STA 141C - Big Data & High Performance Statistical Computing **Spring 2022**

Homework 4

Lecturer: Bo Y.-C. Ning *Due March 09, 2022*

Due **March 09, 2022** by 11:59pm.

A few notes:

1. Submit your homework using the file name "**LastName_FirstName_hw4**"
2. Answer all questions with complete sentences.
3. Your code should be readable; writing a piece of code should be compared to writing a page of a book. Adopt the **one-statement-per-line** rule. Consider splitting a lengthy statement into multiple lines to improve readability. (You will lose one point for each line that does not follow the one-statement-per-line rule)
4. To help understand and maintain code, you should always add comments to explain your code. (homework with no comments will receive 0 points). For a very long comment, break it into multiple lines.
5. Submit your final work with one **.pdf** (or **.html**) file to Canvas. I encourage you to use LaTeX for writing equations. Handwriting is acceptable, you have to scan it and then combine it with the coding part into a single .pdf (or .html) file. Handwriting should be clean and readable.

1 Handwriting recognition

In this homework, we work on a model-based method for handwritten digit recognition. Following figure shows example bitmaps of handwritten digits from U.S. postal envelopes.

Each digit is represented by a 32×32 bitmap in which each element indicates one pixel with a value of white or black. Each 32×32 bitmap is divided into blocks of 4×4, and the number of white pixels are counted in each block. Therefore each handwritten digit is summarized by a vector $x = (x_1, \dots, x_{64})$ of length 64 where each element is a count between 0 and 16.

By a model-based method, we mean to impose a distribution on the count vector and carry out classification using probabilities. The goal is to predict handwritten digit. We separate the dataset into training data and test data. The training set contains 3823 handwritten digits and the test set contains 1797 digits.

A common distribution for count vectors is the multinomial distribution. However, it is not a good model for handwritten digits. Let's work on a more flexible model for count vectors, the Dirichlet-multinomial model.

For a multivariate count vector $x = (x_1, \dots, x_d)$ with batch size $|x|= \sum_{j=1}^d x_j$, the probability mass function for Dirichlet-multinomial distribution is

$$f(x|\alpha) = \binom{|x|}{x} \frac{\prod_{j=1}^d (\alpha_j)_{x_j}}{(|\alpha|)_{|x|}},$$

where $(a)_k = \prod_{i=0}^{k-1}(a+i)$.

Given independent data points $x_1, \dots, x_n$, the log-likelihood is given by

$$L(\alpha) = \sum_{i=1}^n \log \binom{|x_i|}{x_i} + \sum_{i=1}^n \sum_{j=1}^d [\log(\Gamma(\alpha_j + x_{ij}) - \log(\Gamma(\alpha_j))] - \sum_{i=1}^n [\log \Gamma(|\alpha|+|x_i|) - \log \Gamma(|\alpha|)] .$$

How do you calculate the MLE?

In this exercise, we use Newton's method. First, the score function is given by

$$\frac{\partial}{\partial \alpha_j} L(\alpha) = \sum_{i=1}^n [\Psi(x_{ij} + \alpha_j) - \Psi(\alpha_j)] - \sum_{i=1}^n [\Psi(|x_i|+|\alpha|) - \Psi(|\alpha|)] ,$$

where $\Psi(x) = d(\log \Gamma(x)) = \Gamma'(x)/\Gamma(x)$.

Next, the observed information matrix is given by

$$-d^2 L(\alpha) = D - c\mathbb{1}_d \mathbb{1}_d',$$

where D is a diagonal matrix,

$$D_{jj} = \sum_{i=1}^n [\Psi'(\alpha_j) - \Psi'(x_{ij} + \alpha_j)] = \sum_{i=1}^n \sum_{k=0}^{x_{ij}-1} \frac{1}{(\alpha_j + k)^2}$$

and

$$c = \sum_{i=1}^n [\Psi'(|\alpha|) - \Psi'(|x_i|+|\alpha|)] = \sum_{i=1}^n \sum_{k=0}^{|x_i|-1} \frac{1}{(|\alpha|+k)^2}.$$

Then given an initial value for $\alpha^{(0)} = (\alpha_1^{(0)}, \dots, \alpha_n^{(0)})$, the Newton's method keep updating

$$\alpha^{(t)} = \alpha^{(t-1)} + [-d^2 L(\alpha^{(t-1)})]^{-1} dL(\alpha^{(t-1)})$$

for $t = 1, \dots, T$. We stop when $|L(\alpha^{(t)}) - L(\alpha^{(t-1)})| \leq \epsilon$, and then take $\hat{\alpha} = \alpha^{(t)}$.

Note that D is a diagonal matrix, we should use the Sherman-Morrison formula to write

$$[-d^2 L(\alpha)]^{-1} = D^{-1} + \frac{1}{1/c - \sum_j d_j^{-1}} D^{-1} \mathbb{1}\mathbb{1}' D^{-1}$$

In the folder uploaded on Piazza, you will find

- Data containing the training data and the testing data
- 'ddirmult.R', which evaluates the likelihood function (if log = FALSE) or the log-likelihood function (if log = TRUE) of the Dirichlet-multinomial density

- ‘ddirmult.fit.R‘, which estimates the maximum likelihood estimator (MLE) by Newton’s method
- ‘trainingMLE.R‘, which estimates the MLE based on the training data

Question 1. Open ‘trainingMLE.R’ and obtain MLE estimators for each of the 10 handwriting digits $(0, 1, 2, \ldots, 9)$. (You may need to change the path when loading the data)

Question 2. Read in the testing data. Use the estimated MLE for each digit from the training data to predict handwriting digits for the testing data.

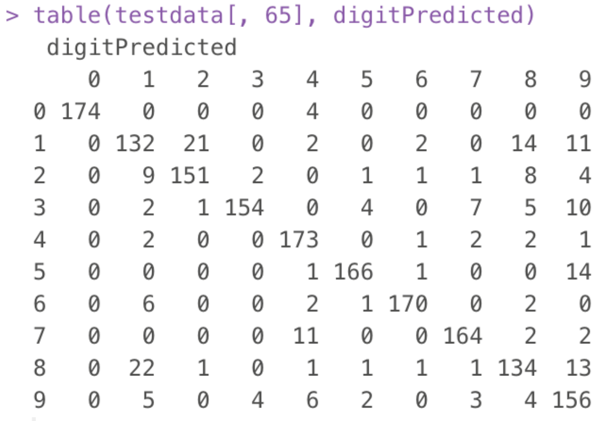
- Hint 1: To predict the handwriting digit, you should use the ‘ddirmult.R’ function. The following code can be helpful

```r
# testDigitProb stores posterior probability of each digit being 0,1,...,9
testdata <- read.table("PATH/optdigits.tes", sep = ",")
testdata <- as.matrix(testdata)
testDigitProb <- matrix(0, dim(testdata)[1], 10)
for (dig in 0:9) {
  testDigitProb[, dig + 1] <-
    ddirmult(testdata[, -65], alphahat[dig + 1, ], log = TRUE)
}
testDigitProb <- testDigitProb +
  rep(log(digitCount / sum(digitCount)), each = nrow(testdata))
digitPredicted <- max.col(testDigitProb) - 1
```

- Hint 2: To summarize the result, you can construct a confusion table using the code

```r
table(testdata[, 65], digitPredicted)
```

The output should look like this:



```
> table(testdata[, 65], digitPredicted)
   digitPredicted
      0   1   2   3   4   5   6   7   8   9
  0 174   0   0   0   4   0   0   0   0   0
  1   0 132  21   0   2   0   2   0  14  11
  2   0   9 151   2   0   1   1   1   8   4
  3   0   2   1 154   0   4   0   7   5  10
  4   0   2   0   0 173   0   1   2   2   1
  5   0   0   0   0   1 166   1   0   0  14
  6   0   6   0   0   2   1 170   0   2   0
  7   0   0   0   0  11   0   0 164   2   2
  8   0  22   1   0   1   1   1   1 134  13
  9   0   5   0   4   6   2   0   3   4 156
```

Question 3. Comment on using gradient descent to obtain the MLE (instead of Newton’s method)? (You do not need to implement this.)

Question 4. What is the advantage and disadvantage of using gradient descent instead of Newton’s method?

Question 5. Do you think the current method is satisfactory for predicting handwriting digits? Do you know any other method(s) that can achieve a higher accuracy?