

# Tractable Bayesian Social Learning on Trees

Yashodhan Kanoria and Omer Tamuz

**Abstract**—We study agents in a social network who learn by observing the actions of their neighbors. The agents iteratively estimate an unknown ‘state of the world’  $s$  from initial private signals, and the past actions of their neighbors in the social network.

First, we consider a set of Bayesian agents, and investigate the computational problem the agents face in implementing the (myopic) Bayesian decision rule. When private signals are independent conditioned on  $s$ , and when the social network graph is a tree, we provide a new ‘dynamic cavity algorithm’ for the agents’ calculations, with computational effort that is exponentially lower than what is currently known. We use our algorithm to perform the first numerical simulations of interacting Bayesian agents on networks with hundreds of nodes.

Second, we investigate a different model of social learning, with naive agents who practice ‘majority dynamics’, i.e., at each round adopt the majority opinion of their neighbors. Under mild conditions, we show that under majority dynamics, agents learn  $s$  with probability  $1 - \epsilon$  in  $O(\log \log(1/\epsilon))$  rounds.

We conjecture that on  $d$ -regular trees, myopic Bayesian agents learn  $s$  as quickly as agents who practice majority dynamics. Using our algorithm for Bayesian agents, the conjecture implies that the computational effort required of Bayesian agents to learn  $s$  is only polylogarithmic in  $1/\epsilon$  on  $d$ -regular trees. Thus, our results challenge the belief that iterative Bayesian learning is computationally intractable.

**Index terms:** social learning, Bayesian agents, computational efficiency, convergence, algorithm, dynamic cavity method.

## I. INTRODUCTION

Consider a graph  $G = (V, E)$ . An edge  $(i, j) \in E$  indicates that nodes  $i \in V$  and  $j \in V$  can observe each other. Nodes attempt to learn the true *state of the world*  $s$ . Each node  $i$  receives noisy information about  $s$  in the form of a ‘private signal’  $x_i$ . In each discrete time period (or round)  $t = 0, 1, \dots$ , each node chooses an action  $\sigma_i(t) \in \mathcal{S}$ , which we call a ‘vote’. Nodes observe the votes cast by their neighbors in  $G$ . Thus, at the time of voting in round  $t \geq 1$ , the information available to a node  $i$  consists of the private signal  $x_i$ , along with the votes cast by neighbors of  $i$  in rounds up to  $t - 1$ .

The above setup is relevant in various contexts, and has thus been studied by diverse communities. On one hand, engineering problems like decentralized detection [23], [24] and distributed decision making [25] have motivated study of this setup, with the purpose of designing appropriate graphs  $G$  along with voting schemes to achieve desired engineering goals such as low infrastructure cost, small error probability, low communication cost, low computation cost, etc. On the other hand, this setup has been considered by sociologists and economists interested in the phenomenon of ‘social learning’ wherein agents (nodes) in a social network learn behavior by

observing the actions of their neighbors in the network. In this case, the focus is on understanding the science of such systems as they are in reality, with  $G$  being a model of a real social network, and the voting rule being a model of the behavior of real agents. In this paper, we focus the discussion on the social learning interpretation of this setup. However, our results and techniques are general, and are also relevant to the corresponding engineering questions.

The importance of social learning in networks has been demonstrated in a wide variety of settings (e.g., adoption of agricultural technology in Ghana [7] and Mozambique [4], choice of contraceptives by European women [16]). Accordingly, understanding mathematical models of social learning by Bayesian agents has been a goal of theoretical economics for the past few decades (cf., Goyal [12]). Typical models in this context assume a *pure information externality*; agent payoffs depend only on the action they choose and an underlying ‘state of the world’, and not on the actions of others. Agents observe the actions of their ‘neighbors’, but do not observe payoffs *ex interim*. Typically, all agents have the same utility function. Each agent receives a private signal that contains noisy information about the state of the world. Agents choose actions to maximize expected payoff, given their own private signal and their observations of the actions chosen by others.

Fully Bayesian models have two advantages over models that assume ‘bounded rationality’ and prescribe thumb rules for agent behavior: First, any bounded rationality approach is bound to involve a somewhat arbitrary decision of which heuristics the agents use. Second, a game theoretic analysis of strategic players is possible only if the players choose actions that are optimal by some criterion. Hence game-theoretic analyses of learning on networks (e.g., [21]) often opt for the more difficult but fully Bayesian model.

Much progress has been achieved in models where Bayesian agents act *sequentially*, such as the herd behavior models of Banerjee [5], Bikhchandani, Hirshleifer and Welch [6], Smith and Sørensen [22] and Acemoglu et al [1]. Here, the interaction is *not bidirectional*: each agent acts only once, taking into account the actions of her predecessors. In comparison, our understanding of Bayesian agents who act repeatedly is much more limited. Gale and Kariv [11] consider Bayesian agents on a network who repeatedly choose actions. They show, in the spirit of Aumann’s Agreement Theorem [2], that agents on a network converge to the same action under some conditions<sup>1</sup>. Related work by Rosenberg, Solan and Vieille [21] and Ménager [17] sheds more light on the phenomenon of agreement on actions and the conditions in which it arises.

Microsoft Research New England, ykanoria@microsoft.com  
Microsoft Research New England and Weizmann Institute,  
omer.tamuz@weizmann.ac.il. Supported by ISF grant 1300/08.

<sup>1</sup>A gap in the proof of Gale and Kariv’s agreement theorem was recently pointed out [18]. However, recent works [21], [19] establish similar results in more general settings.

However, the following questions remain essentially unanswered:

- (I) What action do the agents converge to, e.g., what is the distribution of this consensus action?
- (II) What are the dynamics of such interactions, e.g., what is the rate of agreement/convergence?

There has been a parallel development of non-Bayesian models of reasoning for social learning and social experimentation<sup>2</sup>, e.g., those of Ellison and Fudenberg [10], Bala and Goyal [3], and DeGroot [8]. Such modelling approaches appear to be driven by two primary motivating factors (see, e.g., [10], [3]): (i) Real agents may not be Bayesian. (ii) The desire to “*keep the model mathematically tractable*” [3], and also computationally tractable; since Bayesian models seem to lack these properties. This leads us to another open question in the context of Bayesian agents who act repeatedly:

- (III) Are the computations required of the agents feasible?

We consider a model that features repeated bidirectional interaction between fully Bayesian agents connected by a social network. Our model is a specialization of the model of Gale and Kariv [11]. We consider a group of Bayesian agents, each with a private signal that carries information on an unknown state of the world  $s$ . The individuals form a social network, so that each observes the actions of some subset of others, whom we call her neighbors. The agents must repeatedly choose between a set of possible actions, the relative merit of which depends on the state of the world  $s$ . The agents iteratively learn by observing their neighbors’ actions, and picking an action that is myopically optimal, given their information. Thus, the interaction between agents is not strategic, and is characterized by information externalities.

Even in the simple case of two states of the world, binary private signals and two possible actions, the required calculations appear to be very complicated. A naïve dynamic programming algorithm is exponential in the number of individuals. Although this algorithm seems to be well known, we could not find a complete description of it in the literature and hence supply it for completeness in Section IV. Since at iteration  $t$  one may consider only agents at distance  $t$ , then in graphs of maximum degree  $d$  (on which we focus) the number of individuals to consider is  $O(\min(n, d^t))$ , and the computational effort required of each individual to compute their action at time  $t$  is  $t2^{O(\min(n, d^t))}$ . Obviously, this grows very rapidly. As Gale and Kariv remark [11], “The computational difficulty of solving the model is massive even in the case of three persons.” This prevents them from even simulating networks with more than three nodes.

We describe a novel algorithm for the agents’ calculation in our model, when the social network graph is a tree or nearly a tree. This algorithm has running time that is exponentially smaller than the naïve dynamic program, reducing the computational effort to  $2^{O(\min(n, td))}$ .

Using our algorithm we are able to run numerical simulations of the social learning process. This extends the work

<sup>2</sup>Social experimentation settings are closely related to social learning settings: Here agents can observe (noisy) payoffs received by themselves and their neighbors for different actions, and can use the results of these ‘experiments’ to learn.

of Gale and Kariv [11], who simulated the process for three agents, to much larger networks<sup>3</sup>. We use our algorithm to investigate questions (I) and (II): We numerically evaluate the probability that the agents learn the optimal action, and its progress with time. We observe rapid learning of the optimal action in certain previously unexplored settings: We consider a model with two possible states of the world and two corresponding actions (‘votes’), so the agents are in effect trying to estimate the state of the world and revealing their estimates to their neighbors. The social networks in these analyses were chosen to be  $d$ -regular (infinite) trees, i.e., trees in which each node has  $d$  neighbors. The simulations suggest that, on regular trees, the number of iterations needed under Bayesian learning to estimate  $s$  correctly with probability  $1 - \epsilon$  is  $O(\log \log(1/\epsilon))$ .

We conjecture that the error probability under Bayesian updates is no larger than the error probability under a different ‘majority’ update rule, in which agents adopt the opinion of the majority of their neighbors in the previous round. Our numerical results support this conjecture. We prove that for the majority update rule, the number of iterations needed to estimate  $s$  correctly with probability  $1 - \epsilon$  is  $O(\log \log(1/\epsilon))$ , for regular trees of degree at least five<sup>4</sup>. Our conjecture then implies, again, that the number of iterations needed to estimate  $s$  correctly with probability  $1 - \epsilon$  is  $O(\log \log(1/\epsilon))$ . Thus, assuming the conjecture, the computational effort required of Bayesian agents drops from quasi-polynomial in  $1/\epsilon$  (using the naïve dynamic program) to polynomial in  $\log(1/\epsilon)$  (i.e., polylogarithmic in  $1/\epsilon$ ), making Bayesian learning computationally tractable. Thus, our results shed new light on question (III), suggesting a positive answer in the case of tree graphs.

Our algorithmic approach works provided the local neighborhood of a node is tree structured. The restriction of the discussion to tree or tree-like social networks certainly excludes many natural settings that tend to exhibit highly clustered social graphs. However, in some cases artificially constructed networks have no or few loops by design; these include some highly hierarchical or compartmentalized organizations, as well as some physical communication networks where redundancy is expensive, and the least expensive connected network is a tree. Furthermore, the fact that this non-trivial class of networks does not present a major computational hurdle for fully Bayesian calculations may in itself be somewhat surprising.

A key technique used in this paper is the dynamic cavity method, introduced by Kanoria and Montanari [13] in their study of ‘majority updates’ on trees, a model also motivated by social learning. This technique is a dynamical version of the cavity method of statistical physics and appears promising for the analysis of iterative tree processes in general. The key idea is the following: In a dynamical setting on a tree graph, there is correlation in the trajectories of neighbors of

<sup>3</sup>In each of our numerical analyses, agents receive information (directly or indirectly) from hundreds of distinct nodes.

<sup>4</sup>This result should be of independent interest. Majority dynamics is a reasonable model of social learning with bounded rationality. It is also relevant in other contexts like consensus in distributed systems[20]. We consider the results a main technical contribution of the paper

a node due to a nodes own past actions. The dynamic cavity method allows to exactly account for these correlations. In this work, we use this method for the first time to give a new algorithmic result, enabling efficient computation by nodes. This is in contrast to the case of majority updates, where the update rule is computationally trivial. Our algorithmic and analytical approach leveraging the dynamic cavity method may be applicable to a range of iterative update situations on locally treelike graphs.

A short conference version of this paper [15] contains a description of the dynamic cavity based algorithm and statements of the main results without proofs.

### A. Outline of the paper

We describe and discuss our model in Section II. We state our main results in Section III. Section IV presents a naïve dynamic programming algorithm. Section V presents our main contribution: a dynamic cavity method based algorithm for tree graphs, along with a proof of correctness and analysis of running time. We prove our convergence results in Section VI. Section VII discusses our conjecture regarding convergence (Conjecture III.5) and presents numerical results.

## II. MODEL

The model we consider is a simplified version of the model of social learning introduced by Gale and Kariv [11]. We first give a minimal mathematical description of our model, postponing a discussion on knowledge assumptions and rationality. For ease of exposition, we make use of a simple model that captures the essential features of the problem. In Section II-A, we motivate our model in the context of rational agents, state our knowledge assumptions, and explain how some of our simplifications are merely cosmetic.

Consider a graph  $G = (V, E)$ , representing a network of agents, with  $V$  being the set of agents and  $E$  being the social ties between them. An edge  $(i, j)$  indicates that agents  $i$  and  $j$  can observe each other.

Agents attempt to learn the true *state of the world*  $s \in \mathcal{S}$ , where  $\mathcal{S}$  is finite. Each agent  $i$  receives a private signal  $x_i \in \mathcal{X}$ , where  $\mathcal{X}$  is finite. Private signals are independent conditioned on  $s$ , i.e.,

$$\mathbb{P}[s, x_1, \dots, x_n] = \mathbb{P}[s] \prod_{i \in V} \mathbb{P}[x_i | s].$$

In each discrete time period (or round)  $t = 0, 1, \dots$ , each agent  $i \in V$  chooses an action  $\sigma_i(t) \in \mathcal{S}$ , which we call a ‘vote’. Agents observe the votes cast by their neighbors in  $G$ . Thus, at the time of voting in round  $t \geq 1$ , the information available to an agent consists of the private signal she received initially, along with the votes cast by her neighbors in rounds up to  $t - 1$ . In each round, each agent votes for the most likely state of the world that she currently believes is most likely, given the Bayesian posterior distribution she computes.

We denote by  $\partial i$  the neighbors of agent  $i$ , not including  $i$ , i.e.,  $\partial i \equiv \{j : (i, j) \in E\}$ . We use  $\sigma_i^t \equiv (\sigma_i(0), \sigma_i(1), \dots, \sigma_i(t))$  to denote all of agent  $i$ ’s votes, up to

and including time  $t$ . We call  $\sigma_i \equiv (\sigma_i(0), \sigma_i(1), \dots)$  the ‘trajectory’ of votes at node  $i$ . Denote by  $\mathcal{F}_i^t \equiv (x_i, \sigma_{\partial i}^{t-1}, \sigma_i^{t-1})$  the information available to agent  $i$  prior to voting in round  $t$ . Here  $\sigma_{\partial i}^{t-1}$  denotes the votes cast by nodes in  $\partial i$  up to round  $t - 1$ . Note that this does **not** include her neighbors’ votes at time  $t$ .

The vote  $\sigma_i(t)$  is chosen as  $\arg \max_{s \in \mathcal{S}} \mathbb{P}[s | \mathcal{F}_i^t]$ . We assume a deterministic tie-breaking rule. To differentiate the random variable  $\sigma_i(t)$  from the function used to calculate it, we denote the function by  $g_i(t) : \mathcal{X} \times |\mathcal{S}|^{|\partial i|} \rightarrow \mathcal{S}$ , so that

$$\sigma_i(t) = g_{i,t}(x_i, \sigma_{\partial i}^{t-1})$$

For convenience, we also define the vector function  $g_i^t$  that returns the entire history of  $i$ ’s votes up to time  $t$ ,  $g_i^t \equiv (g_{i,0}, g_{i,1}, \dots, g_{i,t})$ , so that

$$\sigma_i^t = g_i^t(x_i, \sigma_{\partial i}^{t-1}).$$

In case of a deterministic tie-breaking rule,  $\sigma_i(t)$  is a deterministic function of  $(x_i, \sigma_{\partial i}^{t-1})$ , so we can take  $\mathcal{F}_i^t = (x_i, \sigma_{\partial i}^{t-1})$ .

### A. Discussion of our Model

The decision rules can be interpreted/motivated as follows. Suppose  $\mathbb{P}[s]$ ,  $\mathbb{P}[x|s]$  and  $G$  are common knowledge. Suppose that, for each state of the world  $s$ , action  $\sigma$  has utility one when the state of the world is  $s = \sigma$ , and zero otherwise. Then, the action that myopically maximizes the expected utility corresponds to the maximum *a posteriori* probability (MAP) estimator of the state of the world. This leads to the decision rule we consider, with  $\sigma_i(t)$  being chosen as  $\arg \max_{s \in \mathcal{S}} \mathbb{P}[s | \mathcal{F}_i^t]$ . We would like to emphasize that we only restrict the ‘action’ space  $\mathcal{A}$  to  $\mathcal{S}$  (thus calling actions as ‘votes’), with this simple “1 if you vote correctly, 0 otherwise” utility function, for simplicity of presentation. Indeed, our main computational result, Theorem III.2 admits a trivial generalization to the case of a general finite action space  $\mathcal{A}$  and a general common utility function  $U : \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ .

A natural objection to such a model of behavior is that the agents should want to maximize the discounted sum of their future utilities, instead of making the myopic optimal choice. Gale and Kariv [11] deal with this by assuming a continuum of agents at each node, so that no one of them can hope to influence the future by their choice of votes. We can do the same here: Then  $\{\sigma_i(t)\}$  and  $\{\mathcal{F}_i^t\}$  form a weak perfect Bayesian equilibrium (cf. [11, Definition 1]) for the right utility function (see above).

The model presented above is a special case of the Gale-Kariv model [11]. Our choice of a ‘state of the world’  $s$  and conditionally independent private signals, with a utility function dependent only on  $s$  and  $a$ , is typical in herd behavior models (e.g., [5], [6], [22]), but is a specialization of the Gale-Kariv model.

**Scaling regime.** We treat the cardinalities of the sets  $\mathcal{S}$  and  $\mathcal{X}$  as fixed<sup>5</sup>, whereas the scaling parameters are the number of agents  $n \equiv |V|$ , and the number of iterations  $t$ . Later,

<sup>5</sup>Most of this work also treats the maximum degree  $d$  of the network as a fixed parameter.

in Section III, we argue that since agents are trying to learn  $s$ , an alternative scaling parameter to  $t$  is  $1/\epsilon$ , where  $\epsilon > 0$  is the desired probability of error. We will be interested in how the computational effort increases as  $n$  grows, and as  $t$  or  $1/\epsilon$  grow. Such a scaling regime is of much interest with the emergence of massive online networks, where non-expert agents interact on a variety of issues, and individual agents are expected to have limited private information, and typically choose from a (relatively) small set of available actions.

### III. MAIN RESULTS

#### A. Efficient computation

To the best of our knowledge, the literature (e.g., [11], [21], [19]) does not contain an explicit description of an algorithm to compute the actions chosen by agents in our model. However, it seems that a dynamic programming algorithm that performs this computation is well known. The proposition below states the computational complexity of this algorithm.

**Proposition III.1.** *On any graph  $G$ , there is a dynamic programming (DP) based algorithm that allows agents to compute their actions up to time  $t$  with computational effort  $t2^{O(\min(n, (d-1)^t))}$ , where  $d$  is the maximum degree of the graph.*

The algorithm leading to Proposition III.1 is described in Section IV. This proposition provides the baseline or benchmark that we compare our other algorithmic results to. In particular, we do not consider this algorithm a major contribution of this work.

A key advantage of the DP algorithm is that it works for any graph  $G$ . The disadvantage, of course, is that the computational effort required grows doubly exponentially in the number of iterations  $t$ .

Our main result concerns the computational effort needed when the graph  $G$  is a tree<sup>6</sup>. We show that computational effort exponentially lower than that of the naive DP suffices in this case.

**Theorem III.2.** *In a tree graph  $G$  with maximum degree  $d$ , each agent can calculate her actions up to time  $t$  with computational effort  $t2^{O(\min(n, td))}$ .*

The algorithm we use employs a technique called the dynamic cavity method [13], previously used only in analytical contexts. A full description of the algorithm and analysis leading to Theorem III.2 is described in Section V.

An apparent issue is that the computational effort required is exponential in  $t$ ; typically, exponentially growing effort is considered as large. However, in this case, we expect the number of iterations  $t$  to be typically quite small, for two reasons: (1) In many settings, agents appear to converge to the ‘right’ answer in a very small number of iterations [11]. In Section III-B below, we argue that if  $\epsilon$  is the desired probability of error, then the number of rounds required should be only  $O(\log \log(1/\epsilon))$ , leading to computational effort of only  $\text{polylog}(1/\epsilon)$ . Having obtained an approximately correct

estimate, the agents would have little incentive to continue observing their neighbors actions and updating their beliefs.<sup>7</sup>

(2) In many situations we would like to model, we might expect only a small number (e.g., single digit) number of iterative updates to occur, irrespective of network size etc. For instance, voters may discuss an upcoming election with each other over a short period of time, ending on the election day when ballots are cast.

#### B. Convergence

Since an agent gains information at each round, and since she is Bayesian, then the probability that she votes correctly is non-decreasing in  $t$ , the number of rounds. We say that the agent *converges* if this probability converges to one, or equivalently if the probability that the agent votes incorrectly converges to zero<sup>8</sup>.

We say that there is *doubly exponential convergence* to the state of the world  $s$  if the maximum single node error probability  $\max_{i \in V} \mathbb{P}[\sigma_i(t) \neq s]$  decays with round number  $t$  as

$$\max_{i \in V} \mathbb{P}[\sigma_i(t) \neq s] = \exp(-\Omega(b^t)), \quad (1)$$

where  $b > 1$  is some constant.

The following is an immediate corollary of Theorem III.2.

**Corollary III.3.** *Consider iterative Bayesian learning on a tree of with maximum degree  $d$ . If we have doubly exponential convergence to  $s$ , then computational effort that is polynomial in  $\log(1/\epsilon)$  (i.e., polylogarithmic in  $1/\epsilon$ ) suffices to achieve error probability  $\mathbb{P}[\sigma_i(t) \neq s] \leq \epsilon$  for all  $i$  in  $V$ .*

Note that if we weaken our assumption to doubly exponential convergence in only a subset  $V_c \subseteq V$  of nodes, i.e.,  $\max_{i \in V_c} \mathbb{P}[\sigma_i(t) \neq s] = \exp(-\Omega(b^t))$ , we still obtain a similar result with nodes in  $V_c$  efficiently learning  $s$ .

**Remark III.4.** *If computational effort grows only polylogarithmically in an approximation parameter (like  $\epsilon$  here), this is typically considered as very efficient. Even  $\text{poly}(1/\epsilon)$  computational effort is considered reasonably efficient, with the corresponding scheme being called a ‘fully polynomial time approximation scheme’.*

We are handicapped by the fact that very little is known rigorously about convergence of iterative Bayesian learning in this sense (cf. questions (I) and (II) in Section I). Nevertheless, we provide the evidence for doubly exponential convergence on trees: We study a situation with two possible states of the world and two possible private signal values. We state a conjecture and show that it implies doubly exponential convergence of iterative Bayesian learning also on undirected trees. We provide numerical evidence in support of our conjecture.

1) *Bayesian vs. ‘majority’ updates:* We conjecture that iterative Bayesian learning leads to lower error probabilities (in the weak sense) than a very simple alternative update rule we call ‘majority dynamics’[13]. Under this rule, the agents

<sup>7</sup>Thus,  $1/\epsilon$  serves as an alternative scaling parameter to  $t$ .

<sup>8</sup>Note that this notion of ‘convergence’ differs greatly from the ‘agreement on actions’ sense in which the term is sometimes used.

<sup>6</sup>A *tree* graph, in this work, refers to a graph that contains no loops. This is sometimes called a ‘forest’ in the literature.

adopt the action taken by the majority of their neighbors in the previous iteration (this is made precise in Definition VI.1). Our conjecture seems natural since the iterative Bayesian update rule chooses the vote in each round that (myopically) minimizes the error probability. We use  $\widehat{\sigma}_i(t)$  to denote votes under the majority dynamics.

**Conjecture III.5.** *Consider binary  $s \sim \text{Bernoulli}(1/2)$ , and binary private signals that are independent identically distributed given  $s$ , with  $\mathbb{P}[x_i = s] = 1 - \delta$  for some  $\delta \in (0, 1/2)$ . Let the majority dynamics (cf. Definition VI.1) be initialized with the private signals, i.e.,  $\widehat{\sigma}_i(0) = x_i$  for all  $i \in V$ . Then on any infinite regular tree, for all  $t \geq 0$ , we have*

$$\mathbb{P}[\sigma_i(t) \neq s] \leq \mathbb{P}[\widehat{\sigma}_i(t) \neq s]. \quad (2)$$

*In words, the error probability under iterative Bayesian learning is no larger than the error probability under majority dynamics, after the same number of iterations.*

In Section VI, we show doubly exponential convergence for majority dynamics on regular trees.

**Theorem III.6.** *Consider binary  $s \sim \text{Bernoulli}(1/2)$ , and binary initial votes  $\widehat{\sigma}_i(0)$  that are independent identically distributed given  $s$ , with  $\mathbb{P}[\widehat{\sigma}_i(0) \neq s] = 1 - \delta$  for some  $\delta \in (0, 1/2)$ . Let  $i$  be any node in an infinite (undirected)  $d$  regular tree for  $d \geq 5$ . Then, under the majority dynamics,*

$$\mathbb{P}[\widehat{\sigma}_i(t) \neq s] = \exp \left[ -\Omega \left( \left( \frac{1}{2}(d-2) \right)^t \right) \right].$$

when  $\delta < (2e(d-1)/(d-2))^{-\frac{d-2}{d-4}}$ .

Thus, if Conjecture III.5 holds:

- We have doubly exponential convergence for iterative Bayesian learning on regular trees with  $d \geq 5$ , implying that for any  $\epsilon > 0$ , an error probability  $\epsilon$  can be achieved in  $O(\log \log(1/\epsilon))$  iterations with iterative Bayesian learning.
- Combining with Corollary III.3), we see that the computational effort that is polylogarithmic in  $(1/\epsilon)$  suffices to achieve error probability  $\epsilon$ .

This compares favorably with the quasi-poly( $1/\epsilon$ ) (i.e.,  $\exp(\text{polylog}(1/\epsilon))$ ) upper bound on computational effort that we can derive by combining Conjecture III.5 and the naïve dynamic program described in Section IV. Indeed, based on recent results on subexponential decay of error probability with the number of private signals being aggregated [14], it would be natural to conjecture that the number of iterations  $T$  needed to obtain an error probability of  $\epsilon$  obeys  $(d-1)^T \geq C \log(1/\epsilon)$  for any  $C < \infty$ , for  $\epsilon$  small enough. This would then imply that the required computational effort using the naïve DP on a regular tree of degree  $d$  grows faster than any polynomial in  $1/\epsilon$ .

Since we are unable to prove our conjecture, we instead provide numerical evidence for it in Table I. Further numerical results are presented in Section VII, along with a discussion of the difficulties in proving Conjecture III.5. All computations leading to our numerical results are exact (modulo finite precision arithmetic), and were performed using the dynamic cavity

Round	Bayesian	Majority
0	0.15	0.15
1	$2.66119 \cdot 10^{-2}$	$2.66119 \cdot 10^{-2}$
2	$7.61832 \cdot 10^{-4}$	$1.67525 \cdot 10^{-3}$
3	$2.83839 \cdot 10^{-7}$	$8.37462 \cdot 10^{-6}$
4	$1.41065 \cdot 10^{-12}$	$2.48525 \cdot 10^{-10}$

TABLE I  
ERROR PROBABILITY ON A REGULAR TREE WITH  $d = 5$  AND  $\mathbb{P}[x_i \neq s] = 0.15$ , FOR (I) BAYESIAN AND (II) MAJORITY UPDATES. THE AGENTS BREAK TIES BY PICKING THEIR ORIGINAL PRIVATE SIGNALS.

equations. The results are all consistent with our conjecture over different values of  $d$  and  $\mathbb{P}[x_i \neq s]$ .

We would like to emphasize that several of the error probability values could be feasibly computed only because of our new efficient approach to computing the decision functions employed by the nodes. For instance, with  $d = 5$ , computing the decision function at iteration 3 using the dynamic program (cf. Proposition III.1 and Section IV) would require enumeration over  $2^{80} \sim 10^{24}$  possibilities, which is infeasible even on state-of-the-art supercomputers. With our approach, we are able to compute the decision function at iteration 3 and even at iteration 4, on a desktop machine. This aggregates information from the  $\sim 400$  nodes within 4 hops of a given node.

Figure 1 plots decay of error probabilities in regular trees for iterative Bayesian learning with  $\mathbb{P}[x_i \neq s] = 0.3$ , where the agents break ties by picking their original private signals. Each of the curves (for different values of  $d$ ) in the plot of  $\log(-\log \mathbb{P}[\sigma_i(t) \neq s])$  vs.  $t$  appear to be bounded below by straight lines with positive slope, suggesting doubly exponential decay of error probabilities with  $t$ .

The empirical rapidity of convergence, particularly for  $d = 5, 7$ , is noteworthy.

#### IV. A SIMPLE ALGORITHM: PROOF OF PROPOSITION III.1

A sign of the complexity of evaluating the Bayesian decision function  $g_i^t(x_i, \sigma_{\partial_i}^{t-1})$ , is that even the brute-force solution approach to it is not trivial. We therefore describe it here.

One way of thinking of the agents' calculation is to imagine that they keep a long list of all the possible combinations of private signals of all the other agents, and at each iteration cross out entries that are inconsistent with the signals that they've observed from their neighbors up to that point. Then, they calculate the probabilities of the different possible states of the world by summing over the entries that have yet to be crossed out.

This may not be as simple as it seems. To understand which private signal vectors are ruled out by the observed actions of neighbors, an agent "simulates" the network for every possible private signal vector: Each agent calculates the function  $g_i^t$  for every other agent  $i$  and every possible set of observations by  $i$ . We formalize this below.

Let  $\underline{x} \in \mathcal{X}^n$  be the vector of private signals  $(x_i)_{i \in V}$ . The trajectory of  $i$ , denoted by  $\sigma_i$ , is a deterministic function of  $\underline{x}$ . Assume then that up to time  $t - 1$  each agent has calculated

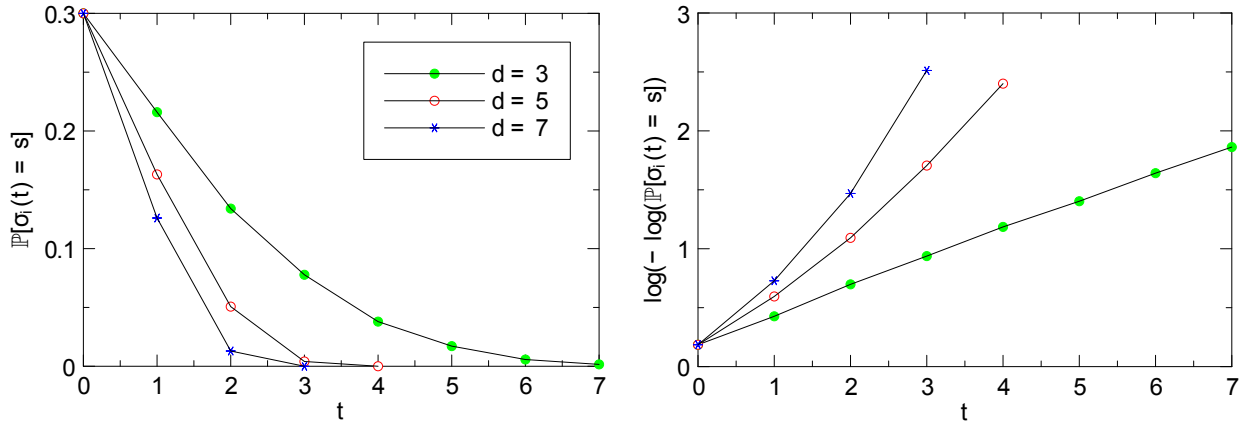


Fig. 1. Error probability decay on regular trees for iterative Bayesian learning, with  $\mathbb{P}[x_i \neq s] = 0.3$  (cf. Section VII). The data used to generate this figure is presented in Table III.

the trajectory  $\sigma_i^{t-1}(\underline{x})$  for all possible private signal vectors  $\underline{x}$  and all agents  $i$ . This is trivial for  $t-1=0$ .

We say that  $\underline{y} \in \mathcal{X}^n$  is feasible for  $i$  at time  $t$  if  $x_i = y_i$  and  $\sigma_{\partial i}^t = \sigma_{\partial i}^t(\underline{y})$ . We denote this set of feasible private signal vectors by  $I_i^t(x_i, \sigma_{\partial i}^t) \subseteq \mathcal{X}^n$ . To calculate  $\sigma_i^t(\underline{x})$ , one observe that for all  $i$ ,  $x_i$  and  $\sigma_{\partial i}^{t-1}$ , we have

$$\begin{aligned} \mathbb{P}[s | \mathcal{F}_i^t] &\propto \mathbb{P}[s] \mathbb{P}[x_i, \sigma_{\partial i}^{t-1} | s] \\ &= \mathbb{P}[s] \sum_{\underline{y} \in I_i^{t-1}(x_i, \sigma_{\partial i}^{t-1})} \mathbb{P}[\underline{x} = \underline{y} | s] \end{aligned}$$

and

$$g_{i,t}(x_i, \sigma_{\partial i}^{t-1}) = \arg \max_{s \in \mathcal{S}} \mathbb{P}[s | \mathcal{F}_i^t]$$

by definition. We use the standard abusive notation  $\mathbb{P}[x_i]$  instead of  $\mathbb{P}[x_i = y_i]$ ,  $\mathbb{P}[\sigma_j^t]$  instead of  $\mathbb{P}[\sigma_j^t = \omega_j^t]$ , etc.

It is easy to verify that using the equations above, the ‘simulation’ can be advanced from  $t-1$  to  $t$  with additional computational effort  $O(n|\mathcal{X}|^n)$ . Thus, the calculation of  $\sigma_i^t(\underline{x})$  for all  $i$  and  $\underline{x}$  requires total effort  $O(tn|\mathcal{X}|^n)$ . This leads to an upper bound of  $t2^{O(n)}$  for this method. Note that up to time  $t$  an agent need only consider agents at distance at most  $t$ , so on a graph with maximum degree  $d$ , we obtain a bound of  $t2^{O((d-1)^t)}$ . This improves the bound above for ‘large’ graphs, i.e., graphs for which  $n > (d-1)^t$  for relevant values of  $t$ . Thus, we obtain the result stated in Proposition III.1.

We call this algorithm ‘the naive dynamic program’.

## V. THE DYNAMIC CAVITY ALGORITHM ON TREES

In this section we develop the dynamic cavity algorithm leading to Theorem III.2. We present the core construction and key technical lemmas in Section V-A. In Section V-B, we show how this leads to an efficient algorithm for the Bayesian computations on tree graphs, and prove Theorem III.2.

Assume in this section that the graph  $G$  is a tree with finite degree nodes. For  $j \in \partial i$ , let  $G_{j \rightarrow i} = (V_{j \rightarrow i}, E_{j \rightarrow i})$  denote the connected component containing node  $j$  in the graph  $G$  with the edge  $(i, j)$  removed. That is,  $G_{j \rightarrow i}$  is  $j$ ’s subtree when  $G$  is rooted at  $i$ .

### A. The Dynamic Cavity Method

We consider a modified process where agent  $i$  is replaced by an *inert agent* who takes a fixed sequence of actions  $\tau_i = (\tau_i(0), \tau_i(1), \dots)$ , and *the true state of the world is assumed to be some fixed  $s$* . Furthermore, this ‘fixing’ goes unnoticed by the agents (except  $i$ , who is inert anyway) who perform their calculations assuming that  $i$  is her regular Bayesian self, and that  $s$  was drawn randomly according to  $\mathbb{P}[s]$ . We denote by  $\mathbb{Q}[A | \tau_i, s]$  the probability of event  $A$  in this modified process.

**Remark V.1.** We emphasize that the modified process with an ‘inert’ agent is a theoretical construct we use to derive an efficient implementation for the iterative Bayesian decision rules. Our algorithm does not involve actual replacement of nodes in the network.

This modified process is easier to analyze, as the processes on each of the subtrees  $V_{j \rightarrow i}$  for  $j \in \partial i$  are independent: Recall that private signals are independent conditioned on  $s$ , and the inert agent ensures that the subtrees stay independent of each other. This is formalized in the following claim, which is immediate to see:

**Claim V.2.** For any  $i \in V$ ,  $s \in \mathcal{S}$  and any trajectory  $\tau_i$ , we have

$$\mathbb{Q}[\sigma_{\partial i}^t | \tau_i, s] = \prod_{j \in \partial i} \mathbb{Q}[\sigma_j^t | \tau_i^t, s]. \quad (3)$$

(Since  $\sigma_j^t$  is unaffected by  $\tau_i(t')$  for all  $t' > t$ , we only need to specify  $\tau_i^t$ , and not the entire  $\tau_i$ .)

Now, it might so happen that for some number of steps the ‘inert’ agent behaves exactly as may be expected of a rational player. More precisely, given  $\sigma_{\partial i}^{t-1}$ , it may be the case that  $\tau_i^t = g_i^t(x_i, \sigma_{\partial i}^{t-1})$ . This event provides the connection between the modified process and the original process, and is the inspiration for the following theorem.

**Proposition V.3.** Consider any  $i \in V$ ,  $s \in \mathcal{S}$ ,  $t \in \mathbb{N}$ , trajectory  $\tau_i$  and  $\sigma_{\partial i}^{t-1}$ . For any  $x_i$  such that  $\mathbb{P}[x_i | s] > 0$ , we have

$$\begin{aligned} \mathbb{P}[\sigma_{\partial i}^{t-1} | s, x_i] \mathbf{1}(\tau_i^t = g_i^t(x_i, \sigma_{\partial i}^{t-1})) \\ = \mathbb{Q}[\sigma_{\partial i}^{t-1} | \tau_i, s] \mathbf{1}(\tau_i^t = g_i^t(x_i, \sigma_{\partial i}^{t-1})). \end{aligned} \quad (4)$$

*Proof:* We couple the original process, after choosing  $s$ , to the modified processes by setting the private signals to be identical in both.

Now, clearly if it so happens that  $\tau_i^t = g_i^t(x_i, \sigma_{\partial i}^{t-1})$  then the two processes will be identical up to time  $t$ . Hence the probabilities of events measurable up to time  $t$  will be identical when multiplied by  $\mathbf{1}(\tau_i^t = g_i^t(x_i, \sigma_{\partial i}^{t-1}))$ , and the theorem follows. ■

Using Eqs. (3) and (4), we can easily write the posterior on  $s$  computed by node  $i$  at time  $t$ , in terms of the probabilities  $\mathbb{Q}[\cdot|\cdot]$ :

$$\begin{aligned} \mathbb{P}[s|\mathcal{F}_i^t] &\propto \mathbb{P}[s] \mathbb{P}[x_i, \sigma_{\partial i}^{t-1}|s] \\ &= \mathbb{P}[s] \mathbb{P}[x_i|s] \mathbb{P}[\sigma_{\partial i}^{t-1}|s, x_i] \\ &= \mathbb{P}[s] \mathbb{P}[x_i|s] \prod_{j \in \partial i} \mathbb{Q}[\sigma_j^{t-1} | \sigma_i^{t-1}, s] \end{aligned} \quad (5)$$

(Recall that  $\sigma_i^{t-1}$  is a deterministic function of  $(x_i, \sigma_{\partial i}^{t-1})$ . Also, note that if  $\mathbb{P}[x_i|s] = 0$ , we simply obtain  $\mathbb{P}[s|\mathcal{F}_i^t] = 0$ . Eq. (5) deals with the non-trivial case  $\mathbb{P}[x_i|s] > 0$ .)

**Remark V.4.** A naive (and incorrect) method to estimate the posterior  $\mathbb{P}[s|\mathcal{F}_i^t]$  would be to treat the trajectories of the neighbors and  $x_i$  as being independent conditioned on  $s$ , leading to the estimate  $\tilde{\mathbb{P}}[s|\mathcal{F}_i^t] \propto \mathbb{P}[s] \mathbb{P}[x_i|s] \prod_{j \in \partial i} \mathbb{P}[\sigma_j^{t-1}|s]$  for posterior beliefs<sup>9</sup>. Eq. (5) gives us a variation on this estimate that is exact on trees. In other words, it provides the right way to ‘combine’ information from neighbors to compute the Bayesian posterior on  $s$ .

The decision function, defined as before, then follows from the posterior:

$$g_{i,t}(x_i, \sigma_{\partial i}^{t-1}) = \arg \max_{s \in \mathcal{S}} \mathbb{P}[s|\mathcal{F}_i^t]. \quad (6)$$

As mentioned earlier, we assume there is a deterministic tie breaking rule.

We are left with the task of calculating  $\mathbb{Q}[\cdot|\cdot]$ . The following theorem is the heart of the dynamic cavity method and allows us to perform this calculation:

**Proposition V.5.** For any  $i \in V$ ,  $j \in \partial i$ ,  $s \in \mathcal{S}$ ,  $t \in \mathbb{N}$ ,  $\tau_i^t$  and  $\sigma_j^t$ , we have

$$\begin{aligned} \mathbb{Q}[\sigma_j^t | \tau_i^t, s] &= \sum_{\sigma_1^{t-1} \dots \sigma_{d-1}^{t-1}} \sum_{x_j} \mathbb{P}[x_j|s] \mathbf{1} \left[ \sigma_j^t = g_j^t \left( x_j, (\tau_i^{t-1}, \sigma_{\partial j}^{t-1}) \right) \right] \\ &\quad \cdot \prod_{l=1}^{d-1} \mathbb{Q}[\sigma_l^{t-1} | \sigma_j^{t-1}, s], \end{aligned} \quad (7)$$

where the neighbors of node  $j$  are  $\partial j = \{i, 1, 2, \dots, d-1\}$ .

We mention without proof that the recursion easily generalizes to the case of a random tie-breaking rule; it is a matter of replacing the expression  $\mathbf{1}[\sigma_j^t = \dots]$  with  $\mathbb{P}[\sigma_j^t = \dots]$ , where this probability is over the randomness of the rule. Eq. (5) continues to be valid in this case.

<sup>9</sup>Thus, the logarithm of this estimated belief is a linear combination of information from neighbors. This has motivated some of the heuristic updates rules studied in the literature [8], [9].

The following proof is similar to the proof of Lemma 2.1 in [13], where the dynamic cavity method is introduced and applied to a different process.

*Proof:* In the modified process, the events in the different branches that  $i$  sees are independent. We therefore consider  $G_{j \rightarrow i}$  only, and view it as a tree rooted at  $j$ . Also, for convenience we define  $\sigma_i^t \equiv \tau_i^t$ ; note that the random variable  $\sigma_i^t$  does not exist in the modified process, as  $i$ 's trajectory is fixed to  $\tau_i$ .

Let  $\underline{x}$  be the vector of private signals of  $j$  and all the vertices up to a depth  $t$  in  $G_{j \rightarrow i}$  (call this set of vertices  $V_{j \rightarrow i}^t$ ). For each  $l \in \{1, \dots, d-1\}$ , let  $\underline{x}_l$  be the vector of private signals of  $V_{l \rightarrow j}^{t-1}$ . Thus,  $\underline{x} = (x_j, \underline{x}_1, \underline{x}_2, \dots, \underline{x}_{d-1})$ .

The trajectory  $\sigma_j^t$  is a function -deterministic, by our assumption- of  $\underline{x}$  and  $\tau_i^t$ . We shall denote this function by  $F_{j \rightarrow i}$  and write  $\sigma_j^t = F_{j \rightarrow i}^t(\underline{x}, \tau_i^t)$ . This function is uniquely determined by the update rules  $g_l^t(x_l, \sigma_{\partial l}^{t-1})$  for  $l \in V_{j \rightarrow i}$ .

We have therefore

$$\mathbb{Q}[\sigma_j^t = \lambda^t | \tau_i^t, s] = \sum_{\underline{x}} \mathbb{P}[\underline{x}|s] \mathbf{1}(\lambda^t = F_{j \rightarrow i}^t(\underline{x}, \tau_i^t)). \quad (8)$$

We now analyze each of the terms appearing in this sum. Since the private signals are independent conditioned on  $s$ , we have

$$\mathbb{P}[\underline{x}|s] = \mathbb{P}[x_j|s] \mathbb{P}[\underline{x}_1|s] \mathbb{P}[\underline{x}_2|s] \dots \mathbb{P}[\underline{x}_{d-1}|s]. \quad (9)$$

The function  $F_{j \rightarrow i}^t(\dots)$  can be decomposed as follows:

$$\begin{aligned} \mathbf{1}(\lambda^t = F_{j \rightarrow i}^t(\underline{x}, \tau_i^t)) &= \sum_{\sigma_1^{t-1} \dots \sigma_{d-1}^{t-1}} \mathbf{1}(\lambda^t = g_j^t(x_j, \sigma_{\partial j}^{t-1})) \\ &\quad \cdot \prod_{l=1}^{d-1} \mathbf{1}(\sigma_l^{t-1} = F_{l \rightarrow j}^{t-1}(\underline{x}_l, \lambda^{t-1})). \end{aligned} \quad (10)$$

Using Eqs. (9) and (10) in Eq. (8) and separating terms that depend only on  $\underline{x}_i$ , we get

$$\begin{aligned} \mathbb{Q}[\sigma_j^t = \lambda^t | \tau_i^t, s] &= \sum_{\sigma_1^{t-1} \dots \sigma_{d-1}^{t-1}} \sum_{x_j} \mathbb{P}[x_j|s] \mathbf{1}(\lambda^t = g_j^t(x_j, \sigma_{\partial j}^{t-1})) \\ &\quad \cdot \prod_{l=1}^{d-1} \sum_{\underline{x}_l} \mathbb{P}[\underline{x}_l|s] \mathbf{1}(\sigma_l^{t-1} = F_{l \rightarrow j}^{t-1}(\underline{x}_l, \lambda^{t-1})). \end{aligned}$$

The recursion follows immediately by identifying that the product over  $l$  in fact has argument  $\mathbb{Q}[\sigma_l^{t-1} | \sigma_j^{t-1}, s]$ . ■

## B. The Agents' Calculations

We now have in place all we need to perform the agents' calculations. At time  $t = 0$  these calculations are trivial. Assume then that up to time  $t$  each agent has calculated the following quantities:

- 1)  $\mathbb{Q}[\sigma_j^{t-1} | \tau_i^{t-1}, s]$ , for all  $s \in \mathcal{S}$ , for all  $i, j \in V$  such that  $j \in \partial i$ , and for all  $\tau_i^{t-1}$  and  $\sigma_j^{t-1}$ .
- 2)  $g_i^t(x_i, \sigma_{\partial i}^{t-1})$  for all  $i, x_i$  and  $\sigma_{\partial i}^{t-1}$ .

Note that these can be calculated without making any observations – only knowledge of the graph  $G$ ,  $\mathbb{P}[s]$  and  $\mathbb{P}[\underline{x}|s]$  is needed.

At time  $t + 1$  each agent makes the following calculations:

- 1)  $\mathbb{Q}[\sigma_j^t | \tau_i^t, s]$  for all  $s, i, j, \sigma_j^t, \tau_i^t$ . These can be calculated using Eq. (7), given the quantities from the previous iteration.
- 2)  $g_i^{t+1}(x_i, \sigma_{\partial i}^t)$  for all  $i, x_i$  and  $\sigma_{\partial i}^t$ . These can be calculated using Eqs. (5) and (6) and the newly calculated  $\mathbb{Q}[\sigma_j^t | \tau_i^t, s]$ .

Since agent  $j$  calculates  $g_i^{t+1}$  for all  $i$ , then she, in particular, calculates  $g_j^{t+1}$ . This allows her to choose the (myopic) Bayes optimal action in rounds up to  $t + 1$ , based on her neighbors' past actions. A simple calculation yields the following lemma.

**Lemma V.6.** *In a tree graph  $G$  with maximum degree  $d$ , the agents can calculate their actions up to time  $t$  with computational effort  $n2^{O(td)}$ .*

In fact, each agent does not need to perform calculations for the entire graph. It suffices for node  $i$  to calculate quantities up to time  $t'$  for nodes at distance  $t - t'$  from node  $i$  (there are at most  $(d - 1)^{t-t'}$  such nodes). A short calculation yields an improved bound on computational effort, stated in Theorem III.2.

*Proof of Theorem III.2:*

Consider an agent  $j$ , who wants to determine her own decision function up to round  $t$ , i.e., she wants to determine  $g_j^t(\cdot, \cdot)$ . The computation is performed in  $t$  steps, that we number  $0, 1, \dots, t - 1$ . Step 0 involves the following: (i) Evaluate  $g_i^0(x_i) = \arg \max_{s \in \mathcal{S}} \mathbb{P}[s | x_i]$  for all  $i$  at a distance at most  $t$  from  $j$ . (ii) Evaluate  $\mathbb{Q}[\sigma_i^0 | \tau_k^0, s]$  for all  $k$  at distance at most  $t - 1$  from  $j$ , for all  $i \in \partial k$ , and for all  $\sigma_i^0, \tau_k^0, s$ , using Eq. (7).

For any  $1 \leq t' \leq t - 1$ , step  $t - t'$  proceeds as follows. Consider any agent  $i$  at distance at most  $t' \geq 1$  from  $j$ . Suppose that we have already computed  $\mathbb{Q}[\sigma_l^{t-t'-1} | \tau_i^{t-t'-1}, s]$  for all such  $i$ , for all  $l \in \partial i$ , and for all possible  $\sigma_l^{t-t'-1}, \tau_i^{t-t'-1}, s$ . Then we can use Eqs. (5) and (6) to compute  $g_i^{t-t'}(x_i, \sigma_{\partial i}^{t-t'-1})$  for all possible  $x_i, \sigma_{\partial i}^{t-t'-1}$ . Using these values, for any  $k$  at a distance  $t' - 1$  from  $j$ , we can compute  $\mathbb{Q}[\sigma_i^{t-t'} | \tau_k^{t-t'}, s]$  for all  $i \in \partial k$ , for all  $\sigma_i^{t-t'}, \tau_k^{t-t'}, s$ , using Eq. (7). The computational effort involved is bounded by  $C(d - 1)^{t'} |\mathcal{S}|^{d(t-t'+1)} |\mathcal{X}|$  for the computation of  $g_i^{t-t'}(\cdot, \cdot)$ 's and bounded by  $C(d - 1)^{t'} |\mathcal{S}|^{(d+1)(t-t'+1)} |\mathcal{X}|$  for the computation of  $\mathbb{Q}[\sigma_i^{t-t'} | \tau_k^{t-t'}, s]$ 's. Here  $d$  is maximum degree, and  $C = C(d) < \infty$  is a constant. Thus, step  $t - t'$  requires effort bounded by  $2^{C'td}$  for some  $C' = C'(d, |\mathcal{S}|, |\mathcal{X}|) < \infty$ . This bound also holds for step 0. Thus, the overall computational effort is bounded by  $t2^{C'td} = 2^{O(td)}$ . ■

## VI. CONVERGENCE OF MAJORITY DYNAMICS: PROOF OF THEOREM III.6

In this section we study a very simple update rule, 'majority dynamics'. We use  $\hat{\sigma}_i(t) \in \{-1, +1\}$  to denote votes under the majority dynamics.

**Definition VI.1.** *Under the majority dynamics, each agent  $i \in V$  chooses her vote in round  $t + 1$  according to the majority*

*of the votes of her neighbors in round  $t$ , i.e.*

$$\hat{\sigma}_i(t + 1) = \text{sign} \left( \sum_{j \in \partial i} \hat{\sigma}_j(t) \right)$$

*Ties are broken by flipping an unbiased coin.*

Let  $s \in \{-1, +1\}$  be drawn from a uniform prior and nodes receive 'private signals'  $\hat{\sigma}_i(0)$  that are correct with probability  $1 - \delta$ , and independent conditioned on  $s$ . We consider an undirected  $d$  regular tree. The analysis is complicated by dependencies which have to be carefully handled. Our analytical approach here is again closely related to the dynamic cavity method.

**Lemma VI.2.** *Consider the setting in Theorem III.6. Let  $i$  and  $j$  be adjacent nodes in the tree. Then for all  $(\hat{\sigma}_i^{t-1}, \hat{\sigma}_j^{t-1}) \in \{-1, +1\}^{2t}$*

$$\mathbb{P}[\hat{\sigma}_i(t) = -1 | \hat{\sigma}_i^{t-1}, \hat{\sigma}_j^{t-1}, s = +1] \leq \delta_t \quad (11)$$

where  $\delta_t$  is defined recursively by  $\delta_0 \equiv \delta$ , and

$$\delta_t \equiv \mathbb{P}[\text{Binomial}(d - 1, \delta_{t-1}) \geq d/2 - 1] \quad (12)$$

*Proof:* We proceed by induction. Clearly Eq. (11) holds for  $t = 0$ . Suppose Eq. (11) holds for some  $t$ . We want to show

$$\mathbb{P}[\hat{\sigma}_i(t + 1) = -1 | \hat{\sigma}_i^t, \hat{\sigma}_j^t, s = +1] \leq \delta_{t+1}, \quad (13)$$

for all  $(\hat{\sigma}_i^t, \hat{\sigma}_j^t) \in \{-1, +1\}^{2(t+1)}$ .

Let  $l_1, l_2, \dots, l_{d-1}$  be the other neighbors of node  $i$  (besides  $j$ ). We will show that, in fact,

$$\mathbb{P}[\hat{\sigma}_i(t + 1) = -1 | \hat{\sigma}_i^t, \hat{\sigma}_j^t, \hat{\sigma}_{l_1}^{t-1}, \dots, \hat{\sigma}_{l_{d-1}}^{t-1}, s = +1] \leq \delta_{t+1}, \quad (14)$$

for all possible  $\xi \equiv (\hat{\sigma}_i^t, \hat{\sigma}_j^t, \hat{\sigma}_{l_1}^{t-1}, \hat{\sigma}_{l_2}^{t-1}, \dots, \hat{\sigma}_{l_{d-1}}^{t-1})$ .

We reason as follows. Fix the state of the world  $s$  and the trajectories  $\hat{\sigma}_i^t$  and  $\hat{\sigma}_j^t$ . Now this induces correlations between the trajectories of the neighbors  $l_1, \dots, l_{d-1}$ , caused by the requirement of consistency with the majority rule at node  $i$ , but *only up to time  $t - 1$* . If we further fix  $\hat{\sigma}_{l_m}^{t-1}$ , then  $\hat{\sigma}_{l_m}(t)$  (and  $\hat{\sigma}_{l_m}$  at all future times) is conditionally independent of  $(\hat{\sigma}_{l_{m'}}^t)_{m' \neq m}$ . Thus, we have<sup>10</sup>

$$\begin{aligned} \mathbb{P}[\hat{\sigma}_{l_m}(t) = -1 | \xi, s = +1] \\ = \mathbb{P}[\hat{\sigma}_{l_m}(t) = -1 | \hat{\sigma}_{l_m}^{t-1}, \hat{\sigma}_i^{t-1}, s = +1], \end{aligned}$$

and therefore, using the induction hypothesis

$$\mathbb{P}[\hat{\sigma}_{l_m}(t) = -1 | \xi, s = +1] \leq \delta_t \quad (15)$$

for all  $m \in \{1, 2, \dots, d - 1\}$ . Also, the actions  $\hat{\sigma}_{l_1}(t), \dots, \hat{\sigma}_{l_{d-1}}(t)$  are conditionally independent of each other given  $\xi, s = +1$ . We have

$$\hat{\sigma}_i(t + 1) = \text{sgn}(\hat{\sigma}_j(t) + \hat{\sigma}_{l_1}(t) + \dots + \hat{\sigma}_{l_{d-1}}(t)),$$

<sup>10</sup>A alternate argument can be constructed using the modified process with an inert agent, mirroring the reasoning used in the proof of Proposition V.5.



with  $\text{sgn}(0)$  being assigned value  $-1$  or  $+1$  with equal probability. This yields

$$\mathbb{P}[\hat{\sigma}_i(t+1) = -1 \mid \xi, s = +1] \leq \mathbb{P}[\text{Binomial}(d-1, \delta_t) \geq d/2 - 1]$$

from Eq. (15) and conditional independence of  $\hat{\sigma}_{l_1}(t), \dots, \hat{\sigma}_{l_{d-1}}(t)$ . Thus, we obtain Eq. (14). Eq. (13) follows by summing over  $\hat{\sigma}_{l_1}^{t-1}, \hat{\sigma}_{l_2}^{t-1}, \dots, \hat{\sigma}_{l_{d-1}}^{t-1}$ . ■

*Proof of Theorem III.6:* By applying the multiplicative version of the Chernoff bound<sup>11</sup> to Eq. (12) we have that

$$\delta_{t+1} \leq e^{(d-2)/2 - (d-1)\delta_t} (2\delta_t(d-1)/(d-2))^{(d-2)/2}$$

Dropping the term  $e^{-(d-1)\delta_t}$ , we obtain

$$\delta_{t+1} \leq (2e\delta_t(d-1)/(d-2))^{\frac{1}{2}(d-2)}. \quad (16)$$

This is a first order non-homogeneous linear recursion in  $\log \delta_t$ . If it were an equality it would yield

$$\begin{aligned} \log \delta_t &= \left( \log \delta + \frac{d-2}{d-4} \log[2e(d-1)/(d-2)] \right) \left[ \frac{1}{2}(d-2) \right]^t \\ &\quad - \frac{d-2}{d-4} \log[2e(d-1)/(d-2)], \end{aligned}$$

and so

$$-\log \delta_t \in \Omega \left( \left( \frac{1}{2}(d-2) \right)^t \right), \quad (17)$$

as long as

$$-\log \delta < \frac{d-2}{d-4} \log[2e(d-1)/(d-2)].$$

Theorem III.6 is non-trivial for  $d \geq 5$ . The upper limit of the ‘noise’  $\delta$  for which it establishes rapid convergence approaches  $(2e)^{-1}$  as  $d$  grows large (see also the discussion below for large  $d$ ). ■

## VII. FURTHER NUMERICAL RESULTS AND DISCUSSION ON CONJECTURE III.5

Table II, together with Table I above, contrast the error probabilities of Bayesian updates with those of majority updates. All cases exhibit lower error probabilities (in the weak sense) for the Bayesian update, consistent with Conjecture III.5. Table III contains the data plotted in Figure 1. Also for these parameters, we found that the Bayesian updates showed lower error probabilities than the majority updates (compare with Table IV).

The running time to generate these tables was less than a minute on a standard desktop machine. We did not proceed with more rounds because of numerical instability issues which begin to appear as error probabilities decrease.

We now discuss briefly the difficulties in proving Conjecture III.5. Order the possible private signals by the implied likelihood ratio of  $s$ , with higher  $x_j$  corresponding to  $s = +1$  being more likely. We say a learning rule with successive rounds of ‘voting’ is *monotonic* if the following occurs: If some  $\underline{x}$  leads to  $\sigma_i(t) = 1$ , then increasing  $x_j$  in  $\underline{x}$  for some  $j \in V$

<sup>11</sup> $\mathbb{P}[X \geq (1+\eta)\mathbb{E}[X]] \leq \left( \frac{\exp \eta}{(1+\eta)^{1+\eta}} \right)^{\mathbb{E}[X]}$ . We substitute  $\mathbb{E}[X] = \delta_t(d-1)$  and  $1+\eta = (d/2 - 1)/[\delta_t(d-1)]$ .

Round	Bayesian	Majority
0	0.15	0.15
1	$6.075 \cdot 10^{-2}$	$6.075 \cdot 10^{-2}$
2	$1.57158 \cdot 10^{-2}$	$2.95136 \cdot 10^{-2}$
3	$2.99170 \cdot 10^{-3}$	$1.59849 \cdot 10^{-2}$
4	$3.39853 \cdot 10^{-4}$	$9.15458 \cdot 10^{-3}$
5	$2.72958 \cdot 10^{-5}$	$5.46501 \cdot 10^{-3}$
6	$2.21981 \cdot 10^{-6}$	$3.35117 \cdot 10^{-3}$

TABLE II  
 $d = 3, \mathbb{P}[x_i \neq s] = 0.15$

Round	$d = 3$	$d = 5$	$d = 7$
0	0.30	0.30	0.30
1	0.216	0.16308	0.126036
2	0.134038	$5.07053 \cdot 10^{-2}$	$1.1966 \cdot 10^{-2}$
3	$7.77755 \cdot 10^{-2}$	$4.06495 \cdot 10^{-3}$	$3.67884 \cdot 10^{-6}$
4	$3.79502 \cdot 10^{-2}$	$1.61786 \cdot 10^{-5}$	
5	$1.71209 \cdot 10^{-2}$		
6	$5.73294 \cdot 10^{-3}$		
7	$1.59587 \cdot 10^{-3}$		

TABLE III  
ERROR PROBABILITIES FOR BAYESIAN AGENTS WITH  
 $\mathbb{P}[x_i \neq s] = 0.3$ , FOR REGULAR TREES OF DIFFERENT DEGREES  $d$ .  
THIS DATA IS DISPLAYED IN FIGURE 1.

leaves  $\sigma_i(t)$  unchanged. One might expect most reasonable learning rules, including iterative Bayesian learning, to satisfy *monotonicity*. For instance, there is a simple proof that the majority rule is monotonic [13]. However, it turns out that iterative Bayesian learning is not always monotonic<sup>12</sup>! It is not very surprising, then, that it is hard to prove convergence of Bayesian learning to the ‘right’ answer, even in simple settings. Controlling the rate of convergence, as in Conjecture III.5, is even harder.

Despite non-monotonicity, it is tempting to hope for a direct proof of Conjecture III.5, by showing inductively (in time) that iterative Bayesian learning is always at least as good as majority dynamics. The difficulty that arises here is that though iterative Bayesian learning minimizes the error probability at a node, given the available information, this is not the case if we condition on the state of the world. After conditioning on the state of the world, iterative Bayesian learning does better than majority dynamics on some nodes, and worse on others. It is very hard to control the difference between the two processes beyond a small number of iterations, making a direct proof of Conjecture III.5 difficult.

## VIII. DISCUSSION

We presented a new algorithmic approach that questions the belief that fully Bayesian computations for agents interacting on a social network are computationally intractable. The chief drawback is that our approach does not seem amenable to graphs with short loops, though many real networks possess this feature. A significant open question suggested by our results is: What is the ‘computational boundary’ between

<sup>12</sup>Elchanan Mossel and Omer Tamuz, private communication.

Round	$d = 3$	$d = 5$	$d = 7$
0	0.30	0.30	0.30
1	0.216	0.16308	0.126036
2	0.170489	0.0733673	0.0232861
3	0.146010	0.0215952	$2.99165 \cdot 10^{-4}$
4	0.130070	$2.61093 \cdot 10^{-3}$	
5	0.119647		
6	0.112267		
7	0.107006		

TABLE IV  
ERROR PROBABILITIES FOR AGENTS USING MAJORITY UPDATES  
WITH  $\mathbb{P}[x_i \neq s] = 0.3$ , FOR REGULAR TREES OF DIFFERENT  
DEGREES  $d$ .

networks where exact Bayesian calculations can be efficiently performed, and networks where this is not possible? In particular, can graphs with a few short loops be handled at some additional computational cost?

**Acknowledgments.** We would like to thank Andrea Montanari, Elchanan Mossel and Allan Sly for valuable discussions.

#### REFERENCES

- [1] D. Acemoglu, M. A. Dahleh, I. Lobel, and A. Ozdaglar. Bayesian learning in social networks. *The Review of Economic Studies*, 78(4), 2011.
- [2] R. Aumann. Agreeing to disagree. *The Annals of Statistics*, 4(6):1236–1239, 1976.
- [3] V. Bala and S. Goyal. Learning from neighbours. *Review of Economic Studies*, 65(3):595–621, July 1998.
- [4] O. Bandiera and I. Rasul. Social networks and technology adoption in northern mozambique\*. *The Economic Journal*, 116(514):869–902, 2006.
- [5] A. V. Banerjee. A simple model of herd behavior. *The Quarterly Journal of Economics*, 107(3):797–817, 1992.
- [6] S. Bikhchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy*, 100(5):992–1026, 1992.
- [7] T. Conley and C. Udry. Social learning through networks: The adoption of new agricultural technologies in ghana. *American Journal of Agricultural Economics*, 83(3), 2001.
- [8] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.
- [9] P. M. DeMarzo, D. Vayanos, and J. Zwiebel. Persuasion bias, social influence, and unidimensional opinions. *Quarterly Journal of Economics*, 118(3):909 – 968, 2003.
- [10] G. Ellison and D. Fudenberg. Rules of thumb for social learning. *Journal of Political Economy*, 110(1):93–126, 1995.
- [11] D. Gale and S. Kariv. Bayesian learning in social networks. *Games and Economic Behavior*, 45(2):329–346, 2003.
- [12] S. Goyal. *Connections: An Introduction to the Economics of Networks*. Princeton University Press, 2007.
- [13] Y. Kanoria and A. Montanari. Majority dynamics on trees and the dynamic cavity method. *Annals of Applied Probability*, 21(5):1694–1748, 2011.
- [14] Y. Kanoria and A. Montanari. Subexponential convergence for information aggregation on regular trees. In *Proceedings of the IEEE Conference on Decision and Control and European Control Conference*, pages 5317–5322, December 2011. Copyright ©2011, IEEE. Reprinted with permission from IEEE and authors.
- [15] Y. Kanoria and O. Tamuz. Tractable bayesian social learning on trees. In *Proceedings of the IEEE International Symposium on Information Theory*, pages 2721–2725, July 2012. Copyright ©2012, IEEE. Reprinted with permission from IEEE and authors.
- [16] H. Kohler. Learning in social networks and contraceptive choice. *Demography*, 34(3):369–383, 1997.
- [17] L. Ménager. Consensus, communication and knowledge: an extension with bayesian agents. *Mathematical Social Sciences*, 51(3):274–279, 2006.
- [18] M. Mueller-Frank. A Comment on ‘Bayesian Learning in Social Networks’. *SSRN eLibrary*, 2011.
- [19] M. Mueller-Frank. Mixed Communication Structures: Opinion Formation in Social Networks with Bayesian and Non-Bayesian Agents. *SSRN eLibrary*, 2011.
- [20] M. Raynal. Fault-tolerant agreement in synchronous message-passing systems. *Synthesis Lectures on Distributed Computing Theory*, 1(1):1–189, 2010.
- [21] D. Rosenberg, E. Solan, and N. Vieille. Informational externalities and emergence of consensus. *Games and Economic Behavior*, 66(2):979 – 994, 2009.
- [22] L. Smith and P. Sørensen. Pathological outcomes of observational learning. *Econometrica*, 68(2):371–398, 2000.
- [23] W. P. Tay, J. Tsitsiklis, and M. Win. Bayesian detection in bounded height tree networks. *IEEE Transactions on Signal Processing*, 57(10), oct. 2009.
- [24] J. Tsitsiklis. Decentralized detection. In H. V. Poor and J. B. Thomas, editors, *Advances in Signal Processing*, volume 2, pages 1–72. JAI Press, 297-344.
- [25] J. Tsitsiklis and M. Athans. Convergence and asymptotic agreement in distributed decision problems. In *Proceedings of the 21st IEEE Conference on Decision and Control*, volume 21, pages 692 –701, dec. 1982.