CS 109B, Spring 2017, Homework 3: Introduction to Bayesian Methods

Jan 2016

Problem 2: Bayesian Logisitic Regression

You are provided with data sets `dataset_2_train.txt` and `dataset_2_test.txt` containing details of contraceptive usage by 1934 Bangladeshi women. There are 4 attributes for each woman, along with a label indicating if she uses contraceptives. The attributes include:

- `district`: identifying code for the district the woman lives in
- `urban`: type of region of residence
- `living.children`: number of living children
- `age_mean`: age of women (in years, centred around mean)

The women are grouped into 60 districts. The task is to build a classification model that can predict if a given woman uses contraceptives.

Use simple visualizations to check if there are differences in contraceptive usage among women across the districts. If so, would we benefit by fitting a separate classification model for each district? To answer this question, you may fit the following classification models to the training set and compare their accuracy on the test set:

- A separate logistic regression model for each district
- A single logistic regression model using the entire training set (ignoring the district information)

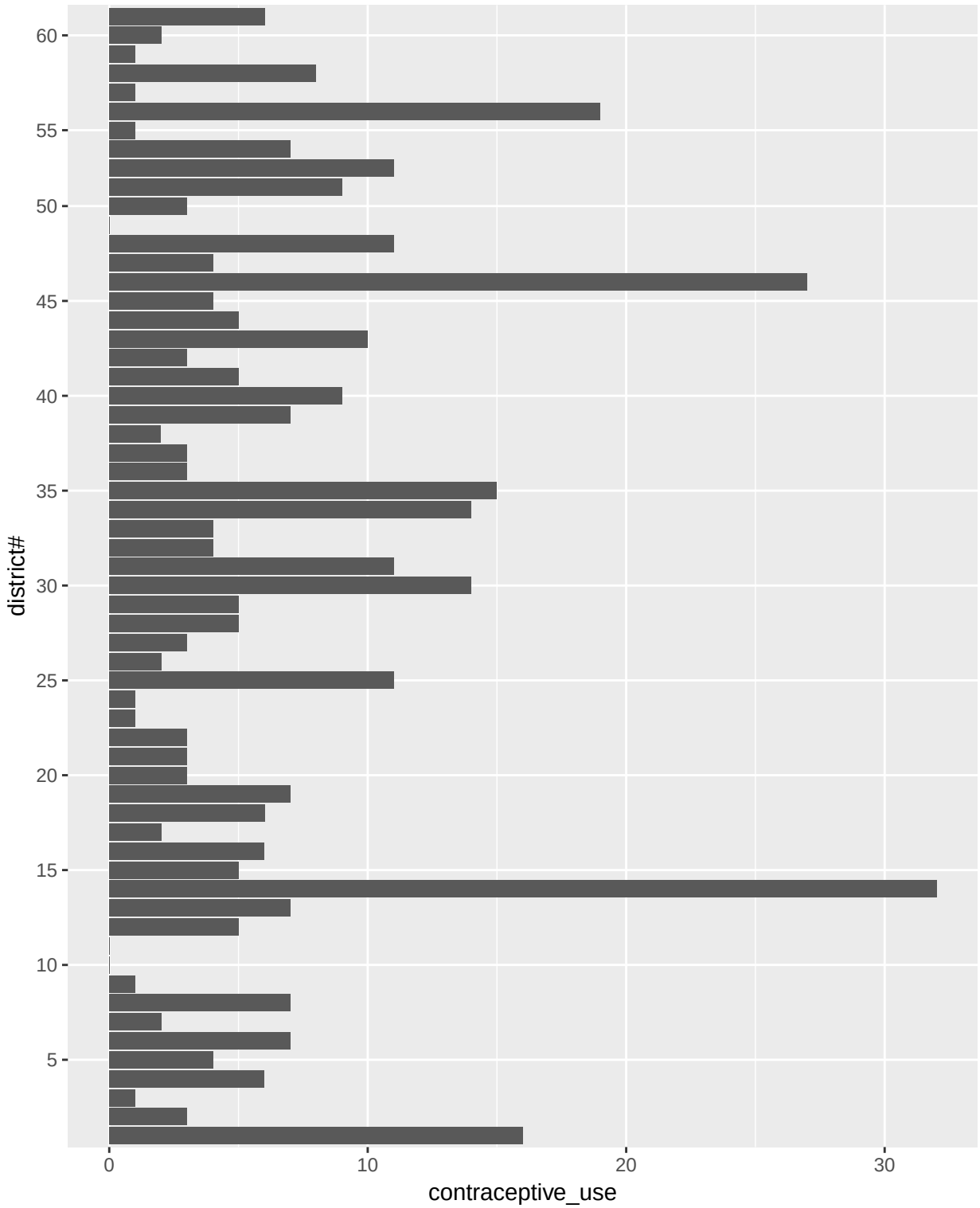
Fit a Bayesian hierarchical logistic regression model to the training set using the district as a grouping variable, and evaluate its accuracy on the test set. Does the Bayesian hierarchical model yield better accuracies than the two models fitted previously? Give an explanation for what you observe. Also, explain why the hierarchical logistic regression is a compromise between fitting separate logistic regressions for each district and a single logistic regression ignoring district membership.

Hint: You may use the `MCMChlogit` function in the `MCMCpack` package for fitting a Bayesian hierarchical logistic regression model.

```
suppressMessages(library(MLmetrics))
suppressMessages(library(e1071))
suppressMessages(library(MCMCpack))
suppressMessages(library(MGLM))
library(ggplot2)
```

```
train = read.csv("./datasets_preprocessed_code/dataset_2_train.txt" , header = TRUE)
test = read.csv('./datasets_preprocessed_code/dataset_2_test.txt', header = TRUE)
```

```
ggplot(data=train,aes(x=factor(district),y=contraceptive_use)) +
  geom_bar(stat="identity") +
  coord_flip() +
  xlab("district#") + scale_x_discrete(breaks=seq(5,60,5))
```




```
# a separate logistic regression model for each district
options(warn=-1)

district_vector = c(unique(train$district))

df = data.frame(y_true_val= integer(0), preds= integer(0))
```

```
for (i in 1:length(district_vector)) {
      model = suppressWarnings(glm(contraceptive_use ~ urban + age_mean + living.children ,
          family = binomial(link="logit"),data=train[train$district==district_vector[i],]))

      y_true = test[test$district==district_vector[i],'contraceptive_use']
      preds = suppressWarnings(predict(model,
                                       newdata=test[test$district==district_vector[i],],
                                       type="response")) > 0.5
      df = rbind(df,do.call(cbind,list(y_true,preds)))
}
Accuracy(df$V1,df$V2) #0.6111686
```

```
## [1] 0.6111686
```

```
# a single logistic regression model using the entire training set (ignoring the district information)

glm.model = glm(contraceptive_use ~ urban + age_mean + living.children ,family = binomial(link="logit")

pred = predict(glm.model, newdata = test, type="response")
pred = ifelse(pred>0.5,1,0)

Accuracy(pred,test$contraceptive_use)
```

```
## [1] 0.647363
```

```
table(pred,test$contraceptive_use)
```

```
##
## pred   0   1
##    0 528 284
##    1  57  98
```

```
library(rstanarm)
```

```
## Loading required package: Rcpp
## rstanarm (Version 2.14.1, packaged: 2017-01-16 18:47:11 UTC)
## - Do not expect the default priors to remain the same in future rstanarm versions.
## Thus, R scripts should specify priors explicitly, even if they are just the defaults.
## - For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores())
```

```
# solution for the optional Bayesian hierarchical logistic regression model using RStan

m <- stan_glmer(contraceptive_use ~ urban + living.children + age_mean + (1 + urban + living.children +
                data = train,
                family = "binomial")
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
##
## Chain 1, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1, Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
## Chain 1, Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1, Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1, Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1, Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1, Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1, Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1, Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1, Iteration: 2000 / 2000 [100%]  (Sampling)
##  Elapsed Time: 92.2041 seconds (Warm-up)
##                54.8471 seconds (Sampling)
##                147.051 seconds (Total)
##
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
##
## Chain 2, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2, Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2, Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2, Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2, Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2, Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2, Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2, Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2, Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2, Iteration: 2000 / 2000 [100%]  (Sampling)
##  Elapsed Time: 110.845 seconds (Warm-up)
##                56.9259 seconds (Sampling)
##                167.771 seconds (Total)
##
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
##
## Chain 3, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3, Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3, Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3, Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3, Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3, Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3, Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3, Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3, Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3, Iteration: 2000 / 2000 [100%]  (Sampling)
##  Elapsed Time: 151.457 seconds (Warm-up)
##                164.349 seconds (Sampling)
##                315.807 seconds (Total)
##
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
##
## Chain 4, Iteration:    1 / 2000 [  0%]  (Warmup)
```

```
## Chain 4, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4, Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4, Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4, Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4, Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4, Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4, Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4, Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4, Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4, Iteration: 2000 / 2000 [100%]  (Sampling)
##  Elapsed Time: 113.821 seconds (Warm-up)
##                55.206 seconds (Sampling)
##                169.027 seconds (Total)
```

```
pd <- apply(posterior_predict(m, newdata = test), 2, mean)
pred.bin = ifelse(pred>0.5,1,0)

table(test$contraceptive_use,
        as.integer(pd > .5))
```

```
##
##       0   1
##   0 519  66
##   1 262 120
```

```
Accuracy(test$contraceptive_use, pred.bin)
```

```
## [1] 0.647363
```

Fitting a single logistic regression on the entire data set is too general; there are clear differences in contraceptive use across districts, and grouping all of the districts together throws away this important distinction, resulting in low performance. On the other hand, fitting a separate logistic regression model for each district and predicting on a case-by-case basis accounts for the variations in contraceptive use explained by variations in district, but suffers from potential underfitting - because each district has a small number of people, the models are imprecise. The hierarchical model is a compromise between the two options because the β_g account for inter-group variation, while the random effects distribution prevents underfitting by weighing small groups toward the population mean.