IITM-CS6745 : Mining Massive Datasets — Release date:08 Oct, 2019

Programming Assignment #2 — Deadline: 8 Nov, 2019, 11:55 pm

- The goal of this assignment is to implement and experiment with Locality Sensitive Hashing (LSH).
- This is an individual assignment. Collaborations and discussions with others are strictly prohibited.
- You may use any Programming Language for the implementation.
- You have to turn in the well documented code along with a detailed report of the results of the experiment electronically in Moodle.
- Be precise for your explanations in the report. Unnecessary verbosity will be penalized.
- You have to check the Moodle discussion forum regularly for updates regarding the assignment.
- Read the submission instructions carefully before submitting.
- **Please start early.**

Data

The data for this assignment comes from the "Bag of Words" dataset at the UCI Machine Learning Repository. This dataset is actually four datasets; we'll use the one called enron. The dataset contains nearly 40,000 internal emails from the company that were released for the public record. You can get the dataset from here. You'll only need to look at the *readme.txt* file, which explains the layout, and the file *docword.enron.txt* which contains the actual data. Note that this file has been compressed via gzip, and so you'll need to use the command gunzip to uncompress it. The file has already converted all words to integers. The words themselves are unnecessary for this exercise, but if you're curious to see a list of them, they're in *vocab.enron.txt.*

Task (1) - Comparing Jaccard Similarity with Minhashing

The objective of this task is to compare the jaccard similarity of any two given documents from the dataset, calculated directly and by using minhashing. Write code that allows you to compute the Jaccard similarity between two particular documents of interest (as identified by their document id). You should then also write code to produce a signature matrix with a specified number of rows. Once you have produced the signature matrix, write code that allows you to use it to estimate the Jaccard similarity between two particular documents (again identified by id).

Questions

1. How does the jaccard similarity estimated using minhashing vary when the number of rows is varied?
2. How much do the calculated jaccard similarity and the estimated similarity vary?

Task (2) - Finding nearest neighbors

Brute-Force Method

Find the k nearest neighbors for a given document ID, using a simple brute-force approach of computing the exact Jaccard similarity between this document and the rest.

LSH Method

Implement an LSH approach for finding the k nearest neighbors for a given document ID by implementing the banding technique described in section 3.4.1 of the textbook.

Questions

3. How does the performance of LSH compare to brute-force, with respect to average similarity and running time?
4. How does this vary with dataset size, k, r, and number of rows in signature matrix?

Things to follow:

1. The code that you submit for both the tasks should allow the user to interact with the application by providing the document ID and the parameters.
2. Be sure to set seed for the random number generators.
3. Your report should show some plots to show how these variables affect the result, and some commentary by you explaining the results you see, and why they are occurring.

Submission Instructions

Submit a single tar/zip file containing the following files in the specified directory structure. Use the following naming convention: 'rollno_PA2.tar.gz'.

Note: '*run.py*' script in each code folder should run everything asked and display the results.

rollno_PA2

- **Code**
  - **q1**
    - *run.py*
    - other code and result files
  - ⋮
  - **q4**
    - *run.py*
    - other code and result files
- **Rollno-report.pdf**
- **README**