2014 Booth Data Mining Midterm

- This is an **INDIVIDUAL** exam: you must work alone.
- The exam must be submitted on chalk before **8am on Monday May 5**.
  This deadline is the same for all sections. You should submit a pdf document on chalk.
- Questions are ordered by class subject, not by difficulty. Each sub-question (i.e., A.B) is worth the same amount.
- The exam is not easy! And it is long. *You may still do fine even if you cannot finish, but you will want to start early.*
- Coding questions should be asked on Piazza, but you must not give away any answers when you formulate your query. We will only answer questions on R coding – not statistics concepts. We will continue to answer conceptual questions if they pertain to homework solutions or lecture examples.
- You will be graded on your written answers and analysis. Please include and refer to plots to illustrate your answers.
- Do not hand in detailed code, but do submit meaningful R output to help us understand your answer (e.g., printed $\hat{\beta}$ values, etc).
- Please be very clear: your answers need to be concise, precise, and easy to understand. Neatness in presentation is expected.

Data and code are on the course site in `midterm.zip`. This includes `yelp_start.R`; use this script as a starting point for your answers. The code can take a bit to run. It might be worth running once and saving so that you can quickly start from scratch. Much of the code you will need has already been written for you in this script.

The data are three tables and two name lists. They describe 8029 Yelp reviews for businesses in Phoenix AZ written from Sept-Dec 2012. The starter script shows how to read the data into R and build associated (sparse) model matrices.

`ReviewStats.csv` is a simple table containing, for each review, its 1-5 star rating (`stars`), the number of words in the review (`nwrd`), and the number of total existing yelp reviews by the author (`nrev`).

`WordFreq.csv` is a simple triplet matrix of word counts from review text, with `i` the review (row of `ReviewStats.csv`) and `j` the word index in our list `words.txt`.

`BizType.csv` is a simple triplet matrix of business type for each review, with `i` the review (row of `ReviewStats.csv`) and `j` the index from our list `categories.txt`. Note that this is a business supplied, non-exclusive taxonomy.

1 Significance and False Discovery

In the starter script, I've extracted a set of p-values from (independent) *marginal regression* of review `stars` on presence in review text for each of 5000 words. That is, we fit

$$\texttt{stars}_i = \alpha + \beta_j \mathbb{1}[x_{ji} > 0] + \epsilon_{ji}$$

for each term j with count x_{ji} in review i, and return the p-value associated with a test of $\beta_j \neq 0$. We'll use these 5000 independent regressions to screen words.

1.1

Plot the p-values and comment on their distribution.

1.2

What is the p-value rejection region associated with a 1% False Discovery Rate?
How many p-values are in this rejection region?

1.3

Suppose you just select the words with the 250 smallest p-values as significant.
How many of these do you expect are false discoveries?

Data for rest of exam

We now consider only words associated with the 250 smallest p-values from [1]. The *counts* for these words are combined in our design matrix '`x`' with `Biz`, the model matrix for business category membership, and `nwrd*Biz`, the interaction of business category and review length.

For the remaining questions, you'll look at models to predict `stars`.

2 Least-Squares and Bootstrapping

For this question you will fit a lasso path of Gaussian regressions for `stars` onto `x`. We will focus on the AICc selected slice of $\hat{\boldsymbol{\beta}}$ along this path.

2.1

What are the in-sample SSE and R^2 for this regression?

2.2

Describe what ratings we'd expect for `Bowling` businesses and `Airports`.

2.3

The starter script has code to bootstrap sampling distribution for the number the of parameters (degrees of freedom) selected by AICc in this regression.

- What is the standard error for selected df?
- What is a 95% CI (just use ±2SE) for df?

3 Model Choice and Logistic Regression

Define a 'bad review' as one with fewer than 4 stars. We'll regress this binary outcome on `x`.

3.1

Use `cv.gamlr` to fit a cross-validated lasso regularization path for this regression.
Plot the in-sample fit.

3.2

Describe the criteria used to choose models under `select="1se"` and `select="min"` rules. What are estimated out-of-sample R^2 for models fit using these λ?

3.3

Compare AICc, AIC, and BIC selection to each other and to the CV rules. Print the top ten best and worst review words under your preferred IC selection rule. Do the results make sense?

3.4

Long story, short question. Your startup idea is to use this yelp model to predict bad reviews on social media (e.g., facebook) that does not include a `star` ranking. The idea is that you'll alert proprietors to bad reviews, and they can respond to the author.

Your business plan envisions healthy profits from charging your clients a flat subscription fee. There are downside risks to your business model, however. Failing to alert a proprietor to a bad review costs on average $190 in lost business. False alarms (telling them a review is bad when it's not) cost $10 each. Correctly alerting or correctly not-alerting have each a cost of zero (people only notice when you screw up).

Say p is the probability that a review is bad. The exected cost of alerting is then $10(1-p)$, and cost of not alerting is $190p$, so that you'll alert at $p > 1/20$.

- What proportion of true bad reviews will you alert to at this cutoff?
- What will be your false alarm rate at this cutoff?
- What are the names we learned in class for the above two rates?

4 Treatment Effects Estimation

Some academic psychologist has a theory that with more reviews the users become tougher raters. We'll now try to measure this effect, while controlling for business type and review content. That is, we want to see if users with higher `nrev` give more `bad` reviews even with the same review content.

4.1

Our treatment variable will be $d = \log(\texttt{nrev})$. Justify this transformation.
How can we interpret the coefficient on this treatment variable?

4.2

Predict d from `x`, and describe how the fit is relevant for causal inference here (assuming `x` contains all relevant confounders).

4.3

Obtain a causal estimate for treatment effect of `nrev` on odds of a bad review, and describe this estimate in plain words.

4.4

Compare the estimate from 4.3 to the marginal (no controls) relationship between d and `y`.

5 Multinomial Regression

Now, we'll consider `stars` as an unordered 5-level factor. Use the code in your starter script to run a multinomial logistic regression.

5.1

What is the change in odds of 1 vs 2-3 stars for an extra count of word `crap` in the review?

5.2

What are the changes in odds of 1 star vs 2 and 1 star vs 5 star ratings for an extra count of word `fantastic` in the review?

Bonus

USE A MAXIMUM OF ONE PAGE, INCLUDING PLOTS, TO ANSWER.

Provide additional analysis of the data. Bonus will handed out only for insightful use of data mining tools, not for scattershot application of techniques.

Do not spend too much (or any!) effort here: it is worth little.