ORF350 (Spring 2017) Analysis of Big Data **Assignment: 4**

Logistic Regression

Lecturer: Han Liu *Email: hanliu@princeton.edu*

Due Date: This assignment is due on Monday, April 10 by 5:00PM.

Submission: Please submit your hard copy at the ORF 350 dropbox in the Sherrerd Hall student lounge. Electronic submissions will not be accepted unless there is an extreme and compelling case.

Question 1. Sentiment Analysis on Amazon Product Reviews (25 points)

Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Sentiment analysis is widely applied to reviews and social media for a variety of applications, ranging from marketing to customer service. In this homework, we'll explore product reviews and ratings from Amazon and build a sentiment analysis classifier based on regularized logistic regression.

(a) First, load in the data `Amazon_SML.RData`. Answer the following basic questions. What are the column names? How many reviews are there? How many unique products are shown? Which product has the most '5' ratings (and how many 5 ratings are there for this product)? Which product has the most '1' ratings (and how many 1 ratings are there for this product)? We suggest to use the `dlply` function in the `plyr` R package for this.

(b) How many reviews of each rating value are there in the entire dataset? Our goal is to build a classifier that reads the review and classifies whether the review was "good" (rating = 5) or "bad" (rating = 1). What is the best performance of a "constant classifier", a classifier that ignores the review and blindly assigns a constant classification?

Now let's build the term-document matrix `dtm.mat` using the script `tdMat.R`. This will help build a matrix where each row represents one document and each column represents a different word. Element (i,j) of the matrix then represents how many times word j appeared in document i (the ith review).

(c) Now we'll run regularized logistic regression on our dataset. We're going to split the `dtm.mat` into two parts: the training set and testing set using the script `splitData.R`. Fit a regularized logistic regression model by `glmnet`. The regularization parameter should be chosen as the `lambda.1se` (1 standard error above the minimizing lambda) calculated by `cv.glmnet`. Please use the random seed and lambda sequence as specified here so that we can check your results.

```
set.seed(10)
lambda<-exp(seq(-20, -1, length.out = 99))
cvfit<-cv.glmnet(x,y,family="binomial",type.measure="class",lambda=lambda)
```

How many covariates have non-zero coefficients in the model selected by lambda.1se? List the twenty words with the most positive coefficients and twenty words with most negative coefficients.

(d) We would like to investigate some of the words that our model selected. We will focus on the words with the most positive and most negative weights that appeared in more than 10 documents. What are these two words? How many documents using either of these two words had a rating of 1 or 5? Print out the first document (according to train.tag) that used this most positive word and the first document that used this most negative word.

(e) Now run the fitted logisitic model on our testing data and report the misclassification rate. How does this compare with the "constant classifier" we originally discussed before (i.e., is it better or worse)?

Q2. Non-existence of MLE for Logistic Regression (10 points)

We have data points $\{(x_1, y_1), ..., (x_n, y_n)\}$ with $y_i \in \{0, 1\}$ from a pair of random variables (X, Y). We assume the data follows a logistic regression model

$$Y \mid X = x \sim \text{Bernoulli}(\eta(x)), \tag{0.1}$$

$$\eta(x) = \frac{e^{\beta x}}{1 + e^{\beta x}}. \tag{0.2}$$

Here, both β and X are scalars.

(a) Write down the log-likelihood function for β.

(b) Suppose the data turn out to be as follows. For every $x_i \leq 0$ we have $y_i = 0$. For every $x_i > 0$ we have $y_i = 1$. Show that the maximum likelihood estimator $\widehat{\beta} = \infty$. [This question shows you a strange behavior of logistic regression. When the data are perfectly separable, the MLE does not exist.]

Q3. LDA and QDA for Multivariate Gaussian (25 points)

In this question, we'll derive the linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) for multivariate Gaussians. We'll provide subquestions that guide you through the derivation.

Let our training set be $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$ where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, 2\}$. Under the LDA framework, we assume

$$p(\boldsymbol{x}_i | y_i = 1) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-(\boldsymbol{x}_i - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_1)/2}$$
$$p(\boldsymbol{x}_i | y_i = 2) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-(\boldsymbol{x}_i - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_2)/2}$$

where $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1}$ denotes the inverse of $\boldsymbol{\Sigma}$. Under the QDA framework, we instead assume

$$p(\boldsymbol{x}_i | y_i = 1) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_1|^{1/2}} e^{-(\boldsymbol{x}_i - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_1)/2}$$
$$p(\boldsymbol{x}_i | y_i = 2) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_2|^{1/2}} e^{-(\boldsymbol{x}_i - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_2)/2}$$

These are multivariate Gaussians with means $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and different covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$. We assume that each data point $(\boldsymbol{x}_i, y_i)$ is independent of each other.

We denote $\eta = \mathbb{P}(y_i = 1)$ (the prior probability of class 1), $n_1 = \sum_{i=1}^n I(y_i = 1)$ and $n_2 = n - n_1$.

Question 3.1: Likelihood Model (5 points)

For LDA, write down the joint log-likelihood of a parameter configuration $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \eta$ with respect to the observed data

$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$.

That is, write down the factorization of

$$\sum_{i=1}^{n} \log p(\boldsymbol{x}_i, y_i).$$

Use the term η in your answer. [This question reviews log-likelihood.]

Hint: (Swapping Determinant and Inverse) Use $\log\big(|\boldsymbol{\Sigma}|^{-1}\big) = \log|\boldsymbol{\Sigma}^{-1}|$. This will be useful in Q4.3.

Question 3.2: MLE Estimation for η and $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ (10 points)

For LDA, using the log-likelihood equation developed in Q3.1, determine the maximum likelihood estimator $\widehat{\eta}$, the estimator for η. Likewise, determine the maximum likelihood estimators $\widehat{\boldsymbol{\mu}}_1$ and $\widehat{\boldsymbol{\mu}}_2$ for $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. Use the terms n_1, n_2 in your answer. [This question reviews MLE estimation.]

(For the rest of this problem, you may assume that the log-likelihood equation is concave.)

Question 3.3: MLE Estimation for $\boldsymbol{\Sigma}$ (10 points)

For LDA, using the log-likelihood equation developed in Q3.1, determine the maximum likelihood estimator $\widehat{\boldsymbol{\Sigma}}$, the estimator for $\boldsymbol{\Sigma}$. Let

$$S_1 = \frac{1}{n_1} \sum_{i:y_i=1}^{n} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1)^T,$$
$$S_2 = \frac{1}{n_2} \sum_{i:y_i=2}^{n} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_2)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_2)^T,$$

where $\widehat{\boldsymbol{\mu}}_1$ and $\widehat{\boldsymbol{\mu}}_2$ are the MLEs obtained in Question 3.2. Use the terms S_1 and S_2 in your answer. (Taking the derivative with respect to $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ might make your life easier.) [This question shows you how to use basic facts in linear algebra to derive MLE.]

Hint 1: Somewhere in your calculation, you will need the following facts. For a symmetric matrix $\boldsymbol{A}$,

$$\frac{\partial \log|\mathbf{A}|}{\partial \mathbf{A}} = \mathbf{A}^{-1}.$$

Hint 2: If c is some scalar computed by $\boldsymbol{b}^T\mathbf{A}\boldsymbol{b}$ for a vector $\boldsymbol{b}$ and matrix $\mathbf{A}$, we can write this term as $c = \text{Tr}(\boldsymbol{b}^T\mathbf{A}\boldsymbol{b})$ where $\text{Tr}(\cdot)$ is the trace of a matrix.

Hint 3: (Commutative Property of Trace): For matrices $\mathbf{A} \in \mathbb{R}^{d_1 \times d_2}$ and $\mathbf{B} \in \mathbb{R}^{d_2 \times d_1}$, $\text{Tr}(\mathbf{BA}) = \text{Tr}(\mathbf{AB})$.

Hint 4: For a matrix $\mathbf{A}$ and $\mathbf{B}$,

$$\frac{\partial \text{Tr}(\mathbf{B}^T\mathbf{A})}{\partial \mathbf{A}} = \mathbf{B}.$$

Q4. Naive Bayes (25 points)

In this question, we will be using parts of the SpamAssassin[1] dataset. Specifically, we will be designing a Naive Bayes (NB) algorithm to classify the text in the body of emails that are in the "easy_ham" and "spam" folders. Our goal is to understand the "naive" (independent features) part of the NB algorithm.

[1]For more information about this dataset, see `http://csmining.org/index.php/spam-assassin-datasets.html`

Unzip the data from `SpamAssassin.zip`. There will be two folders, `easy_ham` and `spam`, and both of these contains various text files. Each file contains metadata and the actual content of a specific email. There are 2188 ham emails (ie: not spam) and 996 spam emails. Design a preprocessing script to extract the "body" of all 3184 emails. Let `mail[[i]]$body` denote the text body (represented as a character vector where each element is a string for one line in the text body) of the i^{th} email. Apply the following set of regular expressions to extract the words from the text body.

```
tmp = mail[[i]]$body
tmp2 = paste(tmp$text,collapse="")
tmp3 = gsub("\\b([[:punct:]|[:digit:]])*[a-zA-Z]*([[:punct:]|[:digit:]])+
              [a-zA-Z]*([[:punct:]|[:digit:]])*"," ",tmp2)
tmp4 = gsub("[^A-Za-z]"," ",tmp3)
```

This regular expression is designed to remove "junk" expressions such as URLs. Then, use the `TermDocumentMatrix` function in the `tm` package to generate a term-document matrix where element (i, j) in this matrix contains the number of instances term i appears in document j. When using this function, use the `control` setting

```
list(removePunctuation = TRUE, stemming = TRUE, wordLengths = c(3, 20))
```

where `stemming` combines similar words such as "run" and "runs." Let `res` refer to the resulting output of `TermDocumentMatrix` with n columns (for $n = 3184$ emails) and m rows (for m different words in our dictionary). Here, `ncol(res)` $= 3184$, the total number of emails.

Question 4.1 (10 points) We define two quantities below. Let $\mathcal{J}_H$ and $\mathcal{J}_S$ denote the set of emails classified as ham and spam respectively. For each word i, calculate the following 2 quantities.

$$\text{Quantity 1:} \qquad \frac{1}{\sum_{j\in\mathcal{J}_H}\mathbb{I}[\texttt{res[i,j]}>0]}\sum_{j\in\mathcal{J}_H}\mathbb{I}[\texttt{res[i,j]}>0]\texttt{res[i,j]} \tag{0.3}$$

$$\text{Quantity 2:} \qquad \frac{1}{|\mathcal{J}_H|}\sum_{j\in\mathcal{J}_H}\mathbb{I}[\texttt{res[i,j]}>0] \tag{0.4}$$

where $\mathbb{I}[x]$ is an indicator function that takes value 1 when the boolean argument x is true, 0 otherwise, and where $|\mathcal{J}_H|$ denotes the size of set $\mathcal{J}_H$. For each quantity, determine the 10 words that obtain the largest value. (This will get you two lists with 10 words in each list.) Next, do the same but replacing $\mathcal{J}_H$ with $\mathcal{J}_S$. (This will get you two more lists of 10 words each. Your final answer to this question will be four lists of words.)

The first quantity is a mean of word counts for word i over all ham emails that contain the word while the second quantity is a frequency of whether or not word i appears in all ham emails. [This question teaches you a way to inspect the data before running any analysis.]

Question 4.2 (10 points) For this question and the next, separate all 3184 emails into the following test set and training set using the following code.

```
set.seed(1)
testingidx = sample(1:ncol(res),100)
trainingidx = 1:ncol(res)
trainingidx = trainingidx[-testingidx]
```

Let $w_{i,j}$ be an indicator based on element (i, j) of `res` that takes value 1 if $\texttt{res}[i, j] > 0$ and 0 otherwise, and let $y_{i,j} = \texttt{res}[i, j]$. Clearly $y_{i,j} = 0$ if and only if $w_{i,j} = 0$. Let $s_j = 1$ if document j is spam, $s_j = 0$ if document j is ham. Let's assume the conditional probability for $w_{i,j}$ and $y_{i,j}$ given $s_j = 1$ is the following

$$p_{\theta_i,\lambda_i}(w_{i,j}, y_{i,j} \mid s_j = 1) = \left[\theta_i \frac{e^{-\lambda_i}\lambda_i^{y_{i,j}-1}}{(y_{i,j}-1)!}\right]^{w_{i,j}} (1-\theta_i)^{(1-w_{i,j})} \tag{0.5}$$

where θ_i and λ_i are parameters specific to word i. Here, $\theta_i \in [0,1]$ and $\lambda_i \in \mathbb{Z}_{+0}$ (a nonnegative integer). A similar probability follows given a ham classification. Fit the following model according to the Naive Bayes algorithm and report the prediction accuracy on the testing data. [This question reviews MLE and helps you appreciate the conditional independence regularization in the Naive Bayes algorithm.]

Hint 1: Recall that

$$p(w_{1:m,1:n}, y_{1:m,1:n}, s_{1:n}) = \prod_{j=1}^{n} p(s_j) \prod_{i=1}^{m} p(w_{i,j}, y_{i,j} \mid s_j)$$

where we omit the subscripts for notational convenience.

Hint 2: Fitting the model will require you to calculate the MLE's. Using sufficient statistics will help you avoid referring to the actual data in `res` too often. Be careful to avoid calculating terms such as $\log x$ or $x!$ when x is 0. You can avoid the former instances by perturbing x to make it a very small positive number and the latter instances by noting that $y_i = 0$ implies that $w_{i,j} = 0$, so the entire term $\theta_i e^{-\lambda_i}\lambda_i^{y_{i,j}-1}\backslash(y_{i,j}-1)!$ is pointless to calculate when $w_{i,j} = 0$.

Remark: Our statistical model is simply embedding a shifted Poisson distribution inside a Bernoulli distribution. We use a shifted Poisson since $w_{i,j} \neq 0$ if and only if $y_{i,j} > 0$.

Question 4.3 (5 points) The method proposed above used a specific set of regular expressions to extract the words out of all the emails. Many words that remained might be considered "useless," and some words that were filtered out might actually contain predictive value. Design your own set of regular expressions to use in the `gsub` and re-run the exact same Naive Bayes algorithm developed in Question 4.2. Report the prediction accuricies based on this custom set of regular expressions. Write 2-3 sentences describing the intuition for your set of regular expressions, and comment on whether or not your prediction results got better or worse from Question 4.2. [This question helps you develop an appreciation for the power of regular expressions.]

Hint: One thing you can do is simply replace `tmp3 = ...` with your own regular expression. We're not expecting you to find a fantastic regular expression since it's (in general) not easy.

Q5. The Bayes Rule (15 points)

Question 5.1 (10 points)

Suppose that $Y \in \{0,1\}$ and $\mathbb{P}(Y=1) = 1/2$. The distribution of $X \mid Y = 0$ is discrete and is specified by

$$\mathbb{P}(X = 1 \mid Y = 0) = 1/3 \quad \mathbb{P}(X = 2 \mid Y = 0) = 2/3.$$

The distribution of $X \mid Y = 1$ is discrete and is given by

$$\mathbb{P}(X = 2 \mid Y = 1) = 1/3 \quad \mathbb{P}(X = 3 \mid Y = 1) = 2/3.$$

Find the Bayes rule and the Bayes risk (Hint: When evaluating the Bayes risk, you may want to use the formula $\mathbb{P}(Y \neq h(X)) = \mathbb{E}_X[\mathbb{P}(Y \neq h(X) \mid X)] = \sum_{k=1}^{3} \mathbb{P}(Y \neq h(X) \mid X = k)\mathbb{P}(X = k)$).

Question 5.2 (5 points)

We consider a naive Bayes classifier with 2 predictors (or input variables) X_1 and X_2 and one output variable Y. All X_1, X_2, and Y are binary variables and can only take binary values either $+1$ or -1. (Note: In the class, we described the Naive Bayes modeling mainly through conditional Gaussian model, but in the homework, you have seen Naive Bayes for discrete distributions. Here is the latter case.)

(a). How many parameters must be estimated to train such a Naive Bayes classifier? (Provide a very brief justification).

(b). How many parameters would have to be estimated to learn the above classifier if we do not make the Naive Bayes assumption? (Provide a very brief justification).

Appendix

A. Relation to Case Study Videoes

Q1 performs sentiment analysis, a topic similar to the "Sentiment Analysis" case study videos. However, the "Sentiment Analysis" case study videos use a different dataset based on news article headlines about the U.S. economy, and uses a Naive Bayes classifier. Overall, the "Sentiment Analysis" case study videos are good to watch for related background and to understand what a term-document matrix, but is not directly related to this homework. Note that the term-document matrix in the "Sentiment Analysis" case study videos is a term-document matrix containing Booleans, unlike the one in our homework which contains word counts.

The "Framingham Heart Study" case study gives background on logistic regression. Our homework uses a regularized logistic regression however, so it is not directly related to this homework. The "Digits Recognition" case study gives background on linear discriminant analysis and quadratic discriminant analysis.