CS 562 - Spring 2018
Homework 3
Due Date: Wednesday, April 25, 2018 at 11:59 PM.

Problem 1. (FastMap)

Using the **FastMap** algorithm to embed the following 4-dimensional points into 2D. Use the Euclidian distance (L_2 norm) as the distance between the points in the original 4-dimensional space. Plot the points on this 2D plane.

```
O1: 5.1,3.5,1.4,0.2
O2: 4.9,3.0,1.4,0.2
O3: 4.7,3.2,1.3,0.2
O4: 6.9,3.1,4.9,1.5
O5: 5.5,2.3,4.0,1.3
O6: 6.5,2.8,4.6,1.5
O7: 7.2,3.0,5.8,1.6
O8: 7.4,2.8,6.1,1.9
```

Problem 2. (SVD and LSI)

In this problem you have to use some existing tools and write some code in order to implement LSI for text documents. The text documents are in:

http://www.cs.bu.edu/faculty/gkollios/cs562s18/datasets/Text/

We explain all the steps in detail next.

1) The first step is to remove the stop words from all the documents. A list of possible stop words can be found here: "http://www.textfixer.com/resources/common-english-words.txt" in CSV format. You can use existing code or write your own code for this.

2) The second step is to use a stemmer to extract the stems (roots) of each word. A very popular stemmer is the Porter stemmer. You can find many stemmers here: "http://tartarus.org/martin/PorterStemmer/". You can use any one that you want. Hint: You should make slight modifications on the code. Do not print the results on the console. You could write the words to a file and each line is a word.

3) Write code to count the frequencies of non-stop words (stems) and construct a document-term matrix A1. For example:

	Term1	Term2	Term3	Term4
Doc1	4	1	0	0
Doc2	0	3	2	5
Doc3	3	1	4	4
Doc4	0	3	1	0
Doc5	1	2	0	4

Another representation is based on tf*idf representation. Create another matrix A2 that uses this representation.
You can also use the term-document representation. In that case, you have to use the appropriate embedding of the query to the LSI space.

4) Use SVD to decompose document-term matrices A1 and A2 and create the LSI representations. You can use python (numpy.linalg.svd or scipy.linalg.svd), Matlab (`svd`) or any other library that you want to compute SVD.

We will provide a dataset with a set of documents in text format and you will have to create the LSI representation of this dataset. Explain what is the meaning of each matrix in the SVD decomposition and if you find some important concepts in the dataset. We will provide some query examples and you should report which documents should be retrieved for these queries. You should also compare the results for A1 and A2 .

5) Explain the steps that you did and the results of your experiments in a report.

Hints and Suggestions:
Using Python: You can use either the Porter or the "Snowball" stemmer to extract the stems of the words.
You can read more at http://www.nltk.org and http://www.nltk.org/howto/stem.html.
You can also vectorize the dataset using this library as well and create the matrices A1 and A2.
Some useful code fragments are here:

```
from nltk.stem.snowball import SnowballStemmer
from nltk.tokenize import word_tokenize, sent_tokenize


stemmed_data = [" ".join(SnowballStemmer("english", ignore_stopwords
=True).stem(word)
         for sent in sent_tokenize(message)
        for word in word_tokenize(sent))
        for message in indata]
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer=TfidfVectorizer(stop_words='english',min_df=4,max_df=0.8)
dtm = vectorizer.fit_transform(indata)
```

Using Matlab:Matlab gives an easy way to compute SVD on a document-term matrix A.
(a) You can load the matrix from a CSV file by using the command ”csvread”. The help webpage is ”http://www.mathworks.com/help/matlab/ref/csvread.html”.
(b) Matlab includes SVD function, so you do not need to implement SVD by yourself. The function command is ”svd”. The help webpage is ”http://www.mathworks.com/help/matlab/ref/svd.html”.
(c) You can use the command: ”dlmwrite(filename,M)” to write the matrix M into a file ”filename”

Problem 3. (MapReduce and Pig)

In this problem you will use Hadoop MapReduce and Pig to implement simple graph algorithms. For the assignment, you need to install Hadoop and Pig on you computer and then run some of the examples that come together with the distribution. After getting familiar with the systems, you have to implement an algorithm to compute the number of triangles per node in a graph that will be provided to you.

1) Install Hadoop on your computer (Single Node Installation): You will find the general directions to download and install Hadoop here:
https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

Here are the directions for installing Hadoop on Windows:
http://wiki.apache.org/hadoop/Hadoop2OnWindows

Now, you can run some examples from the Hadoop distribution. For example, the following operations copy the unpacked conf directory to use as input and then find and displays every match of the given regular expression. Output is written to the given output directory.

```
 $ mkdir input
$ cp conf/*.xml input
$ bin/hadoop jar hadoop-examples-*.jar grep input output 'dfs[a-z.]+'
$ cat output/*
```

2) Install Apache Pig using the instructions from here:

https://pig.apache.org/docs/r0.7.0/setup.html

Run some examples on Local Mode using:

```
$ pig –x local
```

Now you can do the rest of the assignment.

3) Text processing: Run the WordCount example on some text documents. Try to understand how it works.

4) Graph Algorithms: Implement in Map Reduce the naïve and the improved Triangle Counting algorithms that we discussed in class to find the number of triangles incident in each node in the graph. Run the algorithms on the file that is provided to you in the assignments page. Also, count the total number of triangles in the graph.

http://www.cs.bu.edu/faculty/gkollios/cs562s18/datasets/Graph/

5) Implement the triangle counting algorithm in Pig. Here you need to use joins to generate triplets that have three nodes that have edges with each other.

6) Write a report on how you implemented the algorithms and explain the results.

Answers and Code Submission.

Put all your code for all problems in a .gzip or tar.gz file and submit it using gsubmit. Make sure that you put enough comments in your code so the graders can run your code. Use gsubmit to submit your answers as well.