Information and Computation 243 (2015) 145–165

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Arthur–Merlin streaming complexity [☆]

Tom Gur [a,*], Ran Raz [a,b,*]

[a] Weizmann Institute of Science, Israel
[b] Institute for Advanced Study, Princeton, NJ, USA

ARTICLE INFO

Article history:
Received 31 July 2013
Available online 12 December 2014

Keywords:
Probabilistic proof systems
Data streams
Communication complexity

ABSTRACT

We study the power of Arthur–Merlin probabilistic proof systems in the data stream model. We show a canonical $\mathcal{AM}$ streaming algorithm for a class of data stream problems. The algorithm offers a tradeoff between the length of the proof and the space complexity that is needed to verify it.

As an application, we give an $\mathcal{AM}$ streaming algorithm for the *Distinct Elements* problem. Given a data stream of length m over alphabet of size n, the algorithm uses $\tilde{O}(s)$ space and a proof of size $\tilde{O}(w)$, for every s, w such that $s \cdot w \geq n$ (where $\tilde{O}$ hides a polylog(m, n) factor). We also prove a lower bound, showing that every $\mathcal{MA}$ streaming algorithm for the *Distinct Elements* problem that uses s bits of space and a proof of size w, satisfies $s \cdot w = \Omega(n)$. Furthermore, the lower bound also holds for approximating the number of distinct elements within a multiplicative factor of $1 \pm 1/\sqrt{n}$.

As a part of the proof of the lower bound for the *Distinct Elements* problem, we show a new lower bound of $\Omega(\sqrt{n})$ on the $\mathcal{MA}$ communication complexity of the *Gap Hamming Distance* problem, and prove its tightness.

1. Introduction

The data stream computational model is an abstraction commonly used for algorithms that process network traffic using sublinear space [1–3]. In the settings of this model, we have an algorithm that gets a sequence of elements (typically, each element is an integer) as input. This sequence of elements is called a *data stream* and is usually denoted by $\sigma = (a_1, \ldots, a_m)$; where a_1 is the first element, a_2 is the second element, and so forth. The algorithm receives its input (a data stream) element-by-element. After it sees each a_i, it no longer has access to elements with index that is smaller than i. The algorithm is required to compute a function of the data stream, using as little space as possible.

Among the most fundamental problems in the data stream model is the problem of *Distinct Elements*, i.e., the problem of computing the number of distinct elements in a given data stream. The problem has been studied extensively in the last two decades (see, for example, [1,2,4]). Its significance stems both from the vast variety of applications that it spans (covering IP routing, database operations and text compression; e.g., [5,1,6]), and due to the theoretical insight that it gives on the nature of computation in the data stream model.

[☆] Research supported by an Israel Science Foundation grant, by the I-CORE Program of the Planning and Budgeting Committee and the Israel Science Foundation, and by NSF grant numbers CCF-0832797, DMS-0835373.
[*] Corresponding author.
E-mail addresses: tom.gur@weizmann.ac.il (T. Gur), ran.raz@weizmann.ac.il (R. Raz).

http://dx.doi.org/10.1016/j.ic.2014.12.011
0890-5401/© 2014 Elsevier Inc. All rights reserved.

Alon et al. [1] have shown a lower bound of $\Omega(n)$ (where n is the size of the alphabet from which the elements are taken) on the streaming complexity of the computation of the *exact* number of distinct elements in a sufficiently long data stream (i.e., where the length of the data stream is at least proportional to n). The goal of reducing the space complexity of the *Distinct Elements* problem has led to a long line of research on approximation algorithms for the problem, starting with the seminal paper [7] by Flajolet and Martin. Recently, Kane et al. [4] gave the first optimal approximation algorithm for estimating the number of distinct elements in a data stream; for a data stream with alphabet of size n, given $\epsilon > 0$ their algorithm computes a $(1 \pm \epsilon)$ multiplicative approximation using $O(\epsilon^{-2} + \log n)$ bits of space, with 2/3 success probability.

A natural approach for reducing the space complexity of streaming algorithms, *without* settling on an approximation, is by considering a probabilistic proof system. Chakrabarti et al. [3] have shown *data stream with annotations* algorithms for several data stream problems, using a probabilistic proof system that is very similar to $\mathcal{MA}$. This line of work continued in [8], wherein a probabilistic proof system was used in order to reduce the streaming complexity of numerous graph problems. In a subsequent work [9], Cormode et al. provided a practical instantiation of one of the most efficient general-purpose constructions of an interactive proof for arbitrary computations, due to Goldwasser et al. [10].

In this work, we study the power of Arthur–Merlin probabilistic proof systems in the data stream model. We show a canonical $\mathcal{AM}$ streaming algorithm for a class of data stream problems. The algorithm offers a tradeoff between the length of the proof and the space complexity that is needed to verify it. We show that the problem of *Distinct Elements* falls within the class of problems that our canonical algorithm can handle. Thus, we give an $\mathcal{AM}$ streaming algorithm for the *Distinct Elements* problem. Given a data stream of length m over alphabet of size n, the algorithm uses $\tilde{O}(s)$ space and a proof of size $\tilde{O}(w)$, for every s, w such that $s \cdot w \geq n$ (where throughout this paper, $\tilde{O}$ hides a polylog(m,n) factor).

In addition, we give a lower bound on the $\mathcal{MA}$ streaming complexity of *approximating* the *Distinct Elements* problem, which almost matches numerically the upper bound above. Loosely speaking, we show that every streaming algorithm for approximating *Distinct Elements* within a multiplicative factor of $1 \pm 1/\sqrt{n}$ that uses s bits of space and a proof of size w must satisfy $s + w = \Omega(\sqrt{n})$. (We note that a lower bound on the $\mathcal{MA}$ streaming complexity of computing the *exact* number of distinct elements in a stream can be obtained by combining the known $\mathcal{MA}$ lower bound on *set-disjointness* [11] and the known reduction from set-disjointness to *Distinct Elements* [2].) Our lower bound for *Distinct Elements* relies on a new lower bound that we prove on the $\mathcal{MA}$ communication complexity of the *Gap Hamming Distance* problem.

1.1. *Arthur–Merlin probabilistic proof systems*

An $\mathcal{MA}$ (Merlin–Arthur) proof is a probabilistic extension of the notion of proof in complexity theory. Proofs of this type are commonly described as an interaction between two players, usually referred to as Merlin and Arthur. We think of Merlin as an omniscient prover, and of Arthur as a computationally bounded verifier. Merlin is supposed to send Arthur a valid proof for the correctness of a certain statement. After seeing both the input and Merlin's proof, with high probability Arthur can verify a valid proof for a correct statement, and reject every possible alleged proof for a wrong statement.

An $\mathcal{AM}$ proof is defined almost the same as an $\mathcal{MA}$ proof, except that in $\mathcal{AM}$ proof systems we assume that both the prover and the verifier have access to a common source of randomness (alternatively, $\mathcal{AM}$ proof systems can be described as $\mathcal{MA}$ proof systems that start with an extra round, wherein Arthur sends Merlin a random string). We remark that, by definition, $\mathcal{AM}$ proofs are at least as powerful as $\mathcal{MA}$ proofs.

The notion of $\mathcal{AM}$ and $\mathcal{MA}$ proof systems can be extended to many computational models. In this work we consider both the communication complexity analogue of $\mathcal{MA}$, wherein Alice and Bob receive a proof that they use in order to save communication, and the data stream analogues of $\mathcal{MA}$ and $\mathcal{AM}$, wherein the data stream algorithm receives a proof and uses it in order to reduce the required resources for solving a data stream problem.

Recently, probabilistic proof systems for streaming algorithms have been used to provide an abstraction of the notion of *delegation of computation* to a cloud (see [8,9,12]). In the context of cloud computing, a common scenario is one where a user receives or generates a massive amount of data, which he cannot afford to store locally. The user can stream the data he receives to the cloud, keeping only a short certificate of the data he streamed. Later, when the user wants to calculate a function of that data, the cloud can perform the calculations and send the result to the user. However, the user cannot automatically trust the cloud (as an error could occur during the computation, or the service provider might not be honest). Thus the user would like to use the short certificate that he saved in order to verify the answer that he gets from the cloud.

1.2. *Communication complexity and the* Gap Hamming Distance *problem*

Communication complexity is a central model in computational complexity. In its basic setup, we have two computationally unbounded players, Alice and Bob, holding (respectively) binary strings x, y of length n each. The players need to compute a function of both of the inputs, using the least amount of communication between them.

In this work we examine the well known communication complexity problem of *Gap Hamming Distance* (GHD), wherein each of the two parties gets an n bit binary string, and together the parties need to tell whether the Hamming distance of the strings is larger than $\frac{n}{2} + \sqrt{n}$ or smaller than $\frac{n}{2} - \sqrt{n}$ (assuming that one of the possibilities occurs). In [13] a tight linear lower bound was proven on the communication complexity of a randomized communication complexity protocol for GHD. Following [13], a couple of other proofs [14,15] were given for the aforementioned lower bound. Relying on [15,16,11], in this work we give a tight lower bound of $\Omega(\sqrt{n})$ on the $\mathcal{MA}$ communication complexity of GHD.