Assignment2 Report

Name: Yinghao Zhu
StudentID: 887402

1 Introduction

l_0 sampling is a fundamental sampling problem. l_0 sampler is used to uniformly get items with non-zero frequency from a data stream . In this report, the only added stream will be sampled, but it put more emphasis will be put on how to implement l_0 sampler on general stream that means there are items with negative frequency in the dynamic stream.

2 Dataset

In this section, the dataset that is used fr testing in this project will be introduced.

File	Description
dinner.txt	this file includes a only added stream, it is used to test l_0 sampler for only added stream.
dinner2.txt	This file includes a general stream. It is used to test my implementation of l_0 sampler for general stream.
oneSparseMore.txt oneSparseOne.txt oneSparseZero.txt	These three files are used for testing of 1-sparse algorithm.
sSparseMore.txt oneSparseRecover.txt sSparseZero.txt	These three files are used for testing of s-sparse algorithm.

Table 1: The dataset for this project

3 l_0 Sampling Process

In this section, the implementations of l_0 sampler non-turnstile and turnstile stream are introduced.

3.1 l_0 Sampling for Insert-only Stream

For an insert-only stream, a trick will be used. By using a k-wise(k is 9 in this project) hash function that is chose uniformly from a hash family. When an item of the stream comes, this k-wise hash function is used to get its hash value, the item with the smallest hash value will be returned. Due to the hash function is chose randomly every time, the probability of every item to be returned is $\frac{1}{n}$, n is the number of items in this stream.

3.2 l_0 Sampling for Turnstile Stream

While the sampler for a insert-only stream is introduced before, for a turnstile stream that not only allow insertion of data, deletion is also permitted, some approaches are needed.

3.2.1 One-Sparse Recovery

k-Sparse recovery is used to recovery elements from a vector, if the number of no-zero frequency of this vector n satisfy $1 \leq n \leq k$. For One-Sparse Recovery, we compute $\tau = \sum_i x_i \cdot q^i$. x_i is the frequency of this item, i is the item and q is a number that is selected randomly from $\{0, 1, ..., p\}$ where p is a big prime number(In this project, $p = 1073741789$). Finally, we test whether $\tau = F_1 q^{\frac{U}{F_1}} modp$. If succeed, the non-zero frequency item will be returned.

3.2.2 S-Sparse Recovery

For s-sparse recovery, the one-sparse approach will be exploited. There is a 2-dimensional table, it has 2s columns and $\log_2 \frac{s}{\delta}$ rows. For each row, there is a 2-independent hash function selected from hash family. Then, for every coming item we use hash function to hash it into a specific bucket of every row. If the stream s-sparse, the probability that just one coordinate lands in a particular bucket is about 1/2. For these rows, a given items A never along is $(\frac{1}{2})^{\log_2 \frac{s}{\delta}} = \frac{\delta}{s}$. There is s non-zero coordinates, so the probability of failure is δ. Finally, every row of this table, if one row has the most one-sparse bucket n with $1 \leq n \leq k$, these buckets of it will be recovered.

3.2.3 l_0 Sampling for Turnstile Stream

To implement l_0 sampling on a turnstile stream, the stream is divided into nested subsets S. S_j contains $n/2^j$ coordinates and $0 \leq j \leq \log_2 n$. For every coming item, according its hash value that is computed by a k-wise (it is 9 int this project) hash function that maps from n to n^2. It will be put in all subset j, if its hash value less than $\frac{n^2}{2^j}$.

When the stream is finished, I attempt s-sparse recovery with $s = \lceil 4 \log_2 \frac{1}{\delta} \rceil$ for every subset In this project, δ is 1024 . Finally, from the largest-size subset satisfied s-sparse, the item which has the minimum hash value will be selected by the k-wise hash function.

4 Testing Methodology

I used the testing dataset to test my implementation. For l_0 sampling on insertion-only stream, the "dinner.txt" is used, the implementation has been tested for 1 million times and the output of every test was recorded to check whether the proportion of all items appear in the output is same.

For turnstile stream, s-sparse recovery was tested by testing data for 10,000 times and there was no failure. Then "dinner2.txt" was used to test l_0 sampler for 1000,000 times, same to insertion-only stream, I recorded the output and observed the proportion of all items.

5 Output of l_0 Sampler

5.1 Result of l_0 Sampler for Insertion-only Stream

By implementing testing methodology, the result is as Figure1. It is obvious that l_0 sampler for insertion-only stream works well.

5.2 Result of l_0 Sampler for Turnstile Stream

The testing result of l_0 Sampler for Turnstile Stream is Figure2. It shows for 40 items, they are uniformly sampled.

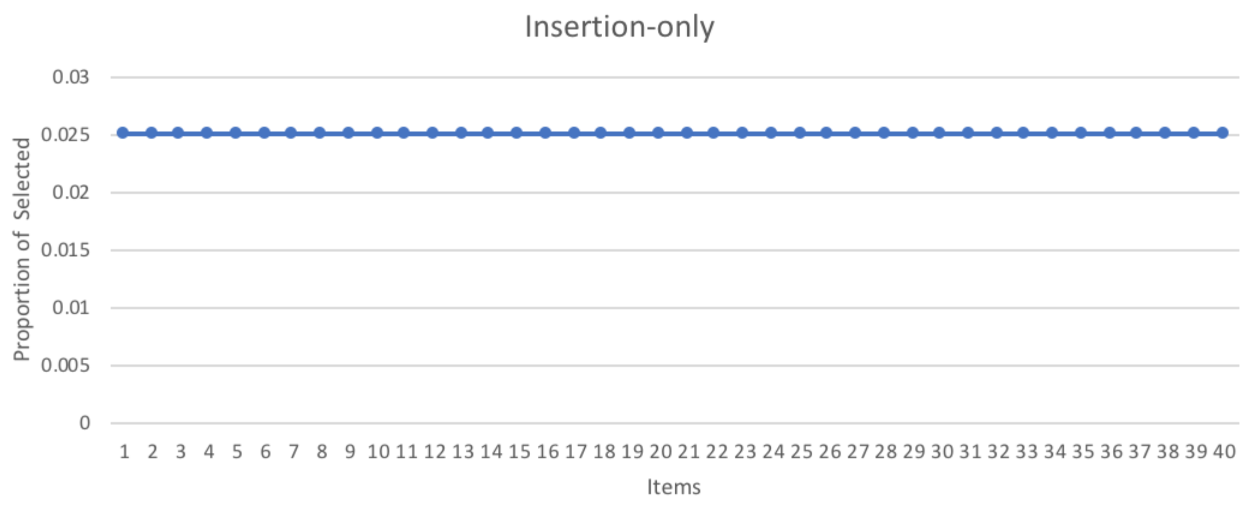

Figure 1: The result of l_0 sampler for Insertion-only stream

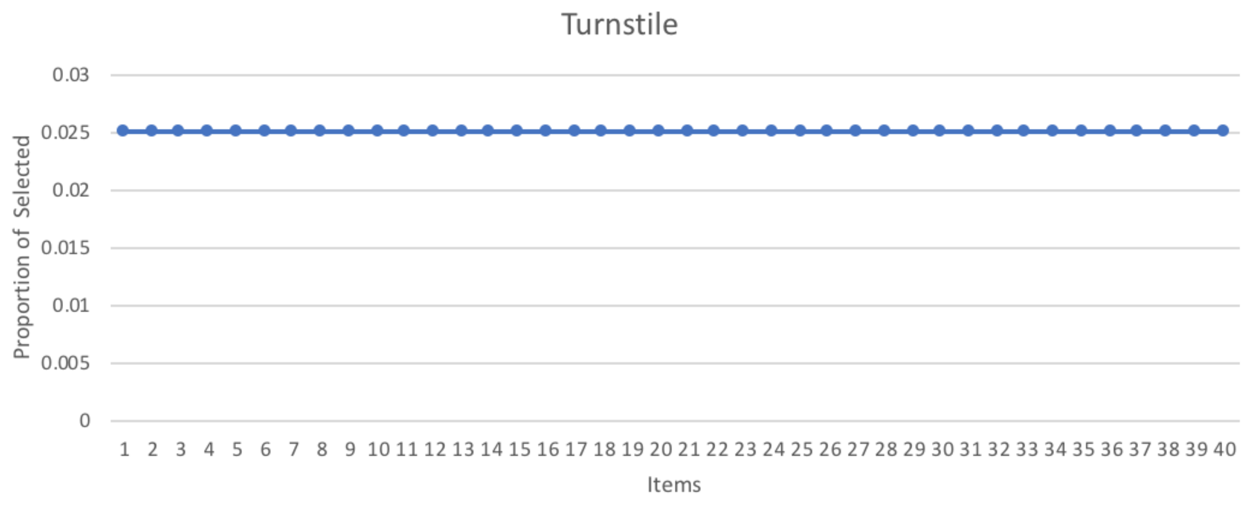

Figure 2: The result of l_0 sampler for turnstile stream

6 Discussion

The l_0 sampler I built in this project cam sample nearly-uniform from a general data stream. It can help to improve the time complexity with cost of less storage space while uniformly select non-zero items will high accuracy. It It has a variety of applications within data analysis, computational geometry and graph algorithms.

7 Conclusion

From the results that are shown in section 5, it is clear that my l_0 samplers works well with a nearly-uniformly output.A conclusion can be drawn that the l_0 samplers I designed in this project can uniformly sample items with non-zero frequency.