8-1998

Recursive Reconstruction on Periodic Trees

Elchanan Mossel
University of Pennsylvania

Recursive Reconstruction on Periodic Trees

Abstract

A periodic tree T_n consists of full n-level copies of a finite tree T. The tree T_n is labeled by random bits. The root label is chosen randomly, and the probability of two adjacent vertices to have the same label is $1-\epsilon$. This model simulates noisy propagation of a bit from the root, and has significance both in communication theory and in biology. Our aim is to find an algorithm which decides for every set of values of the boundary bits of T, if the root is more probable to be 0 or 1. We want to use this algorithm recursively to reconstruct the value of the root of T_n with a probability bounded away from ½ for all n. In this paper we find for all T, the values of ϵ for which such a reconstruction is possible. We then compare the ϵ values for recursive and nonrecursive algorithms. Finally, we discuss some problems concerning generalizations of this model.

Disciplines

Statistics and Probability

Recursive Reconstruction on Periodic Trees

Elchanan Mossel

Hebrew University of Jerusalem

February 21, 2007

Abstract

A periodic tree T_n consists of full n-level copies of a finite tree T. The tree T_n is labeled by random bits. The root label is chosen randomly, and the probability of two adjacent vertices to have the same label is $1 - \epsilon$. This model simulates noisy propagation of a bit from the root, and has significance both in communication theory and in biology. Our aim is to find an algorithm which decides for every set of values of the boundary bits of T, if the root is more probable to be 0 or 1. We want to use this algorithm recursively to reconstruct the value of the root of T_n with a probability bounded away from $1/2$ for all n. In this paper we find for all T, the values of ϵ for which such a reconstruction is possible. We then compare the ϵ values for recursive and non-recursive algorithms. Finally, we discuss some problems concerning generalizations of this model.

1 Introduction and Main Results

Given a finite rooted tree T, having all d leaves at the same level h, we construct recursively a sequence of periodic trees T_n. Set $T_1 = T$, and construct T_n from T_{n-1} by replacing each of the leaves of T_{n-1} with a copy of T. A different method for constructing T_n is to take the n-level d-regular tree and replace each occurrence of the 1-level d-regular tree with the tree T. We call T_n the n-period of T.

On the vertices of the tree T_n, we have a random labeling by bits. The bit on the root is chosen to be 0 or 1 with probability $1/2$ each. For all edges $e = (v_1, v_2)$ the probability that the bit assigned to v_1 is equal to the bit assigned to v_2 is $1 - \epsilon$, independently for all edges, where $\epsilon \leq 1/2$ is fixed. We call this labeling the ϵ-labeling.
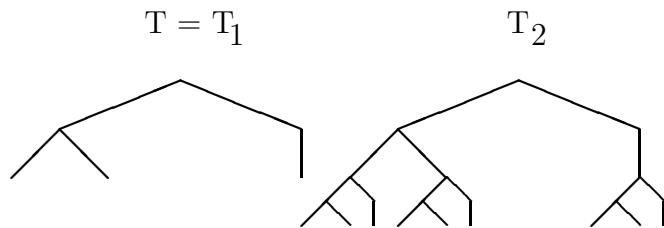

Figure 1: The first two periods of a tree T

This model has several interpretations. In communication theory, one may consider each of the edges as an independent symmetric channel. In biology, one may say that there is some binary property each child inherits from its parent independently for all children. This model is also an example of an Ising model on T_n with constant interaction strength and one of two given Gibbs measures (cf. Lyons [10]).

A *Reconstruction Algorithm* $\mathbf{A}$ for a finite tree T having l leaves, is an algorithm that for given values of the bits on the boundary of T, computes a value for the root of T. Thus, $\mathbf{A}$ is a function, $\mathbf{A} : \{0,1\}^l \to \{0,1\}$. Note that the same reconstruction algorithm may apply to all trees having l leaves. We denote by $r(T, \mathbf{A})$ the probability that $\mathbf{A}$ computes the value of the root of T correctly. For a tree T having all d leaves at the same level h, we say that *the bit reconstruction problem is solvable for ϵ*, if there exist a $\delta > 0$ and algorithms $\mathbf{A}_n$, such that,

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

for all n.

Kamae (unpublished, see Higuchi [9]) has shown that when T_n is the n-level d-regular tree, the reconstruction problem is solvable if $1 - 2\epsilon > d^{-1/2}$. Bleher, Ruiz and Zagrebnov ([2]) have shown that the reconstruction problem for the regular tree is unsolvable if $1 - 2\epsilon \leq d^{-1/2}$. Subsequently, Evans, Kenyon, Peres and Schulman ([4]) studied the same problem for general trees and proved a similar bound involving the *branching number* of the tree. In the current setting, where T is a tree having all d leaves at the same level h, their result shows that the reconstruction problem is solvable if $\epsilon < \epsilon_c$, and unsolvable if $\epsilon > \epsilon_c$, where,

$$\epsilon_c = (1 - d^{-\frac{1}{2h}})/2. \tag{1}$$

For a more detailed history of this problem see [4].

For a finite tree, when ϵ is known, the algorithm which achieves the best reconstruction probability is the *Maximum-Likelihood Algorithm*. When ϵ is small and all the internal vertices have an odd number of children, this algorithm coincides with the *Parsimony Algorithm* - so called in Mathematical Biology (see [7]), or the *Recursive Majority Algorithm* - so called in the theory of computation. This algorithm has the following recursive description: Starting at the boundary, it assigns to each vertex the bit which is the majority of the bits assigned to the children of that vertex. This algorithm has some advantages over the maximum-likelihood algorithm, as the exact value of ϵ is not needed for the algorithm. Moreover, computationally, it has a very simple recursive description. These advantages have made this algorithm popular in biology. When ϵ is not small, the algorithm is not optimal. This motivates us to check how well recursive reconstruction algorithms behave.

Definitions 1.1:

1. Let $\mathbf{A} : \{0,1\}^d \to \{0,1\}$ be a reconstruction algorithm. We construct the *periods* of $\mathbf{A}$, $\mathbf{A}^n : \{0,1\}^{d^n} \to \{0,1\}$, as follows: We set $\mathbf{A}^1 = \mathbf{A}$, and construct $\mathbf{A}^k$ from $\mathbf{A}^{k-1}$ by:

$$\mathbf{A}^k(\sigma_1, \ldots, \sigma_{d^k}) = \mathbf{A}\Big(\mathbf{A}^{k-1}(\sigma_1, \ldots, \sigma_{d^{k-1}}), \ldots, \mathbf{A}^{k-1}(\sigma_{(d-1)d^{k-1}+1}, \ldots, \sigma_{d^k})\Big) \tag{2}$$

Thus, when T is a tree having all d leaves at the same level h, $\mathbf{A}^n$ is a reconstruction algorithm for T_n, with the following recursive description: Starting at

2

the boundary, for each copy of T, we replace the labels $\sigma = (\sigma_1, \ldots, \sigma_d)$ on the boundary of T by a label $\mathbf{A}(\sigma)$ on the root of T, which is the output of the computation of $\mathbf{A}$ on σ.

2. We say that *the bit reconstruction problem is recursively solvable* for ϵ (or T and ϵ) if there exists a reconstruction algorithm $\mathbf{A}$ for T, such that

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

for some $\delta > 0$ and all n. In this case we say that $\mathbf{A}$ *recursively solves the reconstruction problem for* ϵ *(or T and ϵ).*

In our main result we give the exact critical ϵ for recursive reconstruction and exhibit the recursive algorithms which achieve this ϵ.

Theorem 1.2: *Given a tree T having all d leaves at the same level h, the bit reconstruction problem is recursively solvable if and only if $\epsilon < \epsilon_r$, where ϵ_r is defined by,*

$$(\frac{1}{2})^{d-1}\lceil\frac{d}{2}\rceil\binom{d}{\lceil\frac{d}{2}\rceil}(1 - 2\epsilon_r)^h = 1. \tag{3}$$

Moreover, the algorithms $\mathbf{M} : \{0,1\}^d \to \{0,1\}$ which recursively solve the reconstruction problem for all $\epsilon < \epsilon_r$ are those which assign to the root of T the value of the majority of the leaves (when the number of 0 and 1 is equal, it may assign any value).

Note that both in (1) and in the Theorem 1.2 the exact shape of the tree T is unimportant, only the values of d and h play a role in both results. The result of Theorem 1.2 may be surprising at first sight. For example, let T be a tree having 200 stalks of length 100, and T' be a tree having 100 stalks of length 100 and one stalk of length 99 with 100 children (see Figure (2)).
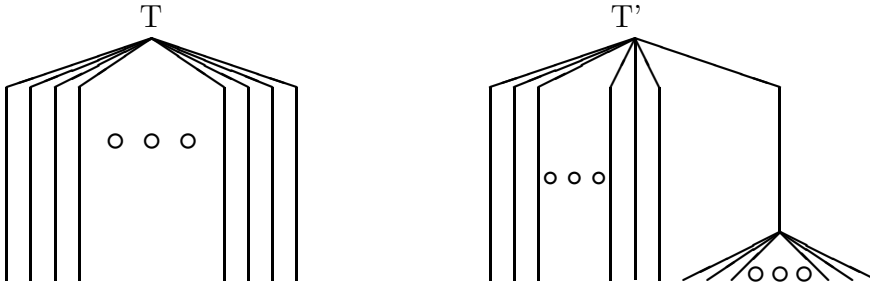

Figure 2: T and T'

By Theorem 1.2 both trees have the same critical ϵ for recursive reconstruction. Moreover, in both trees the 'best' algorithm is a majority algorithm. For T it seems reasonable, since the symmetry of the tree forces the algorithm to be symmetric. But for T', the 100 children of the 99 stalk give us very good information on their parent, but poor information on the root of the tree. It may seem strange that these 100 leaves should get the same weight as the 100 other. We will return to this example after the proof of Theorem 1.2.

3

It is clear that $\epsilon_r \leq \epsilon_c$. However, we may try to approximate algorithms which achieve ϵ_c by recursive algorithms which work on 'large blocks'. More formally, it follows easily from the definitions that $(T_k)_n = T_{kn}$. For the subsequence $\{T_{kn}\}_{n=1}^{\infty}$ we may reconstruct the root using recursively a reconstruction algorithm for T_k. When we take $k \to \infty$, we obtain the following result:

Proposition 1.3: *Given a tree T having all d leaves at the same level h, we denote by $\epsilon_r(k)$ the critical ϵ for recursive solvability of T_k, and by ϵ_c the critical ϵ for which the bit reconstruction problem is solvable for T. Then for all k,*

$$\epsilon_r(k) < \epsilon_c \tag{4}$$

and

$$\lim_{k \to \infty} \epsilon_r(k) = \epsilon_c. \tag{5}$$

The structure of the paper is as follows: In Section 2 we prove Theorem 1.2 for the special case where T_n is the n-level d-regular tree. In Section 3 we use local considerations to prove the lower bound of Theorem 1.2, i.e, we prove that recursive reconstruction is possible for $\epsilon < \epsilon_r$. In Section 4 we use a 'domination lemma' proven in [4], to prove the upper bound, i.e, reconstruction is not possible for $\epsilon \geq \epsilon_r$; complete the proof of Theorem 1.2 and prove Proposition 1.3. In section 5 we show that the 'domination lemma' can not be extended to more general cases, and discuss some remarks and open problems.

2   Regular Trees

In this section we prove Theorem 1.2 in the special case where T_n is the n-level d-regular tree. The proof of this case uses iteration of real functions, as in the classical theory of extinction of branching processes (cf. Atherya and Ney [1], Chapter 1). Similar methods also appear in a work on noisy computation by Evans ([6]).

We begin with some definitions which apply also to the more general (non-regular) case. A basic tool for analyzing the behavior of an algorithm $\mathbf{A}$, is the recursive function of $\mathbf{A}$, which measures how well $\mathbf{A}$ reconstructs the root of T, when we are not given the original boundary bits, but those bits with some additional noise.

Definition 2.1: *For a tree T the (ϵ, p)-labeling of T is the following labeling: First, we label T by an ϵ-labeling. Then for the leaves of T only, we keep each label with probability p, and take the complementary label with probability $1 - p$, independently for all leaves. For an algorithm $\mathbf{A}$ and a tree T, let $R_{\mathbf{A}}^{\epsilon}(p)$ (or $R_{\mathbf{A}, T}^{\epsilon}(p)$) be the probability that $\mathbf{A}$ computes the root of T correctly, when the input to $\mathbf{A}$ is the bits on the boundary of the (ϵ, p)-labeling of T.*

Lemma 2.2: *Let $\mathbf{A}$ be a reconstruction algorithm for T. Then:*

$$r(T_n, \mathbf{A}^n) = R_{\mathbf{A}}^{\epsilon}{}^{(n)}(1),$$

where $f^{(n)}$ denotes the n-times composition of f with itself.

Proof: Immediate by induction. □

Now we focus on the case where T is the 1-level d-regular tree. That is, T_n is the n-level d-regular tree. In this case, as Lemma 2.3 shows, the interaction between ϵ and p in

$R_{\mathbf{A}}^{\epsilon}(p)$ is simple. Moreover, as Lemma 2.4 shows, only majority algorithms should be considered. From now on, we will denote by $\mathbf{M}$ majority algorithms, i.e, algorithms which output the majority of the inputs (if the number of 0 and 1 in the input is equal, we allow the output to be either 0 or 1).

Lemma 2.3: *When T is the 1-level d-regular tree, for all algorithms $\mathbf{A}$ and for all p,*

$$R_{\mathbf{A}}^{\epsilon}(p) = R_{\mathbf{A}}^{0}((1-\epsilon)p + \epsilon(1-p)).$$

Proof: The right hand term expresses the probability that $\mathbf{A}$ will compute the root correctly, when each of the boundary bits is identical to the root with probability $(1-\epsilon)p + \epsilon(1-p)$ independently. But this is also the left hand term. □

Lemma 2.4: *When T is the 1-level d-regular tree, for all algorithms $\mathbf{A}$ and for all $1 \geq p \geq 1/2$*

$$R_{\mathbf{A}}^{\epsilon}(p) \leq R_{\mathbf{M}}^{\epsilon}(p), \tag{6}$$

where $\mathbf{M}$ is any majority algorithm. In particular, the bit reconstruction problem is recursively solvable if and only if $\mathbf{M}$ recursively solves the bit reconstruction problem.

Proof: The second claim follows from the first by Lemma 2.2 and Definition 1.1. In order to prove the first claim, by Lemma 2.3, it is enough to show that

$$R_{\mathbf{A}}^{0}(p) \leq R_{\mathbf{M}}^{0}(p)$$

for all $p \geq 1/2$. Suppose that there are d_0 values of 0 and d_1 values of 1 on the boundary of the 1-level d-regular tree with the p-labeling. If we denote the value of the root bit by $root$, then

$$\frac{\mathbf{P}[root=0|d_0,d_1]}{\mathbf{P}[root=1|d_0,d_1]} = \frac{\mathbf{P}[d_0,d_1|root=0]}{\mathbf{P}[d_0,d_1|root=1]} = \frac{p^{d_0}(1-p)^{d_1}}{p^{d_1}(1-p)^{d_0}} = \left(\frac{p}{1-p}\right)^{d_0-d_1}.$$

Therefore, in order that $R_{\mathbf{A}}^{0}(p)$ will be maximal, we must set $\mathbf{A} = 0$ whenever $d_0 > d_1$, and $\mathbf{A} = 1$ whenever $d_0 < d_1$. Moreover, when $d_0 = d_1$, we may set either $\mathbf{A} = 0$ or $\mathbf{A} = 1$. □

The proof of the Theorem 1.2 for 1-level d-regular trees is based on some properties of the function $f(p) = R_{\mathbf{M}}^{0}(p)$ which will be derived using Russo's Formula (cf. Grimmet [8], Chapter 4). We define $E = \{0,1\}^d$. A function F on E is *increasing* if whenever we change a bit value from 0 to 1 the value of the function increases. We denote by $\mathbf{P}_p$ the probability measure on E which assigns each coordinate 1 with probability p independently, and by $\mathbf{E}_p$ the corresponding expected value function. We have

Russo's Formula: *If F is increasing, then,*

$$\frac{d}{dp}\mathbf{E}_p(F) = \sum_{i=1}^{d} \mathbf{E}_p[\delta_i F],$$

where,

$$\delta_i(F(x_1,\ldots,x_d)) = F(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n) - F(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n).$$

Lemma 2.5: When T is the 1-level d-regular tree, $R_{\mathbf{M}}^{\epsilon}(p)$ is an increasing concave function on $[1/2, 1]$. The derivative of $R_{\mathbf{M}}^{\epsilon}(p)$ at $1/2$ satisfies:

$$\frac{dR_{\mathbf{M}}^{\epsilon}}{dp}(\frac{1}{2}) = (1 - 2\epsilon)(\frac{1}{2})^{d-1} \lceil \frac{d}{2} \rceil \binom{d}{\lceil \frac{d}{2} \rceil}. \tag{7}$$

Proof: Following Lemma 2.3, it is enough to show that $f(p) = R_{\mathbf{M}}^{0}(p)$ is an increasing concave function on $[1/2, 1]$, and that

$$\frac{df}{dp}(\frac{1}{2}) = (\frac{1}{2})^{d-1} \lceil \frac{d}{2} \rceil \binom{d}{\lceil \frac{d}{2} \rceil}. \tag{8}$$

We begin by calculating the derivative of f using Russo's formula. Consider the space $E = \{0, 1\}^d$, where we set 1 at the i-th coordinate if the i-th child of the 1-level d-regular tree has the same label as the root, and set 0 in the i-th coordinate if the i-th child and the root have different labels. In this model, the probability of each coordinate to be 1 is p, independently for all coordinates. We denote by $n_0(x)$ the number of 0 coordinates of x, and by $n_1(x)$ the number of 1 coordinates. By definition,

$$f(p) = \mathbf{P}_p[\{x \in E : n_1(x) > n_0(x)\}] + \frac{1}{2}\mathbf{P}_p[\{x \in E : n_1(x) = n_0(x)\}].$$

Therefore, if we define,

$$F(x) = \begin{cases} 1 & \text{if} & n_1(x) > n_0(x) \\ 1/2 & \text{if} & n_1(x) = n_0(x) \\ 0 & \text{if} & n_1(x) < n_0(x) \end{cases},$$

then $f(p) = \mathbf{E}_p(F)$. F is an increasing function on E. Thus, by Russo's Formula and the definition of F we have for odd d,

$$\begin{aligned} \frac{df}{dp}(p) &= \frac{d+1}{2}\mathbf{P}_p[\{x \in E : |n_1(x) - n_0(x)| = 1\}] \\ &= \frac{d+1}{2}\binom{d}{\frac{d+1}{2}}\left(p^{\frac{d+1}{2}}(1-p)^{\frac{d-1}{2}} + p^{\frac{d-1}{2}}(1-p)^{\frac{d+1}{2}}\right) \\ &= \frac{d+1}{2}\binom{d}{\frac{d+1}{2}}(p(1-p))^{\frac{d-1}{2}}, \end{aligned} \tag{9}$$

and for even d,

$$\begin{aligned} \frac{df}{dp}(p) &= \frac{d}{2}\mathbf{P}_p[\{x \in E : n_1(x) = n_0(x)\}] + \frac{d+2}{4}\mathbf{P}_p[\{x \in E : |n_1(x) - n_0(x)| = 2\}] \\ &= \frac{d}{2}\binom{d}{\frac{d}{2}}p^{\frac{d}{2}}(1-p)^{\frac{d}{2}} + \frac{d+2}{4}\binom{d}{\frac{d+2}{2}}\left(p^{\frac{d+2}{2}}(1-p)^{\frac{d-2}{2}} + p^{\frac{d-2}{2}}(1-p)^{\frac{d+2}{2}}\right) \\ &= \frac{d}{4}\binom{d}{\frac{d}{2}}(p(1-p))^{\frac{d-2}{2}}. \end{aligned} \tag{10}$$

Since the derivatives in (9) and (10) are positive, the function f is increasing in $[1/2, 1]$. Since the function $p(1-p)$ is decreasing in $[1/2, 1]$, both (9) and (10) are decreasing in $[1/2, 1]$ and therefore f is concave in $[1/2, 1]$. (8) follows directly from (9) and (10). $\square$

Proposition 2.6: *For the 1-level d-regular tree the reconstruction problem is recursively solvable if and only if $\epsilon < \epsilon_r$, where ϵ_r is given by*

$$(\frac{1}{2})^{d-1} \lceil \frac{d}{2} \rceil \binom{d}{\lceil \frac{d}{2} \rceil} (1 - 2\epsilon_r) = 1.$$

Moreover, majority algorithms recursively solve the reconstruction problem for all $\epsilon < \epsilon_r$.

Proof: By Lemma 2.4 we should only consider majority algorithms **M**. By Lemma 2.5 the function $R_{\mathbf{M}}^\epsilon(p)$ is increasing and concave in $[1/2, 1]$. Since this function is increasing, the sequence $\{R_{\mathbf{M}}^{\epsilon \, (n)}(1)\}_{n=1}^\infty$ is decreasing, and therefore has a limit:

$$l_\epsilon = \lim_{n \to \infty} R_{\mathbf{M}}^{\epsilon \, (n)}(1).$$

The problem would be recursively solvable if and only if $l_\epsilon > 1/2$. Since $R_{\mathbf{M}}^\epsilon(p)$ is increasing, $l_\epsilon = \max\{t \in [1/2, 1] : R_{\mathbf{M}}^\epsilon(t) = t\}$. Since $R_{\mathbf{M}}^\epsilon(1/2) = f(1/2) = 1/2$, $R_{\mathbf{M}}^\epsilon(1) < 1$, and $R_{\mathbf{M}}^\epsilon(p)$ is concave in $[1/2, 1]$, $l_\epsilon > 1/2$ if and only if

$$\frac{dR_{\mathbf{M}}^\epsilon}{dp}(\frac{1}{2}) > 1,$$

and the claim follows from Lemma 2.5. □

3 The Lower Bound

In the next two sections we prove Theorem 1.2 in the general case. In these sections we fix T to be a finite tree having all d leaves at same level h, and we consider the periods of T, T_n. In the general case, (6) in Lemma 2.4 does not hold, and in order to prove that majority algorithms recursively solve the reconstruction problem for all $\epsilon < \epsilon_r$, a different approach is needed. We start by analyzing the derivative of the majority algorithm at the point $1/2$.

Lemma 3.1: *For majority algorithms:*

$$\frac{dR_{\mathbf{M}}^\epsilon}{dp}(\frac{1}{2}) = (\frac{1}{2})^{d-1} \lceil \frac{d}{2} \rceil \binom{d}{\lceil \frac{d}{2} \rceil} (1 - 2\epsilon)^h. \tag{11}$$

*For any other algorithm **A**,*

$$\frac{dR_{\mathbf{A}}^\epsilon}{dp}(\frac{1}{2}) = C_{\mathbf{A}} (1 - 2\epsilon)^h, \tag{12}$$

where $C_{\mathbf{A}}$ is a constant such that,

$$C_{\mathbf{A}} < (\frac{1}{2})^{d-1} \lceil \frac{d}{2} \rceil \binom{d}{\lceil \frac{d}{2} \rceil}.$$

Proof: We define two auxiliary functions: $R_0^\epsilon(p, \sigma)$ and $R_1^\epsilon(p, \sigma)$, where $R_0^\epsilon(p, \sigma)$ is the probability that the root-value of the (ϵ, p)-labeling of T is 0, when the labeling of the

7

boundary bits is σ. $R_{\mathbf{1}}^{\epsilon}(p, \sigma)$ is defined similarly. Let $P^{\epsilon}(p, \sigma)$ be the probability that the boundary of the (ϵ, p)-labeling of T is σ. It is clear that:

$$R_{\mathbf{A}}^{\epsilon}(p) = \sum_{\sigma} P^{\epsilon}(p, \sigma) R_{\mathbf{A}(\sigma)}^{\epsilon}(p, \sigma). \tag{13}$$

Differentiation of (13) generates:

$$\frac{dR_{\mathbf{A}}^{\epsilon}}{dp}(p) = \sum_{\sigma} \frac{dP^{\epsilon}}{dp}(p, \sigma) R_{\mathbf{A}(\sigma)}^{\epsilon}(p, \sigma) + \sum_{\sigma} P^{\epsilon}(p, \sigma) \frac{dR_{\mathbf{A}(\sigma)}^{\epsilon}}{dp}(p, \sigma).$$

At the point $p = 1/2$, $P^{\epsilon}(1/2, \sigma) = 2^{-d}$ and $R_{\mathbf{A}(\sigma)}^{\epsilon}(1/2, \sigma) = 1/2$ for all σ. Therefore we get:

$$\frac{dR_{\mathbf{A}}^{\epsilon}}{dp}\left(\frac{1}{2}\right) = \frac{1}{2} \sum_{\sigma} \frac{dP^{\epsilon}}{dp}\left(\frac{1}{2}, \sigma\right) + \frac{1}{2^d} \sum_{\sigma} \frac{dR_{\mathbf{A}(\sigma)}^{\epsilon}}{dp}\left(\frac{1}{2}, \sigma\right).$$

However,

$$\sum_{\sigma} \frac{dP^{\epsilon}}{dp}(p, \sigma) = \frac{d}{dp}\left(\sum_{\sigma} P^{\epsilon}(p, \sigma)\right) = \frac{d}{dp} 1 = 0,$$

and therefore

$$\frac{dR_{\mathbf{A}}^{\epsilon}}{dp}\left(\frac{1}{2}\right) = \frac{1}{2^d} \sum_{\sigma} \frac{dR_{\mathbf{A}(\sigma)}^{\epsilon}}{dp}\left(\frac{1}{2}, \sigma\right). \tag{14}$$

In order to evaluate (14), let $(p_1, \ldots, p_d)$ be a vector of probabilities, and consider for a moment the following labeling of T. First, we label T by the ϵ-labeling. Then, only for the leaves of T keep the i-th leaf label with probability p_i, and take the complementary label with probability $1 - p_i$. We define $R_{\mathbf{b}}^{\epsilon}(p_1, \ldots, p_d, \sigma)$ to be the probability that the root of this labeling is $\mathbf{b}$ when the boundary is σ. Using the chain rule we get:

$$\frac{dR_{\mathbf{b}}^{\epsilon}}{dp}\left(\frac{1}{2}, \sigma\right) = \sum_{i=1}^{d} \frac{\partial R_{\mathbf{b}}^{\epsilon}}{\partial p_i}\left(\frac{1}{2}, \ldots, \frac{1}{2}, \sigma\right) \tag{15}$$

for $\mathbf{b} \in \{\mathbf{0}, \mathbf{1}\}$.

$R_{\mathbf{b}}^{\epsilon}(1/2, \ldots, 1/2, p_i, 1/2, \ldots, 1/2, \sigma)$ is the probability that the root-value is $\mathbf{b}$, when the i-th boundary bit is the same as the root with probability

$$\frac{1}{2} + \frac{1}{2}(1 - 2\epsilon)^h (2p_i - 1),$$

and all the other bits are i.i.d $0, 1$ variables which are independent of the root. Therefore,

$$R_{\mathbf{b}}^{\epsilon}\left(\frac{1}{2}, \ldots, \frac{1}{2}, p_i, \frac{1}{2}, \ldots, \frac{1}{2}, \sigma\right) = \begin{cases} \frac{1}{2} + \frac{(1-2\epsilon)^h(2p_i-1)}{2} & \text{if} \quad \mathbf{b} = \sigma_i \\ \frac{1}{2} - \frac{(1-2\epsilon)^h(2p_i-1)}{2} & \text{if} \quad \mathbf{b} \neq \sigma_i \end{cases},$$

and,

$$\frac{\partial R_{\mathbf{b}}^{\epsilon}}{\partial p_i}\left(\frac{1}{2}, \ldots, \frac{1}{2}, \sigma\right) = \begin{cases} +(1 - 2\epsilon)^h & \text{if} \quad \mathbf{b} = \sigma_i \\ -(1 - 2\epsilon)^h & \text{if} \quad \mathbf{b} \neq \sigma_i \end{cases}. \tag{16}$$

Substituting (16) into (15) gives:

$$\frac{dR_{\mathbf{b}}^{\epsilon}}{dp}\left(\frac{1}{2}, \sigma\right) = (1 - 2\epsilon)^h \left(|\{i : \sigma_i = \mathbf{b}\}| - |\{i : \sigma_i \neq \mathbf{b}\}|\right). \tag{17}$$

8

Combining (17) with (14) gives:

$$\frac{dR_{\mathbf{A}}^{\epsilon}}{dp}(\frac{1}{2}) = \frac{(1-2\epsilon)^h}{2^d} \sum_{\sigma} \Big(|\{i : \sigma_i = \mathbf{A}(\sigma)\}| - |\{i : \sigma_i \neq \mathbf{A}(\sigma)\}| \Big). \qquad (18)$$

In order to prove (11), one should calculate the sum on the right of (18) for majority algorithms. An elegant way to finesse this calculation is to observe that for majority algorithms, the sum is a number which depends only on d. Since we already calculated the derivative for $h = 1$ in Lemma 2.5, (11) follows.

The proof of (12) follows since the sum on the right of (18) does not depend on ϵ, and is maximal when $\mathbf{A}$ is a majority algorithm. $\qquad \square$

Using Lemma 3.1 the proof of the lower bound is fairly easy. The derivative at $1/2$ of the majority recursive function determines the behavior of this function near $1/2$. We only have to make sure that this function behaves well far from $1/2$:

Lemma 3.2: *For all $\epsilon < 1/2$ and all $p \in (1/2, 1]$,*

$$R_{\mathbf{M}}^{\epsilon}(p) > 1/2.$$

Proof: Consider the following percolation model on the edges of T. For each edge which is not connected to a leaf, set the edge open with probability $1 - 2\epsilon$ (and closed otherwise). For each edge which is connected to a leaf, set the edge open with probability $(1-2\epsilon)(2p-1)$ (and closed otherwise). Each edge is closed or open independently of all other edges. We now consider the following random labeling. We always label two vertices which share an open edge with the same bit (starting at the root). When the edge connecting a parent and a child is closed, we choose the label of the child randomly (0 and 1 with probability $1/2$ each). Obviously, this gives the same labeling as the (ϵ, p)-labeling.

Let C be the random set of leaves which are connected to the root in the percolation model. We denote by N the random set of leaves. We note that $|C| > d/2$ with positive probability, and on this event the majority algorithm computes the root correctly. On N the number of bits which are the same as the root has the same distribution as the number of bits which are different from the root. Therefore, on the event $|C| \leq d/2$, the probability that the majority algorithm will succeed is at least $1/2$. Combining the last two claims we get the lemma. $\qquad \square$

Proposition 3.3: *For $\epsilon < \epsilon_r$, where ϵ_r is given in (3), majority algorithms solve the recursive reconstruction problem.*

Proof: Let $\mathbf{M}$ be a majority algorithm. We want to show that for $\epsilon < \epsilon_r$ the sequence $\{R_{\mathbf{M}}^{\epsilon}{}^{(n)}(1)\}_{n=1}^{\infty}$ is bounded away from $1/2$. It is enough to show that that there exists a $\delta > 1/2$ such that for all p in $[\delta, 1]$,

$$R_{\mathbf{M}}^{\epsilon}(p) \geq \delta. \qquad (19)$$

By Lemma 3.1, for $\epsilon < \epsilon_r$,

$$\frac{d}{dp} R_{\mathbf{M}}^{\epsilon}(\frac{1}{2}) > 1,$$

therefore, there exists $\delta_1 > 1/2$ such that for all p in $(1/2, \delta_1)$,

$$R_{\mathbf{M}}^{\epsilon}(p) \geq p.$$

By Lemma 3.2

$$\delta_2 = \min_{[\delta_1, 1]} R_{\mathbf{M}}^\epsilon(p) > 1/2.$$

Taking $\delta = \min\{\delta_1, \delta_2\}$ we get (19). □

4   The Upper Bound

In this section we prove the upper bound of Theorem 1.2. The proof is based on a 'domination lemma' from Evans, Kenyon, Peres and Schulman ([4]), which compares the general case to the case of regular trees discussed in Section 2.

Definition 4.1[[4]]: *We say that a labeling on a tree T' dominates a labeling on a tree T, if there exists a noisy channel φ, such that the distribution of the labeling at the leaves of T with root-value B_{root}, equals φ applied to the distribution of the labeling at leaves of T' with root-value B_{root} (for $B_{root} = 0$ and $B_{root} = 1$).*

Definition 4.2[[4]]: *The disjoint tree, $Dis(T)$, associated with a tree T, is a tree which has the same set of root-leaf paths as T but in which these paths are disjoint. Every root-leaf path in T, corresponds to a root-leaf path in $Dis(T)$ of the same length. In addition, the paths in $Dis(T)$ are disjoint sharing only root only.*
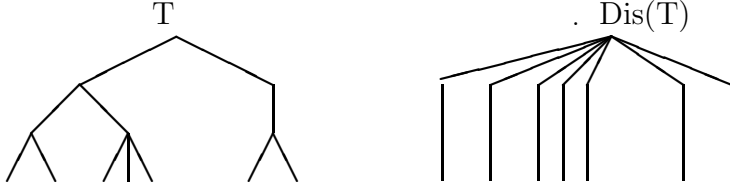

Figure 3: A tree T with Dis(T)

For example, if T has all d leaves at the same level h, then in the ϵ-labeling of $Dis(T)$ there are d leaves each having the same label as the root with probability $(1+(1-2\epsilon)^h)/2$ independently. In the (ϵ, p)-labeling of $Dis(T)$ there are d leaves each having the same label as the root with probability $(1 + (2p - 1)(1 - 2\epsilon)^h)/2$ independently.

Lemma 4.3: *Let T be a finite tree having all d leaves at the same level h, and $\tilde{T}$ be the 1-level d-regular tree. Let $\mathbf{A} : \{0, 1\}^d \to \{0, 1\}$ be a reconstruction algorithm. Then for all p:*

$$R_{\mathbf{A}, Dis(T)}^\epsilon(p) = R_{\mathbf{A}, \tilde{T}}^{1/2 - 1/2(1-2\epsilon)^h}(p).$$

Proof: $R_{\mathbf{A}, Dis(T)}^\epsilon(p)$ is the probability that $\mathbf{A}$ computes the root correctly when there are d leaves each identical to the root with probability $(1 + (2p - 1)(1 - 2\epsilon)^h)/2$ independently. But this is also $R_{\mathbf{A}, \tilde{T}}^{1/2 - 1/2(1-2\epsilon)^h}(p)$. □

The tool for comparing the easy case of $Dis(T)$, and the general case is given in the following lemma.

Lemma 4.4[Evans, Kenyon, Peres and Schulman [4]]: *The (ϵ, p)-labeling on $Dis(T)$ dominates the (ϵ, p)-labeling on T.*

10

For our purposes we need the following lemma which is an easy consequence of domination.

Lemma 4.5[[4]]: *Let T be a tree having all d leaves at the same level h, then for all $\epsilon < 1/2$ and $p \geq 1/2$,*

$$\max_{\mathbf{A}:\{0,1\}^d \to \{0,1\}} R^\epsilon_{\mathbf{A},Dis(T)}(p) \geq \max_{\mathbf{A}:\{0,1\}^d \to \{0,1\}} R^\epsilon_{\mathbf{A},T}(p) \tag{20}$$

Remark 4.6: Note that by Lemma 4.3 and Lemma 2.4, for all $\epsilon < 1/2$ and $p \geq 1/2$,

$$\max_{\mathbf{A}:\{0,1\}^d \to \{0,1\}} R^\epsilon_{\mathbf{A},Dis(T)}(p) = R^\epsilon_{\mathbf{M},Dis(T)}(p)$$

where $\mathbf{M}$ is a majority algorithm.

Proof of Theorem 1.2: We proved the lower bound in Proposition 3.3. In order to prove the upper bound let $\mathbf{A} : \{0,1\}^d \to \{0,1\}$ be an algorithm which recursively solves the reconstruction problem for $1/2 \geq \epsilon \geq \epsilon_r$, where ϵ_r is given in (3). Then, the sequence $\{R^\epsilon_{\mathbf{A},T}{}^{(n)}(1)\}_{n=1}^\infty$ is bounded away from $1/2$. If this sequence is not decreasing, then there exists a p in $(1/2, 1]$ such that

$$R^\epsilon_{\mathbf{A},T}(p) = p. \tag{21}$$

If it is decreasing, it has a limit p in $(1/2, 1]$ which satisfies (21). Let $\mathbf{M} : \{0,1\}^d \to \{0,1\}$ be a majority algorithm. By Lemma 4.5 and Remark 4.6:

$$R^\epsilon_{\mathbf{M},Dis(T)}(p) \geq R^\epsilon_{\mathbf{A},T}(p) = p \tag{22}$$

But by Lemma 4.3 and Lemma 2.5, $R^\epsilon_{\mathbf{M},Dis(T)}(p)$ is an increasing concave function on $[1/2, 1]$ with

$$R^\epsilon_{\mathbf{M},Dis(T)}\left(\frac{1}{2}\right) = \frac{1}{2} \qquad \frac{d}{dp} R^\epsilon_{\mathbf{M},Dis(T)}\left(\frac{1}{2}\right) = \left(\frac{1}{2}\right)^{d-1} \lceil \frac{d}{2} \rceil \binom{d}{\lceil \frac{d}{2} \rceil} (1 - 2\epsilon)^h \leq 1,$$

which contradicts (22).

In order to complete the proof, we must show that only majority algorithms recursively solve the reconstruction problem for all $\epsilon < \epsilon_r$. Suppose that $\mathbf{A} : \{0,1\}^d \to \{0,1\}$ is an algorithm which recursively solves the reconstruction problem for all $\epsilon < \epsilon_r$. We denote $R^\epsilon_{\mathbf{A}} = R^\epsilon_{\mathbf{A},T}$. The argument above shows that for all $\epsilon < \epsilon_r$ there exists a $p_\epsilon \in (1/2, 1]$, such that $R^\epsilon_{\mathbf{A}}(p_\epsilon) = p_\epsilon$. Let $p = \limsup p_\epsilon$. Since $R^\epsilon_{\mathbf{A}}(q)$ is a polynomial in ϵ and q, it is easy to see that, $p = R^{\epsilon_r}_{\mathbf{A}}(p)$. Therefore, by the first part of the proof

$$p = 1/2. \tag{23}$$

We note that since for $\epsilon < \epsilon_r$, $R^\epsilon_{\mathbf{A}}(1/2) = 1/2$, and $R^\epsilon_{\mathbf{A}}(p_\epsilon) = p_\epsilon$, there exists a q_ϵ in $[1/2, p_\epsilon]$ such that $\frac{dR^\epsilon_{\mathbf{A}}}{dp}(q_\epsilon) = 1$. But by (23) $\lim_{\epsilon \to \epsilon_r} q_\epsilon = 1/2$, and therefore, $\frac{dR^{\epsilon_r}_{\mathbf{A}}}{dp}(1/2) = 1$. By Lemma 3.1 this implies that $\mathbf{A}$ is a majority algorithm. $\square$

Proof of Proposition 1.3: By Stirling's formula:

$$\lim_{h \to \infty} \left(\left(\frac{1}{2}\right)^{d^h - 1} \lceil \frac{d^h}{2} \rceil \binom{d^h}{\lceil \frac{d^h}{2} \rceil} \right)^{1/h} = \sqrt{d}$$

11

which implies (5). Given (5), the proof of (4) follows once we show that for every $k \geq 1$ and $l \geq 2$,

$$\epsilon_r(kl) > \epsilon_r(k). \tag{24}$$

Let $\mathbf{A}$ be the l-level recursive application of a majority algorithm on T_k to T_{kl}. It is easy to see that $\mathbf{A}$ is not a majority algorithm on T_{kl}. Therefore, by Theorem 1.2 there exists an $\epsilon < \epsilon_r(kl)$ such that $\mathbf{A}$ fails to recursively solve the reconstruction problem for T_{kl} and ϵ, which implies (24). $\qquad\square$

Remarks:

1. Theorem 1.2 gives the possible values of ϵ for which there is an algorithm $\mathbf{A}$ such that the sequence $r(T_n, \mathbf{A}^n)$ is bounded away from $1/2$ for all n. The proof can be extended to the following model. Instead of taking T_n to be a sequence of periods of a fixed T, fix d and h, and take T_n to be a sequence of trees such that, T_n is constructed from the n-level d-regular tree by replacing each occurrence of the 1-level d-regular tree with some tree T, having all d leaves at the same level h. Here we may replace different copies of the 1-level d-regular tree with different trees.

2. In Proposition 1.3, one may use Stirling's formula to obtain that the sequence $\{\epsilon_r(k)\}_{k=1}^{\infty}$ is a strictly increasing sequence.

3. The proof of Theorem 1.2 provides some intuition why in the example at the introduction (see Figure 2) for both trees T and T' for all $\epsilon < \epsilon_r$ the algorithms which recursively solve the reconstruction problem are majority algorithms. For the tree T it follows from the fact that different bits on the boundary are independent given the root. The point is, that when ϵ is close to ϵ_r, after many iterations of the recursive algorithm, the reconstructed bits are equal to the original label-bits with probability close to $1/2$, and are almost independent given the root in T' as well as in T.

5 Discussion

In this section we discuss several natural generalizations of the model we introduced in section 1. We first discuss models in which we replace the parent-child symmetric binary channel with some other channel.

The main tool in the proof of the upper bound both in the analysis of the non-recursive algorithms (by Evans *et al* [4]) and in the analysis of the recursive algorithms is Lemma 4.4. Extensions of this lemma to other channels, would give upper bounds for the corresponding trees.

We note first, that the notion of domination could be defined for any random labeling on trees. From the *data-processing lemma* it follows that if a labeling on T' dominates a labeling on T, then the mutual information between the root and the leaves in T' is greater than the mutual information between the root and the leaves in T.

5.1 Example of Non-Domination

We give an example of a tree T and a channel M for which the mutual information between all the boundary labels and the root label is positive, while the mutual information between each of the labels on the boundary and the root-label is zero. Thus, in

this example the mutual information between the root and the leaves of T is positive, while the mutual information between the root and the leaves of $Dis(T)$ is 0. Therefore, in this case $Dis(T)$ does not dominate T.

Let the tree T be the 'Y tree'. It consists of 4 vertices: the root R, which has one son C (the center of the tree), which in turn has two sons $L1$ and $L2$ (the leaves of the tree).

Take M to be a Markov-chain on the 4 state-space $\mathbf{Z}_2 \times \mathbf{Z}_2$. M has the following transition matrix:

$$M(x,y) = \begin{cases} (r, x+r \bmod 2) & \text{with probability } 1/2 \\ (r, y+r \bmod 2) & \text{with probability } 1/2 \end{cases}$$

where r takes each of the values 0 and 1 with probability $1/2$, and is independent of anything else.
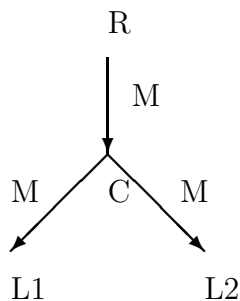

Figure 4: The 'Y tree' with the Markov chain M

Claim 1: *The double iteration of M satisfies:*

$$\mathbf{P}[M(M(x,y)) = (z,w)] = 1/4,$$

for all x, y, z, w. Therefore, the mutual information between R and $L1$ $(L2)$ is zero.
Proof: Immediate. $\qquad\square$

Claim 2: *If the initial distribution on R gives positive probability to every label in $\mathbf{Z}_2 \times \mathbf{Z}_2$, then the mutual information between R and $(L1, L2)$ is positive.*

Proof: Assume that it is given that the label on $L1$ is $(0,0)$, and the label on $L2$ is $(0,1)$. This could happen (and will happen with positive probability) only if the label on C is $(0,1)$ or $(1,0)$. In any case, it would imply that the label on the root R cannot be $(0,0)$. $\qquad\square$

Given the two claims, it is not surprising that for the Markov-chain M, the reconstruction problem solvability depends strongly on the exact shape of the tree. For example we can show that:

Claim 3: Let T be the 2-level 1000-regular tree equipped with the Markov chain M. The recursive reconstruction problem for T is solvable. For $Dis(T)$ equipped with Markov chain M, the reconstruction problem is not (recursively) solvable.

Proof: The second statement follows from Claim 1. We denote the label values by $(x,y) \in \mathbf{Z}_2 \times \mathbf{Z}_2$. In order to prove the first statement, it is enough to show that for the

1-level 1000-regular tree we have a recursive reconstruction algorithm for reconstructing $x+y \mod 2$ with probability greater than 0.99. An example of such a recursive algorithm is the following algorithm. For each vertex v look at the 1000 values of $x + y \mod 2$ for the children of v. If there exists a value which appears more than 700 times, declare the $x + y \mod 2$ of v 0. Otherwise, declare it 1. □

5.2 k-ary Channels

A more natural generalization of the binary symmetric channel, is given by the k-ary symmetric channel in which there are k symbols, each transmitted correctly with probability $1 - (k - 1)\epsilon$, and as any of the other symbols with probability ϵ (where $\epsilon \leq 1/k$). Note that when $k = 2$ we have the (original) binary symmetric channel. When $k = 3$, computer simulations indicate that in the 'Y tree' above there is no domination of $Dis(T)$ over T, and therefore, the methods we have used cannot be generalized to obtain the upper bound even in this case. However, unlike the example above, those simulations suggest that for 3-ary channels the mutual information between the root and the leaves of $Dis(T)$ is greater than the mutual information in T.

Some of the remaining open questions concerning these generalizations are: How do the critical ϵ values for recursive and non-recursive algorithms depend on the shape of the tree? What are the values of these ϵ? As far as I know, even for regular trees the last question is still open.

5.3 Other Generalizations

Another natural generalization is obtained if we allow our recursive algorithms on the $\{0, 1\}$ labeled periodic trees to have as output (and as input after the first level) a set with more than two symbols. Obviously, if we allow an unbounded number of symbols we are back at the case of non-recursive algorithms (since we can recursively 'transmit' the boundary to the root). However, can we say that when the number of symbols is bounded by some constant, the critical ϵ for recursive algorithms is always smaller than for the non-recursive algorithms? Can we find these critical ϵ as a function of the number of symbols? What are the algorithms which achieve these ϵ?

Finally, how can we extend Theorem 1.2 to non-periodic trees? One possible extension is to use the same setting that was used in the paper by Evans *et al* ([4]). Take T to be some infinite tree, and **A** an algorithm which for every h-level tree, computes a root value from the boundary values. Then apply **A** recursively to the first kh levels of T, and check if the reconstruction probability is bounded away from $1/2$ as $k \rightarrow \infty$. However, in this setting the behavior of recursive algorithms as a function of the shape of the tree may be complicated, as can be observed in the following example.

We look at the one level recursive majority algorithm, and consider the following two trees. Take T to be the infinite 3-regular tree. By Theorem 1.2 the critical ϵ for T is $1/6$. Take T' to be an extension of T in which to each vertex we add k stalks, each a copy of **N**. Even though T is a sub-tree of T', the critical ϵ for the one level recursive majority algorithm for T' is smaller than for T, and as $k \rightarrow \infty$ it tends to zero.
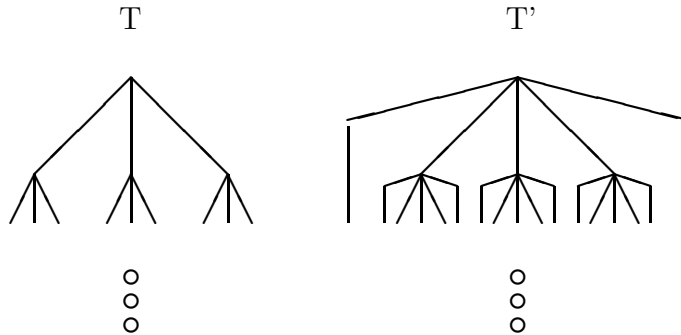
Figure 5: The trees T and T' for k=2

References

[1] Atherya, K. B. and Ney, P. E. (1972) *Branching Processes*, Springer-Verlag.

[2] Bleher, P. M., Ruiz, J. and Zagrebnov V. A. (1995) On the Purity of limiting Gibbs state for the Ising model on the Bethe lattice, *J. Stat. Phys* **79**, 473–482.

[3] Steel, M. and Charelston, M. (1995) Five surprising properties of parsimoniously colored trees, *Bull. of Math. Bio* **57 No 2**, 367–375.

[4] Evans, W., Kenyon, C., Peres, Y., and Schulman, L. J. (1997) Broadcasting on trees and the Ising model, preprint.

[5] Evans, W. and Schulman, L. J (1993) Signal propagation, with application to a lower bound on the depth of noisy formulas, *Proceeding of the 34th Annual Symposium on Foundations on computer Science*, 594–603.

[6] Evans, W. (1994) Information Theory and Noisy Computation. PhD thesis, University of California at Berkeley. Available by **ftp** at **ftp.icsi.berkeley.edu** as file **/pub/techreports/94/tr-94-057.ps**

[7] Farris, J. S. (1973) On the use of parsimony criterion for inferring evolutionary trees, *Syst. Zool* **22**, 250–256.

[8] Gine, E., Grimmett, G. R. and Sallof-Coste, L. (1997) *Lectures on probability theory and statistics*, Springer-Verlag.

[9] Higuchi, Y. (1977) Remarks on the limiting Gibbs state on a $(d + 1)$-tree, *Publ. RIMS Kyoto Univ* **13**, 335–348.

[10] Lyons, R. (1989) The ising Model and Percolation on Trees and Tree-Like Graphs, *Commun. Math. Phys* **125**, 337–352.

ELCHANAN MOSSEL

Department of Mathematics, Hebrew University, Jerusalem. `mossel@math.huji.ac.il`