Midterm Exam 1

Advanced Topics in Data Science II
Harvard University, Spring 2017

Tim Hagmann

February 23, 2017

Contents

The set of questions below address the task of predicting the merit of a restaurant on Yelp. Each restaurant is described by a set of business attributes, and is accompanied by a set of text reviews from customers. For the purpose of the problems below, the average rating (originally on a scale from 0 to 5) was converted into a binary variable depending on whether the average was above 3.5, in which case it is considered "good" (labeled 1), or below 3.5 in which case it is considered "bad" (labeled 0). The overall goal is to predict these binary ratings from the information provided.

The data are split into a training and test set and are in the files `dataset_1_train.txt` and `dataset_1_test.txt` respectively. The first column contains the rating for the restaurant (0 or 1), columns 2-21 contain the business attributes, and columns 22-121 contain text features extracted from the customer reviews for the restaurant. The details about the business attributes are provided in the file `dataset_1_description.txt`.

We use the bag-of-words encoding to generate the text features, where the set of reviews for a restaurant are represented by a vector of word counts. More specifically, we construct a dictionary of 100 frequent words in the customer reviews, and include 100 text features for each restaurant: the *i*-th feature contains the number of times the dictionary word *i* occurs in customer reviews for the restaurant. For example, a text feature 'fantastic' with value 18 for a restaurant indicates that the word 'fantastic' was used a total of 18 times in customer reviews for that restaurant.

Problem 1 [20 points]

Does the location of a restaurant relate to its rating? Construct a compelling visualization to address this question, and write a brief (under 300 words) summary that clearly explains your analysis and conclusions to someone without a data science background.

Data preparation

In the following code chunk all the necessary setup for the modelling environment is done.

Initialize

```
## Options
options(scipen = 10)                          # Disable scientific notation
update_package <- FALSE                       # Use old status of packages

## Init files (always execute, eta: 10s)
source("scripts/01_init.R")                   # Helper functions to load packages
source("scripts/02_packages.R")               # Load all necessary packages
source("scripts/03_functions.R")              # Load project specific functions
```

Load the data

```
## Read data
df_train <- data.frame(read_csv("data/dataset_1_train.txt"))
df_test <- data.frame(read_csv("data/dataset_1_test.txt"))
```

Preprocess data

```
# Extract word count
df_train_words <- df_train[22:length(df_train)]
df_test_words <- df_test[22:length(df_train)]

# Extract words
words <- names(df_train_words)

# Create factor variables
df_train$rating <- factor(df_train$rating, labels=c("bad", "good"))
df_test$rating <- factor(df_test$rating, labels=c("bad", "good"))
```

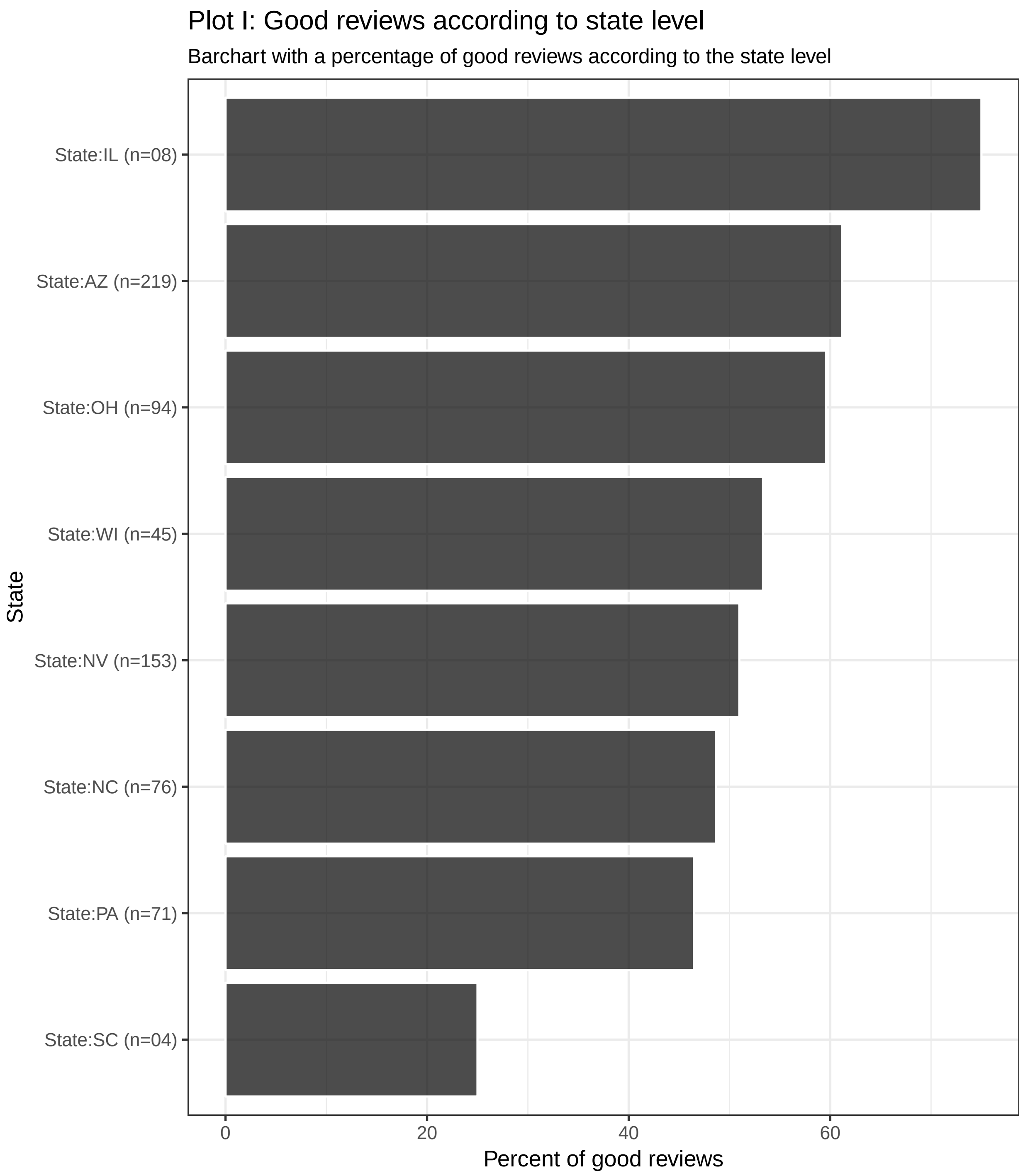
Aggregate

```
# Aggregate data on a districe level
df_state1 <- aggregate(rating ~ state, data=df_train, FUN=length)
df_state2 <- aggregate((as.numeric(df_train$rating) - 1) ~ state,
                       data=df_train, FUN=sum)

# Left join data
df_state <- merge(df_state1, df_state2, by="state", all=TRUE)
names(df_state) <- c("state", "n_review", "sum_good_reviews")

# Calculations
df_state$perc <- round((df_state$sum_good_reviews / df_state$n_review) * 100, 2)

# Change order of the states
df_state$state <- factor(df_state$state,
                          levels=df_state$state[
                            order(df_state$perc, decreasing=FALSE)])
```

```r
df_state <- df_state[order(df_state$perc, decreasing=FALSE), ]

# Add number reviews on a state level (n=xxx) for the visualization below
levels(df_state$state) <- paste0("State:", levels(df_state$state),
                                  " (n=", sprintf("%.2d", df_state$n_review), ")")
rm(df_state1, df_state2)
```

Visualize data

```r
ggplot(df_state, aes(x=state, y=perc)) +
    labs(title="Plot I: Good reviews according to state level",
       subtitle="Barchart with a percentage of good reviews according to the state level") +
  geom_bar(stat="identity", colour="white", fill="black", alpha=0.7) +
  theme_bw() +
  ylab("Percent of good reviews") +
  xlab("State") +
  coord_flip()
```

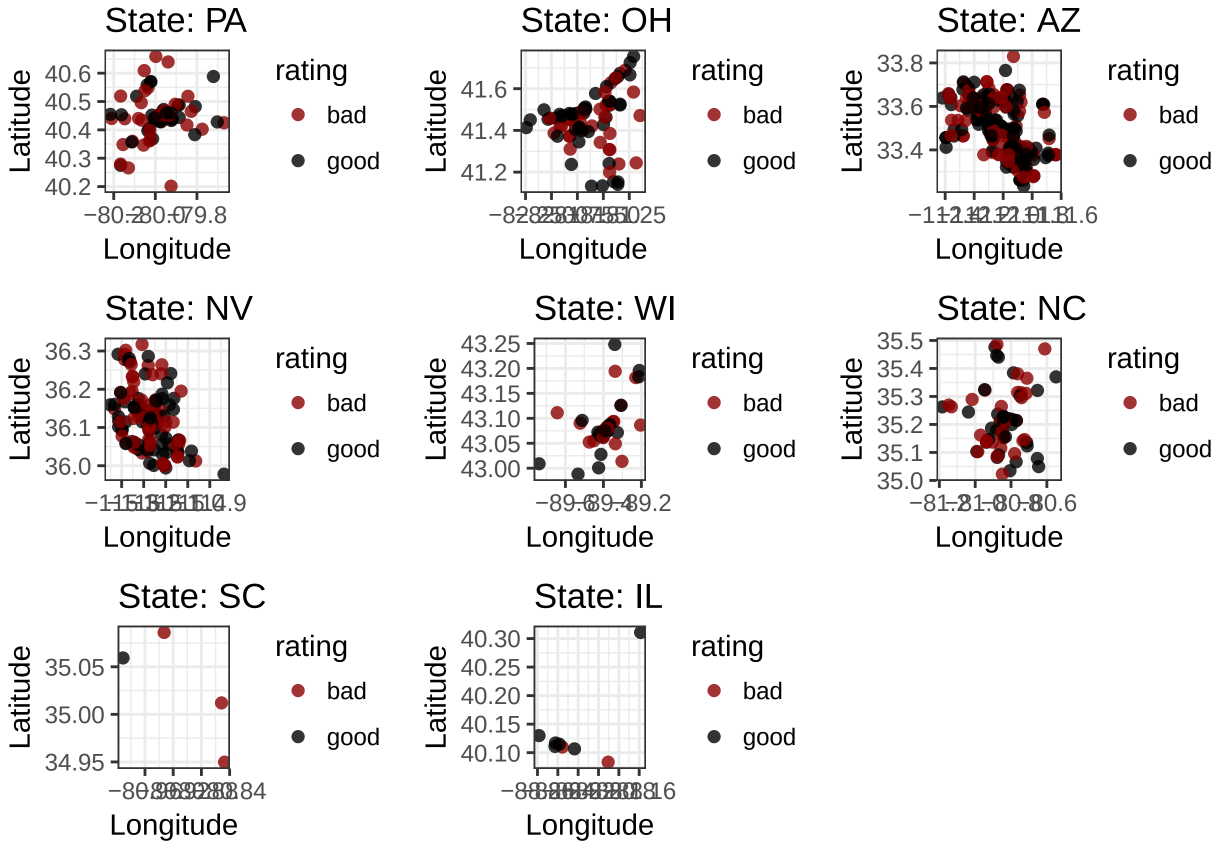


There appears to be difference on a state level basis concerning the review status.

Visualize

```
# Prepare for visualization
p <- c()
```

```
states <- unique(df_train$state)

# Iterate trough the states
for(i in 1:length(states)){
  p[[i]] <- ggplot(df_train[df_train$state == states[i], ],
                   aes(x=longitude, y=latitude, color=rating)) +
    geom_point(stroke=1, size=1, alpha=0.8) +
    theme_bw() +
    labs(title=paste("State:", states[i])) +
    ylab("Latitude") +
    scale_color_manual(values=c("darkred", "black")) +
    xlab("Longitude")
}

do.call(grid.arrange, p)
```



Problem 2 [35 points]

This problem is concerned with predicting a restaurant's rating based on text features. We'll consider the Multinomial-Dirichlet Bayesian model to describe the distribution of text features and finally to predict the binary ratings.

Probability model: Let $(y_1^g, y_2^g, \ldots, y_{100}^g)$ denote the total counts of the 100 dictionary words across the reviews for "good" restaurants, and $(y_1^b, y_2^b, \ldots, y_{100}^b)$ denote the total counts of the 100 dictionary words across the reviews for "bad" restaurants. We assume the following *multinomial* likelihood model:

$$p(y_1^g, y_2^g, \ldots, y_{100}^g \,|\, \theta_1^g, \ldots, \theta_{100}^g) \;\propto\; (\theta_1^g)^{y_1^g} (\theta_2^g)^{y_2^g} \ldots (\theta_{100}^g)^{y_{100}^g}$$

$$p(y_1^b, y_2^b, \ldots, y_{100}^b \,|\, \theta_1^b, \ldots, \theta_{100}^b) \;\propto\; (\theta_1^b)^{y_1^b} (\theta_2^b)^{y_2^b} \ldots (\theta_{100}^b)^{y_{100}^b}.$$

The model parameters $(\theta_1^g, \ldots, \theta_{100}^g)$ and $(\theta_1^b, \ldots, \theta_{100}^b)$ are assumed to follow a *Dirichlet* prior distribution with parameter α. That is

$$p(\theta_1^g, \ldots, \theta_{100}^g) \;\propto\; (\theta_1^g)^{\alpha} \ldots (\theta_{100}^g)^{\alpha}$$

$$p(\theta_1^b, \ldots, \theta_{100}^b) \;\propto\; (\theta_1^b)^{\alpha} \ldots (\theta_{100}^b)^{\alpha}.$$

Hence we can interpret, for example, θ_5^g as the probability the word "perfect" is observed once in a review of "good" restaurants. For the purposes of this problem, set $\alpha = 2$.

(a) Describe briefly in words why the posterior distribution formed from a Dirichlet prior distribution and a multinomial likelihood is a Dirichlet posterior distribution? What are the parameters for the Dirichlet posterior distribution? [5 points]

(b) From a Monte Carlo simulation of the Dirichlet posterior distribution for "good" restaurants, what is the posterior mean probability that the word "chocolate" is used? From a Monte Carlo simulation of the Dirichlet posterior distribution for bad restaurants, what is the posterior mean probability that the word "chocolate" is used? **Hint**: use the `rdirichlet` function in the `MCMCpack` library. [15 points]

(c) For the restaurants in the test data set, estimate the probability based on the results of the Dirichlet-Multinomial model that each is good versus bad. You may want to apply the function `posterior_pA` provided below (in `midterm-1.Rmd`). Create a visual summary relating the estimated probabilities and the actual binary ratings in the test data. [15 points]

Problem 3 [45 points]

This problem is concerned with modeling a restaurant's rating on factors other than word occurrences.

(a) Construct a model for the probability a restaurant is rated "good" as a function of latitude and longitude, average word count, and business attributes. Include quantitative predictor variables as smoothed terms as you see appropriate. You may use default tuning parameter. Summarize the results of the model. Does evidence exist that the smoothed terms are significantly non-linear? Produce visual displays to support your conclusions. [20 points]

(b) For your model in part (a), summarize the predictive ability by computing a misclassification rate. [10 points]

(c) Consider a version of model (a) that does not include the `cuisine` predictor variable. Explain briefly how you would test in your model whether `cuisine` is an important predictor of the probability of a good restaurant rating. Perform the test, and explain your conclusions. [15 points]