CS 422-04: Data Mining (CRN: 19881)

Department of Computer Science
Illinois Institute of Technology
Vijay K. Gurbani, Ph.D.

Fall 2017: Homework 4 (10 points)

Due date: Wednesday, Nov 29, 2017 11:59:59 PM Chicago Time

Please read all of the parts of the homework carefully before attempting any question. If you detect any ambiguities in the instructions, please let me know right away instead of waiting until after the homework has been graded.

In the archive you upload as your homework, PLEASE DO NOT INCLUDE MS-WORD, APPLE PAGES, or any other proprietary document formats. The only accepted document formats are PDF, R scripts, simple text documents that do not need a word processor and Knitted documents in HTML format. ALL OTHER DOCUMENT TYPES WILL NOT BE ACCEPTED. Handwritten documents are fine, but they (a) must be legible to score any points, and (b) must be scanned and converted to PDF.

1 Exercises

1.1 Chapter 8 (0.75 points evenly divided by number of exercises)

Exercises 6, 12, 16

1.2 Leskovec, Ch. 3 (0.75 points evenly divided by number of exercises)

Page 72, Exercise 3.1.1; Page 74, Exercise 3.2.1; Page 81, Exercise 3.3.3; Page 86, Exercise 3.4.1 (evaluate the S curve only for the first two bullets).

1.3 Leskovec, Ch. 9 (0.75 points evenly divided by number of exercises)

Page 320, exercise 9.2.1; page 321, exercise 9.2.3; page 327, exercise 9.3.1.

1.4 Leskovec, Ch. 5 (0.75 points evenly divided by number of exercises)

Page 175, exercise 5.1.1; page 176, exercise 5.1.2; page 177, exercise 5.1.6.

2 Practicum problems

2.1 Topic: Hierarchical clustering (2.3 point evenly divided by number of questions)

For this problem, you will use the "Languages spoken in Europe" dataset available from the Hartigan datasets. This dataset is available at https://people.sc.fsu.edu/~jburkardt/datasets/hartigan/file46.txt. Read this file into R. Note that you will have to pre-process the file since it is not, strictly speaking, a CSV file. Also make sure that the first column is recognized as a row label. (Hint: See the help on read.csv() and look at the row.names parameter.) Recognizing the first column as a row label is important because we want the country names to be printed as labels in the dendograms.

(a) Run hierarchical clustering on the dataset using factoextra::eclust() method. Run the clustering algorithm for three linkages: single, complete, and average. Plot the dendogram associated with each linkage using fviz_dend(). Make sure that the labels (country names) are visible at the leafs of the dendogram.

(b) Examine each graph produced in (a) and understand the dendrogram. Notice which countries are clustered together as two-singleton clusters (i.e., two countries clustered together because they are very close to each other in the shared languages/s). For **each** linkage method, list all the two singleton clusters. For instance, {Great Britain, Ireland} form a two-singleton cluster since they share English as a common language.

(c) Italy is clustered with a larger cluster in the single and average linkage, whereas in complete linkage it is clustered with a smaller cluster. Which linkage strategy do you think accurately reflects how Italy should be clustered? (Hint: Look at the raw data.) Justify your answer in 1-2 sentences.

(d) Let's pick a hierarchical cluster that we will call *pure*, and let's define *purity* as the linkage strategy that produces the most two-singleton clusters. Of the linkage methods you examined in (b), which linkage method would be considered *pure* by our definition?

(e) Using the graph corresponding to the linkage method you chose in (d), at at a height of about 125, how many clusters would you have?

(f) Now, using the number of clusters you picked in (e), re-run the hierarchical clustering using the three linkage modes again, except this time through, specify the number of clusters using the *k* parameter to factoextra::eclust(). Plot the dendogram associated with each linkage using fviz_dend(). Make sure that the labels (country names) are visible at the leafs of the dendogram.

(g) For each cluster obtained by the value of *k* used in (f), print the Dunn and Silhouette width using the fpc::cluster.stats() method. Take a look at the help (or manual) page for fpc::cluster.stats() and see what is the name of the return list component that contains the Dunn index and the average Silhouette width.

(h) From the three clusters in (g), which is the best cluster obtained if you consider the Dunn index only?

(i) From the three clusters in (g), which is the best cluster obtained if you consider the Silhouette width only?

2.2 Topic: Locality sensitive hashing (2.4 point evenly divided by number of questions)

In this problem, you will run LSH or Locality Sensitive Hashing. Make sure you set the seed to be **100** when you invoke textreuse::minhash_generator(). **If you do not do this, your output will not match expectation and points will be taken off!**

You are provided a corpus of 100 documents (see corpus.zip). This corpus is donated by Paul Clough and Mark Stevenson (both from University of Sheffield). They designed this corpus to represent varying degrees of plagiarism and hoped that it will be a useful addition to the set of resources available for the evaluation of plagiarism detection systems.

Read this corpus in by unzipping the ZIP file in a directory. Let's assume you unzip it in /home/vkg/corpus. To read all the files in, use the following R command:

```
file s <- list.files("/home/vkg/corpus", full.names=T)
```

The above command will cause all of the files to be stored in the list files, using the complete path name of the files. You can subsequently pass the files list to textreuse::TexReuseCorpus() function.

Based on this corpus, please answer the following questions.

(a) How many shingles (or tokens) are there in all of the 100 documents? (Hint: look at package TextReuse::tokens()).

(b) What are the dimensions of the *characteristic matrix*?.

(c) Print the first 5 shingles (or tokens) of the file orig_taske.txt.

(d) We will fix our signatures (or hashes, or the rows in the *signature matrix*) at 240. This represents what percentage reduction in the size of the problem?

(e) At 240 signatures (or hashes) we want a probability of 0.888 of getting a candidate pair in at least one band at a Jaccard similarity of 0.3 and above. How many bands will you need to get such a probability?

(f) Using the number of bands you determined in (e), run LSH and find candidate pairs. How many candidate pairs do you get?

(g) Sort the candidate pairs according to their score field, in descending order (i.e., from highest score to lowest score). List the top 5 candidates that are similar.

(h) If you were **not to use** LSH and instead, examined every pair of documents for similarity, (i) how many pairs of documents would you examine? (ii) What is the ratio of this number to the to the number of candidate pairs you found in (f)?

2.3 Topic: Recommender systems (2.3 point evenly divided by number of questions)

In this problem, you will explore recommender systems using the MovieLens database (see https://grouplens.org/datasets/movielens/100k/). The 100K dataset consists of 100,000 ratings (1-5) from 943 users on 1,682 movies. Please read the README.txt file at the MovieLens database to get more information about the contents of the dataset. You will be using two files in the dataset: u.data and u.item. The layout of these files is described in the README.txt file.

(a) Content-based filtering: You will develop a small content-based recommender system. First, you will build a user profile of two users. To build this profile, use the *genre* fields from u.item. The last 19 fields in the u.item file consist of genres. If a genre field has a ‘1’, that implies the movie belongs to that genre. A ‘0’ implies the movie does not belong to that genre. A movie may belong to more than 1 genre. To build the user profile, calculate the mean (average) of all of the movies that the user has seen that belong to a set of genres. For example, let’s say that the movies “Godfather” and “Fargo” belong to genres Drama and Crime, and that “Daddy’s Home” belongs to genre Comedy. Assume that user 1 has seen Godfather and Daddy’s Home. His profile will be a vector corresponding to the genres, where the first element in the vector represents Drama, the second Crime and the third Comedy. User 1’s profile will be the following vector:

[0.5, 0.5, 0.5]

If user 2 has seen Fargo only, her profile will be the following vector:

[1.0, 1.0, 0.0]

The vector for the item (movie) Godfather will be [1, 1, 0] since Godfather belongs to genres Drama and Crime (the first two elements of the vector) but not to genre Comedy.

From the above vectors, you can compute user-user similarity and user-item similarity using the Cosine similarity.

(i) Compute the user-user similarity of users with ID 200 and user with ID 50. (Hint: In order to compute the user-user similarity of the two users, you first have to get a vector of genre similarity of all the movies that each user has watched. This can be accomplished by building the user profile as described above. In R, you can use the following pseudo code to get a genre similarity vector:

```
user200 <- Get all records of user 200 from the u.data file
movies200 <- Get movies user200 has watched from the u.item file
```

```
movie.matrix <- get columns 6-24 of all movies200
genre200 <- apply(movie.matrix, 2, mean)
```

The last line above uses the R apply() function. The apply() function is very versatile; make sure you read the manual page for it. Here it will go down all of the columns of the matrix (the parameter '2' performs computation on the columns) and calculate the computation specified (here, the computation is mean). When the method is done, you have a profile for user 200 in 19 dimensions corresponding to the genres of movie the user watches.)

(ii) Compute the user-item similarity of movie with ID 127 to user 200.

(iii) Compute the user-item similarity of movie with ID 127 to user 50.

(iv) Based on the above two computations, the movie 127 will be recommended to which user?

(b) Collaborative Filtering: Create a utility matrix for users with IDs 1, 21, 44, 59, 72, 82, 102, 234, 268, 409, 486 and movies with IDs 1, 2, 3, 4, 5 and 6. Using this utility matrix, find out the expected rating that user 268 will give to movie 5 using |N| = 3. (Hint: follow the procedure in slides 30-33 of Lecture 11.)