The trace reconstruction problem for spider graphs

Alec Sun[*]   William Yue[†]

September 20, 2022

Abstract

We study the trace reconstruction problem for spider graphs. Let n be the number of nodes of a spider and d be the length of each leg, and suppose that we are given independent traces of the spider from a deletion channel in which each non-root node is deleted with probability q. This is a natural generalization of the string trace reconstruction problem in theoretical computer science, which corresponds to the special case where the spider has one leg. In the regime where $d \geq \log_{1/q}(n)$, the problem can be reduced to the vanilla string trace reconstruction problem. We thus study the more interesting regime $d \leq \log_{1/q}(n)$, in which entire legs of the spider are deleted with non-negligible probability. We describe an algorithm that reconstructs spiders with high probability using $\exp\left(\mathcal{O}\left(\frac{(nq^d)^{1/3}}{d^{1/3}} (\log n)^{2/3} \right) \right)$ traces. Our algorithm works for all deletion probabilities $q \in (0, 1)$.

Keywords: Trace reconstruction, Graph algorithms, Littlewood polynomials

1 Introduction

The *string trace reconstruction problem*, first introduced in 1997 by Levenshtein [17], is concerned with reconstructing an unknown *seed* string using only noisy samples of the data. The unknown seed string is passed into some noisy channel multiple times, and the resulting error-prone copies are referred to as *traces*. The goal is to use multiple traces to reconstruct the original seed string with high probability. Levenshtein solved the trace reconstruction problem for a *substitution channel*, where each symbol of the seed string is mutated independently with constant probability. In 2004,

---

[*]Carnegie Mellon University, `alecsun@andrew.cmu.edu`

[†]Massachusetts Institute of Technology, `willyue@mit.edu`

Batu, Kannan, Khanna, and McGregor [2] analyzed the problem for a *deletion channel*, where symbols of the seed string are each deleted independently with constant probability. The string trace reconstruction problem has applications to computational biology, specifically in the new rapidly-evolving fields of DNA data storage and personalized immunogenics. For example, one might want to reconstruct the correct sequence of nucleotides of a DNA sequence from several traces, each of which has many deletion mutations.

It is critical to minimize the number of traces required to reconstruct the seed string with high probability. For example, in the application of DNA data storage, reducing the number of traces results in lower sequencing cost and time [3]. However, despite a wealth of recent work and attention on the deletion channel string trace reconstruction problem, for example [5, 10, 11, 12, 13, 14, 18, 20, 21], the current best upper and lower bounds for the number of traces necessary to reconstruct the seed string with high probability remain at $\exp(\mathcal{O}(n^{1/5}))$ [4] and $\tilde{\Omega}(n^{3/2})$ [5, 12], respectively, where n is the length of the seed string. We remark that a lower bound of $\exp(\mathcal{O}(n^{1/3}))$ traces was shown for mean-based algorithms, which are algorithms that only use the empirical means of individual bits in the traces for reconstruction [10, 20].

The exponential gap between upper and lower bounds for string trace reconstruction motivates studying variants of the problem for which one may be able to close the gap. Many variants have been recently proposed and studied, for example [1, 6, 7, 9, 16, 19]. We focus on a variant known as the *tree trace reconstruction problem* introduced by Davies, Rácz, and Rashtchian [8]. This is a generalization of the vanilla string trace reconstruction problem where the goal is to learn a node-labeled tree, rather than a single string, using traces from a suitably-defined deletion channel. The tree trace reconstruction problem may be directly applicable as well, as research on DNA nanotechnology has demonstrated that DNA molecule structures can be assembled into trees. Recent research has also shown how to distinguish different molecular topologies, such as spiders with three arms from line DNA, using nanopores [15].

Davies et al. [8] studied the tree trace reconstruction problem for two special classes of trees: complete k-ary trees and spiders. This paper extends their work on spiders. An (n, d)-*spider* consists of a single unlabeled root node with paths of d labeled nodes attached to it. In total, there are n labeled nodes. Consider a deletion channel, formally defined in Section 2.2, in which every node is independently deleted with probability q.

When $d \geq \log_{1/q}(n)$, solving the spider trace reconstruction problem

directly reduces to the string trace reconstruction problem [8, Proposition 24]. This is because in this regime, the legs of the spider are long enough for all of the legs to survive the deletion channel with high probability, so each leg can be considered independently as its own string trace reconstruction problem. Therefore, we assume that $d \leq \log_{1/q}(n)$. In this more interesting regime, entire legs are deleted with non-negligible probability. Hence, if one looks at a single trace, it is unclear which of the legs in the seed spider the legs in the trace come from.

Davies et al. [8] proved that for deletion probabilities $q < 0.7$, there is some constant $C > 0$ that depends only on q such that $\exp(C \cdot d(nq^d)^{1/3})$ traces suffice to reconstruct an (n, d)-spider with probability $1 - \mathcal{O}(1/n)$ (we refer to this as *with high probability*). In this paper, we match this upper bound, up to polylogarithmic factors, but for the full range of deletion probabilities $q \in (0, 1)$. Furthermore, while Davies et al. [8] used a single variable generating function alongside harmonic analysis, we consider a bivariate generating function, which results in considerably simpler analysis. We use a best-match algorithm coupled with some results about bivariate Littlewood polynomials. We remark that Littlewood polynomials have also been used to analyze a different variant of trace reconstruction known as the *matrix reconstruction problem* [16].

Our main result is the following theorem:

Theorem 1.1. *Assume that $d \leq \log_{1/q}(n)$. For any fixed deletion probability $q < 1$, there exists some constant $C > 0$ that depends only on q such that*

$$\exp\left(C \cdot \frac{(nq^d)^{1/3}}{d^{1/3}}(\log n)^{2/3}\right)$$

traces suffice to reconstruct an (n, d)-spider with high probability.

Note that the upper bound in Theorem 1.1 matches the upper bound $\exp(C \cdot d(nq^d)^{1/3})$ in [8] up to polylogarithmic factors and works for all deletion probabilities $q \in (0, 1)$, not just $q < 0.7$. Furthermore, Theorem 1.1 strictly improves upon the upper bound $\exp(C \cdot d(nq^d)^{1/3})$ for all $q \in (0, 1)$ when $d = \omega(\sqrt{\log n})$.
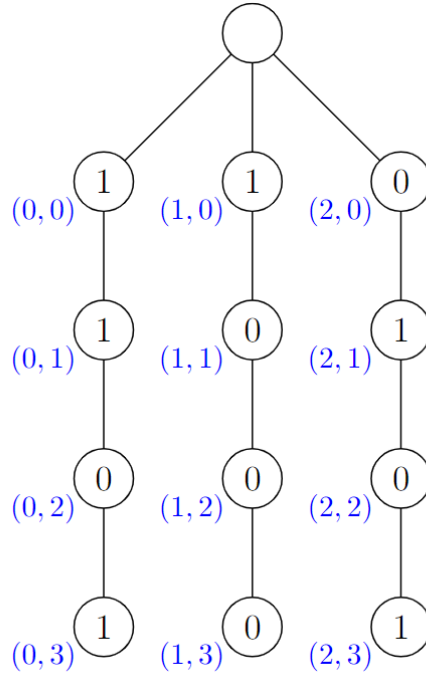
1.1 Acknowledgements

Figure 1: A binary-labeled $(12, 4)$ spider, with indexing system drawn in blue to the bottom left of each vertex.

2 Preliminaries

2.1 Rooted spiders

In this section, we define the objects to be reconstructed: rooted binary-labeled spiders X, as well as an indexing system for their nodes.

Definition 2.1. *Let n and d be positive integers, and for convenience assume that $d \mid n$. An (n, d)-spider X consists of a single unlabeled root node with $\frac{n}{d}$ paths of d nodes with binary labels from $\{0, 1\}$ emanating from it, so there are n labeled nodes in total. We refer to these paths as the* legs *of the spider.*

An example of a binary-labeled $(12, 4)$-spider is shown in Fig. 1. We index each node using two coordinates, where the first coordinate denotes the leg of the spider the node is on, and the second its depth down that leg. This is in contrast to the depth-first-search labelling in [8]. In general,

4

we may denote by $a_{i,j} \in \{0,1\}$ the label of the (i,j)-node and the set of all labels of X as $a = \{a_{i,j}\}_{0 \leq i < \frac{n}{d}, 0 \leq j < d}$. For convenience, we define the set $S := \{(i,j) \mid 0 \leq i < \frac{n}{d}, 0 \leq j < d\}$, so we can write the labels of X as $a = \{a_{i,j}\}_{(i,j) \in S}$.

2.2 Deletion channel for spiders

In the deletion channel for spiders, we start by independently selecting each non-root node for deletion with probability q. Note that we assume the root node is never deleted, as deleting the root node would disconnect the graph. When nodes are deleted, all nodes below it shift upward. If all the nodes in a leg are deleted, the entire leg disappears. If a leg disappears, the remaining legs retain the same left-to-right structure, but it is no longer clear from looking at a trace which leg in the trace corresponds to which leg in the seed.

Remark 2.2. *For trees that are not spiders, one must be more careful with describing the deletion channel. Davies et al. [8] studied two models, the Tree-Edit-Distance (TED) model and the Left-Propagation Model. However, in the case of spiders, both models equivalent to the deletion channel described above.*

For convenience in our analysis, after the deletion process we append nodes labeled 0 to the end of each shortened leg until they are of length d again. Also, if any complete legs were deleted, we add a leg of length d with all nodes labeled 0 to the right of the remaining legs. This pads the trace with 0's to form an (n,d)-spider. We refer to the resulting spider as a *trace*. We remark that this padding process may cause two originally different traces to end up becoming identical. An example of the deletion and padding process is shown in Fig. 2.

2.3 Generating function for traces of spiders

Though the deletion channel for trees is more complicated than that for strings, it turns out that one can still describe the deletion process explicitly using generating functions. These generating functions will then be used to distinguish between candidate spiders.

We begin by defining generating functions which encode the information of the possible traces of a spider:

Definition 2.3. *Let $a = \{a_{i,j}\}_{(i,j) \in S}$ denote the labels of an (n,d)-spider where $a_{i,j} \in \mathbb{R}$, and let the random variable $b = \{b_{i',j'}\}_{(i',j') \in S}$ denote the*
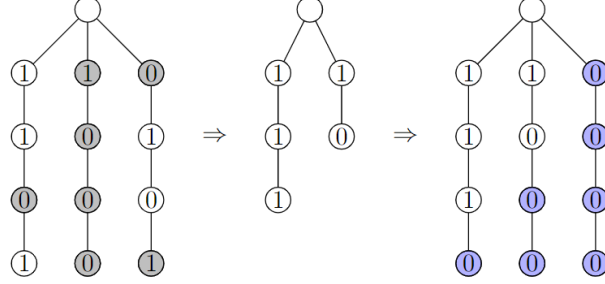
5

Figure 2: An example of the deletion channel applied on a $(12, 4)$-spider. The deleted nodes are colored gray. Then we pad nodes labeled 0, colored blue, to form a $(12, 4)$ spider.

labels of its trace from the deletion channel with deletion probability q. Define a generating function

$$\sum_{(i',j')\in S} b_{i',j'} w_1^{i'} w_2^{j'}$$

for each possible labeling $b = \{b_{i',j'}\}_{(i',j')\in S}$ of a trace.

One of our key observations is that for each (n, d)-spider, we can derive a closed-form formula for the expected value of the generating function of a trace:

Lemma 2.4. *Let $a = \{a_{i,j}\}_{(i,j)\in S}$ denote the labels of an (n, d)-spider where $a_{i,j} \in \mathbb{R}$, and let the random variable $b = \{b_{i',j'}\}_{(i',j')\in S}$ denote the labels of its trace from the deletion channel with deletion probability q. Define*

$$A_a(w_1, w_2) := \mathbb{E}\left[\sum_{(i',j')\in S} b_{i',j'} w_1^{i'} w_2^{j'} \right]$$

to be the expected value of the generating function of a trace, where the expectation is taken over the randomness of the deletion process. Then

$$A_a(w_1, w_2) = (1 - q) \sum_{(i,j)\in S} a_{i,j}(q^d + (1 - q^d)w_1)^i(q + (1 - q)w_2)^j$$

for all $w_1, w_2 \in \mathbb{C}$.

Proof. Note that the coordinates of a specific node can only decrease after the deletion process. We compute the probability that the label $b_{i',j'}$ comes from the label $a_{i,j}$, where $i \geq i'$ and $j \geq j'$. This occurs when:

- $a_{i,j}$ is preserved, which occurs with probability $1 - q$,

- Exactly i' of the first i paths are retained, which occurs with probability
$$\binom{i}{i'}(1 - q^d)^{i'} q^{d(i-i')}.$$

- Exactly j' of the first j nodes in the path of the node with in X with index (i, j) are retained, which occurs with probability
$$\binom{j}{j'}(1 - q)^{j'} q^{j-j'}.$$

Thus the probability that the label $b_{i',j'}$ comes from the label $a_{i,j}$ is

$$(1 - q)\binom{i}{i'}(1 - q^d)^{i'} q^{d(i-i')}\binom{j}{j'}(1 - q)^{j'} q^{j-j'}.$$

We conclude that

$$\mathbb{E}\left[\sum_{(i',j')\in S} b_{i',j'} w_1^{i'} w_2^{j'}\right]$$

$$= (1 - q)\sum_{(i',j')\in S} w_1^{i'} w_2^{j'} \sum_{(i,j)\in S} a_{i,j}\binom{i}{i'}(1 - q^d)^{i'} q^{d(i-i')}\binom{j}{j'}(1 - q)^{j'} q^{j-j'}$$

$$= (1 - q)\sum_{i=0}^{\frac{n}{d}-1}\sum_{j=0}^{d-1} a_{i,j}\sum_{i'=0}^{i}\sum_{j'=0}^{j}\binom{i}{i'}(1 - q^d)^{i'} q^{d(i-i')} w_1^{i'}\binom{j}{j'}(1 - q)^{j'} q^{j-j'} w_2^{j'}$$

$$= (1 - q)\sum_{i=0}^{\frac{n}{d}-1}\sum_{j=0}^{d-1} a_{i,j}(q^d + (1 - q^d)w_1)^i (q + (1 - q)w_2)^j$$

$$= (1 - q)\sum_{(i,j)\in S} a_{i,j}(q^d + (1 - q^d)w_1)^i (q + (1 - q)w_2)^j,$$

where we change the order of summation in the second equality and apply the binomial theorem in the third equality. $\qquad\square$

Example 2.5. *Fig. 3 depicts all $2^4 = 16$ possible deletions that could occur for a specific $(4, 2)$-spider with labels $a_{0,0} = 1$, $a_{1,0} = 0$, $a_{0,1} = 1$, and $a_{1,1} = 1$, shown on the right. The figure also depicts the resulting padded traces and their associated generating functions. Note that $A_a(w_1, w_2)$, which recall*
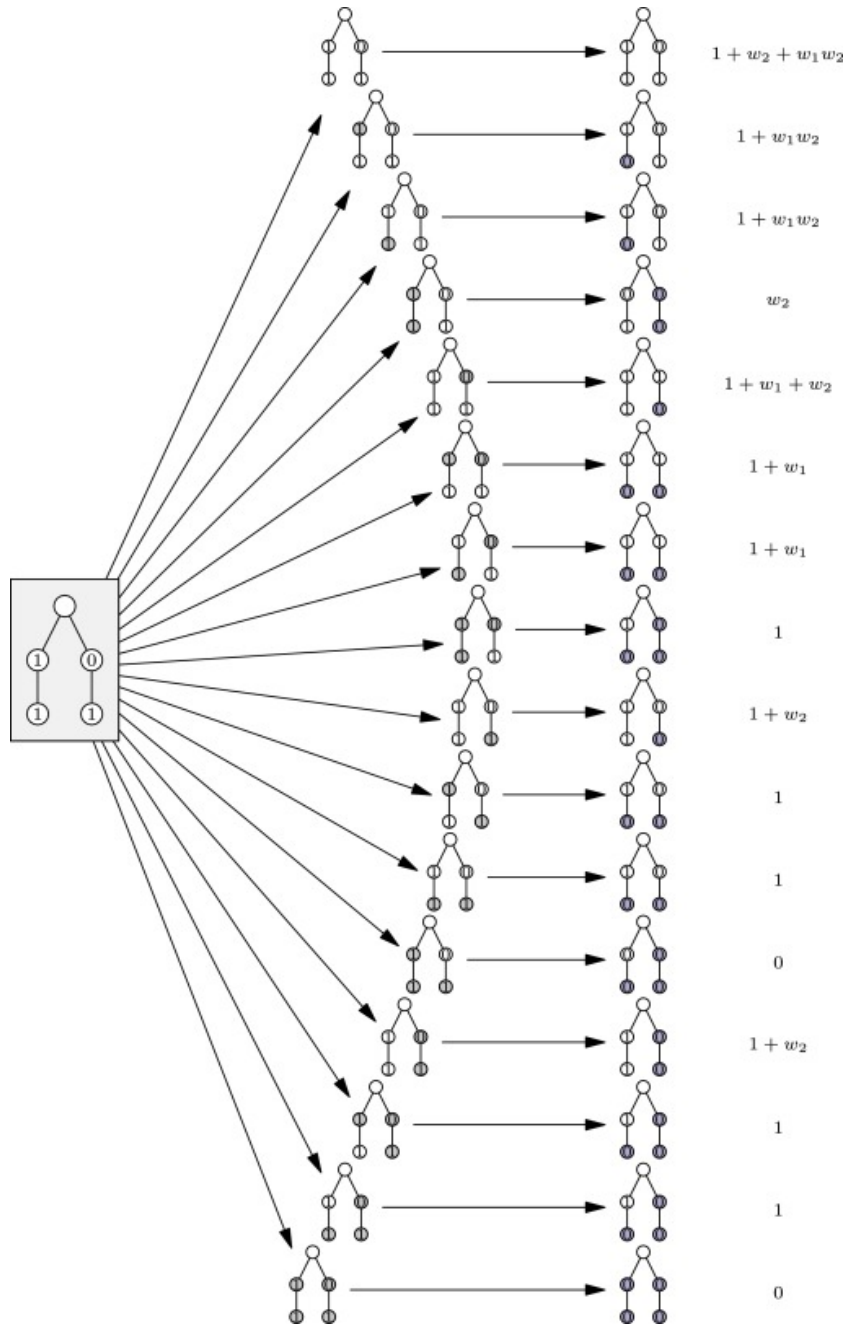
7

Figure 3: All possible padded traces for a certain seed $(4, 2)$-spider after being passed through the deletion channel, with their associated generating functions.

is the expected value of the generating functions of the padded traces, is a weighted average of all the generating functions on the right. For example, if $q = \frac{1}{2}$, we can simply average all the values in Fig. 3 to get

$$
\begin{aligned}
A_a(w_1, w_2) &= \mathbb{E}\left[\sum_{0 \le i' < 2, 0 \le j' < 2} b_{i',j'} w_1^{i'} w_2^{j'}\right] \\
&= \frac{13}{16} + \frac{3}{16} w_1 + \frac{5}{16} w_2 + \frac{3}{16} w_1 w_2
\end{aligned}
\tag{1}
$$

Note that Eq. (1) equals what we expect from Lemma 2.4:

$$
\begin{aligned}
A_a(w_1, w_2) &= (1 - q) \sum_{(i,j) \in S} a_{i,j} (q^d + (1 - q^d) w_1)^i (q + (1 - q) w_2)^j \\
&= \frac{1}{2} \cdot \sum_{0 \le i < 2, 0 \le j < 2} a_{i,j} \left(\frac{1}{4} + \frac{3}{4} w_1\right)^i \left(\frac{1}{2} + \frac{1}{2} w_2\right)^j \\
&= \frac{1}{2} \left[1 + \left(\frac{1}{2} + \frac{1}{2} w_2\right) + \left(\frac{1}{4} + \frac{3}{4} w_1\right) \left(\frac{1}{2} + \frac{1}{2} w_2\right)\right] \\
&= \frac{13}{16} + \frac{3}{16} w_1 + \frac{5}{16} w_2 + \frac{3}{16} w_1 w_2.
\end{aligned}
$$

3   Proof of main result

In this section we prove Theorem 1.1. Like in previous work on string trace reconstruction, we use a best-match algorithm to reconstruct the spider. As is typical in best-match algorithms, we compare every pair $(X^{(1)}, X^{(2)})$ of candidate spiders to see which spider from the pair is more likely to have produced the observed traces. We repeat this process for each pair of candidates and use the results to select a best possible guess for the original seed spider.

3.1   Overview of the algorithm

We consider all 2^n possible candidate spiders X and select a pair of spiders to compare against each other. Suppose we select candidate spiders $X^{(1)}$ and $X^{(2)}$ with labels $a^{(1)} = \{a_{i,j}^{(1)}\}_{(i,j) \in S}$ and $a^{(2)} = \{a_{i,j}^{(2)}\}_{(i,j) \in S}$, respectively. Now, consider the element-wise difference $a = a^{(1)} - a^{(2)}$, which is nonzero since $X^{(1)}$ and $X^{(2)}$ are distinct. Let $Y^{(1)}$ and $Y^{(2)}$ denote the random traces with labels $b^{(1)} = \{b_{i',j'}^{(1)}\}_{(i',j') \in S}$ and $b^{(2)} = \{b_{i',j'}^{(2)}\}_{(i',j') \in S}$, which result from passing $X^{(1)}$ and $X^{(2)}$ respectively through the deletion channel. Now, we

compute the difference of the generating functions corresponding to $X^{(1)}$ and $X^{(2)}$, which is equivalent to plugging a into the expression in Lemma 2.4:

$$\sum_{(i',j')\in S} (\mathbb{E}[b^{(1)}_{i',j'}] - \mathbb{E}[b^{(2)}_{i',j'}]) \cdot w_1^{i'} w_2^{j'}$$
$$= (1-q) \sum_{(i,j)\in S} a_{i,j}(q^d + (1-q^d)w_1)^i (q + (1-q)w_2)^j. \tag{2}$$

Through a process described in Section 3.3, we can select some pair of indices $(I, J) \in S$, depending on $X^{(1)}$ and $X^{(2)}$, such that $|\mathbb{E}[b^{(1)}_{I,J}] - \mathbb{E}[b^{(2)}_{I,J}]|$ is lower bounded substantially. What this means is that the expected value of some label $b_{I,J}$ in the trace differs significantly depending on whether or not the seed spider was $X^{(1)}$ or $X^{(2)}$. We can use this information in combination with the empirical expected value $\mathbb{E}[b_{I,J}]$ among our observed traces to select the better match between $X^{(1)}$ and $X^{(2)}$, that is, which of $X^{(1)}$ or $X^{(2)}$ is more likely to have produced the empirical expected value $\mathbb{E}[b_{I,J}]$. Such a process is known as a *mean-based algorithm*.

We repeat the above comparison for all pairs of spiders and then output the spider X^* that loses against no other spiders, if such a spider exists. If no such spider exists, we can output a uniformly random spider. As the true seed spider is among the 2^n candidate spiders, we can use a Chernoff bound to upper bound the probability that it loses against any other candidate spider by $\mathcal{O}\left(\frac{1}{n}\right)$. Therefore, so long as we are given enough traces, the true seed spider is outputted by the algorithm with high probability.

3.2 Littlewood polynomials

To analyze the expression in Eq. (2), we use bivariate Littlewood polynomials from complex analysis. We begin by defining these polynomials:

Definition 3.1. *A two-variable polynomial $A(z_1, z_2)$ is called a* bivariate Littlewood polynomial *if all of its coefficients are in the set $\{-1, 0, 1\}$.*

Note that in the right hand side of Eq. (2), the coefficients satisfy $a_{i,j} = a^{(1)}_{i,j} - a^{(2)}_{i,j} \in \{-1, 0, 1\}$. If we write the right hand side of Eq. (2) in terms of new variables $z_1 = q^d + (1-q^d)w_1$ and $z_2 = q + (1-q)w_2$, then we get

$$(1-q) \sum_{(i,j)\in S} a_{i,j} z_1^i z_2^j,$$

which is $(1-q)$ times a nonzero bivariate Littlewood polynomial with (z_1)-degree less than $\frac{n}{d}$ and (z_2)-degree less than d. In order to lower bound

this polynomial for some choice of z_1 and z_2, we prove the following lemma concerning bivariate Littlewood polynomials:

Lemma 3.2. *Let $f(z_1, z_2)$ be a nonzero bivariate Littlewood polynomial with degree a in z_1 and degree b in z_2. Then*

$$|f(z_1^*, z_2^*)| \geq \exp\left(-cL_1L_2\log(ab)\right)$$

for some $z_1^* = \exp(i\theta_1)$ and $z_2^* = \exp(i\theta_2)$, where θ_1 and θ_2 lie in the ranges $|\theta_1| \leq \frac{\pi}{L_1}$ and $|\theta_2| \leq \frac{\pi}{L_2}$.

Proof. Define the 2-variable polynomial

$$F(z_1, z_2) = \prod_{\substack{1 \leq x \leq L_1 \\ 1 \leq y \leq L_2}} f\left(z_1 e^{2\pi i x/L_1}, z_2 e^{2\pi i y/L_2}\right).$$

Using the maximum modulus principle, which recall says that the modulus $|F|$ of any holomorphic function F achieves its maximum value at the boundary of its domain, we first show that we can find some z_1' and z_2' on the unit circle such that $|F(z_1', z_2')| \geq 1$. Note that restricting the domain of a holomorphic function to the unit disk leaves the function holomorphic.

Factor $F(z_1, z_2) = z_2^k \cdot G(z_1, z_2)$ so that $G(z_1, z_2)$ no common factors with z_2. Since F has nonzero coefficients, $G(z_1, 0)$ can be viewed as a nonzero polynomial in one variable z_1. We can now factor $G(z_1, 0) = z_1^\ell \cdot H(z_1)$ so that $H(z_1)$ is nonzero and hence satisfies $|H(0)| = 1$. By the maximum modulus principle, we can find some z_1' on the unit circle such that $|H(z_1')| \geq |H(0)| = 1$. We can apply the maximum modulus principle again to find some z_2' on the unit circle such that $|G(z_1', z_2')| \geq |G(z_1', 0)|$. Therefore, we can find z_1' and z_2' such that

$$|F(z_1', z_2')| = |G(z_1', z_2')| \geq |G(z_1', 0)| = |H(z_1')| \geq |H(0)| = 1.$$

Now, applying the definition of F gives

$$1 \leq |F(z_1', z_2')| \leq |f(z_1' e^{2\pi i x/L_1}, z_2' e^{2\pi i y/L_2})| \cdot (ab)^{L_1 L_2 - 1}$$

for all $1 \leq x \leq L_1$ and $1 \leq y \leq L_2$, where we use the fact that $|f(z_1, z_2)| \leq ab$ for $|z_1| = |z_2| = 1$. We can now choose appropriate x and y to rotate z_1' and z_2' along the unit circle in the complex plane so that $z_1^* = z_1' \cdot e^{2\pi i x/L_1} = \exp(i\theta_1)$ and $z_2^* = z_2' \cdot e^{2\pi i y/L_2} = \exp(i\theta_2)$ satisfy $|\theta_1| \leq \frac{\pi}{L_1}$ and $|\theta_2| \leq \frac{\pi}{L_2}$. We conclude that

$$|f(z_1^*, z_2^*)| \geq \frac{1}{(ab)^{L_1 L_2 - 1}} \geq \exp(-L_1 L_2 \log(ab)),$$

where $z_1^* = \exp(i\theta_1)$ and $z_2^* = \exp(i\theta_2)$ satisfy $|\theta_1| \leq \frac{\pi}{L_1}$ and $|\theta_2| \leq \frac{\pi}{L_2}$. $\quad\square$

We remark that Lemma 3.2 is a generalization of [16, Lemma 17].

3.3 Completing the proof

We set the parameters in Lemma 3.2 to be $L_1 = L$ for some constant L to be chosen later, $L_2 = 1$, $a \leq \frac{n}{d}$, and $b \leq d$. By Lemma 3.2 and the triangle inequality, we can lower bound Eq. (2) as

$$\sum_{(i',j') \in S} |\mathbb{E}[b^{(1)}_{i',j'}] - \mathbb{E}[b^{(2)}_{i',j'}]||w^*_1|^{i'}|w^*_2|^{j'} \geq (1-q)\exp\left(-L\log n\right) \qquad (3)$$

for some $z^*_1 = \exp(i\theta_1)$ and $z^*_2 = \exp(i\theta_2)$ such that $|\theta_1| \leq \pi/L$ and $|\theta_2| \leq \pi$. Recall the change of variables

$$w^*_1 = \frac{z^*_1 - q^d}{1 - q^d} \qquad \text{and} \qquad w^*_2 = \frac{z^*_2 - q}{1 - q}.$$

We can upper bound $|w^*_1|$ as

$$|w^*_1| = \frac{|z^*_1 - q^d|}{1 - q^d}$$

$$\leq \frac{\sqrt{\left(\cos\frac{\pi}{L} - q^d\right)^2 + \left(\sin\frac{\pi}{L}\right)^2}}{1 - q^d}$$

$$= \frac{\sqrt{1 - 2q^d\cos\frac{\pi}{L} + q^{2d}}}{1 - q^d}$$

$$= \frac{\sqrt{(1 - q^d)^2 + 2q^d\left(1 - \cos\frac{\pi}{L}\right)}}{1 - q^d}$$

$$= \left(1 + \frac{2q^d\left(1 - \cos\frac{\pi}{L}\right)}{(1 - q^d)^2}\right)^{1/2}$$

$$\leq \exp\left(\frac{q^d\pi^2}{2L^2(1 - q^d)^2}\right),$$

where we use the inequalities $(1 + x)^r \leq e^{rx}$ for $r, x \geq 0$ and $1 - \cos\frac{\pi}{L} \leq \frac{1}{2}\left(\frac{\pi}{L}\right)^2$. Therefore,

$$|w^*_1|^{\frac{n}{d}} \leq \exp\left(\frac{n}{d} \cdot \frac{Cq^d}{L^2(1 - q^d)^2}\right)$$

for some constant C. We can also upper bound $|w^*_2|$ as

$$|w^*_2| = \frac{|z^*_2 - q|}{1 - q} \leq \frac{1 + q}{1 - q},$$

12

so

$$|w_2^*|^d \leq \exp(C'd)$$

for some constant C' depending on q. Therefore, by Eq. (3) and the fact that $|w_1^*|, |w_2^*| \geq 1$, we have

$$\exp\left(\frac{n}{d} \cdot \frac{Cq^d}{L^2(1-q^d)^2} + C'd\right) \sum_{(i',j')\in S} |\mathbb{E}[b_{i',j'}^{(1)}] - \mathbb{E}[b_{i',j'}^{(2)}]| \geq (1-q)\exp(-L\log n).$$

Thus there exists some pair of indices $(I, J) \in S$ such that

$$|\mathbb{E}[b_{I,J}^{(1)}] - \mathbb{E}[b_{I,J}^{(2)}]| \geq \frac{1-q}{n}\exp\left(-\frac{n}{d} \cdot \frac{Cq^d}{L^2(1-q^d)} - C'd - L\log n\right) =: \eta. \quad (4)$$

Denote the right hand side of Eq. (4) by η.

Returning now to the best-match algorithm, given two candidate spiders $X^{(1)}$ and $X^{(2)}$, we define the *better match* to be $X^{(1)}$ if

$$\left|\frac{1}{T}\sum_{t=1}^{T} s_{I,J}^t - \mathbb{E}[b_{I,J}^{(1)}]\right| \leq \left|\frac{1}{T}\sum_{t=1}^{T} s_{I,J}^t - \mathbb{E}[b_{I,J}^{(2)}]\right|,$$

where $s_{I,J}^t \in \{0,1\}$ is the value of the node at position (I, J) of the t-th trace. Now, suppose $X^{(1)} = X^*$ is the true seed spider. For all possible seed spiders $X^{(2)}$, we can use a Chernoff bound to upper bound the failure probability, namely the probability that $X^{(2)}$ is a better match than $X^{(1)}$, by $\exp(-T\eta^2/2)$, where T is the total number of traces. Therefore, by a union bound, the probability that X^* loses to at least one other spider is at most

$$\mathbb{P}[X^* \text{ not chosen by algorithm}] \leq \sum_{X^{(2)} \neq X^*} \mathbb{P}[X^{(2)} \text{ better match than } X^*]$$

$$\leq 2^n \cdot \exp(-T\eta^2/2)$$

$$\leq \exp\left(n\log 2 - \frac{T\eta^2}{2}\right).$$

For this expression to be at most $\frac{1}{n} = \exp(-\log n)$, we set

$$T = \frac{2}{\eta^2}(n\log 2 + \log n) = \Theta(\eta^{-2}n).$$

Plugging in the definition of η from Eq. (4) yields

$$T = \Theta\left(n^3 \cdot \exp\left(\frac{n}{d} \cdot \frac{Cq^d}{L^2(1-q^d)} + C'd + cL\log n\right)\right). \quad (5)$$

13

Note that the n^3 term is negligible. The $C'd$ term is also negligible since we are in the regime $d \leq \log_{1/q}(n)$. Finally, $1 - q^d \geq 1 - q$ depends only on q, so Eq. (5) can be simplified to

$$T = \exp\left(\Theta\left(\frac{nq^d}{dL^2} + L\log n\right)\right).$$

To balance these terms, we set $L = \left(\frac{nq^d}{d\log n}\right)^{1/3}$ to get a final bound of

$$T = \exp\left(C \cdot \frac{(nq^d)^{1/3}}{d^{1/3}}(\log n)^{2/3}\right),$$

where C is a constant that depends only on q. We conclude the proof of Theorem 1.1.

4   Conclusion

We presented a mean-based algorithm using Littlewood polynomials that reconstructs (n, d)-spiders with high probability in the regime $d \leq \log_{1/q}(n)$, where q is the deletion probability. Our algorithm uses $\exp\left(\mathcal{O}\left(\frac{(nq^d)^{1/3}}{d^{1/3}}(\log n)^{2/3}\right)\right)$ traces and works for the full range $q \in (0, 1)$ of deletion probabilities.

In light of recent work improving the string trace reconstruction upper bound to $\exp(\tilde{\mathcal{O}}(n^{1/5}))$ using a non-mean-based algorithm [4], it would be interesting to see whether a similar technique could achieve an upper bound of the form $\exp\left(\tilde{\mathcal{O}}((nq^d)^{1/5})\right)$ for the spider trace reconstruction problem.

References

[1] 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.

[2] Tugkan Batu, Sampath Kannan, Sanjeev Khanna, and Andrew McGregor. Reconstructing strings from random traces. *Symposium on Discrete Algorithms*, pages 910–918, 2004.

[3] Vinnu Bhardwaj, Pavel A Pevzner, Cyrus Rashtchian, and Yana Safonova. Trace reconstruction problems in computational biology. *IEEE Transactions on Information Theory*, 67(6):3295–3314, 2020.

[4] Zachary Chase. New upper bounds for trace reconstruction. *arXiv preprint arXiv:2009.03296*, 2020.

[5] Zachary Chase. New lower bounds for trace reconstruction. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 57(2):627–643, 2021.

[6] Xi Chen, Anindya De, Chin Ho Lee, Rocco A Servedio, and Sandip Sinha. Near-optimal average-case approximate trace reconstruction from few traces. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 779–821. SIAM, 2022.

[7] Mahdi Cheraghchi, Ryan Gabrys, Olgica Milenkovic, and Joao Ribeiro. Coded trace reconstruction. *IEEE Transactions on Information Theory*, 66(10):6084–6103, 2020.

[8] Sami Davies, Miklos Z Racz, and Cyrus Rashtchian. Reconstructing trees from traces. In *Conference On Learning Theory*, pages 961–978. PMLR, 2019.

[9] Sami Davies, Miklós Z Rácz, Benjamin G Schiffer, and Cyrus Rashtchian. Approximate trace reconstruction: Algorithms. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 2525–2530. IEEE, 2021.

[10] Anindya De, Ryan O'Donnell, and Rocco A Servedio. Optimal mean-based algorithms for trace reconstruction. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1047–1056, 2017.

[11] Lisa Hartung, Nina Holden, and Yuval Peres. Trace reconstruction with varying deletion probabilities. In *2018 Proceedings of the Fifteenth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 54–61. SIAM, 2018.

[12] Nina Holden and Russell Lyons. Lower bounds for trace reconstruction. *The Annals of Applied Probability*, 30(2):503–525, 2020.

[13] Nina Holden, Robin Pemantle, Yuval Peres, and Alex Zhai. Subpolynomial trace reconstruction for random strings and arbitrary deletion probability. *Mathematical Statistics and Learning*, 2(3):275–309, 2020.

[14] Thomas Holenstein, Michael Mitzenmacher, Rina Panigrahy, and Udi Wieder. Trace reconstruction with constant deletion probability and related results. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 389–398. Citeseer, 2008.

[15] Philipp Karau and Vincent Tabard-Cossa. Capture and translocation characteristics of short branched dna labels in solid-state nanopores. *ACS sensors*, 3(7):1308–1315, 2018.

[16] Akshay Krishnamurthy, Arya Mazumdar, Andrew McGregor, and Soumyabrata Pal. Trace reconstruction: Generalized and parameterized. *IEEE Transactions on Information Theory*, 67(6):3233–3250, 2021.

[17] V. Levenshtein. Reconstruction of objects from a minimum number of distorted patterns. *Doklady Mathematics*, 55(3):417–420, 1997.

[18] Andrew McGregor, Eric Price, and Sofya Vorotnikova. Trace reconstruction revisited. In *European Symposium on Algorithms*, pages 689–700. Springer, 2014.

[19] Shyam Narayanan and Michael Ren. Circular trace reconstruction. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

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

[21] Krishnamurthy Viswanathan and Ram Swaminathan. Improved string reconstruction over insertion-deletion channels. In *Proceedings of*

the nineteenth annual ACM-SIAM symposium on Discrete algorithms, pages 399–408, 2008.