Modern Dynamic Data Structures

Monika Henzinger
University of Vienna, Department of Computer Science, Vienna, Austria

Abstract
We give an overview of differentially private dynamic data structure, aka differentially private algorithms under continual release.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Differential privacy, data structures

Digital Object Identifier 10.4230/LIPIcs.MFCS.2022.2

Category Invited Talk

Funding This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant agreement No. 101019564 “The Design of Modern Fully Dynamic Data Structures (MoDynStruct)” and from the Austrian Science Fund (FWF) project “Fast Algorithms for a Reactive Network Layer (ReactNet)”, P 33775-N, with additional funding from the netidee SCIENCE Stiftung, 2020–2024

1 Extended Abstract

In the past the main goal when designing data structures was to achieve optimal time per operation and optimal space. However, in recent years new applications have lead to new requirements for data structures, such as differential privacy or fairness.

In this talk I will limit myself to the first topic, namely differential privacy. Since its invention in 2006 by Dwork, McSherry, Nissim, and Smith [3] differentially private algorithms and (static) data structures have been designed for many combinatorial problems (see e.g. [5] for a book on the topic). However, very little work has been done for dynamic data structures. A dynamic data structure is a data structure that supports not only query operations to the stored data, but also update operations, such as insertions and/or deletions. In the differentially privacy research community such data structures are frequently called data structures in the continual release (or continual observation) model.

The problem of binary counting the number of 1s in a binary sequence is equivalent to a dynamic data structure that supports the AppendBit operation and that outputs the number of 1s in the current sequence after each AppendBit operation. This problem has been well-studied in the differentially private setting [4, 1, 8, 2, 7], including a version that outputs a weighted average of the bits in the sequence so far [7]. Another extension of this problem leads to the MaxSum and the SumSelect problem: Assume the input is a sequence of \(d\)-dimensional binary vectors such that the \(t\)-th vector is denoted by \(b_t\). The goal of the SumSelect problem is to output the value of the coordinate \(i\) such that \(i = \text{argmax}_i \sum_t b_t[i]\), the goal of MaxSum is to output max, \(\sum_i b_t[i]\). These problems were studied in the continual release model in [9]: Differentially private partially dynamic graph algorithms were also analyzed in the continual release setting [10, 6].

We will describe the algorithm of [7] in detail and explain why it is superior to the previous solutions for binary counting under continual observation.


