Brief Announcement: Erasure-Resilience Versus Tolerance to Errors

Sofya Raskhodnikova
Boston University, Boston, USA
sofya@bu.edu

Nithin Varma
Boston University, Boston, USA
nvarma@bu.edu

Abstract

We describe work in progress on providing a separation between erasure-resilient and tolerant property testing. Specifically, we are able to exhibit a property which is testable (with the number of queries independent of the length of the input) in the presence of erasures, but is not testable tolerantly.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Property testing, erasures, tolerance to errors, model separation

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.452

Category Brief Announcement

Funding This material is based upon work supported by the National Science Foundation under Grant No. CCF-142297.

1 Description

In this brief announcement, we describe our recent investigation of the effects of adversarial corruption to inputs on the complexity of sublinear-time algorithms. Input corruption occurs either in the form of errors (when some values get changed) or in the form of erasures (when some values go missing). Understanding the relative difficulty of designing algorithms that work in the presence of different forms of corruption is a problem of fundamental importance. It is with this motivation in mind that property testing [5, 7], one of the most widely studied models of sublinear-time algorithms, was generalized to erasure-resilient testing [3] and (error) tolerant testing [6].

Erasure-resilient property testing falls between (standard) property testing and tolerant testing. Specifically, an erasure-resilient tester for a property, in the special case when no erasures occur, is a standard tester for this property. Also, a tolerant tester for a property implies the existence of an erasure-resilient tester with comparable parameters for the same property. Dixit, Raskhodnikova, Thakurta and Varma [3] separate standard and erasure-resilient testing by describing a property that is *easy* to test in the standard model and *hard* to test in the erasure-resilient model. Their separation is based on an earlier result by Fischer and Fortnow [4] that separates standard property testing from tolerant property testing in the same sense. Their main tool is PCPs of proximity (also known as assignment testers) defined by Ben-Sasson, Goldreich, Harsha, Sudan and Vadhan [1] and by Dinur and Reingold [2]. Dixit et al. [3] asked whether it is possible to obtain a separation between

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Don Sannella; Article No. 452; pp. 452:1–452:3
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

erasure-resilient and tolerant testing. Here, we announce such a separation. Specifically, we are able to describe a property testable in the erasure-resilient model with the query complexity independent of the input size, but for which the query complexity of tolerant testing grows with the input size.

1.1 Erasure-Resilient and Tolerant Testing: Definitions

We now describe the erasure-resilient and tolerant models of testing. A *property* $\mathcal{P}$ is a set of strings. A string is α*-erased* for $\alpha \in [0,1)$ if at most an α fraction of its values are erasures (denoted by $\perp$). A *completion* of an α-erased string $x \in \{0,1,\perp\}^n$ is a string $y \in \{0,1\}^n$ that agrees with x on all the positions where x is nonerased. An α*-erasure-resilient* ε*-tester* [3] for a property $\mathcal{P}$ is a randomized algorithm that, given parameters $\alpha \in [0,1), \varepsilon \in (0,1)$ and oracle access to an α-erased string x, accepts with probability at least $2/3$ if x has a completion in $\mathcal{P}$ and rejects with probability at least $2/3$ if, in every completion of x, at least an ε fraction of the **nonerased** positions has to be changed to get a string in $\mathcal{P}$. The property $\mathcal{P}$ is α*-erasure-resiliently* ε*-testable* if there exists an α-erasure-resilient ε-tester for $\mathcal{P}$ with query complexity that depends only on the parameters α and ε (but not on the length of the input string).

A string $x \in \{0,1\}^n$ is ε'-far (α-close) from (to, respectively) a property $\mathcal{P}$, if the normalized Hamming distance of x from $\mathcal{P}$ is at least ε' (at most α, respectively). An (α,ε')-*tolerant tester* [6] for $\mathcal{P}$ is a randomized algorithm that, given parameters $\alpha \in (0,1), \varepsilon' \in (\alpha,1)$ and oracle access to a string x, accepts with probability at least $\frac{2}{3}$, if x is α-close to $\mathcal{P}$ and rejects with probability at least $\frac{2}{3}$, if x is ε'-far from $\mathcal{P}$. The property $\mathcal{P}$ is (α,ε')*-tolerantly testable* if there exists an (α,ε')-tolerant tester for $\mathcal{P}$ with query complexity that depends only on the parameters α and ε' (but not on the length of the input string).

1.2 Comparison of parameters

We remark that, while comparing the above two models, it is appropriate to compare $(\alpha, \alpha+\varepsilon(1-\alpha))$-tolerant testing of a property $\mathcal{P}$ with α-erasure-resilient ε-testing of $\mathcal{P}$ for the same values of α and ε. The parameter α in both the models is an upper bound on the fraction of corruptions (erasures, or errors) that an adversary can make to an input. An α-erasure-resilient ε-tester rejects with probability at least $\frac{2}{3}$ if, for for every way of completing an input string, one needs to change at least an ε fraction of the remaining part of the input to make it satisfy $\mathcal{P}$. Similarly, an $(\alpha, \alpha+\varepsilon(1-\alpha))$-tolerant tester rejects with probability at least $\frac{2}{3}$ if, for every way of *correcting* α fraction of the input values, one needs to change at least an ε fraction of the remaining $(1-\alpha)$ fraction of the input to make it satisfy $\mathcal{P}$.

1.3 Our Results

Our main contribution is the following theorem which states that there exists a property that is erasure-resiliently testable and is not tolerantly testable. This proves that tolerant testing is, in general, a harder problem than erasure-resilient testing.

▶ **Theorem 1** (Main Theorem). *There exists a property* $\mathcal{P}$ *and constants* $\varepsilon, \alpha \in (0,1)$ *such that*

- $\mathcal{P}$ *is* α*-erasure-resiliently* ε*-testable;*
- $\mathcal{P}$ *is not* $(\alpha, \alpha+\varepsilon(1-\alpha))$*-tolerantly testable.*

2 Conclusions

To summarize, we solve an open question proposed by Dixit et al. [3] and prove that tolerant testing is harder than erasure-resilient testing.

References

1. Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.
2. Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006.
3. Kashyap Dixit, Sofya Raskhodnikova, Abhradeep Thakurta, and Nithin Varma. Erasure-resilient property testing. *SIAM Journal on Computing*, 47(2):295–329, 2018.
4. Eldar Fischer and Lance Fortnow. Tolerant versus intolerant testing for boolean properties. *Theory of Computing*, 2(9):173–183, 2006.
5. Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.
6. Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, 72(6):1012–1042, 2006.
7. Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.