibution

```
#parameters<- "theta"

CI_theta <- summary(fit_beta_binomial1,
probs = c(0.025, 0.975))$summary
print(round(CI_theta,3))
```

```
        mean se_mean    sd    2.5%  97.5%    n_eff Rhat
theta  0.668   0.002 0.130   0.402  0.891 3084.455    1
lp__  -8.155   0.014 0.709 -10.153 -7.639 2730.368    1
```

MCMC diagnostics using the bayesplot package

```
traceplot(fit_beta_binomial1, pars = parameters,
          inc_warmup = TRUE)
```

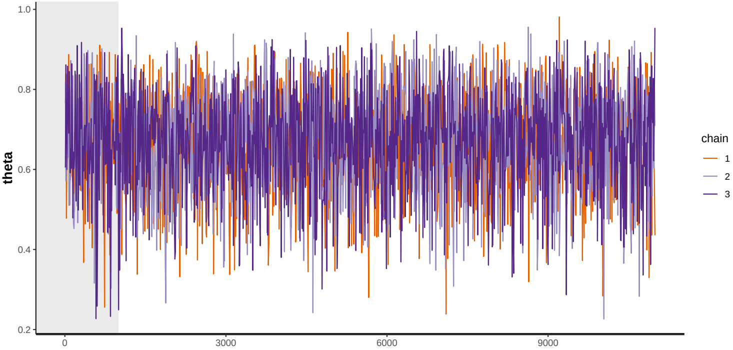


Figure 3: Traceplots for the Beta Binomial example

MCMC diagnostics using the bayesplot package

```
traceplot(fit_beta_binomial1, pars = parameters,
          inc_warmup = FALSE)
```

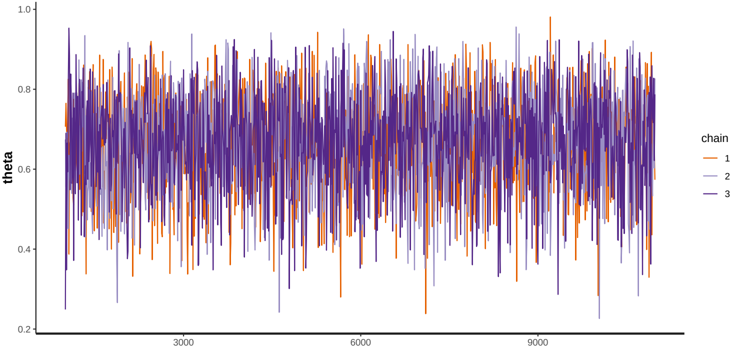


Figure 4: Traceplots for the Beta Binomial example

MCMC diagnostics using the bayesplot package

```
mcmc_combo(mcmc_chain1,pars = parameters)
```

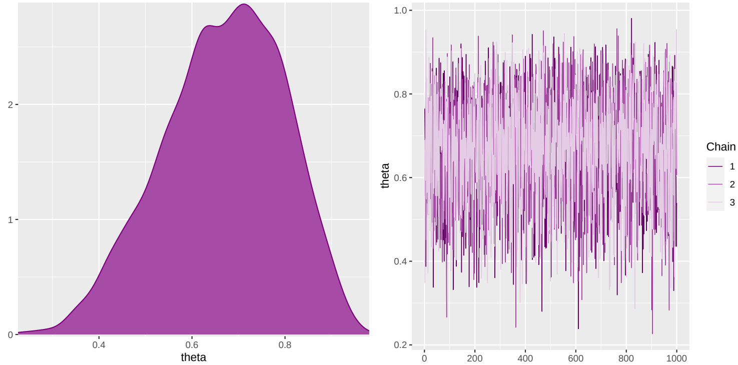


Figure 5: Posterior distributions and traceplots for the beta binomial example

Beta(11,20)+Binomial(10,θ): Stan Code

```
beta_binomial2<-
'data {
  int<lower=0> N;
  int<lower=0> y;
}
parameters {
  real<lower=0,upper=1> theta;
}
model {
  theta ~ beta(11,20);
  y ~ binomial(N,theta);
}
'
```

Beta(11,20)+Binomial(N=10,θ): Stan Code

```
fit_beta_binomial2 <- stan(model_code = beta_binomial2,
               data = list(N = 10,y = 7),
               chain = 3,
               iter = 11000,
               warmup = 1000,
               thin = 10,
               refresh=0)
#save.image("fit_beta_binomial2_beta_11_20.Rdata")
#fit_beta_binomial
```

Summary from the posterior distibution

```
#parameters<- "theta"
CI_theta <- summary(fit_beta_binomial2,
probs = c(0.025, 0.975))$summary
print(round(CI_theta,3))
```

```
         mean se_mean    sd    2.5%   97.5%    n_eff  Rhat
theta   0.439   0.001 0.077   0.290   0.593 2635.699 0.999
lp__  -28.621   0.014 0.724 -30.662 -28.114 2842.964 1.000
```

MCMC diagnostics using the bayesplot package

```
traceplot(fit_beta_binomial2, pars = parameters,
inc_warmup = TRUE)
```

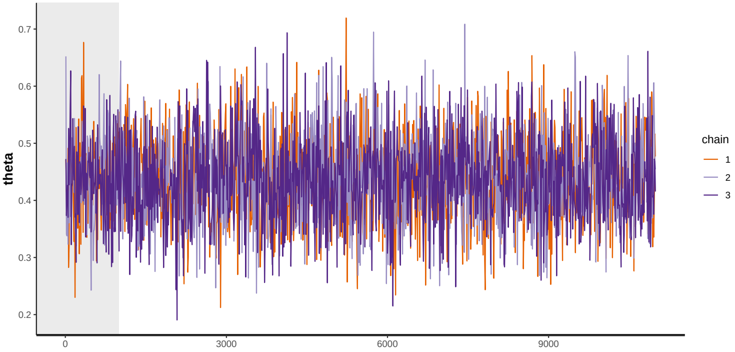


Figure 6: Traceplots for the Beta Binomial example

MCMC diagnostics using the bayesplot package

```
traceplot(fit_beta_binomial2, pars = parameters,
inc_warmup = FALSE)
```

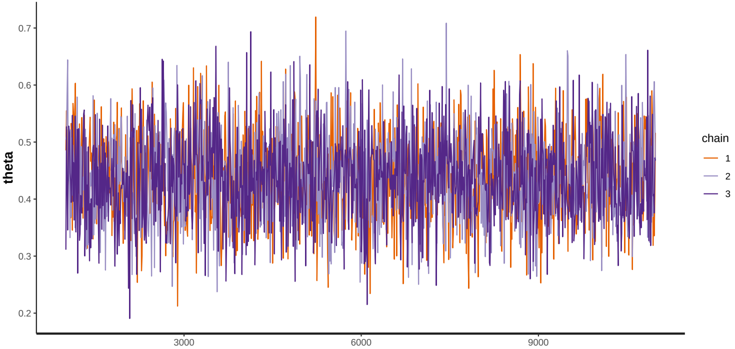


Figure 7: Traceplots for the Beta Binomial example

MCMC diagnostics using the bayesplot package

```
mcmc_combo(mcmc_chain2,pars = parameters,n_warmup=0)
```

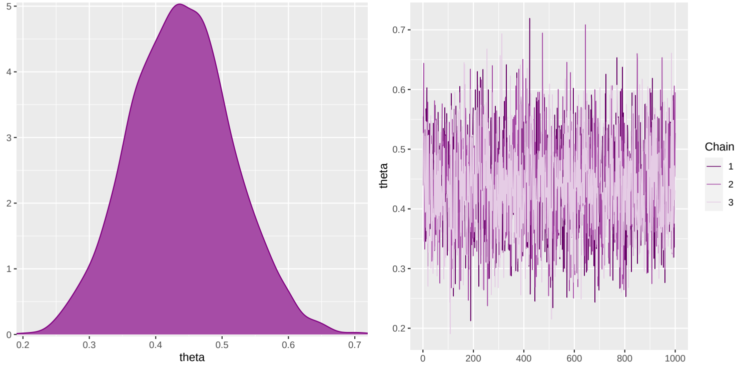


Figure 8: Posterior distributions and traceplots for the beta binomial example

Section 4

Bayesian Logistic Regression

Motivation

These are the number of adult flour beetles which died following a 5-hour exposure to gaseous carbon disulphide. Binomial response with logit link function. Here, we do not specify the prior distribution for each parameter.

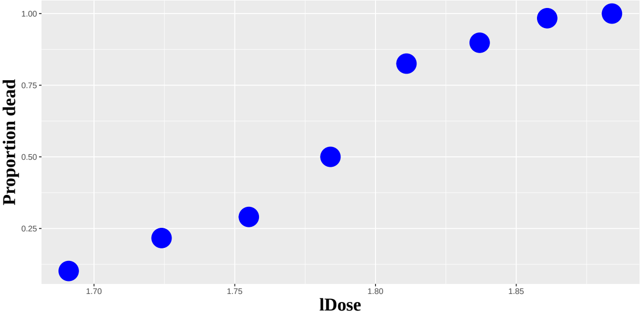


Figure 9: Scatterplot of proportions versus log(Dose)

Stan code

```
logistic_example<- 'data {
int<lower=0> N;
vector[N] x;
int<lower=0> y[N];
int<lower=0> n[N];
}
```

Stan code

```
parameters {
real beta1;
real beta2;
}
```

Stan code

```
transformed parameters  {
real exp_eta[N];
real<lower=0, upper=1> prob[N];
for (i in 1:N) {
exp_eta[i] = exp(beta1 + beta2*x[i]);
prob[i]= exp_eta[i]/(exp_eta[i] + 1);
}
}
```

Stan code

```
model {
beta1 ~ cauchy(0,10);
beta2 ~ cauchy(0,2.5);
y ~ binomial_logit(n, beta1 + beta2 * x);
}
'
#save.image("logistic_fit1.Rdata")
```

Stan code

```
logistic_fit <- stan(model_code = logistic_example,
                     data = list(N = dim(beetleDat)[1],
                                 n = beetleDat$n,
                                 x = beetleDat$lDose,
                                 y = beetleDat$x),
                     chain = 3,
                     iter = 11000,
                     warmup = 1000,
                     thin = 10,
                     refresh=0)
```

Summary from the posterior distribution

```
parameters<- c(paste('beta',1:2, sep=""))

CI_theta <- summary(logistic_fit,
                    pars = parameters,
                    probs = c(0.025, 0.975))$summary
print(round(CI_theta,3))
```

```
         mean se_mean    sd    2.5%   97.5%    n_eff Rhat
beta1 -61.222   0.111 5.292 -72.193 -51.326 2263.911    1
beta2  34.558   0.063 2.974  28.995  40.734 2252.852    1
```

MCMC diagnostics using the bayesplot package

```
traceplot(logistic_fit, pars = parameters,
          inc_warmup = TRUE)
```

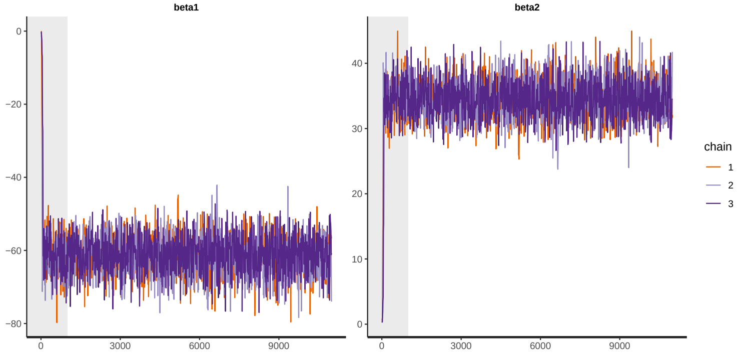


Figure 10: Traceplots for the Logistic regression model

MCMC diagnostics using the bayesplot package

```
mcmc_combo(mcmc_chain,pars = parameters,n_warmup=0)
```

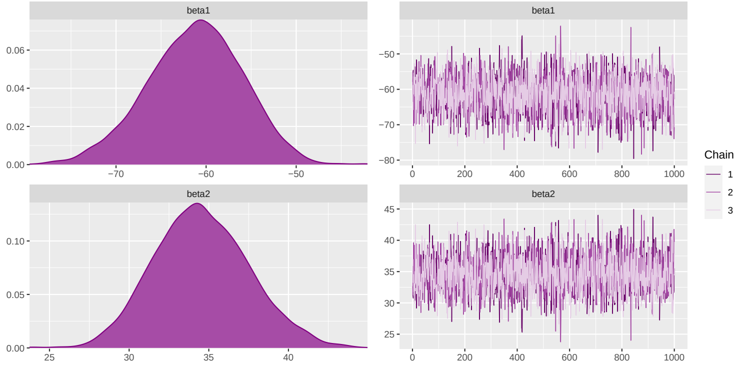


Figure 11: Posterior distributions and traceplots for the Logistic regression model

MCMC diagnostics using the bayesplot package

```
mcmc_pairs(mcmc_chain,pars = parameters)
```

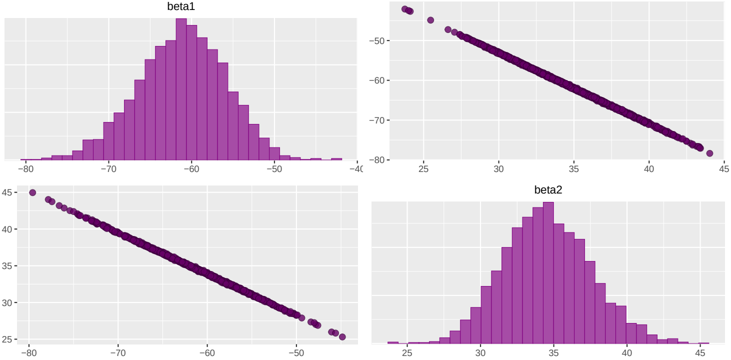


Figure 12: Scatterplots of MCMC draws for the Logistic Regression model

```
#https://www.youtube.com/watch?v=uSjsJg8fcwY
```

Fitted curve based on Bayesian Inference

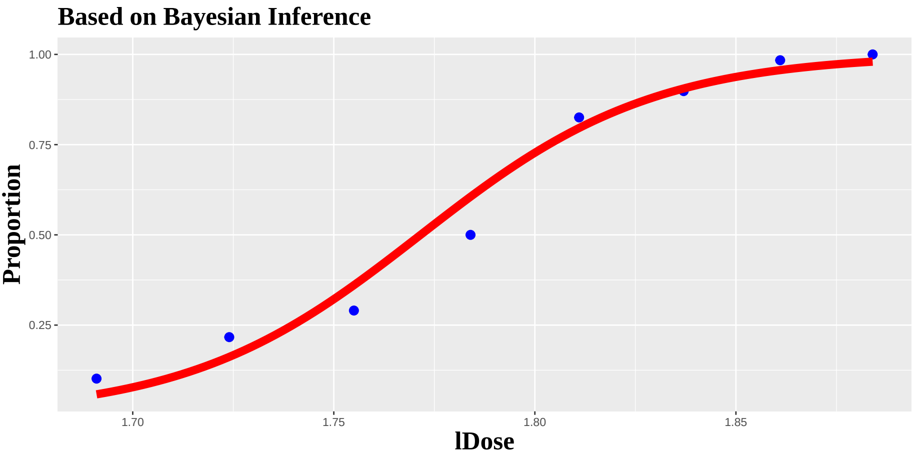


Figure 13: Fitted curves based on bayesian Inference for the Logistic Regression model

More R packages based on Stan

- Bayesian Applied Regression Modeling via Stan: `rstanarm` r package.
- Interactive Visual and Numerical Diagnostics and Posterior Analysis for Bayesian Models `shinystan` r package.

Do I have more time for the shinystan r package?

```r
rm(list=ls())
load("logistic_fit1.Rdata")
launch_shinystan(logistic_fit)
```

References

- Baptiste Auguie (2017). gridExtra: Miscellaneous Functions for “Grid” Graphics. R package version 2.3. https://CRAN.R-project.org/package=gridExtra
- Jonah Gabry and Tristan Mahr (2020). bayesplot: Plotting for Bayesian Models. R package version 1.7.2. https://CRAN.R-project.org/package=bayesplot
- Jonah Gabry (2018). shinystan: Interactive Visual and Numerical Diagnostics and Posterior Analysis for Bayesian Models. R package version 2.5.0. https://CRAN.R-project.org/package=shinystan
- Stan Development Team (2018). RStan: the R interface to Stan. R package version 2.18.2. http://mc-stan.org/.

References

- https://mc-stan.org/docs/2_24/stan-users-guide/index.html
- https://mc-stan.org/docs/2_24/reference-manual/index.html
- https://mc-stan.org/docs/2_24/functions-reference/index.html
- https://cran.r-project.org/web/views/Bayesian.html