Finding Interesting Items in Data Streams

Anonymous Author(s)

ABSTRACT

THis is the abstract.

1 INTRODUCTION

1.1 Background and Motivation

In the study of Big Data, data stream is often the representation form of data. Data stream contains concecutive items whose time of appearances might be zero, one or more. However, under most circumstances, only a small section of data is cared in researches. For instance, people are likely to cared about the items with the highest frequencies. There is another likelihood that the most persistent items are the ones to study. For another example, in security, people want to find the items who are super-spreaders.

Big data is often presented as data streams. A data stream is made of continously arriving items, and the number of appearances of each item can be one or more. Although there are a great number of items, people are often interested in a very limited number of items with some special characteristics, such as frequent items [1–4], persistent items [5, 6], or super-spreader [**?**]. In this paper, we call them interesting items for convenience.

It is signfiicant and chanllenging to find interesint items, because the speed of data streams is often very high. It is often impossible to find interesting items without error. Beside, small and controllable error is often acceptable in practice. Therefore, sketch, a probabilitic data structure obtains wide attention and interests in recent years [7**?**].

1.2 Prior Art and Their Limiations

Existing sketches focus on only one specific interest, such as frequency or persistency. For different interests, existing soltions use different data structures. For example, to find frequent items, there are two kinds of solutions: 1) sketch based solutions; 2) counter based solutions. ... ... In constrast, we aim at designing a generic framework, which can be used for finding any interesting items with high speed and high accuracy at the same time.

1.3 Our Solution

1.4 Key Contribution

This paper makes the following key contributions:

- aaa
- bbb

Roadmap: Section 2 surveys the related work. We present our algorithms in Section 3. We optimize the algorithm in Section 4. We apply the algorithm to three tasks in Section 5. We derive proofs of our algorithms in Section 6. We show the experimental result in Section 7. We conclude this paper in Section 8.

2 RELATED WORK

3 BASIC ALGORITHM

4 OPTIMIZATION

5 APPLICATION

6 PROOF

7 EXPERIMENTS

8 CONCLUSION

REFERENCES

[1] Lukasz Golab, David DeHaan, Erik D Demaine, Alejandro Lopez-Ortiz, and J Ian Munro. Identifying frequent items in sliding windows over on-line packet streams. In *Proc. ACM IMC*, pages 173–178. ACM, 2003.

[2] Richard M Karp, Scott Shenker, and Christos H Papadimitriou. A simple algorithm for finding frequent elements in streams and bags. *ACM Transactions on Database Systems (TODS)*, 28(1):51–55, 2003.

[3] Nishad Manerikar and Themis Palpanas. Frequent items in streaming data: An experimental evaluation of the state-of-the-art. *Data & Knowledge Engineering*, 68(4):415–430, 2009.

[4] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *Automata, Languages and Programming*. Springer, 2002.

[5] Zhewei Wei, Ge Luo, Ke Yi, Xiaoyong Du, and Ji-Rong Wen. Persistent data sketching. In *Proc. ACM SIGMOD*, pages 795–810. ACM, 2015.

[6] Haipeng Dai, Muhammad Shahzad, Alex X Liu, and Yuankun Zhong. Finding persistent items in data streams. *Proceedings of the VLDB Endowment*, 10(4):289–300, 2016.

[7] Pratanu Roy, Arijit Khan, and Gustavo Alonso. Augmented sketch: Faster and more accurate stream processing. In *Proc. ACM SIGMOD*, pages 1449–1463, 2016.