CPSC 4330 Big Data Analytics
Winter 2025

Homework 4

Due: 10:55 am, Monday, March 10

In this homework, you will implement an iterative algorithm (k-means) in Spark to calculate k-means for a set of points.

Prepare the Data

Data scrubbing is a common part of the ETL (Extract, Transform, Load) process. In this step, you will run a python script "DeviceStatusETL.py" (posted on Canvas) to conduct data scrubbing. The script can process the file "devicestatus.txt" (posted on Canvas) to get it into a standardized format for later processing. The file "devicestatus.txt" contains data collected from mobile devices on Loudacre's network, including device ID, current status, location and so on. Because Loudacre previously acquired other mobile provider's networks, the data from different subnetworks has a different format. Note that the records in this file have different field delimiters: some use commas, some use pipe (|) and so on. The python script does the following things.

- Load the dataset
- Use the character at position 19 as the delimiter (since the 1st use of the delimiter is at position 19), parse the line, and filter out bad lines. Each line should have exactly 14 fields, so any line that does not have 14 fields is considered as a bad line.
- Extract the date (first field), manufacturer (second field) (note that the second field contains the device manufacturer and model name, e.g. Ronin S2, which is separated by a white space), device ID (third field), and latitude and longitude (13th and 14th fields respectively).
- Save the extracted data to common delimited text files on HDFS.

Run the script by following the steps below.

1. Download the file "devicestatus.txt" from Canvas and copy it to Docker.

2. On Docker, copy the file "devicestatus.txt" to HDFS.

3. Download the script "DeviceStatusETL.py" from Canvas and copy it to Docker. Run the python script to conduct data scrubbing. You can change the input and output directory if you want to use different directories.

```
#spark-submit  DeviceStatusETL.py    /devicestatus.txt     /devicestatus_etl
```

4. Examine the data in the output dataset. Note that the latitude and longitude are the 4th and 5th fields in the output file. Below are the first 2 lines of the output file.

```
2014-03-15:10:10:20,Sorrento,8cc3b47e-bd01-4482-b500-
  28f2342679af,33.6894754264,-117.543308253
2014-03-15:10:10:20,MeeToo,ef8c7564-0a1a-4650-a655-
  c8bbd5f8f943,37.4321088904,-121.485029632
```

Figure 1 sample data in the output file

Calculate k-means for device location

Write a spark application in python to implement K-means algorithm to calculate K-means for the device location (each location has a latitude and longitude) in the file that is prepared by the previous step. That is to say, the file in /devicestatus_etl on HDFS is the input of your program. Review Figure 1 to see the format of the data. **You cannot use K-means in MLib of Spark to solve the problem.**

In your code,

- Set the number of means (center points) K = 5.
- You also need a variable *convergeDist* to decide when the K-means calculation is done – when the amount the locations of the means changes between iterations is less than convergeDist. Set the variable = 0.1.
- To take a random sample of K location points as starting center points, you can use takeSample(), which is used to return a fixed-size sample subset of an RDD. For example, data.takeSample(False, 5, 34) takes a random sample of 5 location points from the RDD "data" as starting center points and it returns an array of length 5, where "False" means no replacement, and 34 is the value of seed. In your code, use the same values as the arguments (i.e. False, 5, 34) of this function to take 5 starting center points.
- When parsing the input file, only include known locations (that is, filter out (0, 0) locations).
- You need to persist an RDD if necessary. Which RDD needs to be persisted?

Your program should produce the following final K center points (K = 5), if you use the same settings as listed above (i.e. K = 5, *convergeDist* = 0.1, etc.). In the output, the 1st field is latitude, and the 2nd field is longitude.

```
[43.97544955558915, -121.40498164360471]

[34.49209866867575, -118.21258003843369]

[35.0852504610273, -112.57489358662798]

[38.17337679610569, -121.21451034445965]

[33.76035813785657, -116.56967965470706]
```

For this homework, you don't need to use AWS to run your program against a big dataset.

Note that the following output is also correct.

[43.97544955558931, -121.40498164360626]
[34.49209866867549, -118.2125800384333]
[35.085250461027556, -112.57489358662946]
[38.17337679610465, -121.2145103444581]
[33.760358137856706, -116.56967965470677]

Submission

Use Canvas to submit the following:

- Your Spark application (.py)