Project 2 for CS525: Big Data Analytics - Fall 2013

Who are those tweeters, and what topics do they care most about ? ☺

Total Points: 100

Given Out: Tuesday, 29th Oct, 2013

Due Date: Thursday, 14th Nov, 2013

Teams: Project to be done in teams of three (your "Cluster" team) .

Project Overview

In this project, you will work with existing tools as well as create your own large-scale technology for data mining of textual data sources, more specifically, focusing on data clustering/grouping of tweets.

Project Submission

1. You will submit a single zip file containing your project results, including the Java programs, scripts, your project description, your experimental results and findings, via the **mywpi** system. Your report should explain each command you use, as well as document in detail the logic of the programs you wrote.
2. In addition, you need to submit in hardcopy **a signed statement** of the relative contribution of each of your team members in class on the due date. For instance, if each of the 3 of you have done the project independently, and then only at the end you pulled the best of the material together, you need to say so. If each of you have closely collaborated, please state that. Also, indicate how much and which of the work and amount of effort by each team member. All team members must agree to and sign your report at the end confirming to the division of labor indicated in your report.

Project Demonstration

Once completed, each team will schedule an appointment with the instructor to demonstrate their project. In addition, two or three teams will also be asked to provide a brief demonstration and discussion of their results in class to the class.

Project Description

Data Sets, Input Parameters and Output Options

Twitter is a social media system for broadcasting short messages (called tweets) to various channels. In this project, we will provide you with a (parameterized) twitter reader to enable you to easily collect data from twitter. We will generate one data set using this same twitter reader and provide this 'standard' data set to you. In addition, you should take this provided twitter reader and adjust the reader by selecting a second topic of interest to you, for example, politics, sports, medicine, or whatever. Again, rerun your solutions and show what turns up this time around. *You are encouraged (but not required) to employ standard techniques from information retrieval to clean the text messages using the twitter related software. If you do such pre-processing, describe the effect of this preprocessing on your final results.*

About Clustering Task: K-Means clustering is a popular algorithm that we have studied in class for clustering similar objects into *K* groups (clusters). It starts with an initial seed of K points (randomly chosen) as centers, and then the algorithm iteratively tries to enhance these centers. Typically, the algorithm terminates either when two consecutive iterations generate the same K centers (i.e., the output converged or is near conversion), or a maximum number (as indicated by parameter iterate-count) of iterations has been reached. It is no doubt that this is one of the simplest types of clustering available and for this reason it often is used and an easy one to study to understand basics of clustering. You can read up about K-Means at various sites, from most data mining text books to Wikipedia at :

http://en.wikipedia.org/wiki/K-means_clustering#Standard_algorithm

Input: We will provide with a reader that can extract live tweets of your own choosing by specifying keywords. This way you can and should customize this project to work on data that is of interest to you. We will also provide you with one pre-extracted data set using that same reader of your own choosing that we also expect you to work with. In addition, you can *optionally* look for other twitter data available on the internet, such as, the Emotion dataset at http://twittersentiment.appspot.com, and many others.
Beyond the data sets, you need to allow entry of the **parameter R** into your system to help control termination (R = # of rounds) of your iterative clustering tool as well as **parameter K** (K= number of clusters to produce). You should consider entry of the initial K seed data points matching parameter K. One way you could consider to do this is to work with a separate file, called the K-file, that will contain K initial seed points. But the choice is yours, as long as you can show that you can change the parameters K and R on the fly.

Output: There is a variety of output that you should produce:

a. First you should produce the final cluster centers as one output option.
b. The second output should be the content of all items in the same cluster (for example, by attaching a cluster id to each data item).
c. Clearly, for your larger data sets, option (b) may be too huge. So as third option, you would instead output a "description" of each cluster in the form of the top 10 tweets in the cluster or the top 10 keywords in each cluster, as appropriate.
d. Also, produce as result the measurements on the amount of time spend on each execution.

Task 1. Scalable K-Means Clustering the Mahout Way [40 Points]

Mahout is an open-source package that implements a variety of data mining and machine learning techniques on top of Hadoop, including K-Means as well as several other clustering methods. There is a command line interface that you can use to import text files, map them to feature vectors, perform the clustering, and then return results about the clusters.

1. Your task is to first familiarize yourself with the commands that Twitter and in particular Mahout have available to accomplish textual clustering, including data input format, parameters, and data output format. See below a list of the commands that we would expect you will find useful. All commands that you make use of as well as the final process that you used should be documented carefully in your project report.

2. Your main task is then to apply clustering on at least 2 different twitter data sets to produce "interesting" clusters (of size 100MB or later). This will require you to prepare the data for loading into Mahout. More importantly, please experiment with different parameter settings and then report on your findings. Your results should explain what results you found (displaying the actual cluster results and their description in terms of top 10 key words). Discuss how meaningful you could make the results by varying the different parameters. Consider extreme cases of K=2 to K=10,000, and so on. Similarly, also report on the execution times for different parameter settings.

Some References:

http://mahout.apache.org/
https://cwiki.apache.org/confluence/display/MAHOUT/K-Means+Clustering
https://dev.twitter.com/
http://en.wikipedia.org/wiki/Twitter

The virtual machine you had downloaded for the last project includes the needed platform for the project, namely, Mahout library (Version 0.7). As before, you can access the VMWare software available on the Zoo lab machines, if you do not want to use your own laptop.

Some commands you want to look at for your project include:

Below we list some key commands that you should be making yourself familiar with. You will need to be fully documenting each command that you use and its parameters.

a. Convert Text to Mahout Sequence files

```
mahout seqdirectory \
  -c UTF-8 \
  -i examples/textfiles/ \
  -o seqfiles
```

b. Generate tf-idf Vectors from Sequence files:

```
mahout seq2sparse \
  -i seqfiles/ \
  -o vectors/ \
  -ow -chunk 100 \
  -x 90 \
  -seq \
  -ml 50 \
  -n 2 \
  -nv
```

For above, you want to explore the effect of different parameters.
The parameter setting of -n 2 is said to be useful for cosine distance.
The parameter setting -x 90 means that if a token appears in 90% of the documents it is considered a stop-word.
The parameter setting -nv means that named vectors are returned and thus makes it easier for you to inspect the data files (numeric feature vectors wont' make much sense to you).

c. Generate Kmean Clusters from vectors:

```
mahout kmeans \
  -i vectors/tfidf-vectors/ \
  -c kmeans-centroids \
  -cl \
  -o kmeans-clusters \
  -k 20 \
  -ow \
  -x 10 \
  -dm org.apache.mahout.common.distance.CosineDistanceMeasure
```

For above, the parameter setting -x refers to the number of iterations.
The setting kmeans will put random seed vectors into the -c directory.
The setting -cl specified that at the end the actual documents (tweets) will be assigned to clusters.

Task 2. Scalable K-Means Clustering On Your Own [40 Points]

Write map-reduce job(s) to implement your own version of the K-Means clustering algorithm directly on top of Hadoop. Your system should terminate if either of these two conditions become true:

a) The K centers did not change over two consecutive iterations
b) The maximum number of iterations has been reached (parameter R for rounds)

Consider in your design that since the algorithm is iterative, you need your program that generates the map-reduce jobs to also control whether it should start another iteration or not. You are welcome to refine the above termination conditions by considering some delta by how much the center points are allowed to migrate in a given iteration without requiring another iteration to be kicked off. But this is optional.

Please develop various **increasingly sophisticated versions of your clustering strategy** as described below. You should document the key differences from one algorithm to the next, i.e., what changes were made at the mapper, at the reducer, at the number of mappers or reducers, or in the main control program. These algorithms should include:

1. A **single-iteration** only algorithm (you can accomplish that by setting R=1, if you prefer)
2. A basic **multi-iteration** algorithm that terminates after R iterations.
3. A more advanced algorithm that terminates after it converges (or at most after R iterations).
4. An even more advanced algorithm that makes use of **optimization** ideas discussed in class, including the use of combiners to optimize the transmission of data from the mapper to the reducer, and any other optimizations that come to your mind.
5. Further, please design **two variations** of each of your above 4 strategies as per below:
   (a) return only cluster centers along with instructions if convergence has been reached; versus
   (b) return at the very end of your algorithm the final clustered data points along with their cluster centers. For (b), it is up to you what is returned in between intermediate stages of your cluster iterations. Also, note that in task #3 when you utilize twitter data a further extension would be to return "descriptors" of each cluster.

Please document carefully each of your designs of the above solutions.

Experimentation:

- First apply your algorithms on a **very small numeric** data set that you generate yourself (make it a small toy one first for testing), and demonstrate to us that the results make sense. Do not worry about the performance of these runs. This could be as simple as a 2-dimenional data set where each object has an X and a Y integer value in the range from 0 to 100. Each point is in a separate line.

- Then utilize it a **large numeric** data set of your own choosing to demonstrate the performance differences among your alternative runs. Now you should scale the dataset to a size around 100MB or more. For that, you also would increase your domain values for each attribute to be in a larger range, such as, 0 to 10,000.

- Provide a description explaining your respective **design**. Describe the output, as well as relative **performance** of each of your algorithms. Explain whatever your findings may be.

Task 3. Comparison between Roll-your-own and using Mahout [20 Points]

- If not done yet under task 2, now extend your strategies proposed under task 2 to also function for the twitter data sets used in task 1.
    a. This on the one hand means that you need to consider how to convert your data into numeric feature vectors before you start up your clustering for the distance function to be an easier task.
    b. Second, you may want to reconsider the distance function that you use, if you wish.
    c. Third, you ideally would link the numeric output vectors with their corresponding textual tweet objects; so that at the very end you can produce the content of your clusters in a meaningful manner.
    d. And, better yet, the final more advanced task would be to produce descriptors for each of your clusters by providing summaries of them in the form of their frequent keywords or any other descriptors that make sense.

- Then compare the performance numbers of mahout solution versus your own technologies for the identical twitter data set and with each of the parameter settings being as similar as possible.

- Third very briefly compare the capabilities of your solution with those of mahout, in terms of ease of use, your effort, etc.

*****