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

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 erasure-resilient and tolerant testing. Here, we announce such a separation.
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 \( P \) is a set of strings. A string is \( \alpha \)-erased for \( \alpha \in [0,1) \) if at most an \( \alpha \) fraction of its values are erasures (denoted by \( \perp \)). A completion of an \( \alpha \)-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 \( \alpha \)-erasure-resilient \( \varepsilon \)-tester [3] for a property \( P \) is a randomized algorithm that, given parameters \( \alpha \in [0,1), \varepsilon \in (0,1) \) and oracle access to a string \( x \), accepts with probability at least \( 2/3 \) if \( x \) has a completion in \( P \) and rejects with probability at least \( 2/3 \) if, in every completion of \( x \), at least an \( \varepsilon \) fraction of the nonerased positions has to be changed to get a string in \( P \). The property \( P \) is \( \alpha \)-erasure-resiliently \( \varepsilon \)-testable if there exists an \( \alpha \)-erasure-resilient \( \varepsilon \)-tester for \( P \) with query complexity that depends only on the parameters \( \alpha \) and \( \varepsilon \) (but not on the length of the input string).

A string \( x \in \{0,1\}^n \) is \( \varepsilon' \)-far (\( \alpha \)-close) from (to, respectively) a property \( P \), if the normalized Hamming distance of \( x \) from \( P \) is at least \( \varepsilon' \) (at most \( \alpha \), respectively). An \((\alpha,\varepsilon')\)-tolerant tester [6] for \( P \) is a randomized algorithm that, given parameters \( \alpha \in (0,1), \varepsilon' \in (0,1) \) and oracle access to a string \( x \), accepts with probability at least \( 2/3 \), if \( x \) is \( \alpha \)-close to \( P \) and rejects with probability at least \( 2/3 \), if \( x \) is \( \varepsilon' \)-far from \( P \). The property \( P \) is \((\alpha,\varepsilon')\)-tolerantly testable if there exists an \((\alpha,\varepsilon')\)-tolerant tester for \( P \) with query complexity that depends only on the parameters \( \alpha \) and \( \varepsilon' \) (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 \( P \) with \( \alpha \)-erasure-resilient \( \varepsilon \)-testing of \( P \) for the same values of \( \alpha \) and \( \varepsilon \). The parameter \( \alpha \) 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 \( \alpha \)-erasure-resilient \( \varepsilon \)-tester rejects with probability at least \( \frac{2}{3} \) if, for every way of completing an input string, one needs to change at least an \( \varepsilon \) fraction of the remaining part of the input to make it satisfy \( P \). Similarly, an \((\alpha,\alpha + \varepsilon(1-\alpha))\)-tolerant tester rejects with probability at least \( \frac{2}{3} \), if, for every way of correcting \( \alpha \) fraction of the input values, one needs to change at least an \( \varepsilon \) fraction of the remaining \((1-\alpha)\) fraction of the input to make it satisfy \( P \).

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.

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

\begin{itemize}
  \item \( P \) is \( \alpha \)-erasure-resiliently \( \varepsilon \)-testable;
  \item \( P \) is not \((\alpha, \alpha + \varepsilon(1-\alpha))\)-tolerantly testable.
\end{itemize}
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