Scalable Data Mining (Autumn 2018)

Assignment 1 (Full Marks: 100)

Steps for Hadoop installation:

1. Run the following commands in your command line to install Hadoop -

```
$ wget http://apache.mirrors.tds.net/hadoop/common/hadoop-3.0.3/hadoop-3.0.3.tar.gz
$ tar -xzvf hadoop-3.0.3.tar.gz
```

2. Then follow the guidelines given in this link to setup Hadoop in your system:

https://www.digitalocean.com/community/tutorials/how-to-install-hadoop-in-stand-alone-mode-on-ubuntu-16-04

Instructions: Please submit your answers to the following questions as a write-up in a PDF file and your codes via Moodle.

You can download the datasets from the following drive link: https://drive.google.com/open?id=1Ye_23bD_dZ9pVLn04S2F47grgeEBmu3Q
Or use the files attached in Moodle.

Question 1 (Marks = 25+25)

Download the text corpus 'data_Q1.txt' from Moodle/drive link. This file contains the data in the following format: <Sentence ID> <Sentence text> in each line. Perform the following pre-processing on this file to generate the input file:

a. Remove the <Sentence ID> from each line such that it contains only the <Sentence text>
b. Remove all punctuations from <Sentence text> in each line.
c. Perform case-folding such that all words are in lower-case.

Write a MapReduce program in Hadoop to find the bigram count distribution on the text corpus contained in the input file. The output should be a text file containing a bigram followed by its count in each line in the following format:

<Bigram1> <Count>

<Bigram2> <Count>
...

Perform post-processing on this output file to answer the following questions:

1. How many unique bigrams are there?
2. List the top ten most frequent bigrams and their counts.
3. What fraction of all bigrams occurrences does the top ten bigrams account for? That is, what is the cumulative frequency of the top ten bigrams?
4. How many bigrams appear only once?

Question 2 (Marks = 25+25)

Download the file ‘data_Q2.txt’ from Moodle/drive link. This file contains the list of edges in the following format:

<Node 1> <Node 2>
…

denoting an edge per line.

Form the adjacency matrix A and generate the input file where each line denotes an entry in A in the following format:
<rowID> <colID> <value>
…

Implement a MapReduce program in Hadoop using the above input file to perform the following:

1) Perform matrix multiplication on A to compute A*A (denoting number of common friends for a node pair) where each entry (i,j) in A*A denotes the number of common neighbors (number of 2-hop paths) between node i and node j in the social network. Store the values of A*A in the following format in an intermediate text file:
<rowID> <colID> <value>
…

2) Apply another Map Reduce program to find pairwise cosine similarity for all node pairs based on rows of A*A i.e. similarity between node i and node j is the **cosine similarity** between **row**$_i$ and **row**$_j$ of A*A.

Here **row**$_i$ and **row**$_j$ denote the 2-hop path vectors for node i and node j respectively. The input to this step is the intermediate text file obtained in step 1 as Reducer output. Store the output file denoting the similarity of all node pairs in the format:
<rowID> <colID> <value> where <value> is the cosine similarity.
3) Perform post-processing on the output file to return the top-5 most similar connected node pairs.

In your write-up, please provide a description of how you are going to use MapReduce jobs to solve each problem. Don't write more than 3 to 4 sentences for this; we only want a very high-level description of your strategy to tackle the problems.

You will submit 2 files for each question in the following format:

1. Submit your code using the filename *RollNo_AssignmentNo_QuesNo.** where '*' can be .py or .java or .scala
2. Submit the output file using the filename *RollNo_AssignmentNo_QuesNo.txt*
3. Submit the write-up using the filename *RollNo_AssignmentNo_QuesNo.pdf*