# The Suboptimality of Asymmetric Recursive Reconstruction Algorithms

Brian Chen

September 18, 2017

**Abstract**

We examine a variant of tree reconstruction proposed by Mossel. Given a rooted tree $T$ with $d$ leaves, all at the same level, we consider trees $T_n$ consisting of $n$ levels of copies of $T$, and randomly label their vertices with bits as follows. The root is labeled with a random bit, and then for each child, it is labeled with a bit that differs from its parent with probability $\varepsilon$. We analyze algorithms that, given guesses for the $d$ labels of leaves of a copy of $T$, output a guess for the root of $T$, which can be recursively applied starting from labels of the leaves of each $T_n$ to obtain a guess for the label of the root of $T_n$. An algorithm achieves recursive reconstruction if its probability of correctly guessing the label of the root is bounded away from 1/2. In this paper, we show that prior analysis of recursive reconstruction algorithms do not apply to asymmetric algorithms that have different probabilities of reconstructing a 0 versus a 1. In particular, asymmetric majority algorithms that output the most common label among the children but break ties unevenly fail to achieve recursive reconstruction for a range of $\varepsilon$ where symmetric majority algorithms succeed. We also discuss some empirical evidence and theoretical difficulties in studying generalizations of this model.

# 1 Introduction

The tree reconstruction problem is a well-studied problem in probability with ties to communication theory, biology, and statistical physics. It considers a label-propagation model defined as follows. Fix some real number $\varepsilon$ with $0 < \varepsilon \leq 1/2$. Let $T$ be a finite rooted tree with $d$ leaves, all of which are at the same level $h$. We produce a random label for each vertex of the tree as follows. Label the root of the tree with either 0 or 1 at random, each with 1/2 probability; then, for each non-root vertex, label it with the same label as its parent with probability $1-\varepsilon$, and differently with probability $\varepsilon$. A *reconstruction algorithm* for $T$ is an algorithm that, given the labels of the leaves of a tree, computes a guess for the label of the root of the tree. Clearly, as long as $\varepsilon < 1/2$, a good reconstruction algorithm can correctly guess the root more than 1/2 of the time. The tree reconstruction problem is typically considered on an infinite family of trees $(T_n)_{n=1}^{\infty}$, with the goal being to determine whether, as $n$ tends to infinity, the probability of a family of reconstruction algorithms succeeding is bounded away from 1/2.

A commonly considered family of trees is the sequence of "periodic trees" $(T_n)_{n=1}^{\infty}$ built from a finite tree $T$ with $d$ leaves, all at the same level $h$, as follows: Let $T_1 = T$, and for each $n > 1$, construct $T_n$ by replacing each leaf of $T_{n-1}$ with a copy of $T$. Some examples of periodic trees are diagrammed in Figures 1 and 2.

A variety of bounds, positive and negative, have been obtained for this problem and many generalizations and variations thereof. The classical result

Figure 1: Examples of a tree $T$ and its first three periods. The topmost copy of $T$ is highlighted for clarity.

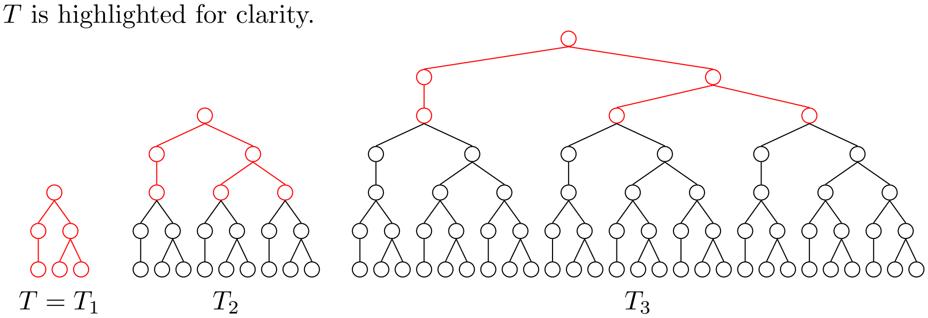
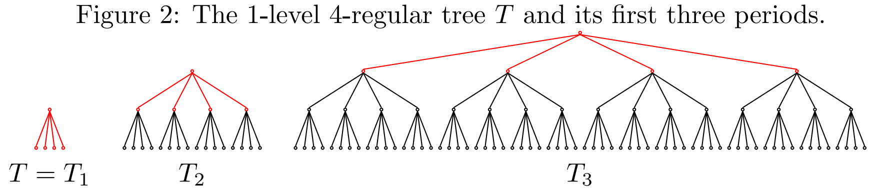



Figure 2: The 1-level 4-regular tree $T$ and its first three periods.



on tree reconstruction is the Kesten-Stigum bound [7], which, when specialized to our model, states that reconstruction is possible when $\varepsilon < \varepsilon_c$, where

$$\varepsilon_c = \frac{1 - d^{-1/2h}}{2}.$$

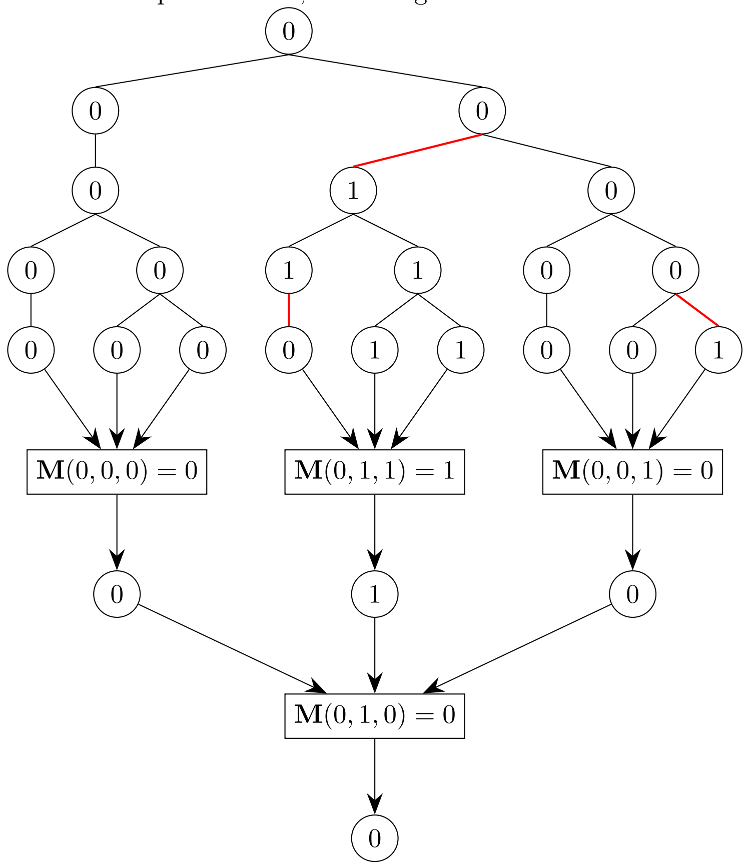
The Kesten-Stigum bound is also tight for our model as described above [1, 4, 5] — that is, reconstruction is not possible if $\varepsilon > \varepsilon_c$ — but there are generalizations of this model where the bound still holds but is not tight [9, 10, 11], as well as other special cases or variations of the generalized model where the bound is tight [10, 6, 2].

In [8], Mossel studies a specific family of reconstruction algorithms called *recursive reconstruction algorithms*. The idea is that, given a reconstruction algorithm $\mathbf{A}$ for $T$, one can create a reconstruction algorithm for all $T_i$ recursively, by taking each tree $T_i$ and repeatedly replacing the bottommost copies of $T$ with single leaves labeled with the output of the algorithm $\mathbf{A}$, until only the root remains. The advantage of recursive reconstruction algorithms is that they are simple to compute and to describe, and that what seems to be the optimal recursive reconstruction algorithm, the *majority algorithms*, does not depend on $\varepsilon$ (as the optimal reconstruction algorithm does). The majority algorithm is the algorithm that outputs the majority of the labels it receives. The question we are concerned with becomes whether, as $n$ tends to infinity, recursive algorithms can succeed with probability bounded away from 1/2.

[8] finds a tight bound on the $\varepsilon$ for which recursive reconstruction is possible, that is, an explicit formula depending on $T$ for a value $\varepsilon_r$, which is less than $\varepsilon_c$, such that recursive reconstruction is possible iff $\varepsilon < \varepsilon_r$. They also prove that the algorithms that achieve recursive reconstruction for all such $\varepsilon$ are precisely the majority algorithms. However, the proof does not account for a large class of algorithms that have different probabilities of reconstructing a 0 versus a 1, which we call *asymmetric* algorithms. In particular, some majority algorithms are asymmetric. Thus, in spite of [8], it remains conceivable that some asymmetric algorithms could reconstruct for some $\varepsilon$ satisfying $\varepsilon_r \le \varepsilon < \varepsilon_c$. In this paper we show that a class of asymmetric algorithms, including asymmetric majority algorithms, in fact fail to achieve recursive reconstruction for a range of $\varepsilon$ that is less than $\varepsilon_r$. This provides evidence that the bound in [8] is tight, while also suggesting that in general, in order for an algorithm to achieve recursive reconstruction, it is important for the algorithm to be symmetric and not favor any labels over any other.

The rest of this paper is structured as follows. In Section 2 we define the "recursive function" whose iterations model the accuracy of periods of a recursive reconstruction algorithm. In Section 3 we specialize to recursive algorithms on the simple 1-level $d$-regular tree, which are easy to analyze and which form the basis for the core results of [8], and show that we can study its convergence by studying the fixed points of the recursive function. In Section 4 we further focus on a class of "uniformly biased" algorithms, which includes majority algorithms that simply break ties biasedly, and prove that these algorithms do not perform as well as the symmetric algorithm. We will prove that there is a range

Figure 3: An example of a possible $\varepsilon$-labeling of $T_2$, where $T$ is the tree of Figure 1, and the majority algorithm **M** processing the labeling. Edges whose incident nodes have different labels are in bold red; this happens for each edge with probability $\varepsilon$ independently. In this case, the root was labeled 0, and the algorithm's final output is also 0, so the algorithm is successful.




of $\varepsilon$ where the symmetric algorithm achieves recursive reconstruction but the asymmetric algorithm fails, which will provide a counterexample to Theorem 1.2 of [8] as written there. The range of $\varepsilon$ for which we prove that recursive reconstruction fails does not appear to be tight; we conjecture a stronger bound, which we prove in Section 5 for the special case of $d = 4$. Finally, in Section 6 we discuss the implications for generalizations of our model, and briefly mention some empirical evidence and difficulties in studying them.

## 2 The Asymmetric Recursive Function

We start by restating the definition of the labelings, trees, and algorithms we are studying formally:

**Definition 2.1.** *For a finite rooted tree $T$ and a real constant $\varepsilon$ satisfying $0 < \varepsilon \leq 1/2$, an* **$\varepsilon$-labeling of** *$T$ is a random labeling of the nodes of $T$ generated as follows. First, the root is randomly labeled with one of 0 or 1, with probability $1/2$ each. Then, for each edge $(v, w)$ with $v$ the parent of $w$, we label $w$ with the same bit as $v$ with probability $1-\varepsilon$, and a different bit with probability $\varepsilon$, independently for each edge.*

*For a finite rooted tree $T$, we define a sequence of trees $(T_n)_{n=1}^{\infty}$ called* **periodic trees** *in terms of $T$ as follows. Let $T_1 = T$. For each $n > 1$, let $T_n$ be the finite rooted tree obtained from $T_{n-1}$ by replacing each leaf with a copy of $T$.*

*A* **reconstruction algorithm A** *for a tree $T$ is a randomized algorithm that, given the labels of the leaves of $T$, computes a guess for the label of the root of $T$. If $T$ has $d$ leaves, then $\mathbf{A}$ is a randomized algorithm that takes inputs from $\{0,1\}^d$ and produces an output in $\{0,1\}$.*

*Denote by $r_\varepsilon(T, \mathbf{A})$ the probability that the reconstruction algorithm $\mathbf{A}$ correctly guesses the label of the root of $T$ when applied to the leaf labels of a random $\varepsilon$-labeling of $T$.*

Note that our model is technically slightly more general than that of [8] and much of the literature in that we allow our algorithm to be randomized, while most other sources define the algorithm as a deterministic function. This does not give our model significantly more power, but it enables us to study algorithms that differ from the symmetric majority algorithm with arbitrarily small probability, which we will show are suboptimal.

In general, we say that the bit reconstruction problem is solvable for $T$ and $\varepsilon$ if there exists a constant $\delta > 0$ and algorithms $(\mathbf{A}_n)_{n=1}^{\infty}$ such that, for every positive integer $n$, we have

$$r_\varepsilon(T_n, \mathbf{A}_n) \geq 1/2 + \delta.$$

As described in the introduction, this problem and its generalizations and variations have received a lot of attention in the literature.

In this paper, we concern ourselves with a restricted class of algorithms called *recursive reconstruction algorithms*, which are algorithms on $T_n$ obtained by recursively applying a reconstruction algorithm $\mathbf{A}$ for $T$. We define recursive reconstruction algorithms as in [8, Definitions 1.1]:

**Definition 2.2.**

- *Suppose $T$ has $d$ leaves and let $\mathbf{A} : \{0,1\}^d \to \{0,1\}$ be a reconstruction algorithm. Then we define a sequence of reconstruction algorithms $(\mathbf{A}^n)_{n=1}^{\infty}$ for the periods $T_n$ of $T$, called the* **periods** *of $\mathbf{A}$, as follows: Let $\mathbf{A}^1 = \mathbf{A}$, and for $k > 1$ let $\mathbf{A}^k$ be defined by*

$$\mathbf{A}^k(\sigma_1, \ldots, \sigma_{d^k}) := \mathbf{A}^{k-1}(\mathbf{A}(\sigma_1, \ldots, \sigma_d), \ldots, \mathbf{A}(\sigma_{d^k-d+1}, \ldots, \sigma_{d^k}))\,.$$

- *We say that* **the bit reconstruction problem is recursively solvable for** $T$ **and** $\varepsilon$ *if there exists a constant* $\delta > 0$ *and a reconstruction algorithm* $\mathbf{A}$ *for* $T$ *such that, for every positive integer* $n$*, we have*

$$r_\varepsilon(T_n, \mathbf{A}^n) \geq 1/2 + \delta.$$

In other words, the bit reconstruction problem is recursively solvable for $T$ and $\varepsilon$ if it is solvable by the periods $(\mathbf{A}^k)_{k=1}^\infty$ of a single algorithm $\mathbf{A}$.

Informally, $\mathbf{A}^k$ can be thought of as follows: Starting at the leaves of $T^n$, for each copy of $T$, we apply the algorithm $\mathbf{A}$ to the labels of the leaves, and replace the copy of $T$ with a single leaf labeled with the output of the algorithm, until there is only one node, the root, left.

In order to analyze the behavior of a recursive algorithm, we define the general analogue of the "recursive function" defined in [8]:

**Definition 2.3.** *For a tree* $T$ *and real constants* $\varepsilon, p_0, p_1$ *satisfying* $0 < \varepsilon \leq 1/2$ *and* $0 \leq p_0, p_1 \leq 1$*, an* $(\varepsilon, p_0, p_1)$**-labeling of** $T$ *is a random labeling of the nodes of* $T$ *generated as follows. First, a random* $\varepsilon$*-labeling of* $T$ *is generated. Next, for each leaf, if its label is* $i$*, then we preserve it with probability* $p_i$ *and replace it with its inverse with probability* $1 - p_i$*.*

*For a reconstruction algorithm* $\mathbf{A}$*, let* $R_\mathbf{A}(p_0, p_1)$ *be a pair of probabilities* $(q_0, q_1)$*, where* $q_i$ *is the probability that, when provided with the leaves of a random* $(\varepsilon, p_0, p_1)$*-labeling of* $T$ *in which the root was labeled* $i$*, the algorithm* $\mathbf{A}$ *correctly outputs the guess* $i$*.*

The function $R_\mathbf{A}$ is important because the reconstruction probability of periods of $\mathbf{A}$ on periods of $T$ is captured by iterating $R_\mathbf{A}$:

**Theorem 2.4.** *Fix a finite rooted tree* $T$ *with* $d$ *leaves, all at depth* $h$*. Let* $(q_0, q_1) = R_\mathbf{A}^{(n)}(1, 1)$*, that is, the result of iterating* $R_\mathbf{A}$ *a total of* $n$ *times starting from the point* $(1, 1)$*. Then* $q_i$ *is the probability that the period* $\mathbf{A}^n$ *of* $\mathbf{A}$ *correctly reconstructs the label* $i$ *when applied to the leaf labels of a random* $\varepsilon$*-labeling of the period* $T^n$*. In particular,*

$$r_\varepsilon(T^n, \mathbf{A}^n) = \frac{q_0 + q_1}{2}.$$

*Proof.* This is proven by induction. For $n = 1$ it is clear from the definition. For the induction step, suppose we are analyzing $T^{n+1}$. Pick a random $\varepsilon$-labeling of $T^{n+1}$, and let $(\tau_1, \ldots, \tau_d)$ be the labels of the nodes at depth $h$, i.e. the leaves of the topmost copy of $T$. Then we can think of this sequence as being drawn from the leaves of a random $\varepsilon$-labeling of $T$.

Let $(p_0, p_1) = R_\mathbf{A}^{(n)}(1, 1)$. For each $\tau_i$, compare it to the label that the recursive reconstruction algorithm will guess for its node: the probability that they are the same is precisely $p_{\tau_i}$. Therefore, if $(q_0, q_1)$ are the probabilities that $\mathbf{A}^{n+1}$ correctly reconstructs the labels 0 and 1 when they are the label of the root, we have

$$(q_0, q_1) = R_\mathbf{A}(p_0, p_1) = R_\mathbf{A}(R_\mathbf{A}^{(n)}(1, 1)) = R_\mathbf{A}^{(n+1)}(1, 1),$$

as desired. □

The function definition and proof above differ from [8, Definition 2.1, Lemma 2.2] in that they account for the possibility that the reconstruction algorithm has different probabilities of reconstructing a 0 versus a 1. The algorithms for which the original paper's analysis still applies can be described as follows.

**Definition 2.5.** *A reconstruction algorithm* **A** *is* **symmetric** *if, whenever* $p_0 = p_1$*, we have* $q_0 = q_1$*, where* $(q_0, q_1) = R_{\mathbf{A}}(p_0, p_1)$*. Otherwise, it is* **asymmetric**.

If we only allow symmetric algorithms, [8] proves that the algorithms that achieve reconstruction for the largest $\varepsilon$ are precisely *majority algorithms*:

**Definition 2.6.** *A reconstruction algorithm* **A** *is a* **majority algorithm** *if it outputs the majority of its inputs whenever such a majority exists. (When there are equally many 0s and 1s in the input,* **A** *may output 0 or 1, or randomly choose between them.)*

**Theorem 2.7.** *([8, Theorem 1.2]) Let $T$ be a tree with $d$ leaves, all at the same level $h$. The bit reconstruction problem is recursively solvable for $T$ and $\varepsilon$ by a symmetric algorithm iff $\varepsilon < \varepsilon_r$, where $\varepsilon_r$ is defined by*

$$\frac{1}{2^{d-1}} \left\lceil \frac{d}{2} \right\rceil \binom{d}{\lceil \frac{d}{2} \rceil} (1 - 2\varepsilon_r)^h = 1.$$

*Furthermore, a symmetric algorithm recursively solves the bit reconstruction problem for all $\varepsilon < \varepsilon_r$ iff it is a majority algorithm.*

We conjecture that the first claim of the theorem is true in general; that is, asymmetric algorithms cannot recursively solve the bit reconstruction problem for $T$ and $\varepsilon$ where symmetric algorithms fail, so the same bound $\varepsilon_r$ on solvable $\varepsilon$ holds across all algorithms. However, the condition of symmetry is essential to the second claim: as we will see, asymmetric majority algorithms cannot solve the bit reconstruction problem for all $\varepsilon < \varepsilon_r$, and in fact there is a range of $\varepsilon$ below $\varepsilon_r$ where all asymmetric majority algorithms fail. We will prove that counterexamples exist on even the 1-level $d$-regular tree when $d$ is even. With this goal in mind, we now turn to analyze recursive reconstruction algorithms on the 1-level $d$-regular tree specifically.

# 3 Regular Trees

When $T$ is the simple 1-level $d$-regular tree, we can describe and analyze $R_{\mathbf{A}}$ particularly precisely. Define

$$L_0(p_0, p_1) := (1 - \varepsilon)p_0 + \varepsilon(1 - p_1) \tag{1}$$
$$L_1(p_0, p_1) := (1 - \varepsilon)p_1 + \varepsilon(1 - p_0); \tag{2}$$

then $L_i(p_0, p_1)$ is the probability that, if the root is initially labeled $i$ in an $(\varepsilon, p_0, p_1)$-labeling of $T$, then a particular leaf will have the label $i$, after propagation and possible replacement. We will sometimes write

$$L(p_0, p_1) := (L_0(p_0, p_1), L_1(p_0, p_1)).$$

Let us point out some simple properties of the above definitions:

**Lemma 3.1.** *Let* $\ell_0 = L_0(p_0, p_1), \ell_1 = L_1(p_0, p_1)$. *Then*

$$\begin{aligned} \ell_0 - \ell_1 &= p_0 - p_1; \\ \ell_0 + \ell_1 &= 2\varepsilon + (1 - 2\varepsilon)(p_0 + p_1). \end{aligned}$$

*In particular,* $\ell_0 + \ell_1$ *is an increasing function of* $p_0 + p_1$.

*Proof.* Immediate from the definitions. □

Next, let $P_{\mathbf{A},i}(\ell_i)$ be the probability that, if $\mathbf{A}$ is provided with $d$ inputs, each of which is independently $i$ with probability $\ell_i$ and $1 - i$ otherwise, then the algorithm will guess $i$. Note that

$$P_{\mathbf{A},0}(\ell_0) = 1 - P_{\mathbf{A},1}(1 - \ell_1). \tag{3}$$

It is straightforward to completely characterize the form of the function $P_{\mathbf{A},i}(\ell_i)$. It simply depends on $d + 1$ real variables $c_j$, each controlling the algorithm's probability of guessing $i$ when $j$ of its inputs are $i$, for each $j = 0, 1, \ldots, d$:

**Lemma 3.2.** *For any reconstruction algorithm* $\mathbf{A}$ *and either* $i \in \{0, 1\}$, *the function* $P_{\mathbf{A},i}$ *is a polynomial of the form*

$$P_{\mathbf{A},i}(\ell) = \sum_{j=0}^{d} c_j \binom{d}{j} \ell^j (1 - \ell)^{d-j} \tag{4}$$

*for real numbers* $c_0, c_1, \ldots, c_d$ *with* $c_j \in [0, 1]$ *for each* $j$. *Furthermore, the choice of these real numbers are unique, and for every sequence* $(c_j)_{j=0}^d$ *satisfying* $c_j \in [0, 1]$ *for each* $j$, *there exists a reconstruction algorithm* $\mathbf{A}$ *such that* (4) *holds.*

*Proof.* To prove the first statement, let $c_j$ be the probability that $\mathbf{A}$ outputs $i$ when provided with inputs that are a random permutation of $j$ occurrences of $i$ and $d - j$ occurrences of $1 - i$. There are $\binom{d}{j}$ distinct permutations, and for any $\ell$, each one would occur in the original model with probability $\ell^j (1 - \ell)^{d-j}$. Equation (4) follows.

To show that the choice of $c_j$ is unique, we just need to prove that $\ell^j (1-\ell)^{d-j}$ for $j = 0, \ldots, d$ are linearly independent polynomials of $\ell$. To see this, note that for each $j'$, each of the polynomials $\ell^j (1 - \ell)^{d-j}$ for $j < j'$ are divisible by $(1-\ell)^{d-j'+1}$, but $\ell^{j'} (1-\ell)^{d-j'}$ is not, so $\ell^{j'} (1-\ell)^{d-j'}$ is linearly independent of the $j'$ polynomials before it in the sequence. Thus, the polynomials are linearly independent. (In fact, they form a spanning basis of the space of degree-at-most-$d$ polynomials of $\ell$.)

Finally, to prove the converse, simply let $\mathbf{A}$ output $i$ with probability $c_j$ and $1-i$ with probability $1-c_j$ on any input with exactly $j$ occurrences of $i$. $\square$

Having defined $L_i$ and $P_{\mathbf{A},i}$, which respectively model the noisy edge/leaf propagation and the algorithm's ability to reconstruct from noise, we can define the recursive reconstruction probability $R_{\mathbf{A}}$ in terms of these functions as

$$R_{\mathbf{A}}(p_0,p_1) = (P_{\mathbf{A},0}(L_0(p_0,p_1)), P_{\mathbf{A},1}(L_1(p_0,p_1))) \,. \tag{5}$$

By Theorem 2.4, in order to study recursive reconstruction, we just need to consider the sequence $R_{\mathbf{A}}^{(n)}(1,1)$, that is, the result of iterating $R_{\mathbf{A}}$ starting from $p_0 = p_1 = 1$. Note, therefore, that for the 1-level $d$-regular tree, recursive reconstruction only depends on the functions $P_{\mathbf{A},i}$. It therefore makes sense to consider two algorithms "the same" if their polynomials are the same:

**Definition 3.3.** *Two reconstruction algorithms* $\mathbf{A}$, $\mathbf{A}'$ *for the 1-level $d$-regular tree are* **equivalent** *if the polynomials* $P_{\mathbf{A},i}$ *and* $P_{\mathbf{A}',i}$ *are identical.*

Also, note that the two polynomials $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ for a single algorithm $\mathbf{A}$ coincide iff the algorithm is symmetric:

**Lemma 3.4.** *A recursive reconstruction algorithm* $\mathbf{A}$ *for the 1-level $d$-regular tree is symmetric iff* $P_{\mathbf{A},0}$ *and* $P_{\mathbf{A},1}$ *coincide.*

*Proof.* Recall that $\mathbf{A}$ is symmetric if, whenever $p_0 = p_1$, we have $q_0 = q_1$, where $(q_0, q_1) = R_{\mathbf{A}}(p_0, p_1)$. Thus, in the 1-level $d$-regular case, $\mathbf{A}$ is symmetric iff

$$P_{\mathbf{A},0}(L_0(p_0,p_1)) = P_{\mathbf{A},1}(L_1(p_0,p_1))$$

whenever $p_0 = p_1$. It's easy to see from Lemma 3.1 that $\ell_0 = \ell_1$ iff $p_0 = p_1$, so the conclusion follows. $\square$

**Corollary 3.5.** *A recursive reconstruction algorithm* $\mathbf{A}$ *for the 1-level $d$-regular tree is symmetric iff in the corresponding sequence $(c_j)_{j=0}^d$ to both $P_{\mathbf{A},i}$ given by Lemma 3.2, we have*

$$c_j + c_{d-j} = 1$$

*for each $j$.*

*Proof.* Let the sequences corresponding to $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ be $(c_j)_{j=0}^d$ and $(c'_j)_{j=0}^d$, respectively. Also note that, as a polynomial identity,

$$1 = \sum_{j=0}^{d} \binom{d}{j} \ell^j (1-\ell)^{d-j}$$

(this is just $(\ell + (1-\ell))^d$ expanded with the binomial theorem). From this and (3), we know that

$$\begin{aligned}\sum_{j=0}^{d} c_j \binom{d}{j} \ell^j (1-\ell)^{d-j} &= 1 - \sum_{j=0}^{d} c'_j \binom{d}{j} (1-\ell)^j \ell^{d-j} \\ &= \sum_{j=0}^{d} (1-c'_j) \binom{d}{j} (1-\ell)^j \ell^{d-j} \\ &= \sum_{j=0}^{d} (1-c'_{d-j}) \binom{d}{j} \ell^j (1-\ell)^{d-j}\end{aligned}$$

which means that $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ coincide iff, as a polynomial identity, we have

$$\sum_{j=0}^{d} c'_j \binom{d}{j} \ell^j (1-\ell)^{d-j} = \sum_{j=0}^{d} (1-c'_{d-j}) \binom{d}{j} \ell^j (1-\ell)^{d-j}.$$

Since the polynomials $\ell^j(1-\ell)^{d-j}$ are independent, as proven in Lemma 3.2, this means that $\mathbf{A}$ is symmetric iff $c'_j = (1-c'_{d-j})$ for all $j \in \{0, \ldots, d\}$. This is also equivalent to $c_j = (1-c_{d-j})$ for all $j \in \{0, \ldots, d\}$. □

When $\mathbf{A}$ is symmetric, we have $P_{\mathbf{A},0} \equiv P_{\mathbf{A},1}$ and $p_0 = p_1$ in every iteration of $R_{\mathbf{A}}^{(n)}(1,1)$. As a result, $L_i$ and $P_{\mathbf{A},i}$ can both be treated as single-variable functions, so $R_{\mathbf{A}}$ is just the composition of these two functions, which is correspondingly easy to analyze. In particular, when $R_{\mathbf{A}}$ is increasing (which is true for any symmetric majority algorithm, for example), we have $R_{\mathbf{A}}(p_0) < p_0$ at each step, so that the sequence $R_{\mathbf{A}}^{(n)}(1)$ converges to the greatest fixed point of $R_{\mathbf{A}}$. In the asymmetric case, it is less obvious that $R_{\mathbf{A}}$ should converge to a fixed point, but fortunately, as long as $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ are increasing, this is still true. In order to prove this, we need to observe how $R_{\mathbf{A}}(p_0, p_1)$ compare to $(p_0, p_1)$:

**Lemma 3.6.** *Suppose $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ are increasing. Let $0 \le p_0, p_1 \le 1$. Let $(q_0, q_1) = R_{\mathbf{A}}(p_0, p_1)$ and $(r_0, r_1) = R_{\mathbf{A}}(q_0, q_1)$. Then:*

- *If $p_0 \ge q_0$ and $p_1 \le q_1$, then $q_0 \ge r_0$ and $q_1 \le r_1$.*
- *If $p_0 \le q_0$ and $p_1 \ge q_1$, then $q_0 \le r_0$ and $q_1 \ge r_1$.*
- *If $p_0 \ge q_0$ and $p_1 \ge q_1$, then either $q_0 \ge r_0$ or $q_1 \ge r_1$.*
- *If $p_0 \le q_0$ and $p_1 \le q_1$, then either $q_0 \le r_0$ or $q_1 \le r_1$.*

*Proof.* The first two statements follow from observing that the first coordinate of $R_{\mathbf{A}}(p_0, p_1)$ is increasing in $p_0$ and decreasing in $p_1$, and the second coordinate is decreasing in $p_0$ and increasing in $p_1$. These can be seen from combining (1), (2), and (5) with the condition that $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ are increasing.

To prove the third statement, let $(\ell_0, \ell_1) = L(p_0, p_1)$ and $(m_0, m_1) = L(q_0, q_1)$. Then, since $p_0 \geq q_0$ and $p_1 \geq q_1$, we know that $p_0 + p_1 \geq q_0 + q_1$, so $\ell_0 + \ell_1 \geq m_0 + m_1$. Therefore either $\ell_0 \geq m_0$, in which case $q_0 = P_{\mathbf{A},0}(\ell_0) \geq P_{\mathbf{A},0}(m_0) = r_0$, or $\ell_1 \geq m_1$, in which case $q_1 = P_{\mathbf{A},1}(\ell_1) \geq P_{\mathbf{A},1}(m_1) = r_1$.

The fourth statement is proven similarly; with the same variables, we know $\ell_0 + \ell_1 \leq m_0 + m_1$, so either $\ell_0 \leq m_0$, in which case $q_0 = P_{\mathbf{A},0}(\ell_0) \leq P_{\mathbf{A},0}(m_0) = r_0$, or $\ell_1 \leq m_1$, in which case $q_1 = P_{\mathbf{A},1}(\ell_1) \leq P_{\mathbf{A},1}(m_1) = r_1$. □

**Theorem 3.7.** *If $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ are increasing, then for any $p_0, p_1$ such that $0 \leq p_0, p_1 \leq 1$, the sequence $R_{\mathbf{A}}^{(n)}(p_0, p_1)$ converges. Furthermore, its limit is a fixed point of $R_{\mathbf{A}}$.*

*Proof.* We first prove that the sequence converges. Let $(p_0^{(n)}, p_1^{(n)}) = R_{\mathbf{A}}^{(n)}(p_0, p_1)$.

- If there exist $N$ such that $p_0^{(N)} \geq p_0^{(N+1)}$ and $p_1^{(N)} \leq p_1^{(N+1)}$, then, by Lemma 3.6, for all $n > N$ we have $p_0^{(n)} \geq p_0^{(n+1)}$ and $p_1^{(n)} \leq p_1^{(n+1)}$. Thus, $\left(p_0^{(n)}\right)_{n=N}^{\infty}$ is a decreasing sequence bounded from below by 0, and $\left(p_1^{(n)}\right)_{n=N}^{\infty}$ is an increasing sequence bounded from above by 1. Thus, both converge, so the original sequence converges.

- If there exist $N$ such that $p_0^{(N)} \leq p_0^{(N+1)}$ and $p_1^{(N)} \geq p_1^{(N+1)}$, then, by the same logic, $\left(p_0^{(n)}\right)_{n=N}^{\infty}$ is an increasing sequence bounded from above by 1, and $\left(p_1^{(n)}\right)_{n=N}^{\infty}$ is a decreasing sequence bounded from below by 0. Thus, both converge, so the original sequence converges.

- If neither of the above holds, then for all $n$, we have either $p_0^{(n)} > p_0^{(n+1)}$ and $p_1^{(n)} > p_1^{(n+1)}$, or $p_0^{(n)} < p_0^{(n+1)}$ and $p_1^{(n)} < p_1^{(n+1)}$.

  - Note that, if $p_0^{(0)} > p_0^{(1)}$ and $p_1^{(0)} > p_1^{(1)}$, then by Lemma 3.6, we have either $p_0^{(1)} \geq p_0^{(2)}$ or $p_1^{(1)} \geq p_1^{(2)}$. Then, in fact, since neither of the above cases apply, both of these inequalities must hold and they must be strict. So by induction, $p_0^{(n)} > p_0^{(n+1)}$ and $p_1^{(n)} > p_1^{(n+1)}$ for all $n$.
  - Similarly, if $p_0^{(0)} < p_0^{(1)}$ and $p_1^{(0)} < p_1^{(1)}$, we have that $p_0^{(n)} < p_0^{(n+1)}$ and $p_1^{(n)} < p_1^{(n+1)}$ for all $n$.

  Thus, once again, $\left(p_0^{(n)}\right)_{n=0}^{\infty}$ and $\left(p_1^{(n)}\right)_{n=0}^{\infty}$ are monotonic and bounded within the interval $[0, 1]$, so they both converge and the original sequence converges.

Combining the above cases, we see that $\left(R_{\mathbf{A}}^{(n)}(1, 1)\right)_{n=0}^{\infty}$ converges. Then by continuity (since $R_{\mathbf{A}}$ is a polynomial),

$$\begin{aligned}\lim_{n\to\infty} R_{\mathbf{A}}^{(n)}(1,1) &= \lim_{n\to\infty} R_{\mathbf{A}}\left(R_{\mathbf{A}}^{(n)}(1,1)\right)\\ &= R_{\mathbf{A}}\left(\lim_{n\to\infty} R_{\mathbf{A}}^{(n-1)}(1,1)\right)\\ &= R_{\mathbf{A}}\left(\lim_{n\to\infty} R_{\mathbf{A}}^{(n)}(1,1)\right)\end{aligned}$$

so the limit is a fixed point of $R_{\mathbf{A}}$. □

## 4 Uniformly Biased Algorithms

In this section we will further narrow our target to a class of asymmetric algorithms called "uniformly biased", which we will define below, and compare them to a symmetric majority algorithm, which will produce a counterexample to Theorem 1.2 of [8].

First, recall that a majority algorithm is an algorithm that outputs the majority of its inputs, with no restriction on its output if there are equally many 0s and 1s in the input. Note that, when $d$ is odd, the latter case cannot happen, so there is only one "majority algorithm", and it is clearly symmetric. In general, we can see that $\mathbf{A}$ is a majority algorithm iff in the corresponding sequence $(c_j)_{j=0}^{d}$ of Lemma 3.2, we have $c_j = 0$ for $j < d/2$ and $c_j = 1$ for $j > d/2$.

**Lemma 4.1.** *When $d$ is even, a majority algorithm is symmetric iff $c_{d/2} = 1/2$.*

*Proof.* Immediate from the above discussion and Corollary 3.5. □

**Lemma 4.2.** *When $d$ is even, all symmetric majority algorithms are equivalent.*

*Proof.* Immediate from the definition, Lemma 3.2, and Lemma 4.1. □

Since all symmetric majority algorithms are equivalent, and equivalent algorithms behave identically for the purpose of recursive reconstruction, we will speak of *the* symmetric majority algorithm and denote it by $\mathbf{M}$. For example, $\mathbf{M}$ could be taken to be the reconstruction algorithm that outputs the majority of its inputs, and when there is a tie, outputs either 0 or 1 with equal probability. (We could also consider a deterministic version of $\mathbf{M}$ that, say, outputs the label of the first leaf when there is a tie.)

Since $\mathbf{M}$ is symmetric, for any $\ell$ we have $(r, r) = P_{\mathbf{M},0}(\ell, \ell) = P_{\mathbf{M},1}(\ell, \ell)$ for some $r$; we will write

$$P_{\mathbf{M}}(\ell) = r$$

for the above. Also, for any $p$ we have $(q, q) = R_{\mathbf{M}}(p, p)$ for some $q$; we will write

$$R_{\mathbf{M}}(p) = q$$

for the above. Explicitly, $P_{\mathbf{M}}(\ell)$ is the probability that the symmetric majority algorithm reconstructs a 0 if provided with $d$ inputs, each of which has probability $\ell$ of being a 0 independently (which is equal to the probability it reconstructs a 1 if each leaf has probability $\ell$ of being a 1 independently); and $R_{\mathbf{M}}(p)$ is the probability that the symmetric majority algorithm correctly reconstructs the root when provided with the $d$ leaves of a random $(\varepsilon, p, p)$-labeling of the tree.

**Lemma 4.3.** *$P_{\mathbf{M}}$ is an increasing concave function on $[1/2, 1]$ and symmetric about the point $(1/2, 1/2)$.*

*Proof.* Clearly, $P_{\mathbf{M}}(1/2) = 1/2$, since every sequence of leaf labels is equally likely as its complement. By [8, Lemma 2.5], the derivative of $P_{\mathbf{M}}$ is of the form

$$\frac{dP_{\mathbf{M}}}{d\ell} = C_d(\ell(1-\ell))^{\lfloor (d-1)/2 \rfloor}$$

for a positive constant $C_d$ depending on $d$. This function is clearly nonnegative and decreasing in the interval $[1/2, 1]$, and symmetric about $1/2$. The conclusion follows. $\square$

**Lemma 4.4.** *Suppose $0 \le \ell_0 \le \ell_1 \le 1$ and $\ell_0 + \ell_1 \ge 1$. Then*

$$P_{\mathbf{M}}\left(\frac{\ell_0 + \ell_1}{2}\right) \ge \frac{1}{2}(P_{\mathbf{M}}(\ell_0) + P_{\mathbf{M}}(\ell_1)).$$

*Proof.* If $1/2 \le \ell_0$, it is immediate from Jensen's inequality. Otherwise, by symmetry of $P_{\mathbf{M}}$ about $(1/2, 1/2)$ we have

$$P_{\mathbf{M}}(1-\ell_0) - P_{\mathbf{M}}(1/2) = P_{\mathbf{M}}(1/2) - P_{\mathbf{M}}(\ell_0).$$

Let $m = \ell_1 + \ell_0 - 1/2$. Then, note that $\ell_1 - m = 1/2 - \ell_0 = (1-\ell_0) - 1/2$, and that $m \ge 1/2$. Thus, again by concavity, we have

$$P_{\mathbf{M}}(\ell_1) - P_{\mathbf{M}}(m) \le P_{\mathbf{M}}(1-\ell_0) - P_{\mathbf{M}}(1/2) = P_{\mathbf{M}}(1/2) - P_{\mathbf{M}}(\ell_0).$$

Rearranging and applying Jensen's inequality again, we have

$$\frac{P_{\mathbf{M}}(\ell_1) + P_{\mathbf{M}}(\ell_0)}{2} \le \frac{P_{\mathbf{M}}(1/2) + P_{\mathbf{M}}(m)}{2} \le P_{\mathbf{M}}\left(\frac{1/2 + m}{2}\right) = P_{\mathbf{M}}\left(\frac{\ell_0 + \ell_1}{2}\right)$$

as desired. $\square$

We will now analyze an asymmetric algorithm $\mathbf{A}$ that is, in some sense, always biased towards one of the possible outputs:

**Definition 4.5.** *A reconstruction algorithm* $\mathbf{A}$ *is* **uniformly biased towards** *$i$ if the sequence $(c_j)_{j=0}^{d}$ corresponding to $P_{\mathbf{A},i}$ satisfies*

$$\begin{aligned} c_j &\ge 0 && \text{if } j < d/2 \\ c_j &\ge 1/2 && \text{if } j = d/2 \\ c_j &= 1 && \text{if } j > d/2 \end{aligned}$$

*and at least one of these inequalities is strict.*

Below, we will assume that $\mathbf{A}$ is uniformly biased towards 1, and that $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ are increasing. Note that these conditions are satisfied by all asymmetric majority algorithms that are more likely to output 1 than 0 when a tie occurs, or even more generally by all algorithms that are uniformly biased towards 1 where the sequence $(c_j)$ corresponding to $P_{\mathbf{A},1}$ is increasing.

We consider the bias

$$B(\ell) := P_{\mathbf{A},1}(\ell) - P_{\mathbf{M}}(\ell),$$

which, by (4), is of the form

$$B(\ell) = \sum_{j=0}^{\lfloor d/2 \rfloor} b_j \ell^j (1-\ell)^{d-j}$$

for some constants $b_j \in [0,1]$, at least one of which is nonzero. Then $B(\ell) \geq 0$ with equality iff $\ell \in \{0,1\}$, and

$$P_{\mathbf{A},0}(\ell) = 1 - P_{\mathbf{A},1}(1-\ell) = 1 - P_{\mathbf{M}}(1-\ell) - B(1-\ell) = P_{\mathbf{M}}(\ell) - B(1-\ell),$$

so, $P_{\mathbf{A},0} \leq P_{\mathbf{A},1}$ on the interval $[0,1]$, also with equality only when $\ell \in \{0,1\}$. Using this, we first observe some simple bounds on the values produced from iterating $R_{\mathbf{A}}$:

**Lemma 4.6.** *Suppose $\mathbf{A}$ is a reconstruction algorithm that is uniformly biased towards 1 such that $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ are increasing. Let $p_0, p_1$ be real numbers such that $0 \leq p_0 \leq p_1 \leq 1$ and $p_0 + p_1 \geq 1$. If $(q_0, q_1) = R_{\mathbf{A}}(p_0, p_1)$, then we also have $0 \leq q_0 \leq q_1 \leq 1$ and $q_0 + q_1 \geq 1$.*

*Proof.* It is clear that $0 \leq q_0, q_1 \leq 1$ because they are probabilities. If $(\ell_0, \ell_1) = L(p_0, p_1)$, we can see from Lemma 3.1 that $\ell_0 \leq \ell_1$ and that $\ell_0 + \ell_1 \geq 1$. Since $P_{\mathbf{A},0}$ is increasing, we have

$$q_0 = P_{\mathbf{A},0}(\ell_0) \leq P_{\mathbf{A},1}(\ell_0) \leq P_{\mathbf{A},1}(\ell_1) = q_1.$$

Finally, by (3),

$$1 = P_{\mathbf{A},0}(1-\ell_1) + P_{\mathbf{A},1}(\ell_1) \leq P_{\mathbf{A},0}(\ell_0) + P_{\mathbf{A},1}(\ell_1) = q_0 + q_1.$$

□

Next, we compare $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ to $P_{\mathbf{M}}$:

**Lemma 4.7.** *Suppose $0 \leq \ell_0 \leq \ell_1 \leq 1$ and $\ell_0 + \ell_1 \geq 1$. Then*

$$P_{\mathbf{M}}(\ell_0) + P_{\mathbf{M}}(\ell_1) \geq P_{\mathbf{A},0}(\ell_0) + P_{\mathbf{A},1}(\ell_1).$$

*Proof.* First, note that we can write

$$B(\ell) = \sum_{j=0}^{\lfloor d/2 \rfloor} b_j (\ell(1-\ell))^j (1-\ell)^{d-2j},$$

and $j$ and $d-2j$ are nonnegative in every term. Next, observe that

$$1-(1-\ell_0)=\ell_0\geq 1-\ell_1,$$

and because $1-\ell_1\leq\ell_0\leq\ell_1$, we have $\ell_0$ is closer to $1/2$ than $\ell_1$, so

$$\ell_0(1-\ell_0)\geq\ell_1(1-\ell_1).$$

By termwise comparison, this shows that $B(1-\ell_0)\geq B(\ell_1)$. Then,

$$\begin{aligned}P_{\mathbf{M}}(\ell_0)+P_{\mathbf{M}}(\ell_1)&\geq P_{\mathbf{M}}(\ell_0)+P_{\mathbf{M}}(\ell_1)+B(\ell_1)-B(1-\ell_0)\\&=P_{\mathbf{A},0}(\ell_0)+P_{\mathbf{A},1}(\ell_1)\end{aligned}$$

as desired. □

**Corollary 4.8.** *Suppose* $0\leq\ell_0\leq\ell_1\leq 1$ *and* $\ell_0+\ell_1\geq 1$. *Then*

$$P_{\mathbf{M}}\left(\frac{\ell_0+\ell_1}{2}\right)\geq\frac{P_{\mathbf{A},0}(\ell_0)+P_{\mathbf{A},1}(\ell_1)}{2}.$$

*Proof.* Immediate from Lemma 4.4 and Lemma 4.7. □

With this corollary, we can show that the asymmetric algorithms we study do no better than symmetric majority on average:

**Lemma 4.9.** *For a nonnegative integer* $n$, *let* $R_{\mathbf{M}}^{(n)}(1)=q$ *and let* $R_{\mathbf{A}}^{(n)}(1,1)=(q_0,q_1)$; *then*

$$\frac{q_0+q_1}{2}\leq q.$$

*Proof.* Suppose we have $p_0,p_1,p$ such that $0\leq p\leq 1$, $0\leq p_0\leq p_1\leq 1$, $p_0+p_1\geq 1$, and $(p_0+p_1)/2\leq p\leq 1$. Let $\ell=(1-\varepsilon)p+\varepsilon(1-p)$ and $(\ell_0,\ell_1)=L(p_0,p_1)$, and let $q=R_{\mathbf{M}}(p)=P_{\mathbf{M}}(\ell)$ and $(q_0,q_1)=R_{\mathbf{A}}(p_0,p_1)=(P_{\mathbf{A},0}(\ell_0),P_{\mathbf{A},1}(\ell_1))$. Then we have $\ell\geq(\ell_0+\ell_1)/2$. By Lemma 4.6, we have $0\leq q\leq 1$, $0\leq q_0\leq q_1\leq 1$, and $q_0+q_1\geq 1$. Finally, by Corollary 4.8 and the fact that $P_{\mathbf{M}}$ is increasing, we have

$$R_{\mathbf{M}}(p)=P_{\mathbf{M}}(\ell)\geq P_{\mathbf{M}}\left(\frac{\ell_0+\ell_1}{2}\right)\geq\frac{P_{\mathbf{A},0}(\ell_0)+P_{\mathbf{A},0}(\ell_1)}{2}=\frac{q_0+q_1}{2}.$$

Since $p_0=p_1=p=1$ satisfy the initial conditions, the result follows from induction. □

Next, we observe that if $p_0$ drops below $1/2$, then at the limit, the algorithm will never succeed at reconstructing the label 0. In other words, no matter what the root is, the algorithm will always guess 1.

**Lemma 4.10.** *Suppose* $0\leq p_0\leq 1/2\leq p_1\leq 1$ *and* $p_0+p_1\geq 1$. *Then the limit of* $R_{\mathbf{A}}^{(n)}(p_0,p_1)$ *as* $n$ *tends to infinity is* $(0,1)$.

*Proof.* Let $\ell_0 = L_0(p_0, p_1) = (1-\varepsilon)p_0 + \varepsilon(1-p_1)$. Note that $1 - p_1 \le p_0$, so

$$\ell_0 \le p_0 \le 1/2.$$

Let $R_{\mathbf{A}}(p_0, p_1) = (q_0, q_1)$. Then, note that

$$q_0 = P_{\mathbf{A},0}(\ell_0) \le P_{\mathbf{M}}(\ell_0) \le \ell_0,$$

with equality in the first inequality only if $\ell_0 = 0$ (or 1, but that's impossible here). Therefore,

$$q_0 \le p_0,$$

with equality only if $p_0 = 0$. In addition, by Lemma 4.6, we have that $q_0 + q_1 \ge 1$.

By induction, this implies that the first coordinates of the sequence $R_{\mathbf{A}}^{(n)}(p_0, p_1)$ are decreasing. By Theorem 3.7, the sequence converges to a limit that is a fixed point of $R_{\mathbf{A}}$, so the limit's first coordinate must be less than the initial $p_0$. By the above argument, we know $R_{\mathbf{A}}$ has no fixed points when $0 < p_0 \le 1/2$, so at the limit we must have $p_0 = 0$. Then it is immediate that $p_1 = 1$. □

We are almost ready to show that our algorithm $\mathbf{A}$ fails to reconstruct for a range of $\varepsilon$ where the majority algorithm succeeds. Let $p_{\mathbf{M}}$ be the greatest fixed point of $R_{\mathbf{M}}$. Let $\ell_{\mathbf{M}} = (1-\varepsilon)p_{\mathbf{M}} + \varepsilon(1-p_{\mathbf{M}})$. Note that, due to the continuity of $R_{\mathbf{M}}$, we have that $p_{\mathbf{M}}$ is a continuous function of $\varepsilon$, and $p_{\mathbf{M}} = 1/2$ precisely when $\mathbf{M}$ cannot achieve recursive reconstruction.

Let $\ell_T$ be the point in $[1/2, 1]$ where $dP_{\mathbf{M}}/d\ell(\ell_T) = 1$. Note that $\ell_T$ does not depend on $\varepsilon$. Since $P_{\mathbf{M}}(1/2) = 1/2$ and $P_{\mathbf{M}}(1) = 1$, this point exists by the Mean Value Theorem, and since $P_{\mathbf{M}}$ is concave on that interval by Lemma 4.3, this value is unique and is greater than 1/2.

**Theorem 4.11.** *If $\ell_{\mathbf{M}} < (1/2 + \ell_T)/2$, then the limit of $R_{\mathbf{A}}^{(n)}(1,1)$ as $n$ tends to infinity is $(0,1)$, and therefore the asymmetric recursive algorithm fails to reconstruct (regardless of how biased it is).*

*Proof.* Let $(q_0, q_1) = R_{\mathbf{A}}^{(n)}(1,1)$ and let $q = R_{\mathbf{M}}^{(n)}(1,1)$. Then let $(\ell_0, \ell_1) = L(q_0, q_1)$ and let $\ell = (1-\varepsilon)q + \varepsilon(1-q)$. As $n$ goes to infinity, $q$ approaches $p_{\mathbf{M}}$, so $\ell$ approaches $\ell_{\mathbf{M}}$; as a result, for all sufficiently large $n$, $\ell < (1/2 + \ell_T)/2$. By Lemma 4.9, we have $(q_0 + q_1)/2 \le q$, which by Lemma 3.1 implies

$$\frac{\ell_0 + \ell_1}{2} \le \ell < \frac{1/2 + \ell_T}{2}$$

for all sufficiently large $n$.

Let $(q_0', q_1')$ be the limit of $R_{\mathbf{A}}^{(n)}(1,1)$ as $n$ goes to infinity, which exists and is a fixed point of $R_{\mathbf{A}}$ by Theorem 3.7. Let $(\ell_0', \ell_1') = L(q_0', q_1')$. By the above, we have

$$\frac{\ell_0' + \ell_1'}{2} < \frac{1/2 + \ell_T}{2}.$$

If $\ell_1' \le \ell_T$, then $dP_{\mathbf{M}}/d\ell \ge 1$ on the interval $[\ell_0', \ell_1']$, which implies $q_1' - q_0' = P_{\mathbf{M}}(\ell_1') - P_{\mathbf{M}}(\ell_0') > \ell_1' - \ell_0' = q_1' - q_0'$, a contradiction. But, if $\ell_1' > \ell_T$, we have $\ell_0 \le 1/2$, which implies $q_0' \le 1/2$, so by Lemma 4.10 we have $(q_0', q_1') = (0,1)$, as desired. □

Since $\ell_{\mathbf{M}}$ is a continuous function of $\varepsilon$, we conclude that there is a range of $\varepsilon$ where $1/2 < \ell_{\mathbf{M}} < (1/2 + \ell_T)/2$. When $\varepsilon$ is in this range, the symmetric majority algorithm achieves recursive reconstruction where all uniformly biased algorithms with increasing $P_{\mathbf{A},i}$ fail. The latter class of algorithms include asymmetric majority algorithms, thus proving that symmetry is essential for the last claim of Theorem 2.7.

The above bound is not tight. We conjecture that $\ell_T$ is actually the exact threshold where asymmetric algorithms start to fail:

**Conjecture 4.12.** *If $\ell_{\mathbf{M}} < \ell_T$, then the limit of $R_{\mathbf{A}}^{(n)}(1,1)$ as $n$ tends to infinity is $(0,1)$, and therefore the asymmetric recursive algorithm fails to reconstruct.*

The intuitive justification is that after sufficiently many iterations of $R_{\mathbf{A}}$, the value of $(\ell_1 + \ell_2)/2$ will fall under $\ell_{\mathbf{M}}$, and once it falls under $\ell_T$, the value of $p_1 - p_0$ starts to grow exponentially with each iteration, which starts a positive feedback loop that drives it to $(0,1)$. Unfortunately it's not clear how to prove that we can't reach a fixed point in this case, where $(\ell_0 + \ell_1)/2$ has settled below $\ell_{\mathbf{M}}$ but $\ell_1$ is enough above $\ell_T$ that $P_{\mathbf{M}}(\ell_1) - P_{\mathbf{M}}(\ell_0) = \ell_1 - \ell_0$ can hold. In the next section, we prove our conjecture with very specific algebraic techniques for the case of $d = 4$.

# 5 Proving the Conjectured Threshold for $d = 4$

As before, we assume $\mathbf{A}$ is a reconstruction algorithm that is uniformly biased towards 1, and that $P_{\mathbf{A},0}$ and $P_{\mathbf{A},1}$ are increasing. We also fix $d = 4$. We will assume $\mathbf{A}$ achieves recursive reconstruction, and derive a bound on $\varepsilon$.

Suppose that $R_{\mathbf{A}}^{(n)}(1,1)$ converges to $(p_0, p_1)$, so that $(p_0, p_1)$ is a fixed point of $R_{\mathbf{A}}$. As usual, let $(\ell_0, \ell_1) = L(p_0, p_1)$. Then we can manipulate the definitions of $L_i$ to get

$$P_{\mathbf{A},i}(\ell_i) - \ell_i = p_i - \ell_i = p_i - ((1-\varepsilon)p_i + \varepsilon(1 - p_{1-i})) = \varepsilon p_0 + \varepsilon p_1 - \varepsilon$$

for $i = 0, 1$. Let $y = \varepsilon p_0 + \varepsilon p_1 - \varepsilon$. Then we have

$$P_{\mathbf{M}}(\ell_1) - \ell_1 < P_{\mathbf{A},1}(\ell_1) - \ell_1 = y = P_{\mathbf{A},0}(\ell_0) - \ell_0 < P_{\mathbf{M}}(\ell_0) - \ell_0. \tag{6}$$

It's clear from induction that $p_0 < p_1$, so $\ell_0 < \ell_1$. Since $R_{\mathbf{A}}$ achieves recursive reconstruction, we know by Lemma 4.10 that $1/2 < \ell_0$. We can explicitly compute that

$$P_{\mathbf{M}}(x) = x^4 + 4x^3(1-x) + 3x^2(1-x)^2 = -2x^3 + 3x^2.$$

Consider the cubic

$$C(x) := P_{\mathbf{M}}(x) - x = -2x^3 + 3x^2 - x.$$

Note that $C(1/2) = C(1) = 0$. We will denote the derivatives of $C$ with primes. We can compute that the second derivative

$$C''(x) = -12x + 6,$$

which is nonpositive in the interval $[1/2, 1]$, so $C$ is concave on this interval.

Let $\ell_T$ be defined as in the previous section, such that $dP_{\mathbf{M}}/dx(\ell_T) = 1$; then we see from the definition of $C$ that $C'(\ell_T) = 0$. Since $C$ is concave on $[1/2, 1]$, we see that $C$ is increasing on the interval $[1/2, \ell_T]$ and decreasing on the interval $[\ell_T, 1]$.

We know from (6) that $0 < y < C(\ell_T)$, so there are two solutions to $C(x) = y$ in the interval $[1/2, 1]$; let them be $x_0$ and $x_1$. By the behavior of $C$ we see that $x_0 < \ell_0$ and $x_1 < \ell_1$.

Since $C$ is a real cubic we know $C(x) = y$ has exactly one other solution $x_{-1}$. Since $C(x) \leq 0$ on the interval $[0, 1/2]$, we see that $x_{-1} \leq 0$, and by Vieta's formula we also have $x_{-1} + x_0 + x_1 = 3/2$.

As $y$ takes values in the interval $[0, C(\ell_T)]$, we can consider $x_{-1}$ as a function of $y$. Since $C'(x) = -6x^2 + 6x = 6x(1-x) < 0$ and $C''(x) = -12x + 6 > 0$ when $x < 0$, we know $y$ is a decreasing convex function of $x_{-1}$, which means $x_{-1}$ is a decreasing convex function of $y$. Next, from Lemma 3.1 we can solve for $y$ in terms of $\ell_i$ to get that

$$y = \frac{\varepsilon(\ell_0 + \ell_1 - 1)}{1 - 2\varepsilon} > \frac{\varepsilon(x_0 + x_1 - 1)}{1 - 2\varepsilon} = \frac{\varepsilon(1/2 - x_{-1})}{1 - 2\varepsilon},$$

from which it follows that

$$x_{-1} > \frac{1}{2} - \frac{(1 - 2\varepsilon)y}{\varepsilon}. \tag{7}$$

However, (7) is false when $y = 0$, because $x_{-1} = 0$ but the RHS is $1/2$. Therefore, since the right-hand side is a linear function of $y$, if (7) is true for some $y \in [0, C(\ell_T)]$, it must be true for $y = C(\ell_T)$. At that point we have $x_{-1} = 3/2 - x_0 - x_1 = 3/2 - 2\ell_T$; rearranging, we get

$$C(\ell_T) - \frac{\varepsilon(2\ell_T - 1)}{1 - 2\varepsilon} > 0.$$

Let

$$f(\ell) = C(\ell) - \frac{\varepsilon(2\ell - 1)}{1 - 2\varepsilon} = P_{\mathbf{M}}(\ell) - \frac{\ell - \varepsilon}{1 - 2\varepsilon}.$$

Then from the above, we have that $f(\ell_T) > 0$. We observe that $(1-\varepsilon)p + \varepsilon(1-p) = \ell$ iff $p = \frac{\ell - \varepsilon}{1 - 2\varepsilon}$; thus, if we define $p$ as such, we see that $f(\ell) = 0$ iff $R_{\mathbf{M}}(p) = p$. Thus, $\ell_{\mathbf{M}}$ is the largest root of $f$ in $[0, 1]$. Since $C$ is concave, $f$ is concave, and since $f(1/2) = 0$ and $\ell_T > 1/2$, we conclude $\ell_T < \ell_{\mathbf{M}}$, proving Conjecture 4.12 when $d = 4$.

# 6 Generalizations

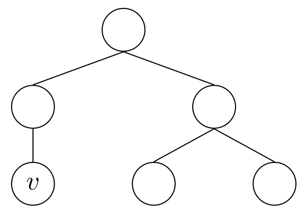
In the analysis above, although we attempted to keep our methods as general as possible, our ultimate results only apply to a small class of asymmetric algorithms for the 1-level $d$-regular tree, although we believe this class of algorithms contains all intuitively reasonable asymmetric algorithms. Since asymmetric algorithms in the $d$-leaf case can be thought of as isomorphic to $[0,1]^{d+1}$ as per Lemma 3.2, we can get some sense of the scale of our results by noting that we have only explored part of a $\lceil d/2 \rceil$-dimensional subspace of this. On the other hand, the results of the original paper only apply to symmetric algorithms, which is also a $\lceil d/2 \rceil$-dimensional subspace.

We have also not proven any results about general trees. The techniques of [8] rely on the useful concept of tree domination from [4] to show that the 1-level $d$-regular tree is the "easiest to reconstruct", in some sense, and that the majority algorithm achieves reconstruction on all trees. It remains conceivable that asymmetric algorithms that take advantage of tree structure could perform well on general trees, although we think it is unlikely.

We can also generalize the tree reconstruction problem to use labels from an alphabet larger than $\{0,1\}$. In this generalized version, if the alphabet has $k$ labels, then each child is given the same label as its parent with probability $1-(k-1)\varepsilon$, and each of the possible different labels with probability $\varepsilon$ each. These were briefly discussed in [8], which points out that the tree domination technique for proving that the 1-level $d$-regular tree is the "easiest to reconstruct" no longer apply, although they note that there is still the most mutual information between the leaves and root of that tree than those of any other tree. However, our results suggest that a different obstacle exists in even the 1-level $d$-regular trees, in that the performance of the "obvious" majority algorithm is very sensitive to tie-breaking behavior. When there are only two labels, a tie is a "global" phenomenon, whereas with three or more labels, there may be ties for the most common label involving only a small fraction of the leaves. In addition, ties can also span more than two labels.

When we examine multilevel trees with more than two labels, there is bad news for generalization as well. Generalizing the results from [8], a natural candidate for the optimal recursive reconstruction algorithm is the "unbiased plurality" algorithm, which outputs the most common label among its inputs, choosing uniformly randomly from among the most common labels if there is more than one. However, if we consider the case of three labels and look at the simplest possible asymmetric tree depicted in Figure 6, we find already that unbiased plurality does not do as well as a plurality algorithm that breaks ties towards the node $v$, i.e. the algorithm outputs the most common label among its inputs, but when each label appears once in its input, it outputs the label of $v$. Note that the latter algorithm is still symmetric — it is not more likely to reconstruct any label over any other. Empirically (with Sage [3]), we find that for a range of $\varepsilon$ including $[0.0485, 0.049]$, this plurality algorithm that breaks ties towards $v$ achieves reconstruction while the unbiased plurality algorithm fails to do so. This is in stark contrast to the 2-label situation studied in [8], where all symmetric plurality algorithms achieve reconstruction for the same range of $\varepsilon$.

Figure 4: The simplest possible asymmetric tree. Ties should be broken in favor of $v$.




# 7 Acknowledgements

This paper developed from a MIT Summer Program in Undergraduate Research (SPUR) project. The author would like to thank his SPUR mentor, Ashwin Narayan, for guidance throughout the project; the SPUR advisors, Ankur Moitra and David Jerison, for many helpful comments and pointers to useful background material; his fellow SPUR students, Ashley Kim and Max Murin, for reading drafts of the paper; and Elchanan Mossel for suggesting the topic, writing a clear paper to investigate, and encouraging the author to finish writing this paper.

# References

[1] Pavel M. Bleher, Jean Ruiz, and Valentin A. Zagrebnov. On the purity of the limiting gibbs state for the ising model on the bethe lattice. *Journal of Statistical Physics*, 79(1-2):473–482, 1995.

[2] Christian Borgs, Jennifer Chayes, Elchanan Mossel, and Sébastien Roch. The kesten-stigum reconstruction bound is tight for roughly symmetric binary channels. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 518–530. IEEE, 2006.

[3] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 7.6)*, 2017. `http://www.sagemath.org`.

[4] William Evans, Claire Kenyon, Yuval Peres, and Leonard J Schulman. Broadcasting on trees and the ising model. *Annals of Applied Probability*, pages 410–433, 2000.

[5] Dmitry Ioffe. On the extremality of the disordered state for the ising model on the bethe lattice. *Letters in Mathematical Physics*, 37(2):137–143, 1996.

[6] Svante Janson and Elchanan Mossel. Robust reconstruction on trees is determined by the second eigenvalue. *The Annals of Probability*, 32(3B):2630–2649, 2004.

[7] Harry Kesten and Bernt P. Stigum. Additional limit theorems for indecomposable multidimensional galton-watson processes. *The Annals of Mathematical Statistics*, 37(6):1463–1481, 1966.

[8] Elchanan Mossel. Recursive reconstruction on periodic trees. *Random Structures and algorithms*, 13(1):81–97, 1998.

[9] Elchanan Mossel. Reconstruction on trees: Beating the second eigenvalue. *Annals of Applied Probability*, pages 285–300, 2001.

[10] Elchanan Mossel and Yuval Peres. Information flow on trees. *The Annals of Applied Probability*, 13(3):817–844, 2003.

[11] Allan Sly. Reconstruction for the potts model. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 581–590. ACM, 2009.