Bishop’s University

CS 450 – Elements of Big Data

CS 550 – Big Data Management and Analytics

Assignment 1: Streaming Algorithms

1. Introduction

The objective of this assignment is to implement a novel streaming algorithm under the constraints of limited memory. Python version 3.7 or later should be used. Template code is provided for each part of the assignment. The template includes all the data science, machine learning, or statistics libraries that are necessary. The intention is for you to implement the algorithms we have gone over, and problem solve to best understand the concepts of this course and their practical application.

Within the templates, all provided method names and classes must be used as provided with the same parameters. However, you may also use additional methods to keep your code clean.

Additional approved libraries that are not in the template will be listed here (if any):

```
import random
from collections import defaultdict
import numpy as np #for numeric algebra and arrays
import math
```

Here, you will be fed a stream of integers representing yearly incomes of individuals (in 10s of thousands, e.g., 1 represents $10,000; 234 represents $2,340,000). Your goal is to summarize the stream in three ways: (a) the approximate distinct number of incomes seen, and (b) the median income, and (c) the most frequent value of income. You can assume the income data approximately follows a power law, the Pareto distribution. As memory you will only be allowed to store a 100 elements array.

Data. Two versions of the data are provided, (1) a small trial version with only 1000 integers to use while developing your method, and (2) a test that goes over 1 million integers to test your data on a larger dataset:

`trial_incomes.csv`
`test_incomes.csv.zip`

Template Code: You will work within a fixed template whereby you should only edit the methods for each task "`def task1ADistinctValues`", "`def task1BMedian`", "`def task1CMostFreqValue`"

Download the code here:

`Assignment1_streamingAlgorithm_CS450_CS550.py`

Steps for code familiarity:

- The code should run with "`python3 Assignment1_streamingAlgorithm_CS450_CS550.py trial_incomes.csv`" but produce bogus results. Test that it does.
- Look at main within the code. You will see that beyond some code to read the input file and set limits on memory (to constrain working with data as a stream). It iterates over elements in the stream and calls each of the three methods passing the value along. At each of the following interactions: $10, 10^2, 10^3, \ldots, 10^6$ it prints the current calculation from the method.
- Your job will be to edit those methods such that they return the approximate value, and only use the 100 elements array (technically, a deque object with a `maxsize` but it operates as an array) provided to them as a memory. Your 100 elements array may only contain ints or floats as values (i.e., you are not allowed to store dictionaries, other arrays, or any other objects/data structures as the values in this

array). During streaming, the current size of the array will be printed. It should remain < 8,000.

Of course, there are ways to work around the memory limitation. However, your goal is to implement approaches that work well even within this limitation.

2. Task I.A) Approximate the count of distinct incomes (20 points)

For this subtask, you must approximate the number of distinct incomes seen thus far in the stream. Use the *Flajolet-Martin algorithm* as described in class. You must use the median of means approach to average the estimates resulting from all hash functions. You should decide and justify the number of hash functions to use, although it must be < 100 since you may only store up to 100 elements at a time in memory.

3. Task I.B) Approximate the median of the incomes (20 points)

For this subtask, you must approximate the median of the incomes seen thus far – i.e. the income value, m, such that 50% of incomes seen thus far will be $<\ m$ and 50% will have been > m. Typically, approximating the median of streaming data without storing many values is quite difficult. However, here you can assume the data follows a Pareto type 1 distribution, and the following is true of such a distribution:

- The function that yields the probability that a value is < a given x (i.e. the cumulative distribution function, CDF) is:

  $$P(X < x) = 1 - \left(\frac{1}{x}\right)^{\alpha}$$

  where one can assume the minimum value is 1.

- can be estimated from the following function (which was derived from maximum likelihood estimation):

  $$\hat{\alpha} = \frac{n}{\sum_{i=1}^{n} \ln x_i}$$

4. Task I.C) Approximate the median of the incomes (20 points)

Theoretically, the "mode" (i.e., most frequent value) of a Pareto distribution should be its smallest possible value. However, practically speaking, real-world data rarely follow the theoretical distribution perfectly and you've been tasked more precisely with finding the most frequent income seen thus far in the stream. Propose an approximate solution for calculating the most frequently witnessed income within the restrictions of the streaming template (i.e. without using more than 1 array of size 100).

Hint: There is no perfect solution to this. This subtask is intended to push the boundaries of your thinking. Thus, you may be awarded points for creativity and demonstration of effort even if your approximate answers are not very accurate.

5. Submission

You must submit the *pdf* file of your report. In fact, it must contain the responses in form of outputs for the tasks I.A), I.B), and I.C. In addition, you must submit your implementation of `Assignment1_streamingAlgorithm_CS450_CS550.py`. ***Please, do not submit your assignment in .zip or .rar files.***

6. Useful external references

- NumPy documentation
- SciPy documentation
- scikit-learn
- Python Dictionaries
- Python programming language