Com S 435/535 Programming Assignment 4
600 Points
Due: Dec 12 , 11:59PM
Late Submission Nov 15, 11:59 (1% Penalty, Yes it is 1)

In this programming assignment, you will implement two parts of a search engine—computing page rank and creating an Inverted Index to allow ranked retrieval of documents related to a query. Note that the description of a programming assignment is not a linear narrative and may require multiple readings before things start to click. You are encouraged to consult instructor/Teaching Assistant for any questions/clarifications regarding the assignment.

For this assignment, you may work in groups of two.

In this PA, you will implement streaming algorithms. Your programs must process the data as a stream.

1 Streaming Algorithms

In this PA, you will implement streaming algorithms. Your programs must process the data as a stream.

1.1 Count Min Sketch

Design a class named `CMS` to implement count min sketch. This class will have following constructors and methods.

`CMS(float epsilon, float delta, ArrayList<Integer> s, float q, float r)` Process the array list s as a stream and build sufficient data structures (based on Count-Min Sketch) so that the methods below work properly. You may assume that $q \geq r + epsilon$

`approximateFrequency(int x)` Return the approximate value of f_x by consulting the data structures built by the constructor.

Given a multi set S consisting of N items. Recall the following definitions. A q-heavy hitter is the set of all elements x for which $f_x \geq qN$. A set L is called *approximate* $\langle q, r\rangle$ *Heavy Hitter*, if the following holds: For every $x \in L$, $f_x \geq qN$ and every x with $f_x < rN$ does not belong to L. Recall that count-min sketch can be used used to keep track of heavy hitters.
`approximateHH()` Returns an array of integers which is a $\langle q, r\rangle$ heavy hitter. Assume that $q \geq r+\epsilon$.

1.2 Count Sketch

Design a class named `CountSketch` to implement count sketch. This class will have following constructors and methods.

`CountSketch(float epsilon, float delta, ArrayList<Integer> s)` Process the array list s as a stream and build sufficient data structures (based on Count-Min Sketch) so that the methods below work properly.

`approximateFrequency(int x)` Return the approximate value of f_x by consulting the data structures built by the constructor.

1.3 F_2

Here you will design a streaming algorithm to estimate F_2 of a stream. Design a class named `AMS` that will have the following `static` method.
`secondFreqMoment(ArrayList<Integer> s, float epsilon, float delta)` Return an estimate for F_2 of the stream s by using AMS algorithm using ϵ, and δ as approximation and probability error parameters. You must use AMS algorithm.

1.4 Dimensionality Reduction

Recall from lectures and notes that AMS algorithm can also be viewed as an algorithm to reduce the dimension os a data set while approximately preserving the distances.

You are provided a class named `Vector` to represent vectors in (high) dimensional spaces. Use this class to implement dimensionality reduction.

Design a class named `AMSDimRed` that will have the following `static` method.

`ArrayList<Vector> reduceDim(ArrayList<Vector> inputVectors, float epsilon, float delta)`. Returns an array list obtained by reducing dimensionality using ϵ and δ as parameters. The i vector of the output must correspond to the ith vector of `inputVectors`.

1.5 Evaluation

Design experiments to evaluate the algorithms implemented.

- Design experiment to evaluate the performance of Count min Sketch algorithm. How well it approximates frequencies? Which frequencies (high or low) are well approximated? What is the quality of the Heavy Hitters returned by this algorithm ? Design similar experiments to evaluate count sketch.
- Design a experiment to compare performances of count and count min sketch algorithms. Given the same amount memory, which one is more accurate?
- Consider data set from PA2. Each document can be view as a point in a m- dimensional space, where m is the total number of terms in the document collection (after removal of stop words). Given a document d, the corresponding vector is $[f_1, f_2, \cdots f_m]$ where f_i is the number of times ith term appears in the document d. Run the dimensionality reduction in this data set. How well are L_2 distances (between pairs of documents) preserved? For what fraction of documents the distances are approximately preserved?

Prepare a report that describes your design experiments and your conclusions.

2 What to Submit

Please submit the following .java files and the report. Your report should be in pdf format. Please zip all .java files and the report, name the file `PA4YourUserID.zip`.

- CMS
- CountSketch
- AMS
- AMSDimRed

Only one submission per group please. Please do not submit .class files
Have Fun!