BIG DATA COMPUTING (Classes A and B, proff. Pietracaprina and Silvestri) 2021-2022

Home / My courses / BDC21/22 / Homework 3 / Assignment of Homework 3: DEADLINE June 13, 23.59pm

Assignment of Homework 3: DEADLINE June 13, 23.59pm

In this homework, you will run a Spark program on the CloudVeneto cluster. The core of the Spark program will be the implementation of **2-round coreset-based MapReduce algorithm for k-center with z outliers**, which works as follows: in **Round 1**, separately for each partition of the dataset, a *weighted coreset* of k+z+1 points is computed, where the weight assigned to a coreset point p is the number of points in the partition which are closest to p (ties broken arbitrarily); in **Round 2**, the L weighted coresets (one from each partition) are gathered into one weighted coreset of size (k+z+1)*L, and one reducer runs the sequential algorithm developed for Homework 2 (**SeqWeightedOutliers**) on this weighted coreset to extract the final solution. In the homework you will test the accuracy of the solution and the scalability of the various steps.

Using CloudVeneto

A brief description of the cluster available for the course, together with instructions on how to access the cluster and how to run your program on it are given in this User guide for the cluster on CloudVeneto.

Assignment

You must perform the following tasks.

TASK 1. Download the template (TemplateHW3.java for Java users, and TemplateHW3.py for Python users). The template contains a skeleton of the implementation of the 2-round MapReduce algorithm described above. Specifically, the template is structured as follows:

- Receives in input the following command-line (CLI) arguments:
  - **A path to a text file containing point set in Euclidean space**. Each line of the file contains, separated by commas, the coordinates of a point. Your program should make no assumptions on the number of dimensions!
  - **4 integers**: k (number of centers), z (number of outliers), and L (number of partitions).
- Reads the input points into and RDD of Vector called **inputPoints**, subdivided into L partitions, sets the Spark configuration, and prints various statistics.
- Runs a method **MR_kCenterOutliers** to compute the solution (i.e., a set of at most k centers) to the k-center problem with z outliers for the input dataset. The method implements the 2-round algorithm described. In Round 1 it extracts k+z+1 coreset points from each partition using method **kCenterFFT** which implements the Farthest-First Traversal algorithm, and compute the weights of the coreset points using method **computeWeights**. In Round 2, it collects the weighted coreset into a local data structure and runs method **SeqWeightedOutliers**, "recycled" from Homework 2, to extract and return the final set of centers (you must fill in this latter part).
- Computes the value of the objective function for the returned solution (i.e., the maximum distance of a point from its closest center, excluding the z largest distances), using method **computeObjective**.
- Prints the value of the objective function and the time taken by computeObjective.

TASK 2. Rename the template into **GxxxHW3.java** (or **GxxxHW3.py**), where xxx is your 3-digit group number, and **complete the code** as follows (*you will see the parts where you must add code clearly marked in the template*):

1. Insert the code for SeqWeightedOuliers from your Homework 2.
2. Complete Round 2 of MR_kCenterOutliers to extract and return the final solution. **IMPORTANT: you must run SeqWeightedOutliers on the weighted coreset using alpha=2**
3. Add suitable istructions to MR_kCenterOutliers, so to measure and print separately the time required by Round 1 and Round 2. Please be aware of the Spark's lazy evaluation.
4. Write the code for method computObjective. It is important that you keep in mind that the input dataset may be very large and that, in this case, any structure of the size of this dataset may not fit into local memory.

The output of your code should use the following OutputFormat. (Note that the lines "Initial guess", "Final Guess" and "Number of guesses" are those prinited by SeqWeightedOutliers, as in Homework 2).

TASK 3. Test and debug your program in local mode on your PC *to make sure that it runs correctly.* For this local test you can use the 16-point dataset testdataHW3.txt which you can download here and the datasets uber-small.csv and artificial9000.txt that you used also for Homework 2, available in this page.

TASK 4. Test your program on the cluster using the datasets which have been preloaded in the HDFS available in the cluster. Use various configurations of parameters and report your results using the tables given in this word file: TableHW3-Java.docx (for Java users) and TableHW3-Python.docx (for Python users).

WHEN USING THE CLUSTER, YOU MUST STRICTLY FOLLOW THESE RULES:

- **To avoid congestion, groups with even (resp., odd) group number must use the clusters in even (resp., odd) days.**
- **Do not run several instances of your program at once.**
- **Do not use more than 16 executors.**
- **Try your program on a smaller dataset first.**
- **Remember that if your program is stuck for more than 1 hour, its execution will be automatically stopped by the system.**

SUBMISSION INSTRUCTIONS. Each group must submit a **zipped folder GxxxHW3.zip**, where xxx is your group number. The folder must contain the program (**GxxxHW3.java** or **GxxxHW3.py**) and a file **GxxxHW3table.docx** with the aforementioned table. Only one student per group must do the submission using the link provided in the Homework3 section. Make sure that your code is free from compiling/run-time errors and that you comply with the specification, otherwise your grade will be penalized.

*If you have questions about the assignment, contact the teaching assistants (TAs) by email to bdc-course@dei.unipd.it . The subject of the email must be "**HW3 - Group xxx**", where xxx is your group number. If needed, a zoom meeting between the TAs and the group will be organized*.

Last modified: Wednesday, 25 May 2022, 5:53 PM

◄ Submission Homework 2

Jump to...

User guide for the cluster on CloudVeneto ►