

Bayesian Logistic Regression

Eric C. Brown

September 23, 2013

1 Model

$$\begin{aligned}\mathbf{y} &\sim \text{Bern}(\boldsymbol{\theta}) \\ \boldsymbol{\theta} &= \frac{1}{1 + \exp(-\boldsymbol{\mu})} \\ \boldsymbol{\mu} &= \beta_0 + \mathbf{X}\boldsymbol{\beta} \\ \beta_0 &\sim \mathcal{N}(0, 100) \\ \beta_j &\sim \mathcal{N}(0, 100), \quad j = 1, \dots, J\end{aligned}$$

2 Code

```
bayesianLogisticRegression <- function(y, X = NULL, Xnew = NULL, iter = 5000,
  thin = 1, chains = 1, burnin = floor(iter/2), betameanpriors = if (is.null(X)) {
    0
  } else {
    rep(0, ncol(X) + 1)
  }, betasdpriors = if (is.null(X)) {
    100
  } else {
    rep(100, ncol(X) + 1)
  }, inits = NULL, seed = 1) {
  if (is.null(X)) {
    J <- 0
  } else {
    J <- ncol(X)
    if (!(nrow(X) == length(y))) {
      message("ERROR: X and y must have the same number of rows.")
      message(paste(c("ERROR: nrow(X) =", nrow(X), ", length(y) =", length(y)),
        collapse = " "))
      return(NA)
    }
  }
  if (!(is.null(Xnew))) {
    if (!(ncol(Xnew) == J)) {
      message("ERROR: X and Xnew must have the same number of columns.")
      message(paste(c("ERROR: ncol(X) =", ncol(X), ", ncol(Xnew) =", ncol(Xnew)),
        collapse = " "))
      return()
    }
  }
}
```

```

}
if (is.null(X)) {
  form <- formula("y ~ 1")
  mm <- model.matrix(form, data.frame(y = y))
} else {
  form <- formula(paste("y ~ ", paste(names(X), collapse = "+"), sep = ""))
  mm <- model.matrix(form, X)
}
if (is.null(inits)) {
  fit <- glm.fit(mm, y, family = binomial(link = "logit"))
  inits <- coef(fit)
}
if (is.null(Xnew)) {
  model <- "\n data {\n   int<lower=0> N;\n   int<lower=1> J;\n   in
data <- list(N = nrow(mm), J = ncol(mm), y = y, X = mm, betameanpriors = betameanpriors,
  betasdpriors = betasdpriors)
  set.seed(seed)
  print(seed)
  fit <- stan(model_code = model, data = data, chains = chains, iter = iter,
    warmup = burnin, thin = thin, init = function(x) {
      list(beta = inits)
    }, nondiag_mass = TRUE, seed = seed)
} else {
  mmnew <- model.matrix(form, Xnew)
  model <- "\n data {\n   int<lower=1> N;\n   int<lower=1> J;\n
data <- list(N = nrow(mm), J = ncol(mm), y = y, X = mm, betameanpriors = betameanpriors,
  betasdpriors = betasdpriors, Nnew = nrow(mmnew), Xnew = mmnew)
  set.seed(seed)
  print(seed)
  fit <- stan(model_code = model, data = data, chains = chains, iter = iter,
    warmup = burnin, thin = thin, init = function(x) {
      list(beta = inits)
    }, nondiag_mass = TRUE, seed = seed)
}
return(fit)
}

```

3 Example

Predict whether the car will have an automatic (0) or manual (1) transmission.

```

y <- mtcars[, 'am']
X <- as.data.frame(mtcars[, c('mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'vs', 'gear', 'carb')])
Xnew <- X
fit <- bayesianLogisticRegression(y=y, X=X,
  # Xnew=Xnew,
  iter=10000, chains=4, inits=rep(0, ncol(X)+1))

## [1] 1
##
## TRANSLATING MODEL 'model' FROM Stan CODE TO C++ CODE NOW.
## COMPILING THE C++ CODE FOR MODEL 'model' NOW.

```

```

## /Users/brown/.R/Makevars:123: warning: overriding commands for target `.c.o'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:123: warning: ignoring old commands
## /Users/brown/.R/Makevars:125: warning: overriding commands for target `.c.d'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:125: warning: ignoring old commands
## /Users/brown/.R/Makevars:128: warning: overriding commands for target `.cc.o'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:128: warning: ignoring old commands
## /Users/brown/.R/Makevars:130: warning: overriding commands for target `.cpp.o'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:130: warning: ignoring old commands
## /Users/brown/.R/Makevars:132: warning: overriding commands for target `.cc.d'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:132: warning: ignoring old commands
## /Users/brown/.R/Makevars:135: warning: overriding commands for target `.cpp.d'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:135: warning: ignoring old commands
## /Users/brown/.R/Makevars:138: warning: overriding commands for target `.m.o'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:138: warning: ignoring old commands
## /Users/brown/.R/Makevars:140: warning: overriding commands for target `.m.d'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:140: warning: ignoring old commands
## /Users/brown/.R/Makevars:143: warning: overriding commands for target `.mm.o'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:143: warning: ignoring old commands
## /Users/brown/.R/Makevars:145: warning: overriding commands for target `.M.o'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:145: warning: ignoring old commands
## /Users/brown/.R/Makevars:147: warning: overriding commands for target `.f.o'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:147: warning: ignoring old commands
## /Users/brown/.R/Makevars:149: warning: overriding commands for target `.f95.o'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:149: warning: ignoring old commands
## /Users/brown/.R/Makevars:151: warning: overriding commands for target `.f90.o'
## /opt/local/Library/Frameworks/R.framework/Resources/etc/Makeconf:151: warning: ignoring old commands
## SAMPLING FOR MODEL 'model' NOW (CHAIN 1).
##
Iteration:    1 / 10000 [ 0%] (Adapting)
Iteration: 1000 / 10000 [ 10%] (Adapting)
Iteration: 2000 / 10000 [ 20%] (Adapting)
Iteration: 3000 / 10000 [ 30%] (Adapting)
Iteration: 4000 / 10000 [ 40%] (Adapting)
Iteration: 5000 / 10000 [ 50%] (Adapting)
Iteration: 6000 / 10000 [ 60%] (Sampling)
Iteration: 7000 / 10000 [ 70%] (Sampling)
Iteration: 8000 / 10000 [ 80%] (Sampling)
Iteration: 9000 / 10000 [ 90%] (Sampling)
Iteration: 10000 / 10000 [100%] (Sampling)
##
## SAMPLING FOR MODEL 'model' NOW (CHAIN 2).
##
Iteration:    1 / 10000 [ 0%] (Adapting)
Iteration: 1000 / 10000 [ 10%] (Adapting)
Iteration: 2000 / 10000 [ 20%] (Adapting)
Iteration: 3000 / 10000 [ 30%] (Adapting)
Iteration: 4000 / 10000 [ 40%] (Adapting)
Iteration: 5000 / 10000 [ 50%] (Adapting)
Iteration: 6000 / 10000 [ 60%] (Sampling)
Iteration: 7000 / 10000 [ 70%] (Sampling)
Iteration: 8000 / 10000 [ 80%] (Sampling)
Iteration: 9000 / 10000 [ 90%] (Sampling)
Iteration: 10000 / 10000 [100%] (Sampling)

```

```
##
## SAMPLING FOR MODEL 'model' NOW (CHAIN 3).
##
Iteration:    1 / 10000 [  0%] (Adapting)
Iteration: 1000 / 10000 [ 10%] (Adapting)
Iteration: 2000 / 10000 [ 20%] (Adapting)
Iteration: 3000 / 10000 [ 30%] (Adapting)
Iteration: 4000 / 10000 [ 40%] (Adapting)
Iteration: 5000 / 10000 [ 50%] (Adapting)
Iteration: 6000 / 10000 [ 60%] (Sampling)
Iteration: 7000 / 10000 [ 70%] (Sampling)
Iteration: 8000 / 10000 [ 80%] (Sampling)
Iteration: 9000 / 10000 [ 90%] (Sampling)
Iteration: 10000 / 10000 [100%] (Sampling)
##
## SAMPLING FOR MODEL 'model' NOW (CHAIN 4).
##
Iteration:    1 / 10000 [  0%] (Adapting)
Iteration: 1000 / 10000 [ 10%] (Adapting)
Iteration: 2000 / 10000 [ 20%] (Adapting)
Iteration: 3000 / 10000 [ 30%] (Adapting)
Iteration: 4000 / 10000 [ 40%] (Adapting)
Iteration: 5000 / 10000 [ 50%] (Adapting)
Iteration: 6000 / 10000 [ 60%] (Sampling)
Iteration: 7000 / 10000 [ 70%] (Sampling)
Iteration: 8000 / 10000 [ 80%] (Sampling)
Iteration: 9000 / 10000 [ 90%] (Sampling)
Iteration: 10000 / 10000 [100%] (Sampling)
```