Foundations of Computational Math – Final Exam

Sam Reeves

```
library(dplyr)
library(ggplot2)
library(igraph)
```

Part 1. Playing with Pagerank

1. The A and B Matrices (5)

Form the A matrix. Then, introduce decay and form the B matrix as we did in the course notes.

$$B = 0.85 \times A + \frac{0.15}{n}$$

```
n <- 6
A <- matrix(c(0, 1/2, 1/2, 0, 0, 0,
              0, 0, 0, 0, 0, 0,
              1/3, 1/3, 0, 0, 1/3, 0,
              0, 0, 0, 0, 1/2, 1/2,
              0, 0, 0, 1/2, 0, 1/2,
              0, 0, 0, 1, 0, 0),
            ncol = n, byrow = TRUE,
            dimnames = list(c(1:6), c(1:6)))

B <- (A * 0.85) + (0.15 / n)
```

2. The Rank Vector (5)

Start with a uniform rank vector r and perform power iterations on B till convergence. That is, compute the solution $r = B^n \times r$. Attempt this for a sufficiently large n so that r actually converges.

```
r_i <- rep(1/6, 6)

power_iteration <- function(old_matrix) {
  new_matrix <- old_matrix
    for (j in seq(1:nrow(old_matrix))) {
      for (k in seq(1:ncol(old_matrix))) {
      new_matrix[j, k] <- old_matrix[j,] %*% old_matrix[,k]
      }
    }
```

```
    return(new_matrix)
}

converge <- function(old_matrix, n = 1) {
  new_matrix <- power_iteration(old_matrix)
  n <- n + 1

  if (identical(new_matrix, old_matrix)) {
    return(list(n, new_matrix))
  } else {
    converge(new_matrix, n)
  }
}

converge(B)
```

```
## [[1]]
## [1] 16
##
## [[2]]
##   1 2 3 4 5 6
## 1 0 0 0 0 0 0
## 2 0 0 0 0 0 0
## 3 0 0 0 0 0 0
## 4 0 0 0 0 0 0
## 5 0 0 0 0 0 0
## 6 0 0 0 0 0 0
```

3. Eigen-Decomposition (10)

> Compute the eigen-decomposition of B and verify that you indeed get an eigenvalue of 1 as the largest eigenvalue and that its corresponding eigenvector is the same vector that you obtained in the previous power iteration method. Further, this eigenvector has all positive entries and it sums to 1.

```
eigen_B <- eigen(B)

eigen_B$values[1]
```

```
## [1] 0.9516527+0i
```

```
eigen_B$vectors[,1]
```

```
## [1] -0.2159123+0i -0.0572329+0i -0.2980792+0i -0.5358032+0i -0.5358032+0i
## [6] -0.5358032+0i
```

```
eigen(A)
```

```
## eigen() decomposition
## $values
```

```
## [1]  1.0000000 -0.5000000 -0.5000000  0.4082483 -0.4082483  0.0000000
##
## $vectors
##           [,1]       [,2]       [,3]      [,4]       [,5]       [,6]
## [1,] 0.1118034 -0.5345225 -0.5345225 0.7745967  0.7745967 -0.5773503
## [2,] 0.0000000  0.0000000  0.0000000 0.0000000  0.0000000  0.5773503
## [3,] 0.2236068  0.5345225  0.5345225 0.6324555 -0.6324555 -0.5773503
## [4,] 0.5590170 -0.2672612 -0.2672612 0.0000000  0.0000000  0.0000000
## [5,] 0.5590170 -0.2672612 -0.2672612 0.0000000  0.0000000  0.0000000
## [6,] 0.5590170  0.5345225  0.5345225 0.0000000  0.0000000  0.0000000
```

4. Graph Algorithm (10)

Use the *graph* package in R and its *page.rank* method to compute the PageRank of the graph as given in A. Note that you don't need to apply decay. The package starts with a connected graph and applies decay internally. Verify that you do get the same PageRank vector as the two approaches above.

Part 2. The Venerable MNIST Dataset

1. Getting Started

Get a Kaggle.com account.

Username: samreeves

2. The Dataset

Download the data for /c/digit-recognizer/overview – The MNIST dataset of handwritten images.

```
mnist_train <- tibble(read.csv('prob2/train.csv'))
mnist_test <- tibble(read.csv('prob2/test.csv'))
```

3. Min-Max Scaling (5)

Using the training.csv file, plot representations of the first 10 images to understand the data format. Go ahead and divide all pixels by 255 to produce values between 0 and 1.

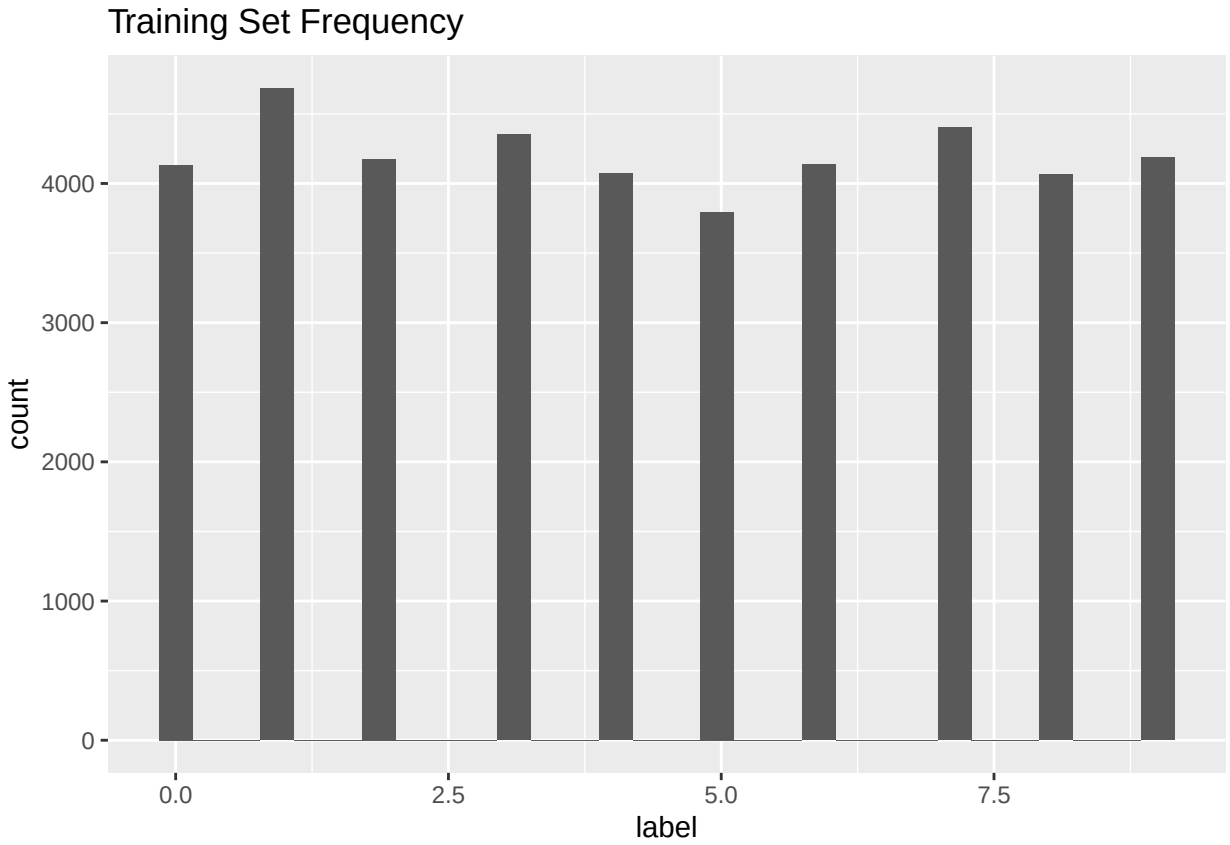
```
mnist_transformed <- cbind(mnist_train[,1],
                           mnist_train[,2:785]/255)
```

4. Understanding the Data Part 1 (5)

What is the frequency distribution of the numbers in the dataset? (5)

```
mnist_train %>% ggplot(aes(x = label)) +
  geom_histogram() +
  labs(title = 'Training Set Frequency')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The frequency distribution of the training set is basically uniform.

5. Understanding the Data Part 2 (5)

For each number, provide the mean pixel intensity. What does this tell you?

```
intensity <- mnist_transformed %>%
  group_by(label) %>%
  summarize_all("mean")

rowMeans(intensity[,2:785])
```

```
##  [1] 0.17323133 0.07597272 0.14941526 0.14165760 0.12121210 0.12923129
##  [7] 0.13873008 0.11470210 0.15098113 0.12281788
```

This tells you "how much ink" is used to write each number on average. It makes sense that "0" and "8" would have the greatest average intensity, while "1" would have the lowest.

6. Principal Component Analysis (5)

Reduce the data by using principal components that account for 95% of the variance. How many components did you generate? Use PCA to generate all possible components (100% of the variance). How many components are possible? Why?

7. Plotting and Noise (5)

Plot the first 10 images generated by PCA. They will appear to be noise. Why?

8. Accounting for Variance (5)

Now, select only those images that have labels that are 8's. Re-run PCA that accounts for all of the variance (100%). Plot the first 10 images. What do you see?

9. Multinomial Model (10)

An incorrect approach to predicting the images would be to build a linear regression model with y as the digit values and X as the pixel matrix. Instead, we can build a multinomial model that classifies the digits. Build a multinomial model on the entirety of the training set. Then, provide its classification accuracy (percent correctly identified) as well as a matrix of observed versus forecast values (confusion matrix). This matrix will be a 10 x 10, and correct classifications will be on the diagonal.

Part 3. Kaggle Competition – House Prices: Advanced Regression Techniques

```
house_train <- tibble(read.csv('prob3/train.csv'))
house_test <- tibble(read.csv('prob3/test.csv'))
```

Descriptive and Inferential Statistics (5)

Provide univariate descriptive statistics and appropriate plots for the training data set. Provide a scatterplot for at least two of the independent variables and the dependent variable. Derive a correlation matrix for any three quantitative variables in the dataset. Test the hypothesis that the correlations between each pairwise set of variables is 0 and provide an 80% confidence interval. Discuss the meaning of your analysis. Would you be worried about familywise error? Why or why not?

Linear Algebra and Correlation (5)

Invert your correlation matrix from above. (This is known as the precision matrix and contains variance inflation factors on the diagonal.) Multiply the correlation matrix by the precision matrix, and then multiply the precision matrix by the correlation matrix. Conduct LU decomposition on the matrix.

Calculus-Based Probability and Statistics (10)

Many times it makes sense to fit a closed form distribution to data. Select a variable in the Kaggle.com training dataset that is skewed to the right, shift it so that the minimum value is absolutely above zero if necessary. Then load the $MASS$ package and run $fitdistr$ to fit an exponential probability density function. Find the optimal value of λ for this distribution, and then take 1000 samples from this exponential distribution using this value (e.g., rexp(1000, λ)).

Plot a histogram and compare it with a histogram of your original variable. Using the exponential PDF, find the 5^th and 95^th percentiles using the CDF. Also, generate a 95% confidence interval from the empirical data, assuming normality. Finally, provide the empirical 5^th percentile of the data. Discuss.

Modeling (10)

Build some type of *multiple* regression model and submit your model to the competition board. Provide your complete model summary and results with analysis. Report your Kaggle.com username and score, and provide a screenshot.