Big Data Analytics Assignment II

Due: 23:55 December 11th, 2020

November 17, 2020

Note: You should submit a report including the main results, and submit the code with readme.txt in a separate folder.

1 Text Classification

We have two categories of emails, in which one category is about hockey and the other is about baseball. The data is in the folder classification.

a) Firstly preprocess the documents into numerical data, *e.g.*, you can tf-idf or word2vec.

b) Use Naive Bayes to classify the documents and test the classification results with 5-fold cross validation. You should report the precision, recall, and F1-measure of each fold and the average values.

c) Implement Sequential Minimal Optimization (SMO) by following the introductive slides.

2 Clustering for Amyotrophic Lateral Sclerosis (ALS)

This case-study examines the patterns, symmetries, associations and causality in a rare but devastating disease, amyotrophic lateral sclerosis (ALS). ALS demands conducting clinical trials and collecting big, multi-source and heterogeneous datasets that can be interrogated to derive potential biomarkers. Overcoming many scientific, technical and infrastructure barriers is required to establish complete, efficient, and reproducible protocols (pipelines/workflows) starting with acquiring raw data, preprocessing, aggregation, harmonization, analysis, visualization and result interpretation. The dataset is ALS.csv.

The clinical data shows that the rate of ALS progression varies significantly among patients. Majority of the patients die within 3 to 5 years after ALS onset, however, a few are able survive for over 10 years. This heterogeneity of disease course hinders demonstration of its biological mechanism and development of effective treatment. We need to develop reliable predictive models of ALS progression to understand the pathophysiology of the disease.

Now perform the clustering analysis according to the following procedures for the ALS dataset.

- Load and prepare the data.
- Perform summary and preliminary visualization.
- Train a K-Means model on the data, select an appropriate K by using the Elbow method. (Tips: *1. Compute clustering algorithm (e.g., K-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters. 2. For each k, calculate the total within-cluster sum of squared errors (SSE). 3. Plot the curve of SSE according to the number of clusters k. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.*)
- Evaluating the model performance by report the center of clusters and explain details. You can also interpret the clustering results by inspecting the data near the center.

Reference: Tang, M., Gao, C, Goutman, SA, Kalinin, A, Mukherjee, B, Guan, Y, and Dinov, ID. (2018) Model-Based and Model-Free Techniques for Amyotrophic Lateral Sclerosis Diagnostic Prediction and Patient Clustering, Neuroinformatics, 1-15, DOI: 10.1007/s12021-018-9406-9.

3 Social Networks

Enron email communication network covers all the email communication within a dataset of around half million emails (`Enron.txt`). This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation. Nodes of the network are email addresses and if an address i sent at least one email to address j, the graph contains an undirected edge from i to j.

i) Create an undirected network with adjacent list. Each node can be represented by a list of its neighbors.

ii) Plot the degree distribution. You may use log to transfer the data to make the figure clearer, and you should submit the code of how to compute the degree distribution and the figure of distribution.

iii) Calculate the betweenness centrality for each node and return the top 100 nodes with highest betweenness centrality values.

Please refer to Brandes Algorithm and try to implement the algorithm by yourself. (Will get bonus points if you implement the Brandes algorithm.)

4 Information Diffusion

Given the `Enron` network constructed above, perform information diffusion experiments in the network.

i) Randomly select 5 nodes as initially infected, set the diffusion probability to be 0.1, apply *Independent Cascade Model* to simulate the diffusion process. Show the number of infections over time.

ii) Design a more appropriate diffusion probability by yourself, describe the reason why you design the diffusion probability in this way. Implement the diffusion process with *Independent Cascade Model* with the newly designed probability and show the number of infections over time.

Tips: How you can program probability distribution through random module. Assume that we have a discrete distribution $\{p_1, p_2, p_3\}$ to select objects x, y, z, in which $p_1+p_2+p_3 = 1$. Now we have the rule that with probability p_1, we select x; with probability p_2, we select y; with probability p_3, we select z. We can mimic the event of drawing an object as follows, we firstly transfer the distribution to a cumulative distribution $\{p_1, p1 + p2, 1\}$, then we generate a random number from a uniform distribution $v \sim U[0, 1]$. If $v \leq p_1$, we choose x; if $p_1 < v \leq p_1 + p_2$, we choose y; if $v > p1 + p_2$, we choose z.

5 Recommender Systems

The data contains information about TV shows (`TV-show`). More precisely, for 9985 users and 563 popular TV shows, we know if a given user watched a given show over a 3 months period. The files are:

- `user-shows.txt` This is the ratings matrix R, where each row corresponds to a user and each column corresponds to a TV show. $R_{ij} = 1$ if user i watched the show j over a period of three months. The columns are separated by a space.
- `shows.txt` This is a file containing the titles of the TV shows, in the same order as the columns of R.

We will compare the user-based and item-based collaborative filtering recommendations for the 500^{th} user of the dataset. Let's call him Alex.

In order to do so, we have erased the first 100 entries of Alex's row in the matrix, and replaced them by 0s. This means that we don?t know which of the first 100 shows Alex has watched. Based on Alex's behaviour on the other shows, we will give Alex recommendations on the first 100 shows. We will then see if our recommendations match what Alex had in fact watched.

Use **Cosine Similarity** to measure the similarity between two vectors. Implement user-based CF and item-based respectively and return the top-5 TV shows that Alex is most likely to watch.