INP7079233 - BIG DATA COMPUTING (Proff. A: Pietracaprina and F. Silvestri) 2022-2023

Assignment of Homework 3 (DEADLINE: June 18, 23.59)

In this homework, you will use the Spark Streaming API to devise a program which processes a stream of items and assesses experimentally the space-accuracy tradeoffs featured by the count sketch to estimate the individual frequencies of the items and the second moment F_2.

Spark streaming setting that will be used for the homework

For the homework, we created a server which generates a continuous stream of **integer items**. The server has been already activated on the machine **algo.dei.unipd.it** and emits the items as strings on **port 8888**. Your program will define a **Spark Streaming context** that accesses the stream through the method **socketTextStream** which transforms the input stream (coming from the specified machine and port number) into **DStream** (*Discretized Stream*) of batches of items arrived during a time interval whose duration is specified at the creation of the context. A method **foreachRDD** is then invoked to process the batches one after the other. Each batch is seen as an RDD and a set of RDD methods are available to process it. Typically, the processing of a batch entails the update of some data structures stored in the driver's local space (i.e., its *working memory*) which are needed to perform the required analysis. The beginning/end of the stream processing will be set by invoking **start/stop** methods from the context sc. For the homework, the stop command will be invoked after (approximately) 10M items have been read. The **threshold 10M** will be hardcoded as a constant in the program.

To learn more about Spark Streaming you may refer to the official Spark site. Relevant links are:

- Spark Streaming Programming Guide (full documentation)
- Transformations on Streams (list of transformations applicable to the RDDs of a DStream)

Running the program and template

Your program will be run in local mode, exactly as the one devised for Homework 1. The **master should be set to local[*]** (however, take notice that if you do not set the master it is also ok, since local[*] is the default master).

In order to see a concrete application of the above setting you can download the following **example program** which computes the exact number of distinct elements in the stream:

- (**Java version**) DistinctItemsExample.java
- (**Python version**) DistinctItemsExample.py (updated 29/05/2023 at 23.30).

We strongly encourage to use this program as a template for your homework.

TASK for HW3.

You must write a program **GxxxHW3.java** (for Java users) or **GxxxHW3.py** (for Python users), where xxx is your 3-digit group number (e.g., 004 or 045), which receives in input the following **6 command-line arguments (in the given order):**

- **An integer** D: the number of rows of the count sketch
- **An integer** W: the number of columns of the count sketch
- **An integer** *left*: the left endpoint of the interval of interest
- **An integer** *right*: the right endpoint of the interval of interest
- **An integer** K: the number of top frequent items of interest
- **An integer** *portExp*: the port number

The program must read the first (approximately) 10M items of the stream Σ generated from **machine algo.dei.unipd.it** at port *portExp* and compute the following statistics. Let R denote the interval $[left, right]$ and let Σ_R be the substream consisting of all items of Σ belonging to R. The program must compute

- A $D \times W$ count sketch for Σ_R. To this purpose, you can use the same family of hash functions used in Homeworks 1 and 2, namely ((ax+b) mod p) mod C, where p=8191, a is a random integer in [1,p-1] and b is a random integer in [0,p-1]. The value C depends on the range you want for the result.
- The exact frequencies of all distinct items of Σ_R
- The true second moment F_2 of Σ_R. To avoid large numbers, normalize F_2 by dividing it by $|\Sigma_R|^2$.
- The approximate second moment $\tilde{F}_2$ of Σ_R using count sketch, also normalized by dividing it by $|\Sigma_R|^2$.
- The average relative error of the frequency estimates provided by the count sketch where the average is computed over the items of $u \in \Sigma_R$ whose true frequency is $f_u \geq \phi(K)$, where $\phi(K)$ is the K-th largest frequency of the items of Σ_R. Recall that if $\tilde{f}_u$ is the estimated frequency for u, the relative error of is $|f_u - \tilde{f}_u|/f_u$.

The program should print:

- The input parameters provided as command-line arguments
- The lengths of the streams ($|\Sigma|$ and $|\Sigma_R|$)
- The number of distinct items in Σ_R
- The average relative error of the frequency estimates for the items of Σ_R with the top-K highest true frequencies
- (Only if $K \leq 20$) True and estimated frequencies of the items of Σ_R with the top-K highest true frequencies (no specific order required).

This file shows how to format your output. Make sure that your program complies with this format (*the link will be added soon*).

The program that you submit should run without requiring additional files. Test your program on your local or virtual machine using various configurations of parameters, and **report your results using the table given in** this word file (*the link will be added soon*).

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 Homework 3 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, 31 May 2023, 5:18 PM

◄ Submission form for HW2

Jump to...