J. Appl. Probab. **60,** 629–641 (2023)
*doi:*10.1017/jpr.2022.81

TREE TRACE RECONSTRUCTION USING SUBTRACES

TATIANA BRAILOVSKAYA ,* AND
MIKLÓS Z. RÁCZ,** *Princeton University*

Abstract

Tree trace reconstruction aims to learn the binary node labels of a tree, given independent samples of the tree passed through an appropriately defined deletion channel. In recent work, Davies, Rácz, and Rashtchian [10] used combinatorial methods to show that $\exp(\mathrm{O}(k \log_k n))$ samples suffice to reconstruct a complete k-ary tree with n nodes with high probability. We provide an alternative proof of this result, which allows us to generalize it to a broader class of tree topologies and deletion models. In our proofs we introduce the notion of a subtrace, which enables us to connect with and generalize recent mean-based complex analytic algorithms for string trace reconstruction.

Keywords: Trace reconstruction; statistical error correction; tree graphs

2020 Mathematics Subject Classification: Primary 60C05
Secondary 94D99

1. Introduction

Trace reconstruction is a fundamental statistical reconstruction problem that has received much attention lately. Here the goal is to infer an unknown binary string of length n, given independent copies of the string passed through a deletion channel. The deletion channel deletes each bit in the string independently with probability q and then concatenates the surviving bits into a *trace*. The goal is to learn the original string with high probability using as few traces as possible.

The trace reconstruction problem was introduced two decades ago [3, 20], and despite much work over the past two decades [6, 7, 8, 12, 13, 14, 15, 16, 17, 18, 24], understanding the sample complexity of trace reconstruction remains wide open. Specifically, the best known upper bound is due to Chase [6] who showed that $\exp\big(\mathrm{O}\big(n^{1/5} \log^5 n\big)\big)$ samples suffice; this work builds upon previous breakthroughs by De, O'Donnell, and Servedio [12, 13] and Nazarov and Peres [24], who simultaneously obtained an upper bound of $\exp\big(\mathrm{O}\big(n^{1/3}\big)\big)$. In contrast, the best known lower bound is $\Omega\big(n^{3/2}/\log^7 n\big)$ (see [7] and [16]). Considering average-case strings, as opposed to worst-case ones, reduces the sample complexity considerably, but the large gap remains: the current best known upper and lower bounds are $\exp\big(\mathrm{O}\big(\log^{1/3} n\big)\big)$ (see [17]) and $\Omega\big(\log^{5/2} n/(\log\log n)^7\big)$ (see [7]), respectively. As we can see, the bounds are exponentially far apart for both the worst-case and average-case problems.

Received 29 September 2021; revision received 6 July 2022.
* Postal address: Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ, USA. Email address: tatianab@princeton.edu
** Postal address: Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, USA. Email address: mracz@princeton.edu

Given the difficulty of the trace reconstruction problem, several variants have been introduced, in part to study the strengths and weaknesses of various techniques. These include generalizing trace reconstruction from strings to trees [10] and matrices [19], coded trace reconstruction [5, 9], population recovery [1, 2, 22], and more [8, 11, 23].

In this work we consider tree trace reconstruction, introduced recently by Davies, Rácz, and Rashtchian [10]. In this problem we aim to learn the binary node labels of a tree, given independent samples of the tree passed through an appropriately defined deletion channel. The additional tree structure makes reconstruction easier; indeed, in several settings Davies, Rácz, and Rashtchian [10] show that the sample complexity is polynomial in the number of bits in the worst case. Furthermore, Maranzatto [21] showed that strings are the hardest trees to reconstruct; that is, the sample complexity of reconstructing an arbitrary labeled tree with n nodes is no more than the sample complexity of reconstructing an arbitrary labeled n-bit string.

As demonstrated in [10], tree trace reconstruction provides a natural testbed for studying the interplay between combinatorial and complex analytic techniques that have been used to tackle the string variant. Our work continues in this spirit. In particular, Davies, Rácz, and Rashtchian [10] used combinatorial methods to show that $\exp(\mathrm{O}(k \log_k n))$ samples suffice to reconstruct complete k-ary trees with n nodes, and here we provide an alternative proof using complex analytic techniques. This alternative proof also allows us to generalize the result to a broader class of tree topologies and deletion models. Before stating our results we first introduce the tree trace reconstruction problem more precisely.

Let X be a rooted tree with unknown binary labels (i.e. $\{0, 1\}$) on its n non-root nodes. We assume that X has an ordering of its nodes, and the children of a given node have a left-to-right ordering. The goal of tree trace reconstruction is to learn the labels of X with high probability, using as few traces as possible, knowing only the deletion model, the deletion probability $q < 1$, and the tree structure of X. Throughout this paper we write 'with high probability' (w.h.p.) to mean with probability tending to 1 as $n \to \infty$.

While for strings there is a canonical model of the deletion channel, there is no such canonical model for trees. Previous work in [10] considered two natural extensions of the string deletion channel to trees: the *tree edit distance* (TED) deletion model and the *left-propagation* (LP) deletion model; see [10] for details. Here we focus on the TED model, while also introducing a new deletion model, termed *all-or-nothing* (AON), which is more 'destructive' than the other models. In both models the root never gets deleted.

Tree edit distance (TED) deletion model. Each non-root node is deleted independently with probability q and deletions are associative. When a node v gets deleted, all of the children of v now become children of the parent of v. Equivalently, contract the edge between v and its parent, retaining the label of the parent. The children of v take the place of v in the left-to-right order; in other words, the original siblings of v that are to the left of v and survive are now to the left of the children of v, and the same holds to the right of v.

All-or-nothing (AON) deletion model. Each non-root node is marked independently with probability q. If a node v is marked, then the whole subtree rooted at v is deleted. In other words, a node is deleted if and only if it is marked or it has an ancestor which is marked.

Figure 1 illustrates these two deletion models. We refer to [10] for motivation and further remarks on the TED deletion model. While the AON deletion model is significantly more destructive than the TED deletion model, an advantage of the tools we develop in this work is

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

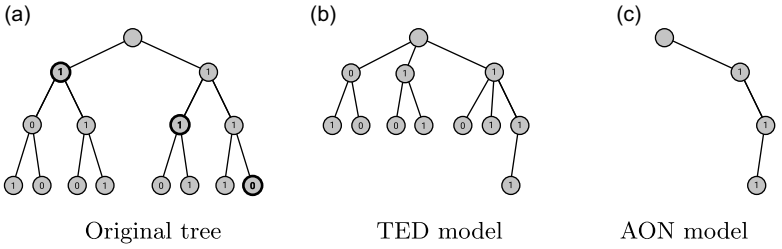

FIGURE 1. Actions of deletion models on a sample tree. (a) Original tree, with bold nodes to be deleted. (b) Resulting trace in the TED model, and (c) the AON model.

that we are able to obtain similar results for arbitrary tree topologies under the AON deletion model.

Before we state our results, we recall a result of Davies, Rácz, and Rashtchian [10, Theorem 4].

Theorem 1.1. ([10].) *In the TED model, there exists a finite constant C depending only on q such that* $\exp(Ck\log_k n)$ *traces suffice to reconstruct a complete k-ary tree on n nodes with high probability (here $k \geq 2$).*

In particular, note that the sample complexity is polynomial in n whenever k is a constant. Our first result is an alternative proof of the same result, under some mild additional assumptions, as stated below in Theorem 1.2.

Theorem 1.2. *Fix $c \in \mathbb{Z}^+$ and let $q < c/(c+1)$. There exists a finite constant C, depending only on c and q, such that for any $k > c$ the following holds: in the TED model,* $\exp(Ck\log_k n)$ *traces suffice to reconstruct a complete k-ary tree on n nodes with high probability.*

The additional assumptions compared to Theorem 1.1 are indeed mild. For instance, with $c = 1$ in the theorem above, Theorem 1.1 is recovered for $q < 1/2$. Theorem 1.2 also allows q to be arbitrarily close to 1, provided that k is at least a large enough constant.

In [10], the authors use combinatorial techniques to prove Theorem 1.1. Our proof of Theorem 1.2 uses a mean-based complex analytic approach, similar to [12], [13], [19], and [24]. In fact we can prove a slightly more general statement than Theorem 1.2 using our technique.

Theorem 1.3. *Let X be a rooted tree on n nodes with binary labels, with nodes on level ℓ all having the same number of children k_ℓ. Let $k_{\max} := \max_\ell k_\ell$ and $k_{\min} := \min_\ell k_\ell$, where the minimum goes over all levels except the last one (containing leaf nodes). If $q < c/(c+1)$ and $k_{\min} > c$ for some $c \in \mathbb{Z}^+$, then there exists a finite constant C, depending only on c and q, such that* $\exp(Ck_{\max}\log_{k_{\min}} n)$ *traces suffice to reconstruct X with high probability.*

Furthermore, with some slight modifications, our proof of Theorem 1.2 also provides a sample complexity bound for reconstructing arbitrary tree topologies in the AON deletion model.

Theorem 1.4. *Let X be a rooted tree on n nodes with binary labels, let $k_{\max}$ denote the maximum number of children a node has in X, and let d be the depth of X. In the AON model, there exists a finite constant C depending only on q such that* $\exp(Ck_{\max}d)$ *traces suffice to reconstruct X with high probability.*

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

The key idea in the above proofs is the notion of a *subtrace*, which is the subgraph of a trace that consists only of root-to-leaf paths of length d, where d is the depth of the underlying tree. In the proofs of Theorems 1.2 and 1.3 we essentially only use the information contained in these subtraces and ignore the rest of the trace. This trick is key to making the setup amenable to the mean-based complex analytic techniques.

The rest of the paper follows the following outline. We start with some preliminaries in Section 2, where we state basic tree definitions and define the notion of a subtrace more precisely. In Section 3 we present our proof of Theorem 1.3. In Section 4 we extend the methods of Section 3 to another deletion model, proving Theorem 1.4. We conclude in Section 5.

2. Preliminaries

In what follows, X denotes an underlying rooted tree of known topology alongx with binary labels associated with the n non-root nodes of the tree.

Basic tree terminology. A *tree* is an acyclic graph. A rooted tree has a special node that is designated as the *root*. A *leaf* is a node of degree 1. We say that a node v is at *level* ℓ if the graph distance between v and the root is ℓ. We say that node v is at *height* h if the largest graph distance from v to a leaf is h. *Depth* is the largest distance from the root to a leaf. We say that node u is a *child* of node v if there is an edge between u and v and v is closer to the root than u in graph distance. Similarly, we also call v the *parent* of u. More generally, v is an *ancestor* of u if there exists a path $v = x_0, \ldots, x_n = u$ such that x_i is closer to the root than x_{i+1} for every $i \in \{0, 1, \ldots, n-1\}$. A *complete k-ary tree* is a tree in which every non-leaf node has k children.

Subtrace augmentation. Above we defined the subtrace Z as the subgraph of the trace Y containing all root-to-leaf paths of length d, where d is the depth of X. In what follows, it will be helpful to slightly modify the definition of the subtrace by augmenting Z to Z' such that Z' is a complete k-ary tree that contains Z as a subgraph. Given Z, we construct Z' recursively as follows. We begin by setting $Z' := Z$. If the root of Z' currently has fewer than k children, then add more child nodes to the root to the right of the existing children and label them 0. Now, consider the leftmost node in level 1 of Z'. If it has fewer than k children, add new children to the right of the existing children of this node and label them 0. Then repeat the same procedure for the second leftmost node in level 1. Continue this procedure left to right for each level, moving from top to bottom of the tree. See Figure 2 for an illustration of this process. In Section 3, when we mention the subtrace of X we mean the augmented subtrace, constructed as described here in the case of k-ary trees. We will slightly modify the notion of an augmented subtrace for the other tree topologies we will be considering.

3. Reconstruction under the TED model

In this section we prove Theorem 1.3. Then Theorem 1.2 will follow as an immediate corollary with $k_\ell = k$. The proof takes inspiration from [12], [13], [19], and [24]. We begin by computing, for every node in the original tree, its probability of survival in a subtrace. We then derive a multivariate complex generating function for every level ℓ, with random coefficients corresponding to the labels of nodes in the subtrace. Finally, we show how we can 'average' the subtraces to determine the correct labeling for each level of the original tree with high probability.

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

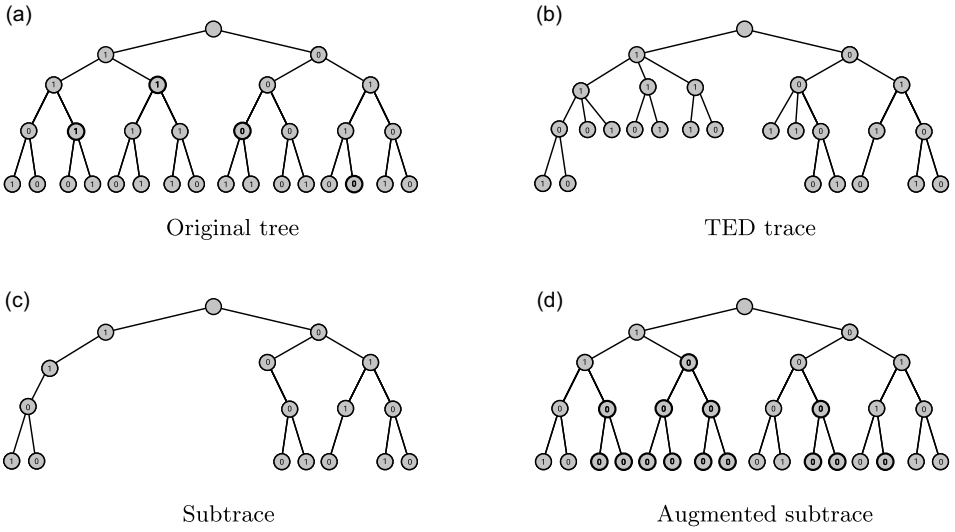

FIGURE 2. Construction of an augmented subtrace. (a) Original tree, with bold nodes to be deleted. (b) Resulting trace under the TED deletion model. (c) Subtrace. (d) Augmented subtrace, with bold nodes corresponding to the padding 0s.

Before we proceed with the proof, we must clarify the notion of a subtrace. In Section 2 we described the notion of an augmented subtrace for a k-ary tree. More generally, for trees in the setting of Theorem 1.3, we define an augmented subtrace in a similar way; the key point is that the underlying tree structure of the augmented subtrace is the same as the underlying tree structure of X. That is, we start with the root of the subtrace Z, and if it has less than k_0 children, we add nodes with 0 labels to the right of its existing children, until the root has k_0 children in total. We then move on to the leftmost node on level 1 and add new children with label 0 to the right of its existing children, until it has k_1 children. We continue in this fashion from left to right on each level, ensuring that each node on level ℓ has k_ℓ children, moving from top to bottom of the tree.

3.1. Computing the probability of node survival in a subtrace

Let d denote the depth of the original tree X. Let Y denote a trace of X and let Z denote the corresponding subtrace obtained from Y. Observe that a node v at level ℓ of X survives in the subtrace Z if and only if a root-to-leaf path that includes v survives in the trace Y. Furthermore, there exists exactly one path from the root to v, which survives in Y with probability $(1-q)^{\ell-1}$ (since each of the $\ell-1$ non-root ancestors of v has to survive independently). Let $p_{d-\ell}$ denote the probability that no v-to-leaf path survives in Y. Thus

$$\mathbb{P}(v \text{ survives in } Z) = (1-q)^{\ell-1}(1-p_{d-\ell}).$$

Thus we can see that it suffices to compute p_h for $h \in \{0, 1, \ldots, d-1\}$ in order to compute the probability of survival of v in a subtrace. The rest of this subsection is thus dedicated to understanding $\{p_h\}_{h=0}^{d-1}$. We will not find an explicit expression for p_h, but rather derive a recurrence relation for p_h, which will prove to be good enough for us.

Let v denote a vertex at height h, which is the root of the subtree under consideration. There are two events that contribute to p_h. Either v gets deleted (this happens with probability q) or all

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

of the k_{d-h} (recall that vertices on level ℓ each have k_ℓ children) subtrees rooted at the children of v do not have a surviving root-to-leaf path in the subtrace $\left(\text{this happens with probability } p_{h-1}^{k_{d-h}}\right)$. Thus we have the following recurrence relation: for every $h \geq 0$ we have

$$p_{h+1} = q + (1-q)p_h^{k_{d-h-1}}; \tag{3.1}$$

furthermore, the initial condition satisfies $p_0 = q$. This recursion allows us to compute $\{p_h\}_{h=0}^{d-1}$. We now prove the following statement about this recursion, which will be useful later on.

Lemma 3.1. *Suppose that* $0 < q < c/(c+1)$ *and* $k_{\min} > c$ *for some* $c \in \mathbb{Z}^+$. *There exists* $p' < 1$, *depending only on* c *and* q, *such that* $p_i \leq p' < 1$ *for every* $i \geq 0$.

Proof. The function $f(p) := 1 + p + \cdots + p^c$ is continuous and strictly increasing on $[0, 1]$ with $f(0) = 1$ and $f(1) = c + 1$. The assumption $q \in (0, c/(c+1))$ implies that $1/(1-q) \in (1, c+1)$, so there exists a unique $p' \in (0, 1)$ such that $f(p') = 1/(1-q)$. By construction p' is a function of c and q. We will show by induction that $p_i \leq p'$ for every $i \geq 0$.

First, observe that $f(p) \leq \sum_{m \geq 0} p^m = 1/(1-p)$. Thus $1/(1-p') \geq 1/(1-q)$ and so $q \leq p'$. Since $p_0 = q$, this proves the base case of the induction.

For the induction step, first note that if $p \in (0, 1)$, then $f(p) = \left(1 - p^{c+1}\right)/(1-p)$. Therefore the equation $f(p') = 1/(1-q)$ implies that $(1-q)(p')^{c+1} = p' - q$. So if $p_i \leq p'$, then (3.1) implies that $p_{i+1} = q + (1-q)p_i^{k_{d-i-1}} \leq q + (1-q)(p')^{c+1} = q + (p'-q) = p'$, where we also used the assumption $k_{\min} \geq c + 1$ in the inequality. □

We can interpret the result of Lemma 3.1 as follows. If we restrict that $k_{\min} > c$ and $q < c/(c+1)$, then there exists at least one surviving root-to-leaf path in the trace with positive probability even as $n \to \infty$. Existence of root-to-leaf paths in the trace is essential for the subtrace to carry any meaningful information, which is why this lemma will become important later in the proof.

3.2. Generating function derivation

We begin by introducing some additional notation. Let $b_{\ell,i}$ denote the label of the node located at level ℓ in position i from the left in the original tree X; we start the indexing of i from 0, so $b_{\ell,0}$ is the label of the leftmost vertex on level ℓ. Similarly, let $a_{\ell,i}$ denote the label of the node located at level ℓ in position i from the left in the subtrace Z. Observe that for $i \in \{0, 1, \ldots, k_0 \ldots k_{\ell-1} - 1\}$ we can write $i = i_{\ell-1}k_0 \ldots k_{\ell-2} + i_{\ell-2}k_0 \ldots k_{\ell-3} + \cdots + i_0$ with $i_\ell \in \{0, \ldots, k_\ell - 1\}$, that is, if, say, all $k_\ell = k$, we have that $i_{\ell-1}i_{\ell-2} \ldots i_0$ is the base k representation of i. To abbreviate notation, we will write $a_{\ell,i} = a_{\ell,i_{\ell-1}k_0 \ldots k_{\ell-1} + i_{\ell-2}k_0 \ldots k_{\ell-2} + \cdots + i_0}$ as simply $a_{\ell,i_{\ell-1} \ldots i_0}$.

We introduce, for every level ℓ, a multivariate complex generating function whose coefficients are the labels of the nodes at level ℓ of a subtrace Z. Specifically, we introduce complex variables $w_0, \ldots, w_{\ell-1}$ for each position in the representation of i, and define

$$A_\ell(w) := \sum_{i_0=0}^{k_0-1} \cdots \sum_{i_{\ell-1}=0}^{k_{\ell-1}-1} a_{\ell,i_{\ell-1} \ldots i_0} w_{\ell-1}^{i_{\ell-1}} \ldots w_0^{i_0}. \tag{3.2}$$

We are now ready to state the main result of this subsection, which computes the expectation of this generating function.

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

Lemma 3.2. *For every $\ell \in \{1, \dots, d\}$, we have*

$$\mathbb{E}[A_\ell(w)] = (1-q)^{\ell-1}\big(1-p_{d-\ell}\big) \sum_{i_0=0}^{k_0-1} \cdots \sum_{i_{\ell-1}=0}^{k_{\ell-1}-1} b_{\ell,i_{\ell-1}\dots i_0} \prod_{m=0}^{\ell-1} \big((1-p_{d-\ell+m})w_m + p_{d-\ell+m}\big)^{i_m}. \tag{3.3}$$

This lemma is useful because the right-hand side of (3.3) contains the labels of the nodes on level ℓ of X, while the left-hand side can be estimated by averaging over subtraces.

Proof of Lemma 3.2. By linearity of expectation we have

$$\mathbb{E}[A_\ell(w)] = \sum_{i_0=0}^{k_0-1} \cdots \sum_{i_{\ell-1}=0}^{k_{\ell-1}-1} \mathbb{E}\big[a_{\ell,i_{\ell-1}\dots i_0}\big] w_{\ell-1}^{i_{\ell-1}} \dots w_0^{i_0}, \tag{3.4}$$

so our goal is to compute $\mathbb{E}[a_{\ell,i_{\ell-1}\dots i_0}]$. For node $i = i_{\ell-1}\dots i_0$ on level ℓ, we may interpret each digit i_m in the representation of i as follows: consider node i's ancestor on level $\ell - m$; the horizontal position of this node amongst its siblings is i_m. Thus, if the original bit $b_{\ell,j}$ survives in the subtrace, it can only end up in position $i = i_{\ell-1}i_{\ell-2}\dots i_0$ on level ℓ satisfying $i_m \le j_m$ for every m. If the mth digit of the location of $b_{\ell,j}$ in the subtrace is $i_{\ell-m}$, then exactly $j_{\ell-m} - i_{\ell-m}$ siblings left of the ancestor of $b_{\ell,j}$ on level $\ell - m$ must have been deleted and the ancestor of $b_{\ell,j}$ on level $\ell - m$ must have survived in the subtrace. Thus the probability that the bit $b_{\ell,j}$ of the original tree is sent to $a_{\ell,i}$ in the subtrace is given by

$$\begin{aligned}
\mathbb{P}\big(b_{\ell,j_{\ell-1}\dots j_0} \to a_{\ell,i_{\ell-1}\dots i_0}\big) &= \binom{j_0}{i_0} p_{d-\ell}^{j_0-i_0}\big(1-p_{d-\ell}\big)^{i_0+1} \times \binom{j_1}{i_1} p_{d-\ell+1}^{j_1-i_1}\big(1-p_{d-\ell+1}\big)^{i_1}(1-q) \\
&\quad \times \cdots \times \binom{j_{\ell-1}}{i_{\ell-1}} p_{d-1}^{j_{\ell-1}-i_{\ell-1}}(1-p_{d-1})^{i_{\ell-1}}(1-q) \\
&= (1-q)^{\ell-1}\big(1-p_{d-\ell}\big) \prod_{m=0}^{\ell-1} \binom{j_m}{i_m} p_{d-\ell+m}^{j_m-i_m}\big(1-p_{d-\ell+m}\big)^{i_m}.
\end{aligned}$$

Summing over all j satisfying $i_m \le j_m$ for every m, and plugging into (3.4), we obtain

$$\begin{aligned}
\mathbb{E}[A_\ell(w)] &= (1-q)^{\ell-1}\big(1-p_{d-\ell}\big) \\
&\quad \times \sum_{i_0=0}^{k_0-1} \cdots \sum_{i_{\ell-1}=0}^{k_{\ell-1}-1} \sum_{j_0=i_0}^{k_0-1} \cdots \sum_{j_{\ell-1}=i_{\ell-1}}^{k_{\ell-1}-1} b_{\ell,j_{\ell-1}\dots j_0} \prod_{m=0}^{\ell-1} \binom{j_m}{i_m} p_{d-\ell+m}^{j_m-i_m}\big(1-p_{d-\ell+m}\big)^{i_m} w_m^{i_m}.
\end{aligned}$$

Interchanging the order of summations and using the binomial theorem (ℓ times), we obtain (3.3). □

3.3. Bounding the modulus of the generating function

Here we prove a simple lower bound on the modulus of a multivariate Littlewood polynomial. This bound will extend to the generating function computed above for appropriate choices of w_m. The argument presented here is inspired by the method of proof of [19, Lemma 4]. Throughout the paper we let $\mathbb{D}$ denote the unit disk in the complex plane and let $\partial\mathbb{D}$ denote its boundary.

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

Lemma 3.3. *Let $F(z_0, \dots, z_{\ell-1})$ be a non-zero multivariate polynomial with monomial coefficients in $\{-1, 0, 1\}$. Then*

$$\sup_{z_0,\dots,z_{\ell-1}\in\partial\mathbb{D}} |F(z_0, \dots, z_{\ell-1})| \geq 1.$$

Proof. We define a sequence of polynomials $\{F_i\}_{i=0}^{\ell-1}$ inductively as follows, where F_i is a function of the variables $z_i, \dots, z_{\ell-1}$. First, let t_0 be the smallest power of z_0 in a monomial of F and let $F_0(z_0, \dots, z_{\ell-1}) := z_0^{-t_0} F(z_0, \dots, z_{\ell-1})$. By construction, F_0 has at least one monomial where z_0 does not appear. For $i \in \{1, \dots, \ell-1\}$, given F_{i-1} we define F_i as follows. Let t_i be the smallest power of z_i in a monomial of $F_{i-1}(0, z_i, \dots, z_{\ell-1})$ and let $F_i(z_i, \dots, z_{\ell-1}) := z_i^{-t_i} F_{i-1}(0, z_i, \dots, z_{\ell-1})$. Observe that this construction guarantees, for every i, that the polynomial $F_i(z_i, \dots, z_{\ell-1})$ has at least one monomial where z_i does not appear. In particular, the univariate polynomial $F_{\ell-1}(z_{\ell-1})$ has a non-zero constant term. Since the coefficients of the polynomial are in $\{-1, 0, 1\}$, this means that the constant term has absolute value 1, i.e. $|F_{\ell-1}(0)| = 1$.

We will now use induction to prove that for every $i \leq \ell - 1$, we have

$$\sup_{z_{\ell-1-i},\dots,z_{\ell-1}\in\partial\mathbb{D}} |F_{\ell-1-i}(z_{\ell-1-i}, \dots, z_{\ell-1})| \geq 1. \tag{3.5}$$

We begin with $i = 0$. By the maximum modulus principle, $\sup_{z\in\partial\mathbb{D}} |F_{\ell-1}(z)| \geq |F_{\ell-1}(0)| = 1$, so we move on to the inductive step. Suppose the claim holds for some i. Now let $(z^*_{\ell-1-i}, \dots, z^*_{\ell-1})$ denote the maximizer of $|F_{\ell-1-i}(z_{\ell-1-i}, \dots, z_{\ell-1})|$ with $z^*_m \in \partial\mathbb{D}$. Again, by the maximum modulus principle and the inductive hypothesis,

$$\left|F_{\ell-i}\left(0, z^*_{\ell-1-i} \dots, z^*_{\ell-1}\right)\right| = \left|\left(z^*_{\ell-1-i}\right)^{t_{\ell-1-i}}\right| \left|F_{\ell-1-i}\left(z^*_{\ell-1-i}, \dots, z^*_{\ell-1}\right)\right| \geq 1.$$

By another application of maximum modulus principle and the inequality of the previous display, it follows that

$$\sup_{z_{\ell-i},\dots,z_{\ell-1}\in\partial\mathbb{D}} |F_{\ell-i}(z_{\ell-i}, \dots, z_{\ell-1})| \geq \left|F_{\ell-i}\left(0, z^*_{\ell-1-i}, \dots, z^*_{\ell-1}\right)\right| \geq 1.$$

Thus (3.5) holds and the desired conclusion follows with $i = \ell - 1$. □

3.4. Finishing the proof of Theorem 1.3

Proof of Theorem 1.3. Let X' and X'' be two trees on n non-root nodes with different binary node labels such that every vertex on level ℓ has k_ℓ children. Our first goal is to distinguish between X' and X'' using subtraces. At the end of the proof we will then explain how to estimate the original tree X using subtraces.

Since X' and X'' have different labels, there exists at least one level of the tree where the node labels differ. Call the minimal such level $\ell_* = \ell_*(X', X'')$; we will use this level of the subtraces to distinguish between X' and X''. Let

$$\left\{b'_{\ell_*,i} \colon i \in \left\{0, 1, \dots, k_0 \dots k_{\ell_*-1} - 1\right\}\right\} \quad \text{and} \quad \left\{b''_{\ell_*,i} \colon i \in \left\{0, 1, \dots, k_0 \dots k_{\ell_*-1} - 1\right\}\right\}$$

denote the labels on level ℓ_* of X' and X'', respectively. Furthermore, for every i define $b_{\ell_*,i} := b'_{\ell_*,i} - b''_{\ell_*,i}$. By construction, $b_{\ell_*,i} \in \{-1, 0, 1\}$ for every i, and there exists i such that $b_{\ell_*,i} \neq 0$. Let Z' and Z'' be subtraces obtained from X' and X'', respectively, and let

$$\left\{Z'_{\ell_*,i} \colon i \in \left\{0, 1, \dots, k_0 \dots k_{\ell_*-1} - 1\right\}\right\} \quad \text{and} \quad \left\{Z''_{\ell_*,i} \colon i \in \left\{0, 1, \dots, k_0 \dots k_{\ell_*-1} - 1\right\}\right\}$$

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

denote the labels on level ℓ_* of Z' and Z'', respectively. By Lemma 3.2 we have

$$\mathbb{E}\left[\sum_{t_0=0}^{k_0-1}\cdots\sum_{t_{\ell_*-1}=0}^{k_{\ell_*-1}-1} Z'_{\ell_*,t_{\ell_*-1}\ldots t_0}\prod_{m=0}^{\ell_*-1} w_m^{t_m}\right]-\mathbb{E}\left[\sum_{t_0=0}^{k_0-1}\cdots\sum_{t_{\ell_*-1}=0}^{k_{\ell_*-1}-1} Z''_{\ell_*,t_{\ell_*-1}\ldots t_0}\prod_{m=0}^{t_{\ell_*-1}} w_m^{t_m}\right]$$
$$=(1-q)^{\ell_*-1}(1-p_{d-\ell_*})\sum_{t_0=0}^{k_0-1}\cdots\sum_{t_{\ell_*-1}=0}^{k_{\ell_*-1}-1} b_{\ell_*,t_{\ell_*-1}\ldots t_0}\prod_{m=0}^{\ell_*-1}\left((1-p_{d-\ell_*+m})w_m+p_{d-\ell_*+m}\right)^{t_m}.$$

Now define the multivariate polynomial $B(z)$ in the variables $z=(z_0,\ldots,z_{\ell_*-1})$ as follows:

$$B(z):=\sum_{t_0=0}^{k_0-1}\cdots\sum_{t_{\ell_*-1}=0}^{k_{\ell_*-1}-1} b_{\ell_*,t_{\ell_*-1}\ldots t_0}\prod_{m=0}^{\ell_*-1} z_m^{t_m}.$$

Lemma 3.3 implies that there exists $z^*=\left(z_0^*,\ldots,z_{\ell_*-1}^*\right)$ such that $z_m^*\in\partial\mathbb{D}$ for every $m\in\{0,\ldots,\ell_*-1\}$ and

$$|B(z^*)|\geq 1.$$

For $m\in\{0,\ldots,\ell_*-1\}$, let

$$w_m^*:=\frac{z_m^*-p_{d-\ell_*+m}}{1-p_{d-\ell_*+m}}.$$

Note that the polynomial B is a function of X' and X'', and thus so is z^* and also $w^*=\left(w_0^*,\ldots,w_{\ell_*-1}^*\right)$. Putting together the four previous displays and using the triangle inequality, we obtain

$$\sum_{t_0=0}^{k_0-1}\cdots\sum_{t_{\ell_*-1}=0}^{k_{\ell_*-1}-1}\left|\mathbb{E}\left[Z'_{\ell_*,t_{\ell_*-1}\ldots t_0}\right]-\mathbb{E}\left[Z''_{\ell_*,t_{\ell_*-1}\ldots t_0}\right]\right|\prod_{m=0}^{\ell_*-1}|w_m^*|^{t_m}\geq(1-q)^{\ell_*-1}\left(1-p_{d-\ell_*}\right).\tag{3.6}$$

Next we estimate $|w_m^*|$. By the definition of w_m^* and the triangle inequality, we have

$$|w_m^*|=\frac{|z_m^*-p_{d-\ell_*+m}|}{1-p_{d-\ell_*+m}}\leq\frac{|z_m^*|+p_{d-\ell_*+m}}{1-p_{d-\ell_*+m}}\leq\frac{2}{1-p'},\tag{3.7}$$

where in the last inequality we used that $|z_m^*|=1$ and that $p_{d-\ell_*+m}\leq p'<1$ (from Lemma 3.1). Note that p' is a constant that depends only on c and q (recall that c is an input to the theorem). The bound in (3.7) implies that

$$\prod_{m=0}^{\ell_*-1}|w_m^*|^{t_m}\leq\left(\frac{2}{1-p'}\right)^{k_{\max}\ell_*}.$$

Plugging this back into (3.6) (and using that $p_{d-\ell_*}\leq p'$), we get

$$\sum_{t_0=0}^{k-1}\cdots\sum_{t_{\ell_*-1}=0}^{k-1}\left|\mathbb{E}\left[Z'_{\ell_*,t_{\ell_*-1}\ldots t_0}\right]-\mathbb{E}\left[Z''_{\ell_*,t_{\ell_*-1}\ldots t_0}\right]\right|\geq(1-q)^{\ell_*-1}(1-p')^{k_{\max}\ell_*+1}(1/2)^{k_{\max}\ell_*}.$$

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

Thus, by the pigeonhole principle, there exists $i_* \in \{0, 1, \dots, k_0 \dots k_{\ell_*-1} - 1\}$ such that

$$\begin{aligned} \left|\mathbb{E}[Z'_{\ell_*,i_*}] - \mathbb{E}[Z''_{\ell_*,i_*}]\right| &\geq \frac{(1-q)^{\ell_*-1}(1-p')^{k_{\max}\ell_*+1}(1/2)^{k_{\max}\ell_*}}{k_{\max}^{\ell_*}} \\ &\geq \exp\left(-Ck_{\max}\ell_*\right) \\ &\geq \exp\left(-Ck_{\max}\log_{k_{\min}} n\right), \end{aligned} \tag{3.8}$$

where the second inequality holds for a large enough constant C that depends only on c and q, while the third inequality is because the depth of the tree is $\leq \log_{k_{\min}} n$. Note that i_* is a function of X' and X''.

Now suppose that we sample T traces of X from the TED deletion channel and let $Z^1, \dots, Z^T$ denote the corresponding subtraces. Let X' and X'' be two complete labeled trees of appropriate topologies with different labels, and recall the definitions of $\ell_* = \ell_*(X', X'')$ and $i_* = i_*(X', X'')$ from above. We say that X' *beats* X'' (with respect to these samples) if

$$\left|\frac{1}{T}\sum_{t=1}^{T} Z^t_{\ell_*,i_*} - \mathbb{E}[Z'_{\ell_*,i_*}]\right| < \left|\frac{1}{T}\sum_{t=1}^{T} Z^t_{\ell_*,i_*} - \mathbb{E}[Z''_{\ell_*,i_*}]\right|.$$

We are now ready to define our estimate $\widehat{X}$ of the labels of the original tree. If there exists a tree X' that beats every other tree of the same topology (with respect to these samples), then we let $\widehat{X} := X'$. Otherwise, define $\widehat{X}$ arbitrarily.

Finally, we show that this estimate is correct with high probability. Let

$$\eta := \exp\left(-Ck_{\max}\log_{k_{\min}} n\right).$$

By a union bound and a Chernoff bound (using (3.8)), the probability that the estimate is incorrect is bounded by

$$\begin{aligned} \mathbb{P}(\widehat{X} \neq X) &\leq \sum_{X' :\, X' \neq X} \mathbb{P}(X' \text{ beats } X) \\ &\leq 2^n \exp\left(-T\eta^2/2\right) \\ &= 2^n \exp\left(-\frac{T}{2}\exp\left(-2Ck_{\max}\log_{k_{\min}} n\right)\right). \end{aligned}$$

Choosing $T = \exp\left(3Ck_{\max}\log_{k_{\min}} n\right)$, the right-hand side of the display above tends to 0. □

4. Reconstruction under the AON model

The method of proof shown in the previous section naturally lends itself to the results of Theorem 1.4 for the AON deletion model. The proof is almost entirely identical to the one presented above, so we will only highlight the new ideas below and leave the details to the reader.

We begin by first proving Theorem 1.4 for complete k-ary trees. We will then generalize to arbitrary tree topologies. Importantly, in the AON model we will work directly with the tree traces, as opposed to the subtraces as we did previously. As described in Section 2, we augment each trace Y with additional nodes with 0 labels to form a k-ary tree. In what follows, when we say 'trace' we mean this augmented trace.

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

Theorem 4.1. *In the AON model, there exists a finite constant C depending only on q such that* $\exp(Ck \log_k n)$ *traces suffice to reconstruct a complete k-ary tree on n nodes w.h.p. (here $k \geq 2$).*

Proof. We may define $A_\ell(w)$, the generating function for level ℓ, exactly as in (3.2). The following lemma is the analog of Lemma 3.2; we omit its proof, since it is analogous to that of Lemma 3.2. □

Lemma 4.1. *For every $\ell \in \{1, \ldots, d\}$, we have*

$$\mathbb{E}[A_\ell(w)] = (1-q)^\ell \sum_{t_0=0}^{k-1} \cdots \sum_{t_{\ell-1}=0}^{k-1} b_{\ell,t_{\ell-1}\ldots t_0} \prod_{m=0}^{\ell-1} ((1-q)w_m + q)^{t_m}.$$

With this lemma in place, the remainder of the proof is almost identical to Section 3.4. The polynomial $B(z)$ and hence also z^* are as before. Now we define $w_m^* := \left(z_m^* - q\right)/(1-q)$. The right-hand side of (3.6) becomes $(1-q)^{\ell_*}$. The analog of (3.7) becomes the inequality $|w_m^*| \leq 2/(1-q)$; moreover, wherever p' appears in Section 3.4, it is replaced by q here. Altogether, we find that there exists $i_* \in \{0, 1, \ldots, k^{\ell_*} - 1\}$ such that

$$\left|\mathbb{E}\left[Z'_{\ell_*,i_*}\right] - \mathbb{E}\left[Z''_{\ell_*,i_*}\right]\right| \geq \exp(-Ck\ell_*) \geq \exp(-Ckd) \geq \exp(-Ck \log_k n),$$

where the first inequality holds for a large enough constant C that depends only on q. The rest of the proof is identical to Section 3.4, showing that $T = \exp(3Ckd) = \exp(3Ck \log_k n)$ traces suffice. □

Proof of Theorem 1.4. Suppose that X is a rooted tree with arbitrary topology and let $k_{\max}$ denote the largest number of children a node in X has. Once we sample a trace Y from X, we form an augmented trace similarly to how we do it when X is a k-ary tree, except now we add nodes with 0 labels to ensure that each node has $k_{\max}$ children. Thus each augmented trace is a complete $k_{\max}$-ary tree. Now let X' denote a $k_{\max}$-ary tree obtained by augmenting X to a $k_{\max}$-ary tree in the same fashion that we augment traces of X to a $k_{\max}$-ary tree.

As before, for each node i on level ℓ of X, there is a unique representation $i = i_{\ell-1} \ldots i_0$ where i_m is the position of node i's ancestor on level $\ell - m$ among its siblings. Importantly, for every node in X, its representation in X' is the same. This fact, together with the augmentation construction, implies that $\mathbb{E}[a_{\ell,t_{\ell-1}\ldots t_0}]$ for the node $a_{\ell,t_{\ell-1}\ldots t_0}$ in Y is identical to $\mathbb{E}[a_{\ell,t_{\ell-1}\ldots t_0}]$ for the node $a_{\ell,t_{\ell-1}\ldots t_0}$ in Y', which is a trace sampled from X'. Therefore we can use the procedure presented in Theorem 4.1 to reconstruct X' w.h.p. using $T = \exp(Ck_{\max}d)$ traces sampled from X. By taking the appropriate subgraph of X', we can thus reconstruct X as well. □

Remark 4.1. We can observe that the distribution of the subtraces obtained from TED and AON deletion models applied to a k-ary tree is identical. Thus Theorem 4.1 (with the addition of a mild condition on k) can also be viewed as a direct consequence of the argument of Theorem 1.2 since the previous observation implies that the generating function (3.2) will be identically distributed for both AON and TED models. However, viewed in this way, Theorem 4.1 would not lend itself easily to the generalization obtained in Theorem 1.4. The basic idea of Theorem 1.4 is that we can think of trace reconstruction of a tree with an arbitrary topology under the AON deletion model as trace reconstruction of a $k_{\max}$-ary tree produced by padding the original tree with additional leaves with label 0. This operation does not change the

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

augmented trace distribution of the underlying tree, but it does change the augmented subtrace distribution, which is why this argument would fail if we tried using subtrace reconstruction. Hence we present a more complex proof of Theorem 4.1 here, despite there being a simpler argument available.

5. Conclusion

In this work we introduce the notion of a subtrace and demonstrate its utility in analyzing traces produced by the deletion channel in the tree trace reconstruction problem. We provide a novel algorithm for the reconstruction of complete k-ary trees, which matches the sample complexity of the combinatorial approach of [10], by applying mean-based complex analytic tools to the subtrace. This technique also allows us to reconstruct trees with more general topologies in the TED deletion model, specifically trees where the nodes at every level have the same number of children (with this number varying across levels).

However, many questions remain unanswered; we hope that the ideas introduced here will help address them. In particular, how can we reconstruct, under the TED deletion model, arbitrary trees where all leaves are on the same level? Since the notion of a subtrace is well-defined for such trees, we hope that the proof technique presented here can somehow be generalized to answer this question.

In the spirit of understanding the interplay between combinatorial and complex analytic methods, one may ask if Theorem 1.3 can also be proved using combinatorial methods (either by extending ideas in [10] or by using new ideas). Finally, for various applications it may be relevant to consider other deletion models. For instance, consider the AON model where subtrees are not deleted, but rather the node labels are replaced with random labels; this is a generalization of the TrimSuffixAndExtend model for strings (see [4]). The proof technique in Section 4 may be extended to this model (we leave the details to the reader), and it is natural to explore how broadly it is applicable.

Acknowledgements

We thank Sami Davies and Cyrus Rashtchian for helpful feedback and discussions. We also thank an anonymous reviewer for their careful and helpful feedback.

Funding information

Miklós Z. Rácz is supported in part by NSF grant DMS 1811724 and by a Princeton SEAS Innovation Award.

Competing interests

There were no competing interests to declare which arose during the preparation or publication process of this article.

References

[1] Ban, F., Chen, X., Freilich, A., Servedio, R. A. and Sinha, S. (2019). Beyond trace reconstruction: population recovery from the deletion channel. In *60th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 745–768.

[2] Ban, F., Chen, X., Servedio, R. A. and Sinha, S. (2019). Efficient average-case population recovery in the presence of insertions and deletions. In *Approximation, Randomization, and Combinatorial Optimization:*

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press

Algorithms and Techniques (APPROX/RANDOM) (LIPIcs **145**), pp. 44:1–44:18. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[3] Batu, T., Kannan, S., Khanna, S. and McGregor, A. (2004). Reconstructing strings from random traces. In *Proceedings of the 15th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pp. 910–918.

[4] Bhardwaj, V., Pevzner, P. A., Rashtchian, C. and Safonova, Y. (2021). Trace reconstruction problems in computational biology. *IEEE Trans. Inf. Theory* **67**, 3295–3314.

[5] Brakensiek, J., Li, R., and Spang, B. (2020). Coded trace reconstruction in a constant number of traces. In *Proceedings of the IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 482–493.

[6] Chase, Z. (2020). New upper bounds for trace reconstruction. Available at arXiv:2009.03296.

[7] Chase, Z. (2021). New lower bounds for trace reconstruction. *Ann. Inst. H. Poincaré Prob. Statist.*, to appear.

[8] Chen, X., De, A., Lee, C. H., Servedio, R. A. and Sinha, S. (2021). Polynomial-time trace reconstruction in the smoothed complexity model. In *Proceedings of the 32nd Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pp. 54–73.

[9] Cheraghchi, M., Gabrys, R., Milenkovic, O. and Ribeiro, J. (2020). Coded trace reconstruction. *IEEE Trans. Inf. Theory* **66**, 6084–6103.

[10] Davies, S., Rácz, M. Z. and Rashtchian, C. (2021). Reconstructing trees from traces. *Ann. Appl. Prob.*, to appear.

[11] Davies, S., Racz, M. Z., Rashtchian, C. and Schiffer, B G. (2020). Approximate trace reconstruction. Available at arXiv:2012.06713.

[12] De, A., O'Donnell, R. and Servedio, R. A. (2017). Optimal mean-based algorithms for trace reconstruction. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC),* pp. 1047–1056.

[13] De, A., O'Donnell, R. and Servedio, R. A. (2019). Optimal mean-based algorithms for trace reconstruction. *Ann. Appl. Prob.* **29**, 851–874.

[14] Grigorescu, E., Sudan, M. and Zhu, M. (2020). Limitations of mean-based algorithms for trace reconstruction at small distance. Available at arXiv:2011.13737.

[15] Hartung, L., Holden, N. and Peres, Y. (2018). Trace reconstruction with varying deletion probabilities. In *Proceedings of the 15th Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pp. 54–61.

[16] Holden, N. and Lyons, R. (2020). Lower bounds for trace reconstruction. *Ann. Appl. Prob.* **30**, 503–525.

[17] Holden, N., Pemantle, R., Peres, Y. and Zhai, A. (2020). Subpolynomial trace reconstruction for random strings and arbitrary deletion probability. *Math. Statist. Learning* **2**, 275–309.

[18] Holenstein, T., Mitzenmacher, M., Panigrahy, R. and Wieder, U. (2008). Trace reconstruction with constant deletion probability and related results. In *Proceedings of the 19th ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pp. 389–398.

[19] Krishnamurthy, A., Mazumdar, A., McGregor, A. and Pal, S. (2019). Trace reconstruction: generalized and parameterized. In *27th Annual European Symposium on Algorithms (ESA 2019)*, pp. 68:1–68:25.

[20] Levenshtein, V. I. (2001). Efficient reconstruction of sequences. *IEEE Trans. Inf. Theory* **47**, 2–22.

[21] Maranzatto, T. J. (2020). Tree trace reconstruction: some results. Thesis, New College of Florida, Sarasota, FL.

[22] Narayanan, S. (2021). Population recovery from the deletion channel: nearly matching trace reconstruction bounds. In *Proceedings of the ACM–SIAM Symposium on Discrete Algorithms (SODA).*

[23] Narayanan, S. and Renk, M. (2021). Circular trace reconstruction. In *Proceedings of the 12th Innovations in Theoretical Computer Science Conference (ITCS)* (LIPIcs 185), pp. 18:1–18:18. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[24] Nazarov, F. and Peres, Y. (2017). Trace reconstruction with $\exp(O(n^{1/3}))$ samples. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC),* pp. 1042–1046.

https://doi.org/10.1017/jpr.2022.81 Published online by Cambridge University Press