Optimal mean-based algorithms for trace reconstruction

Anindya De*
Northwestern University
anindya@eecs.northwestern.edu

Ryan O'Donnell†
Carnegie Mellon University
odonnell@cs.cmu.edu

Rocco A. Servedio‡
Columbia University
rocco@cs.columbia.edu

December 12, 2016

Abstract

In the *(deletion-channel) trace reconstruction problem*, there is an unknown n-bit *source string* x. An algorithm is given access to independent *traces* of x, where a trace is formed by deleting each bit of x independently with probability δ. The goal of the algorithm is to recover x exactly (with high probability), while minimizing samples (number of traces) and running time.

Previously, the best known algorithm for the trace reconstruction problem was due to Holenstein et al. [HMPW08]; it uses $\exp(\widetilde{O}(n^{1/2}))$ samples and running time for any fixed $0 < \delta < 1$. It is also what we call a "mean-based algorithm", meaning that it only uses the empirical means of the individual bits of the traces. Holenstein et al. also gave a lower bound, showing that any mean-based algorithm must use at least $n^{\widetilde{\Omega}(\log n)}$ samples.

In this paper we improve both of these results, obtaining matching upper and lower bounds for mean-based trace reconstruction. For any constant deletion rate $0 < \delta < 1$, we give a mean-based algorithm that uses $\exp(O(n^{1/3}))$ time and traces; we also prove that any mean-based algorithm must use at least $\exp(\Omega(n^{1/3}))$ traces. In fact, we obtain matching upper and lower bounds even for δ subconstant and $\rho := 1 - \delta$ subconstant: when $(\log^3 n)/n \ll \delta \leq 1/2$ the bound is $\exp(-\Theta(\delta n)^{1/3})$, and when $1/\sqrt{n} \ll \rho \leq 1/2$ the bound is $\exp(-\Theta(n/\rho)^{1/3})$.

Our proofs involve estimates for the maxima of Littlewood polynomials on complex disks. We show that these techniques can also be used to perform trace reconstruction with random insertions and bit-flips in addition to deletions. We also find a surprising result: for deletion probabilities $\delta > 1/2$, the presence of insertions can actually *help* with trace reconstruction.

*Supported by start-up grant from Northwestern University.
†Supported by NSF grant CCF-1618679.
‡Supported by NSF grants CCF-1420349 and CCF-1563155.

1 Introduction

Consider a setting in which a string x of length n over an alphabet Σ is passed through a *deletion channel* that independently deletes each coordinate of x with probability δ. The resulting string, of length somewhere between 0 and n, is referred to as a *trace* of x, or as a *received string*; the original string x is referred to as the *source string*. The *trace reconstruction problem* is the task of reconstructing x (with high probability) given access to independent traces of x. This is a natural and well-studied problem, dating back to the early 2000's [Lev01b, Lev01a, BKKM04], with some combinatorial variants dating even to the early 1970's [Kal73]. However, perhaps surprisingly, much remains to be discovered both about the information-theoretic and algorithmic complexity of this problem. Indeed, in a 2009 survey [Mit09, Section 7], Mitzenmacher wrote that "the study of [trace reconstruction] is still in its infancy".

Before discussing previous work, we briefly explain why one can assume a binary alphabet without loss of generality. In case of a general Σ, drawing $O(\frac{\log n}{1-\delta})$ traces will with high probability reveal the entire alphabet $\Sigma' \subseteq \Sigma$ of symbols that are present in x. For each symbol $\sigma \in \Sigma'$ we may consider the binary string $x|_\sigma$ whose i-th character is 1 iff $x_i = \sigma$; a trace of x is easily converted into a trace of $x|_\sigma$, so the trace reconstruction problem for x can be solved by solving the binary trace reconstruction problem for each $x|_\sigma$ and combining the results in the obvious way. For this reason, our work (and most previous work) focuses on the case of a binary alphabet.

1.1 Prior work

As described in [Mit09], the trace reconstruction problem can arise in several natural domains, including sensor networks and biology. However, the apparent difficulty of the problem means that there is not too much published work, at least on the problem of "worst-case" trace reconstruction problem ("worst-case" in the sense that the source string may be any element of $\{0,1\}^n$). Because of this, several prior authors have considered an "average-case" version of the problem in which the source string is assumed to be uniformly random over $\{0,1\}^n$ and the algorithm is required to succeed with high probability over the random draw of the traces and over the uniform random choice of x. This average-case problem seems to have first been studied by Batu et al. [BKKM04], who showed that a simple efficient algorithm which they call Bitwise Majority Alignment succeeds with high probability for sufficiently small deletion rates $\delta = O(1/\log n)$ using only $O(\log n)$ traces. Subsequent work of Kannan and McGregor [KM05] gave an algorithm for random x that can handle both deletions and insertions (both at rates $O(1/\log^2 n)$ as well as bit-flips (with constant probability bounded away from 1/2) using $O(\log n)$ traces. Viswanathan and Swaminathan [VS08] sharpened this result by improving the deletion and insertion rates that can be handled to $O(1/\log n)$. Finally, [HMPW08] gave a poly(n)-time, poly(n)-trace algorithm for random x that succeeds with high probability for any deletion rate δ that is at most some sufficiently small absolute constant.

Several researchers have considered, from an information-theoretic rather than algorithmic perspective, various reconstruction problems that are closely related to the (worst-case) trace reconstruction problem. Kalashnik [Kal73] showed that any n-bit string is uniquely specified by its k-*deck*, which is the multiset of all its length-k subsequences, when $k = \lfloor n/2 \rfloor$; this result was later reproved by Manvel et al. [MMS+91]. Scott [Sco97] subsequently showed that $k = (1+o(1))\sqrt{n \log n}$ suffices for reconstruction from the k-deck for any x, and simultaneously and independently Krasnikov and Roditty [KR97] showed that $k = \lfloor \frac{16}{7}\sqrt{n} \rfloor + 5$ suffices. (McGregor et al. observed in [MPV14] that the result of [Sco97] yields an information-theoretic algorithm using $\exp(\tilde{O}(n^{1/2}))$ traces for any deletion rate $\delta \leq 1 - O(\sqrt{\log(n)/n})$, but did not discuss the running time of such an al-

gorithm.) On the other side, successively larger $\Omega(\log n)$ lower bounds on the value of k that suffices for reconstruction of an arbitrary $x \in \{0,1\}^n$ from its k-deck were given by Manvel et al. [MMS+91] and Choffrut and Karhumäki [CK97], culminating in a lower bound of $2^{\Omega(\sqrt{\log n})}$ due to Dudík and Schulman [DS03].

Surprisingly few algorithms have been given for the worst-case trace reconstruction problem as defined in the first paragraph of this paper. Batu et al. [BKKM04] showed that a variation of their Bitwise Majority Alignment algorithm succeeds efficiently using $O(n \log n)$ traces if the deletion rate δ is quite low, at most $O(1/n^{1/2+\varepsilon})$. Holenstein et al. [HMPW08] gave a "mean-based" algorithm (we explain precisely what is meant by such an algorithm later) that runs in time $\exp(\tilde{O}(\sqrt{n}))$ and uses $\exp(\tilde{O}(\sqrt{n}))$ traces for any deletion rate δ that is bounded away from 1 by a constant; this is the prior work that is most relevant to our main positive result. [HMPW08] also gave a lower bound showing that for any δ bounded away from 0 by a constant, at least $n^{\Omega(\frac{\log n}{\log\log n})}$ traces are required for any mean-based algorithm. Since the result of [HMPW08], several researchers (such as [Mos13]) have raised the question of finding (potentially inefficient) algorithms which have a better sample complexity; however, no progress had been made until this work.

One may also ask (as was done in the "open questions" of [Mit09, Section 7]) for trace reconstruction for more general channels, such as those that allow deletions, insertions, and bit-flips. The only work we are aware of along these lines is that of Andoni et al. [ADHR12], which gives results for trace reconstruction for *average-case* words in the presence of insertions, deletions, and substitutions on a tree.

1.2 Our results

Theorem 1.1 (Deletion channel positive result)**.** *There is an algorithm for the trace reconstruction problem which, for any constant $0 < \delta < 1$, uses $\exp(O(n^{1/3}))$ traces and running time.*

Theorem 1.1 significantly improves the running time and sample complexity of the [HMPW08] algorithm, which is $\exp(\tilde{O}(n^{1/2}))$ for fixed constant δ. Furthermore, we can actually extend Theorem 1.1 to the case of $\delta = o(1)$ or $\delta = 1 - o(1)$; see Theorem 1.3 below.

The algorithm of Theorem 1.1 is a "mean-based" algorithm, meaning that it uses only the empirical mean of the trace vectors it receives. We prove an essentially matching lower bound for such algorithms:

Theorem 1.2 (Deletion channel negative result)**.** *For any constant $0 < \delta < 1$, every mean-based algorithm must use at least $\exp(\Omega(n^{1/3}))$ traces.*

As mentioned, we can also treat $\delta = o(1)$ and $\delta = 1 - o(1)$:

Theorem 1.3 (Deletion channel general matching bounds)**.** *The matching bounds in Theorems 1.1 and 1.2 extend as follows: For $O(\log^3 n)/n \le \delta \le 1/2$, the matching bound is $\exp(\Theta(\delta n)^{1/3})$ (and for any smaller δ we have a $\mathrm{poly}(n)$ upper bound). Writing $\rho = 1-\delta$ for the "retention" probability, for $O(1/n^{1/2}) \le \rho \le 1/2$ the matching bound is $\exp(\Theta(n/\rho)^{1/3})$.*

For simplicity in the main portion of the paper we consider only the deletion channel and prove the above results. In Appendix A we consider a more general channel that allows for deletions, insertions, and bit-flips, and prove the following result, which extends Theorem 1.1 to that more general channel and includes Theorem 1.1 as a special case.

Theorem 1.4 (General channel positive result)**.** *Let $\mathcal{C}$ be the general channel described in Section A.1 with deletion probability $\delta = 1 - \rho$, insertion probability σ, and bit-flip probability $\gamma/2$. Define*

$$r := \frac{\rho + \delta\sigma}{1 + \sigma}.$$

Then there is an algorithm for $\mathcal{C}$-channel trace reconstruction using samples and running time bounded by

$$\mathrm{poly}(\tfrac{1}{1-\delta}, \tfrac{1}{1-\sigma}, \tfrac{1}{1-\gamma}) \cdot \begin{cases} \exp(O(n/r)^{1/3}) & \text{if } C/n^{1/2} \le r \le 1/2, \\ \exp(O((1-r)n)^{1/3}) & \text{if } O(\log^3 n)/n \le 1 - r \le 1/2. \end{cases}$$

Since some slight technical and notational unwieldiness is incurred by dealing with the more general channel, we defer the proof of Theorem 1.4 to Appendix A; however, we note here that the main core of the proof is unchanged from the deletion-only case. We additionally note that, as discussed in Appendix A, a curious aspect of the upper bound given by Theorem 1.4 is that having a constant insertion rate can make it possible to perform trace reconstruction in time $\exp(O(n^{1/3}))$ even when the deletion rate is much higher than Theorem 1.3 could handle in the absence of insertions. A possible intuitive explanation for this is that having random insertions could serve to "smooth out" worst-case instances that are problematic for a deletion-only model.

1.3 Independent and concurrent work

At the time of writing, we have been informed [Per] that Fedor Nazarov and Yuval Peres have independently obtained results that are substantially similar to Theorems 1.1 and 1.2. Also, Elchanan Mossel has informed us [Mos] that around 2008, Mark Braverman, Avinatan Hassidim and Elchanan Mossel had independently proven (unpublished) superpolynomial lower bounds for mean-based algorithms.

1.4 Our techniques

For simplicity of discussion, we restrict our focus in this section to the question of upper bounding the sample complexity of trace reconstruction for the deletion channel, where every bit gets deleted independently with probability δ. (As discussed above, generalizing the results to channels which also allow for insertions and flips is essentially a technical exercise that does not require substantially new ideas.) As we discuss in Section 3.2, an efficient *algorithm* follows easily from a sample complexity upper bound via the observation that the minimization problem whose solution yields a sample complexity upper bound, extends to a slightly larger *convex* set, and thus one can use convex (in fact, linear) programming to get an algorithmic result. Hence the technical meat of the argument lies in upper bounding the sample complexity.

The key enabling idea for our work is to take an analytic view on the combinatorial process defined by the deletion channel. More precisely, consider two distinct strings $x, x' \in \{-1, 1\}^n$. A necessary (and sufficient) condition to upper bound the sample complexity of trace reconstruction is to lower bound the statistical distance between the two distributions of traces of x versus x' (let us write $\mathcal{C}(x)$ and $\mathcal{C}(x')$ to denote these two distributions). Since analyzing the statistical distance $d_{\mathrm{TV}}(\mathcal{C}(x), \mathcal{C}(x'))$ between the distributions $\mathcal{C}(x)$ and $\mathcal{C}(x')$ turns out to be a difficult task, we approach it by considering a limited class of statistical tests.

In [HMPW08] the authors consider "mean-based" algorithms; such algorithms correspond to statistical tests that only use 1-bit marginals of the distribution of the received string. More

precisely, for any $1 \le j \le n$, consider the quantities $\mathbf{Pr}_{\boldsymbol{y} \leftarrow \mathcal{C}(x)}[\boldsymbol{y}_j = 1]$ and $\mathbf{Pr}_{\boldsymbol{y}' \leftarrow \mathcal{C}(x')}[\boldsymbol{y}'_j = 1]$. The difference $|\mathbf{Pr}_{\boldsymbol{y} \leftarrow \mathcal{C}(x)}[\boldsymbol{y}_j = 1] - \mathbf{Pr}_{\boldsymbol{y}' \leftarrow \mathcal{C}(x')}[\boldsymbol{y}'_j = 1]|$ is a lower bound on $d_{\mathrm{TV}}(\mathcal{C}(x), \mathcal{C}(x'))$.

Let us define the vector $\beta_{x,x'} = (\beta_{x,x'}(1), \ldots, \beta_{x,x'}(n)) \in [-1,1]^n$ by

$$\beta_{x,x'}(j) = \mathop{\mathbf{Pr}}_{\boldsymbol{y} \leftarrow \mathcal{C}(x)}[\boldsymbol{y}'_j = 1] - \mathop{\mathbf{Pr}}_{\boldsymbol{y}' \leftarrow \mathcal{C}(x')}[\boldsymbol{y}_j = 1].$$

In this terminology, giving a sample complexity upper bound on mean-based algorithms correspond to showing a lower bound on $\min_{x \neq x' \in \{-1,1\}^n} \|\beta_{x,x'}\|_1$. A central idea in this paper is to analyze $\|\beta_{x,x'}\|_1$ by studying the Z-transform of the vector $\beta_{x,x'}$. More precisely, for $z \in \mathbb{C}$, we consider $\widehat{\beta}_{x,x'}(z) := \sum_{j=1}^{n} \beta_{x,x'}(j) \cdot z^{j-1}$. Elementary complex analysis can be used to show that

$$\sup_{|z|=1} |\widehat{\beta}_{x,x'}(z)| \le \|\beta_{x,x'}\|_1 \le \sqrt{n} \cdot \sup_{|z|=1} |\widehat{\beta}_{x,x'}(z)|.$$

Thus, for our purposes, it suffices to study $\sup_{|z|=1} |\widehat{\beta}_{x,x'}(z)|$. By analyzing the deletion channel and observing that $\widehat{\beta}_{x,x'}(z)$ is a polynomial in z, we are able to characterize this supremum as the supremum of a certain polynomial (induced by x and x') on a certain disk in the complex plane. Thus giving a sample complexity upper bound amounts to lower bounding $\sup_{|z|=1} |\widehat{\beta}_{x,x'}(z)|$ across all polynomials $\widehat{\beta}_{x,x'}$ induced by distinct $x, x' \in \{-1,1\}^n$ (essentially, across a class of polynomials closely related to *Littlewood polynomials: those polynomials with all coefficients in* $\{-1,0,1\}$*)*. The technical heart of our sample complexity upper bound is in establishing such a lower bound. Finally, similar ideas and arguments are used to lower bound the sample complexity of mean-based algorithms, by upper bounding $\sup_{|z|=1} |\widehat{\beta}_{x,x'}(z)|$ across all polynomials $\widehat{\beta}_{x,x'}$ induced by distinct $x, x' \in \{-1,1\}^n$.

2 Preliminaries and terminology

Throughout this paper we will use two slightly nonstandard notational conventions. Bits will be written as $\{-1,1\}$ rather than $\{0,1\}$, and strings will be indexed starting from 0 rather than 1. Thus the *source string* will be denoted $x = (x_0, x_1, \ldots, x_{n-1}) \in \{-1,1\}^n$; this is the unknown string that the reconstruction algorithm is trying to recover.

We will write $\mathcal{C}$ for the *channel* through which x is transmitted. In the main body of the paper our main focus will be on the *deletion channel* $\mathcal{C} = \mathrm{Del}_\delta$, in which each bit of x is independently δeleted with probability $\delta < 1$. We will also often consider $\rho = 1 - \delta > 0$, the ρetention probability of each coordinate. In Appendix A we will see that a more general channel that also involves *insertions* and *bit-flips* can be handled in a similar way.

We will use boldface to denote random variables. We typically write $\boldsymbol{y} \leftarrow \mathcal{C}(x)$ to denote that $\boldsymbol{y} = (\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\boldsymbol{n}-1})$ is a random *trace* (or *received string* or *sample*), obtained by passing x through the channel $\mathcal{C}$. Notice the slight inconvenience that the length of $\boldsymbol{y}$ is a random variable (for the deletion channel this length is always between 0 and n); we denote this length by $\boldsymbol{n}$.

We define a *trace reconstruction algorithm for channel* $\mathcal{C}$ to be an algorithm with the following property: for any unknown source string $x \in \{-1,1\}^n$, when given access to independent strings $\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots$ each distributed according to $\mathcal{C}(x)$, it outputs x with probability at least (say) 99%. The *sample complexity* of the trace reconstruction algorithm is the number of draws from $\mathcal{C}(x)$ that it uses (in the worst case across all $x \in \{-1,1\}^n$ and all draws from $\mathcal{C}(x)$). We are also interested in the algorithm's (worst-case) running time.

As mentioned earlier we will use basic complex analysis. The following notation will be useful:

Notation 2.1. We write $D_r(c)$ for the closed complex disk of radius r centered at c; i.e., $\{z \in \mathbb{C} : |z - c| \leq r\}$. We write $\partial D_r(c)$ for the boundary of this disk; thus, e.g., $\partial D_1(0) = \{z \in \mathbb{C} : |z| = 1\}$ is the complex unit circle.

3 Mean traces

We now come to a key definition, that of the *mean trace*. For now we restrict our focus to $\mathcal{C}$ being the deletion channel Del_δ (we consider a more general channel in Appendix A).

Although a random trace $\boldsymbol{y} \leftarrow \mathrm{Del}_\delta(x)$ does not have a fixed length, we can simply define the mean trace of a source string $x \in \{-1, 1\}^n$ to be

$$\mu_{\mathrm{Del}_\delta}(x) = \mathop{\mathbf{E}}_{\boldsymbol{y} \leftarrow \mathrm{Del}_\delta(x)}[\boldsymbol{y}'] \in [-1, 1]^n, \tag{1}$$

where $\boldsymbol{y}'$ is $\boldsymbol{y}$ padded with zeros so as to be of length exactly n. Here "0" has a natural interpretation as a "uniformly random bit" (indeed, a trace reconstruction algorithm could always pad deletion-channel traces with random bits by itself, and this would not change the definition of the mean trace $\mu_{\mathrm{Del}_\delta}(x)$).

The following is immediate:

Proposition 3.1. *Viewing the domain of $\mu_{\mathrm{Del}_\delta}$ as the real vector space $\mathbb{R}^n$, $\mu_{\mathrm{Del}_\delta}(x)$ is a (real-)linear function of x; that is, each $\mu_{\mathrm{Del}_\delta}(x)_j$ can be written as $\sum_i a_{i,j} x_i$ for some constants $a_{i,j} \in \mathbb{R}$.*

3.1 The mean-based (deletion-channel) trace reconstruction model

One of the most basic things that a trace reconstruction algorithm can do is calculate an empirical estimate of the mean trace. A simple Chernoff/union bound shows that, with $\mathrm{poly}(n/\epsilon)$ samples and time, an algorithm can compute an estimator $\widehat{\mu}_{\mathrm{Del}_\delta}(x) \in [-1, 1]^n$ satisfying $\|\widehat{\mu}_{\mathrm{Del}_\delta}(x) - \mu_{\mathrm{Del}_\delta}(x)\|_1 \leq \epsilon$ with very high probability. The algorithm might then proceed to base its reconstruction solely on $\widehat{\mu}_{\mathrm{Del}_\delta}(x)$, without relying on further traces. We call such algorithms "mean-based trace reconstruction algorithms" (Holenstein et al. [HMPW08] called them algorithms based on "summary statistics"). We give a formal definition:

Definition 3.2. An algorithm in the *mean-based (deletion-channel) trace reconstruction model* works as follows. Given an unknown source string $x \in \{-1, 1\}^n$, the algorithm first specifies a parameter $T \in \mathbb{N}$. The algorithm is then given an estimate $\widehat{\mu}_{\mathrm{Del}_\delta}(x) \in [-1, +1]^n$ of the mean trace satisfying

$$\|\widehat{\mu}_{\mathrm{Del}_\delta}(x) - \mu_{\mathrm{Del}_\delta}(x)\|_1 \leq 1/T. \tag{2}$$

We define the "cost" of this portion of the algorithm to be T. Having been given $\widehat{\mu}_{\mathrm{Del}_\delta}(x)$, the algorithm has no further access to x, but may do further "postprocessing" computation involving $\widehat{\mu}_{\mathrm{Del}_\delta}(x)$. The algorithm should end by outputting x.

From the above discussion, we see that an algorithm in the mean-based trace reconstruction model with cost T_1 and postprocessing time T_2 may be converted into a normal trace reconstruction algorithm using $\mathrm{poly}(n, T_1)$ samples and $\mathrm{poly}(n, T_1) + T_2$ time.

3.2 The complexity of mean-based (deletion-channel) trace reconstruction

As discussed in [HMPW08], the sample complexity of mean-based trace reconstruction is essentially determined by the minimum distance between the mean traces $\mu_{\mathrm{Del}_\delta}(x)$ and $\mu_{\mathrm{Del}_\delta}(x')$ of two distinct source strings $x, x' \in \{-1,1\}^n$. Furthermore, one can get an upper bound on the *time* complexity of mean-based trace reconstruction if a certain "fractional relaxation" of this minimum mean trace distance is large. We state these observations from [HMPW08] here, using slightly different notation.

Definition 3.3. Given n and $0 \le \delta < 1$, we define:

$$\epsilon_{\mathrm{Del}_\delta}(n) \coloneqq \min_{\substack{x,x' \in \{-1,1\}^n \\ x \neq x'}} \|\mu_{\mathrm{Del}_\delta}(x) - \mu_{\mathrm{Del}_\delta}(x')\|_1 = 2 \min_{\substack{b \in \{-1,0,+1\}^n \\ b \neq 0}} \|\mu_{\mathrm{Del}_\delta}(b)\|_1;$$

$$\epsilon^{\mathrm{frac}}_{\mathrm{Del}_\delta}(n) \coloneqq \min_{0 \le i < n} \min_{\substack{x,x' \in [-1,+1]^n \\ x_j = x'_j \in \{-1,1\} \forall j < i \\ x_i = -x'_i \in \{-1,1\}}} \|\mu_{\mathrm{Del}_\delta}(x) - \mu_{\mathrm{Del}_\delta}(x')\|_1 = 2 \min_{d \in [n]} \min_{b \in \{0\}^{d-1} \times \{1\} \times [-1,+1]^{n-d}} \|\mu_{\mathrm{Del}_\delta}(b)\|_1.$$

In both cases, the equality on the right uses Proposition 3.1.

It's easy to see that in the mean-based trace reconstruction model, it is information-theoretically possible for an algorithm to succeed if and only if its cost T exceeds $2/\epsilon_{\mathrm{Del}_\delta}(n)$. Thus characterizing the sample complexity of mean-based trace reconstruction essentially amounts to analyzing $\epsilon_{\mathrm{Del}_\delta}(n)$. For example, to establish our lower bound Theorem 1.2, it suffices to prove that the $\epsilon_{\mathrm{Del}_\delta}(n) \le \exp(-\Omega(n^{1/3}))$ for constant $0 < \delta < 1$.

Furthermore, as observed in [HMPW08], given an $\epsilon^{\mathrm{frac}}_{\mathrm{Del}_\delta}(n)/4$-accurate estimate of $\mu_{\mathrm{Del}_\delta}(x)$, as well as the ability to compute the linear function $\mu_{\mathrm{Del}_\delta}(x')$ for any $x' \in [-1,+1]^n$ (or even estimate it to $\epsilon^{\mathrm{frac}}_{\mathrm{Del}_\delta}(n)/4$-accuracy), one can recover x exactly in $\mathrm{poly}(n, \log(1/\epsilon^{\mathrm{frac}}_{\mathrm{Del}_\delta}(n)))$ time by solving a sequence of n linear programs.[1] Thus to establish our Theorem 1.1, it suffices to prove that $\epsilon^{\mathrm{frac}}_{\mathrm{Del}_\delta}(n) \ge \exp(-O(n^{1/3}))$ for constant $0 < \delta < 1$.

3.3 Reduction to complex analysis

Our next important definition is of a polynomial that encodes the components of $\mu_{\mathcal{C}}(x)$ in its coefficients — kind of a generating function for the channel. We think of its parameter z as a complex number.

Definition 3.4. Given $x \in \{-1,1\}^n$ and $0 \le \delta < 1$, we define the *deletion-channel polynomial*

$$P_{\mathrm{Del}_\delta, x}(z) = \sum_{j<n} \mu_{\mathrm{Del}_\delta}(x)_j \cdot z^j,$$

a polynomial of degree less than n. We extend this definition to $x \in [-1,+1]^n$ using the linearity of $\mu_{\mathrm{Del}_\delta}$.

We now make the step to elementary complex analysis, by relating the size of a mean trace difference $\mu_{\mathrm{Del}_\delta}(b)$ to the maximum modulus of $P_{\mathrm{Del}_\delta, b}(z)$ on the unit complex circle (or equivalently, the unit complex disk, by the Maximum Modulus Principle):

[1] If the algorithm "knows" δ it can efficiently compute $\mu_{\mathrm{Del}_\delta}(x')$ exactly. But even if it doesn't "know" δ, it can estimate δ to sufficient accuracy so that $\mu_{\mathrm{Del}_\delta}(x')$ can be estimated to the necessary accuracy, with no significant algorithmic slowdown.

Proposition 3.5. *For any $b \in [-1,1]^n$, we have*

$$\max_{z \in \partial D_1(0)} |P_{\mathrm{Del}_\delta, b}(z)| \le \|\mu_{\mathrm{Del}_\delta}(b)\|_1 \le \sqrt{n} \max_{z \in \partial D_1(0)} |P_{\mathrm{Del}_\delta, b}(z)|.$$

Proof. Recall that $\mu_{\mathrm{Del}_\delta}(b)$ is the length-n vector of coefficients for the polynomial $P_{\mathrm{Del}_\delta, b}(z)$. The lower bound above is immediate from the triangle inequality. For the upper bound, we use

$$\|\mu_{\mathrm{Del}_\delta}(b)\|_1^2 \le n \|\mu_{\mathrm{Del}_\delta}(b)\|_2^2 = n \operatorname*{avg}_{z \in \partial D_1(0)} |P_{\mathrm{Del}_\delta, b}(z)|^2 \le n \left(\max_{z \in \partial D_1(0)} |P_{\mathrm{Del}_\delta, b}(z)| \right)^2.$$

Here the first inequality is Cauchy–Schwarz, the equality is an elementary fact about complex polynomials (or Fourier series), and the final inequality is obvious. □

Let us reconsider Definition 3.3. As a factor of $\sqrt{n}$ is negligible compared to the bounds we will prove (which are of the shape $\exp(-\Theta(n^{1/3}))$, we may as well analyze $\max_{z \in \partial D_1(0)} |P_{\mathrm{Del}_\delta, b}(z)|$ rather than $\|\mu_{\mathrm{Del}_\delta}(b)\|_1$ in the definition of $\epsilon_{\mathrm{Del}_\delta}(n)$ and $\epsilon^{\mathrm{frac}}_{\mathrm{Del}_\delta}(n)$. We therefore take a closer look at the deletion-channel polynomial.

4 The deletion-channel polynomial

In this section we compute the deletion-channel polynomial. When the deletion channel is applied to some source string x, each bit x_i is either deleted with probability δ or else is transmitted at some position $j \le i$ in the received string $\boldsymbol{y}$. Let us introduce (non-independent) random variables $\boldsymbol{J}_0, \dots, \boldsymbol{J}_{n-1}$, where $\boldsymbol{J}_i = \perp$ if x_i is deleted and otherwise $\boldsymbol{J}_i$ is the position in $\boldsymbol{y}$ at which x_i is transmitted. We thus have

$$P_{\mathrm{Del}_\delta, x}(z) = \sum_{j<n} \mathop{\mathbf{E}}_{\boldsymbol{y} \leftarrow \mathcal{C}(x)} [\boldsymbol{y}_j] \cdot z^j = \sum_{j<n} z^j \cdot \sum_{i<n} \mathbf{Pr}[\boldsymbol{J}_i = j] x_i = \sum_{i<n} x_i \cdot \sum_{j<n} \mathbf{Pr}[\boldsymbol{J}_i = j] z^j = \sum_{i<n} x_i \cdot \text{“}\mathbf{E}\text{”}[z^{\boldsymbol{J}_i}].$$

Here we put the expectation $\mathbf{E}$ in quotation marks because the expression should count 0 whenever $\boldsymbol{J}_i = \perp$. Observing that $\mathbf{Pr}[\boldsymbol{J}_i \ne \perp]$ equals the retention probability $\rho = 1 - \delta$, if we define the conditional random variable

$$\widetilde{\boldsymbol{J}}_i = (\boldsymbol{J}_i \mid \boldsymbol{J}_i \ne \perp)$$

(so $\widetilde{\boldsymbol{J}}_i$ is an $\mathbb{N}$-valued random variable), then we have

$$P_{\mathrm{Del}_\delta, x}(z) = \rho \sum_{i<n} x_i \cdot \mathbf{E}[z^{\widetilde{\boldsymbol{J}}_i}]. \tag{3}$$

Observing that $\widetilde{\boldsymbol{J}}_i$ is distributed as $\mathrm{Binomial}(i, \rho)$, and letting $\boldsymbol{B}_1, \dots, \boldsymbol{B}_i$ denote independent Bernoulli random variables with “success” probability ρ, we easily compute

$$\mathbf{E}[z^{\widetilde{\boldsymbol{J}}_i}] = \mathbf{E}[z^{\boldsymbol{B}_1 + \cdots + \boldsymbol{B}_i}] = \mathbf{E}[z^{\boldsymbol{B}_1}]^i = ((1-\rho) + \rho z)^i.$$

Denoting

$$w = 1 - \rho + \rho z,$$

we conclude that

$$P_{\mathrm{Del}_\delta, x}(z) = \rho \sum_{i<n} x_i w^i.$$

As z ranges over the unit circle $\partial D_1(0)$, w ranges over the radius-ρ circle $\partial D_\rho(1-\rho)$. Recalling Definition 3.3 and Proposition 3.5, we are led to consider the following two quantities for $0 < \rho < 1$ (note that by the Maximum Modulus Principle, these quantities are unchanged whether the max is over $D_\rho(1-\rho)$ or $\partial D_\rho(1-\rho)$):

$$\kappa_{\text{Littlewood}}(\rho, n) = \min\left\{ \max_{w \in D_\rho(1-\rho)} |P(w)| : P(w) = b_0 + b_1 w + \cdots + b_{n-1} w^{n-1},\ b_i \in \{0, \pm 1\} \text{ not all } 0 \right\},$$

$$\kappa^{\text{frac}}_{\text{bounded}}(\rho, d) = \min\left\{ \max_{w \in D_\rho(1-\rho)} |P(w)| : P(w) = w^d + b_{d+1} w^{d+1} + \cdots + b_N w^N,\ N \geq d,\ b_i \in D_1(0) \right\}.$$

By the Maximum Modulus Principle, both $\kappa_{\text{Littlewood}}(\rho, n)$ and $\kappa^{\text{frac}}_{\text{bounded}}(\rho, d)$ are nondecreasing functions of $0 < \rho < 1$. It's also easy to see that both are nonincreasing functions of their second argument for all $0 < \rho < 1$ (for $\kappa^{\text{frac}}_{\text{bounded}}(\rho, d)$, consider replacing $P(w)$ by $wP(w)$) and observe that $|wP(w)| \leq |P(w)|$ for all $w \in D_\rho(1-\rho)$). It thus follows that

$$\kappa^{\text{frac}}_{\text{bounded}}(\rho, d) \leq \kappa_{\text{Littlewood}}(\rho, d).$$

Our main technical theorems are the following:

Theorem 4.1. *There is a universal constant $C \geq 1$ such that:*

$$\text{for } 1/d \leq \delta \leq 1/2, \qquad \kappa^{\text{frac}}_{\text{bounded}}(1-\delta, d) \geq \exp(-C(\delta d)^{1/3});$$
$$\text{for } 1/d^{1/2} \leq \rho \leq 1/2, \qquad \kappa^{\text{frac}}_{\text{bounded}}(\rho, d) \geq \exp(-C(d/\rho)^{1/3}).$$

Theorem 4.2. *There is a universal constant $C \geq 1$ such that:*

$$\text{for } C(\log^3 n)/n \leq \delta \leq 1/2, \qquad \kappa_{\text{Littlewood}}(1-\delta, n) \leq \exp(-\Omega(\delta n)^{1/3});$$
$$\text{for } C/n^{1/2} \leq \rho \leq 1/2, \qquad \kappa_{\text{Littlewood}}(\rho, n) \leq \exp(-\Omega(n/\rho)^{1/3}).$$

By Definition 3.3, Proposition 3.5, and the discussion at the end of Section 3.2, we have that Theorem 4.2 implies both Theorem 1.2 and the more general sample complexity lower bound in Theorem 1.3. Regarding the algorithmic upper bounds in Theorems 1.1 and 1.3, again from Definition 3.3 and Proposition 3.5 we get that

$$\begin{aligned}\epsilon^{\text{frac}}_{\text{Del}_\delta}(n) &\geq 2\rho \cdot \min_{0 \leq d < n} \left\{ \max_{w \in D_\rho(1-\rho)} |P(w)| : P(w) = w^d + b_{d+1} w^{d+1} + \cdots + b_{n-1} w^{n-1},\ b_i \in [-1, +1] \right\} \\ &\geq 2\rho \cdot \min_{0 \leq d < n} \kappa^{\text{frac}}_{\text{bounded}}(\rho, d) \geq 2\rho \cdot \kappa^{\text{frac}}_{\text{bounded}}(\rho, n).\end{aligned}$$

Thus the upper bounds Theorems 1.1 and 1.3 likewise follow from Theorem 4.1 and the discussion at the end of Section 3.2. (Note that if $\delta \leq O(\log^3 n)/n$, we can always pay the bound for the larger value $\delta = \Theta(\log^3 n)/n$), which is $\text{poly}(n)$.)

5 Proof of Theorem 4.1

We will need the following:

Theorem 5.1. *([BE97], Corollary 3.2, $M = 1$ case.) Let $Q(w)$ be a polynomial with constant coefficient 1 and all other coefficients bounded by 1 in modulus. Fix any $0 < \theta \leq \pi$, and let A be the arc $\{e^{it} : -\theta \leq t \leq \theta\}$. Then $\sup_{w \in A} |Q(w)| \geq \exp(-C_1/\theta)$ for some universal constant C_1.*

We remark that for any $0 < r < 1$, Theorem 5.1 holds for the arc $A = \{re^{it} : -\theta \le t \le \theta\}$ with no change in the constant C_1. This is immediate by applying the theorem to $\widetilde{Q}(w) = Q(rw)$.

Proof of Theorem 4.1. Fix $d \ge 2$ (else the hypotheses are vacuous) and $\delta + \rho = 1$. We call Case I when $1/d \le \delta < 1/2$, and we call Case II when $1/d^{1/2} \le \rho \le 1/2$. Select

$$\theta = \begin{cases} \frac{1}{2(\delta d)^{1/3}} & \text{in Case I,} \\ \left(\frac{\rho}{d}\right)^{1/3} & \text{in Case II.} \end{cases}$$

In Case I we have $\theta \le 1/2$, and in Case II we have $\theta \le \rho \le 1/2$.

Let $P(w) = w^d \cdot Q(w)$, where $Q(w)$ is a polynomial with constant coefficient 1 and all other coefficients bounded by 1 in modulus. We need to show

$$\max_{w \in D_\rho(\delta)} |P(w)| \ge \begin{cases} \exp(-C(\delta d)^{1/3}) & \text{in Case I,} \\ \exp(-C(d/\rho)^{1/3}) & \text{in Case II.} \end{cases} \tag{4}$$

In Case I, the ray $\{re^{i\theta} : r > 0\}$ intersects $\partial D_\rho(\delta)$ at a unique point, call it w_0. In Case II, the same ray intersects $D_\rho(\delta)$ twice (this uses $\theta \le \rho$); call the point of larger modulus w_0. In either case, consider the triangle formed in the complex plane by the points 0, δ, and w_0; it has some acute angle α at w_0 and an angle of θ at 0. By the Law of Sines,

$$\frac{\rho}{\sin\theta} = \frac{\delta}{\sin\alpha} = \frac{|w_0|}{\sin(\pi - \theta - \alpha)} = \frac{|w_0|}{\sin(\theta + \alpha)} = \frac{|w_0|}{\sin\theta\cos\alpha + \sin\alpha\cos\theta}$$

$$\implies |w_0| = \delta\cos\theta + \rho\cos\alpha = \delta\cos\theta + \rho\sqrt{1 - (\tfrac{\delta}{\rho})^2\sin^2\theta} \ge \delta(1-\theta^2) + \rho(1 - (\tfrac{\delta}{\rho})^2\theta^2) = 1 - \tfrac{\delta}{\rho}\theta^2.$$

(The last inequality used $\theta \le \rho$ in Case II.) Writing $r_0 = |w_0|$, Theorem 5.1 (and the subsequent remark) implies that

$$\max_{w \in A} |Q(w)| \ge \exp(-C_1/\theta) \quad \text{for } A = \{r_0 e^{it} : -\theta \le t \le \theta\} \subset D_\rho(\delta). \tag{5}$$

Thus

$$\max_{w \in D_\rho(\delta)} |P(w)| \ge \max_{w \in A} |P(w)| \ge r_0^d \cdot \exp(-C_1/\theta) \ge (1 - (\delta/\rho)\theta^2)^d \cdot \exp(-C_1/\theta) \ge \exp(-2(\delta/\rho)\theta^2 d - C_1/\theta)$$

(the last inequality again using $\theta \le \rho$ in Case II). Substituting in the value of θ yields (4). □

5.1 An improved version

Although we don't need it for our application, we can actually provide a stronger version of the results in the previous section that is also self-contained — i.e., it does not rely on Borwein and Erdélyi's Theorem 5.1. We used that theorem to establish (5); but more strongly than (5), we can show there exists an arc $A \subset D_\rho(\delta)$ such that

$$\mathrm{GM}_{w \in A} |Q(w)| \ge \exp(-O(1/\theta)),$$

where the left-hand side here denotes the *geometric mean* of $|Q|$ along A. (Of course, this is at most the max of $|Q|$ along A.) To keep the parameters simpler, we will assume $\rho \le 1/3$ (this is

the more interesting parameter regime anyway, and it is sufficient to yield our Theorem 1.1). Our alternate arc A will be

$$A = \{1/3 + re^{it} : -\theta \le t \le \theta\},$$

where $0 < r < 2/3$ is the larger real radius such that $1/3 + re^{\pm i\theta} \in \partial D_\rho(\delta)$. We remark that still $A \subset D_\rho(\delta)$, by virtue of $\theta \le \rho \le 1/3$, and it is not hard to show that the the endpoint of A, call it $w' = 1/3 + re^{i\theta} \in \partial D_\rho(\delta)$, again satisfies $|w'| \ge 1 - \Omega(\frac{\delta}{\rho}\theta^2)$. Thus instead of using Theorem 5.1 as a black box, we could have completed our proof of Theorem 4.1 using the following:

Theorem 5.2. *Let $Q(w)$ be a polynomial with constant coefficient 1 and all other coefficients in $D_1(0)$. Fix any $0 < \theta \le \pi$, $0 \le r \le 2/3$, and let A be the arc $\{1/3 + re^{it} : -\theta \le t \le \theta\}$. Then $\mathrm{GM}_{w\in A}(|Q(w)|) \ge 9/18^{\pi/\theta}$.*

Our proof will require one standard fact from the theory of "Mahler measures":

Fact 5.3. *Let Q be a complex polynomial and let $\mathcal{O}$ be a circle in the complex plane with center c. Then $\mathrm{GM}_{w\in\mathcal{O}}(|Q(w)|) \ge |Q(c)|$.*

Proof. By a linear transformation we may assume $\mathcal{O}$ is the unit circle $\partial D_1(0)$. Express $Q(w) = a_0 \prod_i (w - \alpha_i)$, where the α_i's are the roots of Q. Then $\mathrm{GM}_{w\in\mathcal{O}}(|Q(w)|)$ — known as Q's *Mahler measure*, see e.g. [Smy08] — is exactly equal to $|a_0| \prod_{i\in I} |\alpha_i|$, where $I = \{i : |\alpha_i| \ge 1\}$. (Since $\mathrm{GM}_{w\in\mathcal{O}}(|\cdot|)$ is multiplicative, this statement follows immediately from the elementary fact that $\mathrm{GM}_{w\in\mathcal{O}}(|w - \alpha|) = \max\{|\alpha|, 1\}$.) But clearly we have $|a_0| \prod_{i\in I} |\alpha_i| \ge |a_0| \prod_i |\alpha_i| = |Q(0)|$. □

We can now establish Theorem 5.2:

Proof of Theorem 5.2. Using the bounds on Q's coefficients we have:

$$|Q(w)| \le 1 + |w| + |w|^2 + \cdots = \frac{1}{1 - |w|} \text{ for } w \in D_1(0); \tag{6}$$

$$|Q(1/3)| \ge 1 - |1/3| - |1/3|^2 - \cdots = 1/2. \tag{7}$$

Let us apply Fact 5.3 with $\mathcal{O} = \partial D_r(1/3) \supset A$, writing A' for the complementary arc to A in $\mathcal{O}$. We get

$$1/2 \le \mathrm{GM}_{w\in\mathcal{O}}(|Q(w)|) = \mathrm{GM}_{w\in A}(|Q(w)|)^{\theta/\pi} \cdot \mathrm{GM}_{w\in A'}(|Q(w)|)^{1-\theta/\pi}. \tag{8}$$

And by (6) we have

$$\operatorname*{GM}_{w\in A'}(|Q(w)|) \le \operatorname*{GM}_{w\in A'}(\tfrac{1}{1-|w|}) \le \operatorname*{GM}_{w\in\mathcal{O}}(\tfrac{1}{1-|w|}) \le \operatorname*{GM}_{w\in\partial D_{2/3}(1/3)}(\tfrac{1}{1-|w|}), \tag{9}$$

where the second inequality is because the points $w \in A$ only have larger $\frac{1}{1-|w|}$ than the points in A', and the third inequality is because increasing the radius of $\mathcal{O}$ from r to $2/3$ only increases the value of $\frac{1}{1-|w|}$ for points on $\mathcal{O}$. But now for $-\pi < t \le \pi$, the point $w = 1/3 + (2/3)e^{it} \in D_{2/3}(1/3)$ has $|w|^2 = 1 - \frac{4}{9}(1 - \cos t)$ and hence

$$\frac{1}{1 - |w|} = \frac{1}{1 - \sqrt{1 - \frac{4}{9}(1 - \cos t)}} \le \frac{9}{2(1 - \cos t)}.$$

Thus

$$\operatorname*{GM}_{w\in\partial D_{2/3}(1/3)}(\tfrac{1}{1-|w|}) \le \exp\left(\tfrac{1}{2\pi}\int_{-\pi}^{\pi} \ln\left(\tfrac{9}{2(1-\cos t)}\right) dt\right) = \frac{9}{2}\exp\left(-\tfrac{1}{2\pi}\int_{-\pi}^{\pi} \ln(1 - \cos t)\, dt\right) = 9, \tag{10}$$

the last integral being known. (One can get a much easier integral, with a slightly worse constant, by lower-bounding $1 - \cos t \ge (2/\pi^2)t^2$.) Combining (8), (9), (10) yields the theorem. □

6 Proof of Theorem 4.2

The key ingredient is the following theorem from [BEK99]. (Recall that a *Littlewood polynomial* has all nonzero coefficients either -1 or 1.)

Theorem 6.1 ([BEK99], Theorem 3.3)**.** *For all $k \geq 2$ there is a nonzero Littlewood polynomial Q_k of degree at most k satisfying $|Q_k(t)| \leq \exp(-c_0\sqrt{k})$ for all real $0 \leq t \leq 1$. Here $c_0 > 0$ is a universal constant.*

By a simple use of the Hadamard Three-Circle Theorem and Maximum Modulus Principle, Borwein and Erdélyi proved in [BE97] that the polynomials in Theorem 6.1 establish tightness of their Theorem 5.1 (up to the constant C_1). We quote a result that appears within their proof:

Theorem 6.2 ([BE97], in the first proof of Theorem 3.3 in the "special case", p. 11)**.** *There are universal constants $c_1, c_2, c_3 > 0$ such that the following holds: For all $0 < a \leq c_1$ there exists an integer $2 \leq k \leq c_2/a^2$ such that $\max_{w \in D_{6a}(1)} |Q_k(w)| \leq \exp(-c_3/a)$, where Q_k is the nonzero Littlewood polynomial from Theorem 6.1.*

Remark 6.3. Actually, Borwein and Erdélyi proved this with an elliptical disk $\mathcal{E}_a$ in place of $D_{6a}(1)$, where $\mathcal{E}_a$ has foci at $1-8a$ and 1 and major axis $[1-14a, 1+6a]$. It is easy to see that $D_{6a}(1) \subset \mathcal{E}_a \subset D_{14a}(1)$, so we wrote $D_{6a}(1)$ in Theorem 6.2 for simplicity and because it loses almost nothing.

We can now prove Theorem 4.2. We state here a slightly more precise version:

Theorem 6.4. *Using the notation $\delta = 1-\rho$, and the notation $\mathrm{Exp}(t) = \exp(c \cdot t)$ for an unspecified universal constant $c > 0$, we have*

$$\kappa_{\mathrm{Littlewood}}(\rho, n) \leq \begin{cases} \mathrm{Exp}(-(\delta n)^{1/3}) & \text{in Case I:} \quad C(\log^3 n)/n \leq \delta \leq 1/2, \\ \mathrm{Exp}(-(n/\rho)^{1/3}) & \text{in Case II:} \quad C/n^{1/2} \leq \rho \leq 1/2, \end{cases}$$

provided $n \geq n_0$. Here $n_0, C \geq 1$ are universal constants.

Proof of Theorem 4.2. With $C \geq 1$ to be specified later, select

$$a = \begin{cases} C_1/(\delta n)^{1/3} & \text{in Case I:} \quad C(\log^3 n)/n < \delta \leq 1/2, \\ C_1(\rho/n)^{1/3} & \text{in Case II:} \quad 1/n^{1/2} < \rho < 1/2, \end{cases}$$

where $C_1 \geq 1$ is a universal constant to be specified later. Assuming $n_0 = n_0(C_1)$ is sufficiently large we get that $a \leq c_1$, where c_1 is as in Theorem 6.2. Applying that theorem, we obtain

$$\max_{w \in A} |Q_k(w)| \leq \mathrm{Exp}(-1/a), \qquad \text{where } A \coloneqq D_{6a}(1), \quad k \leq c_2/a^2 < n/2. \tag{11}$$

Here the inequality $c_2/a^2 < n/2$ holds in Case I by assuming $n_0 = n_0(C_1, c_2)$ large enough, and in Case II by taking $C_1 = C_1(c_2)$ large enough. Now define

$$P(w) = w^{\lfloor n/2 \rfloor} \cdot Q_k(w), \quad \text{a nonzero Littlewood polynomial of degree less than } n.$$

We wish to bound

$$\max_{w \in R} |P(w)|, \qquad R \coloneqq D_\rho(\delta)$$

by the expression in the theorem statement. For the points $w \in R \cap A$, we are done by (11) (and the fact that $|w^{\lfloor n/2 \rfloor}| \leq 1$). For the points in $w \in R \setminus A$, we claim that

$$|w|^2 \leq 1 - 36\tfrac{\delta}{\rho}a^2 \leq \exp(-36\tfrac{\delta}{\rho}a^2) \quad \forall w \in R \setminus A. \tag{12}$$

Assuming (12), we get

$$\max_{w \in R \setminus A} |P(w)| \leq \max_{w \in R \setminus A} |w|^{\lfloor n/2 \rfloor} \cdot \max_{w \in R \setminus A} |Q_k(w)| \leq \exp(-18\tfrac{\delta}{\rho}a^2)^{\lfloor n/2 \rfloor} \cdot (n/2+1) \leq \mathrm{Exp}(-n\tfrac{\delta}{\rho}a^2) \cdot (n/2+1),$$

where the factor $n/2+1$ is an upper bound on $|Q_k(w)|$ over all of $D_1(0)$ (recall that Q_k is a Littlewood polynomial of degree less than $n/2$). By inspection, this is sufficient to complete the proof in both Case I and Case II (in Case I we need to assume C large enough to absorb the factor of $(n/2+1)$).

It remains to establish (12). For this we first note that $\rho > 3a$ in both Case I and Case II (Case I is easier to check; for Case II we need to use that $C = C(C_1)$ is sufficiently large). This in particular means that $R \setminus A \neq \emptyset$. Writing w_0 for either of the intersection points of ∂R and ∂A, we have $\max_{w \in R \setminus A} |w| \leq |w_0|$. Thus it suffices to upper-bound $|w_0|^2$.

In the complex plane, consider the triangle formed by δ, 1, and w_0. Note that w_0 has distance ρ from δ and distance $6a$ from 1. Let θ denote the triangle's angle at δ. By the Cosine Law, $(6a)^2 = \rho^2 + \rho^2 - 2\rho^2 \cos\theta$ and hence $\cos\theta = 1 - 18a^2/\rho^2$. Now consider the triangle formed by δ, 0, and w_0. Its angle at δ is $\pi - \theta$ and the adjacent sides have length δ, ρ. Thus by the Cosine Law,

$$|w_0|^2 = \delta^2 + \rho^2 - 2\delta\rho\cos(\pi - \theta) = \delta^2 + \rho^2 + 2\delta\rho\cos\theta = (\delta+\rho)^2 - 36\delta\rho a^2/\rho^2 = 1 - 36\tfrac{\delta}{\rho}a^2,$$

as needed for (12). □

7 Conclusions

A natural direction for future work is to go beyond mean-based algorithms. For example, an efficient algorithm can estimate the covariances of all *pairs* of trace bits. If different sources strings lead to sufficiently different trace-covariances, one could potentially get a more efficient trace reconstruction algorithm. Analyzing this strategy is equivalent to analyzing a certain problem concerning the maxima of Littlewood-like polynomials on $\mathbb{C}^2$; however we could not make any progress on this problem. It would also be interesting to develop lower bound techniques that apply to a broader class of algorithms than just mean-based algorithms.

Finally, we mention that the authors have applied the techniques in this paper (specifically, the technique used in Section 5.1) to several aspects of the population recovery problem. Details will appear in a forthcoming work.

Acknowledgments

The authors would like to thanks the Simons Foundation for sponsoring the symposium on analysis of Boolean functions where the authors began work on this project. A. D. would like to thank Aravindan Vijayaraghavan for useful discussions about this problem.

References

[ADHR12] Alexandr Andoni, Constantinos Daskalakis, Avinatan Hassidim, and Sebastien Roch. Global alignment of molecular sequences via ancestral state reconstruction. *Stochastic Processes and their Applications*, 122(12):3852–3874, 2012. 1.1

[BE97] Peter Borwein and Tamás Erdélyi. Littlewood-type polynomials on subarcs of the unit circle. *Indiana University Mathematics Journal*, 46(4):1323–1346, 1997. 5.1, 6, 6.2

[BEK99] Peter Borwein, Tamás Erdélyi, and Géza Kós. Littlewood-type problems on $[0, 1]$. *Proceedings of the London Mathematical Society*, 3(79):22–46, 1999. 6, 6.1

[BKKM04] Tuğkan Batu, Sampath Kannan, Sanjeev Khanna, and Andrew McGregor. Reconstructing strings from random traces. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 910–918, 2004. 1, 1.1

[CK97] Christian Choffrut and Juhani Karhumäki. Combinatorics of words. In *Handbook of Formal Languages, Volume I*, pages 329–438. Springer, 1997. 1.1

[DS03] Miroslav Dudík and Leonard Schulman. Reconstruction from subsequences. *Journal of Combinatorial Theory, Series A*, 103(2):337–348, 2003. 1.1

[HMPW08] Thomas Holenstein, Michael Mitzenmacher, Rina Panigrahy, and Udi Wieder. Trace reconstruction with constant deletion probability and related results. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 389–398, 2008. (document), 1.1, 1.2, 1.4, 3.1, 3.2, 3.2

[Jan14] Svante Janson. Tail bounds for sums of geometric and exponential variables, 2014. http://www2.math.uu.se/~svante/papers/sjN14.pdf. A.2

[Kal73] V. V. Kalashnik. Reconstruction of a word from its fragments. *Computational Mathematics and Computer Science (Vychislitel'naya matematika i vychislitel'naya tekhnika), Kharkov*, 4:56–57, 1973. 1, 1.1

[KM05] Sampath Kannan and Andrew McGregor. More on reconstructing strings from random traces: Insertions and deletions. In *IEEE International Symposium on Information Theory*, pages 297–301, 2005. 1.1, A.1

[KR97] Ilia Krasikov and Yehuda Roditty. On a reconstruction problem for sequences,. *Journal of Combinatorial Theory, Series A*, 77(2):344–348, 1997. 1.1

[Lev01a] Vladimir Levenshtein. Efficient reconstruction of sequences. *IEEE Transactions on Information Theory*, 47(1):2–22, 2001. 1

[Lev01b] Vladimir Levenshtein. Efficient reconstruction of sequences from their subsequences or supersequences. *Journal of Combinatorial Theory Series A*, 93(2):310–332, 2001. 1

[Mit09] Michael Mitzenmacher. A survey of results for deletion channels and related synchronization channels. *Probability Surveys*, 6:1–33, 2009. 1, 1.1

[MMS+91] Bennet Manvel, Aaron Meyerowitz, Allen Schwenk, Ken Smith, and Paul Stockmeyer. Reconstruction of sequences. *Discrete Mathematics*, 94(3):209–219, 1991. 1.1

[Mos] Elchanan Mossel. Personal communication, October 2016. 1.3

[Mos13] Elchanan Mossel. MSRI open problem session, 2013. https://www.msri.org/c/document_library/get_file?uuid=4a885484-bcdd-4238-a3da-21c057
1.1

[MPV14] Andrew McGregor, Eric Price, and Sofya Vorotnikova. Trace reconstruction revisited. In *Proceedings of the 22nd Annual European Symposium on Algorithms*, pages 689–700, 2014. 1.1

[Per] Yuval Peres. Personal communication, October 2016. 1.3

[Sco97] Alexander Scott. Reconstructing sequences. *Discrete Mathematics*, 175(1):231–238, 1997. 1.1

[Smy08] Chris Smyth. The Mahler measure of algebraic numbers: A survey. In *Number Theory and Polynomials*, pages 322–349. London Mathematical Society Lecture Note Series 352, 2008. 5.1

[VS08] Krishnamurthy Viswanathan and Ram Swaminathan. Improved string reconstruction over insertion-deletion channels. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 399–408, 2008. 1.1

A Results on channels that allow insertions, deletions and flips

A.1 Defining the general channel

We now describe the most general channel $\mathcal{C}$ that we analyze, which we subsequently refer to as "the general channel". As stated earlier, this channel allows for three different types of corruptions: deletions with probability δ, insertions with probability σ, and bit-flips with probability $\gamma/2$. We comment that for mean-based algorithms, the presence of bit-flips makes hardly any difference; thus the reader may focus just on the combination of deletions and insertions.

Our definition of this general channel is essentially the same as that of Kannan and McGregor [KM05]. More precisely, for parameters $\delta, \sigma, \gamma \in [0, 1)$, we define how the channel acts on a single source bit $b \in \{-1, 1\}$:

1. First, the channel performs "insertions"; i.e., it repeatedly does the operation "with probability σ, transmit a uniformly random bit; with probability $1 - \sigma$, stop".

2. Having stopped, the channel "deletes" (completes transmission without sending b or $-b$) with probability δ.

3. Otherwise (with probability $1 - \delta$), the channel transmits one more bit: namely, b with probability $1 - \gamma/2$, or $-b$ with probability $\gamma/2$.

As usual, the channel operates on an entire source string $x \in \{-1, 1\}^n$ by operating on its individual bits independently, concatenating the results. That is,

$$\mathcal{C}(x) = \mathcal{C}(x_0)\mathcal{C}(x_1)\cdots\mathcal{C}(x_{n-1}) \in \{-1, 1\}^*.$$

Of course, if we set $\sigma = \gamma = 0$, we get the deletion channel Del_δ that was analyzed in the main body of the paper.

An alternative description of the channel's operation on a single bit x_i is as follows:

$$\mathcal{C}(x_i) = \begin{cases} \boldsymbol{w} & \text{with probability } \delta, \\ (\boldsymbol{w}, \boldsymbol{a}) & \text{with probability } (1-\delta)\cdot\gamma, \\ (\boldsymbol{w}, x_i) & \text{with probability } (1-\delta)\cdot(1-\gamma), \end{cases} \tag{13}$$

where $\boldsymbol{a} \in \{-1,1\}$ is a uniformly random bit, and where $\boldsymbol{w} \in \{-1,1\}^{\boldsymbol{G}}$ is a uniformly random string of $\boldsymbol{G}$ bits, with $\boldsymbol{G}$ in turn being a Geometric random variable of parameter $1-\sigma$.[2] From this description one can see that in a received word $\boldsymbol{y} \leftarrow \mathcal{C}(x)$, each received bit either "comes from a properly transmitted source bit x_i", or else is uniformly random. (The probability each x_i comes through is $(1-\delta)(1-\gamma)$.) As a consequence, we have that Proposition 3.1 continues to hold for $\mathcal{C}$: for every $j \in \mathbb{N}$, the mean value $\mathbf{E}_{\boldsymbol{y}\leftarrow\mathcal{C}(x)}[\boldsymbol{y}_j]$ is a (real-)linear function of x.

Note that when the insertion probability σ is positive, the received word $\boldsymbol{y} \leftarrow \mathcal{C}(x)$ does not have an a priori bounded length. This is a minor annoyance can be handled in several different ways; we choose one way in the next section.

A.2 Mean traces for the general channel

We revisit some of our definitions and observations about mean traces from Section 3, in our new context of the general channel. We begin with (1), the definition of the mean trace. Since the length of a received word may now be arbitrarily large, the mean trace is now an infinite vector. We deal with this by truncating it at what we call the "effective trace length bound N".

Definition A.1. For the general channel $\mathcal{C}$ with insertion probability $0 \le \sigma < 1$, we define the *effective trace length bound* $N = N(\sigma)$ to be $N = \left\lceil 10 \cdot \frac{n+\ln(1/(1-\sigma))}{1-\sigma} \right\rceil \le \mathrm{poly}(n, \frac{1}{1-\sigma})$.

Definition A.2. For the general channel $\mathcal{C}$ and a source string $x \in \{-1,1\}^n$, we define the *idealized mean trace* to be the infinite sequence

$$\mu_{\mathcal{C}}^{\mathrm{ideal}}(x) = \mathop{\mathbf{E}}_{\boldsymbol{y}\leftarrow\mathcal{C}(x)}[(\boldsymbol{y}, 0, 0, 0, \dots)] \in [-1,+1]^{\mathbb{N}}.$$

We define just the *mean trace* to be its truncation to length N:

$$\mu_{\mathcal{C}}(x) = (\mu_{\mathcal{C}}^{\mathrm{ideal}}(x)_0, \mu_{\mathcal{C}}^{\mathrm{ideal}}(x)_1, \dots, \mu_{\mathcal{C}}^{\mathrm{ideal}}(x)_{N-1}) \in [-1,+1]^N.$$

Recalling (13), we see that the length $\boldsymbol{n}$ of a received word is stochastically dominated by $(\boldsymbol{G}_1+1)+\cdots+(\boldsymbol{G}_n+1)$, where the $\boldsymbol{G}_i$'s are i.i.d. random variables distributed as Geometric$(1-\sigma)$. We upper bound this using Janson's bound on the sum of independent Geometric random variables (Theorem 2.1 of [Jan14]), noting that his Geometric random variables count the number of "trials", which aligns precisely with our $(\boldsymbol{G}_i+1)$'s. His bound gives that $\mathbf{Pr}[\boldsymbol{n} \ge N+j] \le \exp(-(N+j)(1-\sigma)/2)$ for any $j \ge 0$, and hence we have the following: for any $x \in [-1,1]^n$,

$$\begin{aligned}
\|\mu_{\mathcal{C}}(x) - \mu_{\mathcal{C}}^{\mathrm{ideal}}(x)\|_1 &= \sum_{\ell=N}^{\infty} |\mu_{\mathcal{C}}^{\mathrm{ideal}}(x)|_\ell \le \sum_{\ell=N}^{\infty} \mathbf{Pr}[\boldsymbol{n} \ge \ell] = \sum_{j=0}^{\infty} \mathbf{Pr}[\boldsymbol{n} \ge N+j] \\
&= \exp(-N(1-\sigma)/2) \cdot \frac{1}{1-\exp(-(1-\sigma)/2)} < \frac{4\exp(-N(1-\sigma)/2)}{1-\sigma} \\
&\le 4\exp(-n), \qquad \text{by our choice of } N. \qquad (14)
\end{aligned}$$

The mean-based trace reconstruction model for the general channel. Definition 3.2 has a natural analogue for the general channel: an algorithm in the mean-based general-channel model specifies a cost parameter $T \in \mathbb{N}$ and is given an estimate $\widehat{\mu}_{\mathcal{C}}(x) \in [-1,1]^N$ of the mean

[2] Here we use the convention that Geometric random variables take values $0, 1, 2, \dots$ (equal to the number of "failures"); i.e., $\mathbf{Pr}[\boldsymbol{G} = t] = \sigma^t(1-\sigma)$ for each $t \ge 0$.

trace satisfying $\|\widehat{\mu}_{\mathcal{C}}(x) - \mu_{\mathcal{C}}(x)\|_1 \leq 1/T$. It is clear that an algorithm in the mean-based general-channel trace reconstruction model with cost T_1 and postprocessing time T_2 may be converted into a normal trace reconstruction algorithm using $\mathrm{poly}(N, T_1) = \mathrm{poly}(n, \frac{1}{1-\sigma}, T_1)$ samples and $\mathrm{poly}(n, \frac{1}{1-\sigma}, T_1) + T_2$ time. Note that since we will be studying algorithms with cost $T \ll 2^n$, by (14) there is no real difference between getting an estimate of $\mu_{\mathcal{C}}(x)$ or of $\mu_{\mathcal{C}}^{\mathrm{ideal}}(x)$.

The complexity of mean-based trace reconstruction for the general channel. Regarding the complexity of mean-based trace reconstruction, for the general channel we define $\epsilon_{\mathcal{C}}(n)$ and $\epsilon_{\mathcal{C}}^{\mathrm{frac}}(n)$ in the obvious way, replacing each occurrence of the length-n vector $\mu_{\mathrm{Del}_\delta}(\cdot)$ in Definition 3.3 with the length-N vector $\mu_{\mathcal{C}}(\cdot)$. As in Section 3.2, to show that trace reconstruction can be performed under the general channel in time $\mathrm{poly}(N, M) = \mathrm{poly}(n, \frac{1}{1-\sigma}, M)$ it suffices to show that $\epsilon_{\mathcal{C}}^{\mathrm{frac}}(n) \geq 1/M$.[3]

Reduction to complex analysis for the general channel. For $x \in \{-1, 1\}^n$ the *general-channel polynomial* is defined entirely analogously to Definition 3.4:

$$P_{\mathcal{C},x}(z) = \sum_{j<N} \mu_{\mathcal{C}}(x)_j \cdot z^j;$$

note that this is a polynomial of degree less than N. This definition extends to $x \in [-1, +1]^n$ using the linearity of $\mu_{\mathcal{C}}$. Similarly, we may define the *idealized general-channel "polynomial"* by

$$P_{\mathcal{C},x}^{\mathrm{ideal}}(z) = \sum_{j \in \mathbb{N}} \mu_{\mathcal{C}}^{\mathrm{ideal}}(x)_j \cdot z^j;$$

this will actually be a rational function of z.

Entirely analogous to Proposition 3.5, we get that for every $b \in [-1, 1]^n$,

$$\max_{z \in \partial D_1(0)} |P_{\mathcal{C},b}(z)| \leq \|\mu_{\mathcal{C}}(b)\|_1 \leq \sqrt{N} \max_{z \in \partial D_1(0)} |P_{\mathcal{C},b}(z)|.$$

Similar to Section 3.3, a factor of $\sqrt{N} = \mathrm{poly}(n, \frac{1}{1-\sigma})$ is negligible compared to the bounds we will prove, so it suffices to analyze $\max_{z \in \partial D_1(0)} |P_{\mathcal{C},b}(z)|$ rather than $\|\mu_{\mathcal{C}}(b)\|_1$ in the definitions of $\epsilon_{\mathcal{C}}(n)$ and $\epsilon_{\mathcal{C}}^{\mathrm{frac}}(n)$. Moreover, since by (14) we have that $|P_{\mathcal{C},b}^{\mathrm{ideal}}(z) - P_{\mathcal{C},b}(z)| \leq 2^{-n}$ for all $b \in [-1, 1]^n$ and all $z \in \partial D_1(0)$, it suffices to analyze $\max_{z \in \partial D_1(0)} \left| P_{\mathcal{C},b}^{\mathrm{ideal}}(z) \right|$; we do this in the next subsection.

A.3 Channel polynomial for general channels

We now compute the ideal channel polynomial for the general channel defined in Section A.1, using the same technique as in Section 4 and recalling the discussion around the alternative channel description (13). As usual, let $\rho = 1 - \delta$. Let $\boldsymbol{J}_i$ be the random variable whose value is $\perp$ if x_i is either deleted (probability δ) or is replaced by a random bit (probability $(1-\delta) \cdot \gamma$), or else is the position j such that coordinate x_i of the source string ends up in coordinate j in the received string $\boldsymbol{y}$. As before we let $\widetilde{\boldsymbol{J}}_i$ denote the random variable $\boldsymbol{J}_i$ conditioned on not being $\perp$. Since $\mathbf{Pr}[\boldsymbol{J}_i \neq \perp] = (1-\delta) \cdot (1-\gamma)$, a derivation identical to that of (3) yields

$$P_{\mathcal{C},x}^{\mathrm{ideal}}(z) = (1-\delta)(1-\gamma) \sum_{i<n} x_i \cdot \mathbf{E}[z^{\widetilde{\boldsymbol{J}}_i}]. \tag{15}$$

[3] Again, to carry out the linear-programming algorithm, we can either assume that the channel parameters δ, σ, γ are known to the algorithm, or else they should estimated; we omit the details here.

To compute $\mathbf{E}[z^{\widetilde{\boldsymbol{J}}_i}]$, it is straightforward to see that each coordinate $x_{i'}$ with $i' < i$ independently generates a random number of received positions distributed as $\boldsymbol{G}+\boldsymbol{B}$, where $\boldsymbol{G} \sim \text{Geometric}(1-\sigma)$ and independently $\boldsymbol{B} \sim \text{Bernoulli}(\rho)$. Further, conditioned on x_i not being deleted, x_i generates a number of received positions distributed as $\boldsymbol{G}+1$, where the final "+1" is for x_i (or $-x_i$) itself. Thus $\widetilde{\boldsymbol{J}}_i$ is distributed as

$$\boldsymbol{G}_0 + \cdots + \boldsymbol{G}_i + \boldsymbol{B}_0 + \cdots + \boldsymbol{B}_{i-1},$$

where the $\boldsymbol{G}_k$'s are independent copies of $\boldsymbol{G}$ and the $\boldsymbol{B}_k$'s are independent copies of $\boldsymbol{B}$. We therefore obtain

$$\mathbf{E}[z^{\widetilde{\boldsymbol{J}}_i}] = \mathbf{E}[z^{\boldsymbol{G}}]^{i+1} \cdot \mathbf{E}[z^{\boldsymbol{B}}]^i = \left(\mathbf{E}[z^{\boldsymbol{G}}] \cdot \mathbf{E}[z^{\boldsymbol{B}}]\right)^i \cdot \mathbf{E}[z^{\boldsymbol{G}}].$$

Let $F_G(z)$ denote $\mathbf{E}[z^{\boldsymbol{G}}]$ and let $F_B(z)$ denote $\mathbf{E}[z^{\boldsymbol{B}}]$. It is easy to calculate that $F_G(z) = \frac{1-\sigma}{1-\sigma z}$, and we saw earlier that $F_B(z) = (1-\rho) + \rho z = \delta + \rho z$. For brevity, let us write

$$w = F_G(z)F_B(z) = \frac{(1-\sigma)\cdot(\delta+\rho z)}{1-\sigma z},$$

which is a Möbius transformation of z. Thus w ranges over a complex circle as z ranges over $\partial D_1(0)$. More specifically, as z ranges over $\partial D_1(0)$ we have that w ranges over $\partial D_r(1-r)$, where

$$r = \frac{\rho + \delta\sigma}{1+\sigma}.$$

Plugging this back into (15) using $\mathbf{E}[z^{\widetilde{\boldsymbol{J}}_i}] = F_G(z)\cdot w^i$, we obtain

$$P^{\text{ideal}}_{\mathcal{C},x}(z) = (1-\delta)\cdot(1-\gamma)\cdot F_G(z)\cdot\sum_{i<n} x_i \cdot w^i = (1-\gamma)\cdot(1-\delta)\cdot\frac{1-\sigma}{1-\sigma z}\cdot\sum_{i<n} x_i\cdot w^i.$$

We use the bound $\left|\frac{1-\sigma}{1-\sigma z}\right| \geq \frac{1-\sigma}{2}$ for $z \in \partial D_1(0)$. Now by the analysis of $\kappa^{\text{frac}}_{\text{bounded}}(r,d)$ given in Section 4 we get the following algorithmic result for general-channel trace reconstruction, which is our most general positive result:

Theorem 1.4, restated. *Let $\mathcal{C}$ be the general channel described in Section A.1 with deletion probability $\delta = 1-\rho$, insertion probability σ, and bit-flip probability $\gamma/2$. Define*

$$r := \frac{\rho+\delta\sigma}{1+\sigma}.$$

Then there is an algorithm for $\mathcal{C}$-channel trace reconstruction using samples and running time bounded by

$$\text{poly}\left(\tfrac{1}{1-\delta}, \tfrac{1}{1-\sigma}, \tfrac{1}{1-\gamma}\right)\cdot\begin{cases}\exp(O(n/r)^{1/3}) & \text{if } C/n^{1/2} \leq r \leq 1/2,\\ \exp(O((1-r)n)^{1/3}) & \text{if } O(\log^3 n)/n \leq 1-r \leq 1/2.\end{cases}$$

Let us make some observations about this result. First, our Theorem 1.1 for the deletion channel is the special case of Theorem 1.4 obtained by setting $\sigma = \gamma = 0$. Next, for fixed δ,

if $\delta \leq 1/2$, $\quad r$ ranges from $1-\delta$ down to $1/2$ as σ ranges from 0 up to 1;

if $\delta \geq 1/2$, $\quad r$ ranges from $1-\delta$ *up to* $1/2$ as σ ranges from 0 up to 1.

The second statement is rather peculiar: it implies that when the deletion rate is high, the ability to perform trace reconstruction actually *improves*, the more insertions there are. Indeed, when we have deletions only, our ability to do trace reconstruction in time $\exp(O(n^{1/3}))$ is limited to retention probability $\rho \geq \Omega(1)$. But as soon as the insertion rate σ satisfies $\sigma \geq \Omega(1)$, we can do trace reconstruction in time $\exp(O(n^{1/3}))$ as long as the retention rate $\rho = 1-\delta$ satisfies $\rho \geq \exp(-O(n^{1/3}))$.