Tractable Bayesian Social Learning on Trees

Yashodhan Kanoria* and Omer Tamuz†

April 16, 2012

Abstract

We study a model of Bayesian agents in social networks who learn from the actions of their neighbors. Most results concerning social learning in networks have been achieved either in 'herd behavior' models, where each agent acts only once, or in models where agents are not Bayesian and use rules of thumb, or are boundedly rational. Models of Bayesian agents who act repeatedly have posed two related problems: (1) they have proved notoriously difficult to analyze; and (2) the calculations required of interacting Bayesian agents often seem intractable.

We consider a set of Bayesian agents who are attempting to iteratively estimate an unknown 'state of the world' s from their initial private signals, and the past actions of their neighbors in a social network. When private signals are independent conditioned on s, and when the social network graph is a tree, we provide an algorithm for the agents' calculations with running time that is exponentially lower than what is currently known.

We use this algorithm to perform the first numerical simulations of interacting Bayesian agents on networks with hundreds of nodes, and observe rapid learning of s in some settings.

Keywords: social learning, Bayesian agents, computational efficiency, convergence, algorithm, dynamic cavity method.

1 Introduction

The importance of social learning in networks has been demonstrated in a wide variety of settings (e.g., adoption of agricultural technology in Ghana [9] and Mozambique [6], choice of contraceptives by European women [19]). Accordingly, understanding mathematical models of social learning by Bayesian agents has been a goal of theoretical economics for the past few decades (cf., Goyal [15]). 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

*Microsoft Research New England and Department of Electrical Engineering, Stanford University. Supported by 3Com Corporation Stanford Graduate Fellowship.

†Microsoft Research New England and Weizmann Institute. Supported by ISF grant 1300/08. Omer Tamuz is a recipient of the Google Europe Fellowship in Social Computing, and this research is supported in part by this Google Fellowship.

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., [25]) 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 [7], Bikhchandani, Hirshleifer and Welch [8], Smith and Sørenson [27] and Acemoglu et al [2]. 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 [13] consider Bayesian agents on a network who repeatedly choose actions. They show, in the spirit of Aumann's Agreement Theorem [4], that agents on a network converge to the same action under some conditions[1]. Related work by Rosenberg, Solan and Vieille [25] and Ménager [20] sheds more light on the phenomenon of agreement on actions and the conditions in which it arises.

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?[2]

(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[3], e.g., those of Ellison and Fudenberg [12], Bala and Goyal [5], and DeGroot [10]. Such modelling approaches appear to be driven by two primary motivating factors (see, e.g., [12], [5]): (i) Real agents may not be Bayesian. (ii) The desire to *"keep the model mathematically tractable"* [5], and also computationally tractable[4]; 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 [13]. 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[5] is exponential in the number of individuals. 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

[1] A gap in the proof of Gale and Kariv's agreement theorem was recently pointed out [21]. However, recent works [25, 22] establish similar results in more general settings.

[2] In fact, convergence is *not* always obtained [25]. The exact conditions under which convergence occurs are not known.

[3] 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.

[4] Mathematical and computational tractability often go together, e.g., see [16].

[5] 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 4.

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 [13], "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 of Gale and Kariv [13], who simulated the process for three agents, to much larger networks[6]. 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[7]. 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 (see Section 5.3). 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.

Besides computational issues, another difficulty of the Gale and Kariv model is that it requires the social graph to be common knowledge. A possible alternative to this is a modified model that allows the agents to know only their own neighborhoods and the distribution from which the rest of the graph was picked. We pursue this for a standard model of random graphs and show that our computational approach extends to this case (see Section 5.3.2). We also consider that nodes may not all be 'active' in each round, and that nodes may observe only a random subset of active neighbors. We show that these features can be handled when 'inactive' edges/nodes occur independently of each other and in time.

A key technique used in this paper is the dynamic cavity method, introduced by Kanoria and

[6]In each of our numerical analyses, agents receive information (directly or indirectly) from hundreds of distinct nodes.

[7]This result could 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[24].

Montanari [17] 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 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.

Our algorithm may be of interest beyond its computational implications. The form of the Bayesian posterior belief that we obtain for trees (see Section 5), appears closely related 'weighted average' type heuristic update rules [10, 11] that have been studied before. This suggests an interesting direction for further study: What if agents on a loopy graph depart from rationality only in assuming that their neighborhood is treelike? The heuristic update rule they employ is then identical to the one we obtain. Thus, our work may suggest new 'almost rational' heuristic update models for study on general loopy graphs. This is discussed in more detail in Section 8.2.

1.1 Outline of the paper

We describe and discuss our model in Section 2. We state our main results in Section 3. Section 4 presents a naïve dynamic programming algorithm. Section 5 presents our main contribution: a dynamic cavity method based algorithm for tree graphs, along with a proof of correctness and analysis of running time. Section 5.3 extends our algorithm in various directions. We prove our convergence results in Section 6. Section 7 discusses our conjecture regarding convergence (Conjecture 3.6) and presents numerical results. We conclude with a discussion in Section 8.

2 Model

The model we consider is a simplified version of the model of social learning introduced by Gale and Kariv [13]. 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 2.1, we motivate our model in the context of rational agents, state our knowledge assumptions, and explain how some of our simplifications are merely cosmetic. The scaling regime we consider is described in Section 2.2. Finally, Section 2.3 compares our model with other models, including that of Gale and Kariv [13].

Consider a directed graph $G = (V, E)$, representing a network of agents, with V being the set of agents and E being the social ties between them. A directed edge (i, j) indicates that agent i observes agent j. (In most of this paper, we study the special case of undirected graphs, where relationships between agents are bidirectional.)

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.,

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

In each discrete time period (or round) $t = 0, 1, \ldots$, the agents choose must choose an action

$\sigma_i(t) \in \mathcal{S}$, which we call a 'vote'[8]. 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 ∂i the neighbors of agent i, not including i, i.e., $\partial i \equiv \{j : (i,j) \in E\}$. We use $\sigma_i^t \equiv (\sigma_i(0), \sigma_i(1), \ldots, \sigma_i(t))$ to denote all of agent i's votes, up to and including time t. We call $\sigma_i \equiv (\sigma_i(0), \sigma_i(1), \ldots)$ 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 ∂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 $\arg\max_{s \in \mathcal{S}} \mathbb{P}\left[s \middle| \mathcal{F}_i^t\right]$. 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 $g_i(t) : \mathcal{X} \times |\mathcal{S}|^{t|\partial i|} \to \mathcal{S}$, 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, $g_i^t \equiv (g_{i,0}, g_{i,1}, \ldots, g_{i,t})$, 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 $\mathcal{F}_i^t = (x_i, \sigma_{\partial i}^{t-1})$.

2.1 Discussion of our Model

The decision rules can be interpreted/motivated as follows. Suppose $\mathbb{P}[s]$, $\mathbb{P}[\underline{x}|s]$ and G are common knowledge. Suppose that, for each state of the world s, action σ 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 $\arg\max_{s \in \mathcal{S}} \mathbb{P}\left[s \middle| \mathcal{F}_i^t\right]$. 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 3.2, and its extensions in Section 5.3, admit a trivial generalization to the case of a general finite action space $\mathcal{A}$ and a general common utility function $U : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. Section 5 includes a precise description of why this is the case.

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 [13] 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. [13, Definition 1]) for the right utility function (see above).

Rosenberg, Solan and Vieille[25] consider a model with fully rational agents -one per node- maximizing the discounted sum of payoffs. In this setting strategic behavior arises, and so they study the properties of the model's Nash equilibria. They extend to this model many agreement

[8] We choose to use "vote" rather than the more usual "act" since, in this simplified model, the set of actions and the set of states of the world are identical, so choosing an action is equivalent to picking a possible state of the world as a guess for the true s.

results (e.g., those of Gale and Kariv [13], Parikh and Krasucki [23] and Sebenius and Geanakoplos [26]) that were previously known to apply to myopic agents. Our results apply only to the myopic model, which can be thought of as a stepping stone towards the strategic model, being a special case corresponding to discount factor 0.

2.2 Scaling regime

A major objective of this work is to examine whether the computations required of agents in this model of Bayesian social learning can be *efficiently* performed. In defining a problem of efficient computability, it is important to distinguish between parameters that are 'fixed' and parameters that 'scale'; also termed the *scaling regime*. The goal, then, is to obtain a reasonably slow growth in computational effort needed as the scaling parameters become larger, while treating the fixed parameters as constants. We treat the cardinalities of the sets $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{X}$ as fixed[9], whereas the scaling parameters are the number of agents $n \equiv |V|$, and the number of iterations t. Later, in Section 3, 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.

Our choice of scaling variables is in contrast, for instance, to the work of Aaronson on the complexity of agreement [1]. Aaronson focuses on the case of two agents, and allows the set of possible private signals to grow, aiming to reach agreement with minimum communication and computational cost. In our case, the objective must clearly be the computational cost, since the 'protocol' is defined by the model itself, leading to a fixed communication cost.

2.3 Comparison with other models

The model presented above is a special case of the Gale-Kariv model [13], which we refer to as the GK model henceforth.

In the GK model there is an all-encompassing 'state of nature' $\omega \in \Omega$, the agents have a set of possible actions $\mathcal{A}$, and the utility of the actions is a general function $U : \mathcal{A} \times \Omega \to \mathbb{R}$. The utility functions of all agents are the same. The agents receive general private signals $(x_1(\omega), \ldots, x_n(\omega))$.

We specialize the GK model as follows:

- We reduce ω to $(x_1, \ldots, x_n, s)$, where $x_i \in \mathcal{X}$ is agent i's private signal and s belongs to a given (finite) set $\mathcal{S}$. We restrict the utility function to depend only on s and a, i.e., $U : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. In particular, U does not depend directly on the x_i's.
- We demand that the x_i's be conditionally independent of each other, given s.
- We demand that $\mathcal{S}$, $\mathcal{X}$ and $\mathcal{A}$ be *finite* sets with bounded cardinalities.

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., Banerjee [7] Bikhchandani, Hirshleifer and Welch [8], Smith and Sørensen [27]). It is also the basis of the model of boundedly-rational agents on social networks studied by Bala and Goyal [5]. Nevertheless it is important to note that our first and second assumptions represents an important specialization of the GK model. The third assumption corresponds to our choice of scaling regime (cf. Section 2.2).

[9]Most of this work also treats the maximum degree d of the network as a fixed parameter.

Our assumptions play a crucial role in the efficient approach we develop to enable the computation of Bayesian posteriors. As discussed above, we allow the number of agents n to scale. Hence, one might expect a general utility function that depends on all private signals to cause a computational burden that grows exponentially in n, just to enumerate the different utility functions possible. Our assumption 1 above eliminates this difficulty. Similarly, dependent private signals might lead to the problem of summing over exponentially many different possibilities.

While we know of no formal computational hardness results for Bayesian calculations on social networks, we conjecture that the removal of any of the first two assumptions, or the consideration of general graphs (i.e., not tree or tree-like graphs), makes the agents' calculations #P hard. A proof of this conjecture would be a natural complement to this work.

3 Main results

We make use of the following notations:

- For positive valued functions f_1, f_2, we write $f_1(z) = O(f_2(z))$ or $f_1(z) \in O(f_2(z))$ as $z \to \infty$, if there exist $C < \infty$ and $z_0 < \infty$ such that $f_1(z) \leq C f_2(z)$ for all $z > z_0$.
- For positive valued functions f_1, f_2, we write $f_1(z) = \Omega(f_2(z))$ or $f_1(z) \in \Omega(f_2(z))$ as $z \to \infty$, if there exist $C > 0$ and $z_0 < \infty$ such that $f_1(z) \geq C f_2(z)$ for all $z > z_0$.

The qualifier "as $z \to \infty$" is often omitted for brevity. In this work, z corresponds to the scaling variables n, t or $1/\epsilon$ (or combinations of these, e.g., in $O(\min(n, td))$ the scaling parameter is $\min(n, td)$). The constant C may depend on fixed variables like $|\mathcal{S}|$, $|\mathcal{X}|$, and the maximum degree d. We adopt the convention that *C should not depend on the network G*. We remark that $O(\cdot)$ and $\Omega(\cdot)$ are the only asymptotic notations that we use. For instance, ω is *not* used as an asymptotic notation.

3.1 Efficient computation

To the best of our knowledge, the literature (e.g., [13, 25, 22]) 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 3.1. *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 3.1 is described in Section 4. 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[10]. We show that computational effort exponentially lower than that of the naive DP suffices in this case.

[10] A *tree* graph, in this work, refers to a graph that contains no loops. This is sometimes called a 'forest' in the literature.

Theorem 3.2. *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 [17], previously used only in analytical contexts. A full description of the algorithm and analysis leading to Theorem 3.2 is described in Section 5.

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 [13]. In Section 3.2 below, we argue that if ϵ 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.[11] (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.

3.2 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[12].

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

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

where $b>1$ is some constant.

The following is an immediate corollary of Theorem 3.2.

Corollary 3.3. *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 $\mathbb{P}\left[\sigma_i(t)\neq s\right]\leq\epsilon$ for all i in V.*

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

Remark 3.4. *If computational effort grows only polylogarithmically in an approximation parameter (like ϵ 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 in known rigorously about convergence of iterative Bayesian learning in this sense (cf. questions (I) and (II) in Section 1). Nevertheless, we

[11] Thus, $1/\epsilon$ serves as an alternative scaling parameter to t.

[12] Note that this notion of 'convergence' differs greatly from the 'agreement on actions' sense in which the term is sometimes used.

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. First, on a regular *directed* tree we show that except for the case of very noisy signals, we have doubly exponential convergence if the degree is at least five. Second, 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.

3.2.1 Directed trees

We consider an infinite directed d-ary tree. By this we mean a tree graph where each node i has one 'parent' who observes i and d 'children' whom i observes, but who do not observe i. Learning in such a tree is much easier to analyze (than in an undirected tree) because the trajectories of the d children are uncorrelated conditioned on s.

Proposition 3.5. *Consider a directed d-ary tree, binary $s \sim$ Bernoulli$(1/2)$, and binary private signals that are independent identically distributed given s, with $\mathbb{P}[x_i \neq s] = 1 - \delta$ for some $\delta \in (0, 1/2)$. For any symmetric tie breaking rule (e.g., "follow your private signal"), we have*

$$\mathbb{P}[\sigma_i(t) \neq s] = \exp\Big[-\Omega\left((d/2)^t\right)\Big]. \tag{2}$$

Proposition 3.5 is proved in Section 6.

3.2.2 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'[17]. Under this rule, the agents adopt the action taken by the majority of their neighbors in the previous iteration (this is made precise in Definition 6.2). 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 3.6. *Consider binary $s \sim$ Bernoulli$(1/2)$, and binary private signals that are independent identically distributed given s, with $\mathbb{P}[x_i \neq s] = 1 - \delta$ for some $\delta \in (0, 1/2)$. Let the majority dynamics (cf. Definition 6.2) 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*

$$\mathbb{P}[\sigma_i(t) \neq s] \leq \mathbb{P}[\widehat{\sigma}_i(t) \neq s]\ . \tag{3}$$

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 6, we show doubly exponential convergence for majority dynamics on regular trees:

Theorem 3.7. *Consider binary $s \sim$ 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 (undirected) d regular tree for $d \geq 5$. Then, under the majority dynamics,*

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

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

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 1: 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.

Thus, if Conjecture 3.6 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 ϵ can be achieved in $O(\log\log(1/\epsilon))$ iterations with iterative Bayesian learning.
- Combining with Corollary 3.3), we see that the computational effort that is polylogarithmic in $(1/\epsilon)$ suffices to achieve error probability $1/\epsilon$.

This compares favorably with the quasi-poly$(1/\epsilon)$ (i.e., $\exp\big(\text{polylog}(1/\epsilon)\big)$) upper bound on computational effort that we can derive by combining Conjecture 3.6 and the naïve dynamic program described in Section 4. Indeed, based on recent results on subexponential decay of error probability with the number of private signals being aggregated [18], it would be natural to conjecture that the number of iterations T needed to obtain an error probability of ϵ obeys $(d-1)^T \geq C\log(1/\epsilon)$ for any $C < \infty$, for ϵ 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 1. Further numerical results are presented in Section 7, along with a discussion of the difficulties in proving Conjecture 3.6. All computations leading to our numerical results are exact (modulo finite precision arithmetic), and were performed using the dynamic cavity 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 3.1 and Section 4) 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 ~ 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.

4 A Simple Algorithm: Proof of Proposition 3.1

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.

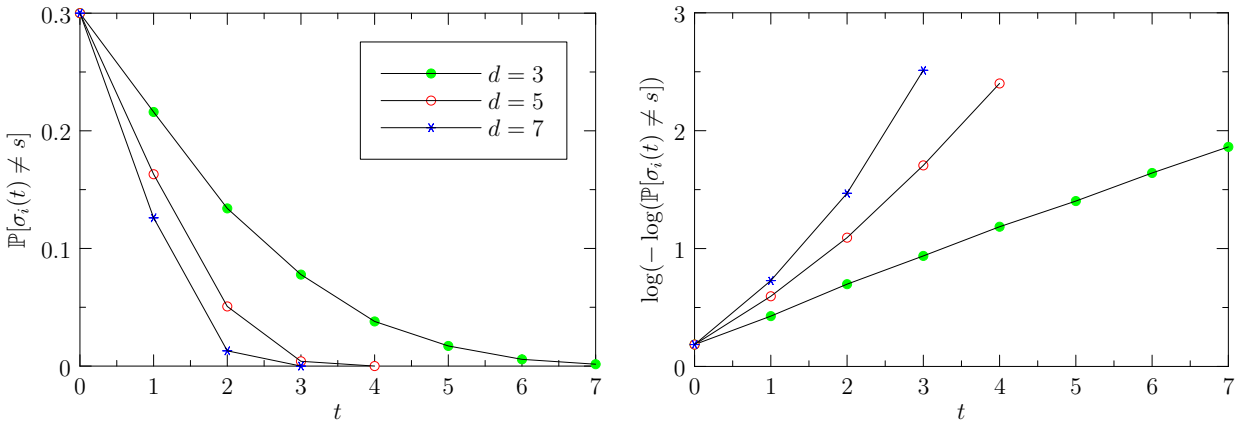

Figure 1: Error probability decay on regular trees for iterative Bayesian learning, with $\mathbb{P}[x_i \neq s] = 0.3$ (cf. Section 7). The data used to generate this figure is presented in Table 3.

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 σ_i, is a deterministic function of $\underline{x}$. Assume then that up to time $t-1$ each agent has calculated 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

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

and

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

by definition. We use the standard abusive notation $\mathbb{P}[x_i]$ instead of $\mathbb{P}[x_i = y_i]$, $\mathbb{P}\left[\sigma_j^t\right]$ instead of $\mathbb{P}\left[\sigma_j^t = \omega_j^t\right]$, 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 3.1.

We call this algorithm 'the naïve dynamic program'.

5 The Dynamic Cavity Algorithm on Trees

In this section we develop the dynamic cavity algorithm leading to Theorem 3.2. We present the core construction and key technical lemmas in Section 5.1. In Section 5.2, we show how this leads to an efficient algorithm for the Bayesian computations on tree graphs, and prove Theorem 3.2. We extend our results in various directions in Section 5.3.

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

5.1 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 $\tau_i = (\tau_i(0), \tau_i(1), \ldots)$, 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 5.1. *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 $V_{j\to i}$ 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 5.2. *For any $i \in V$, $s \in \mathcal{S}$ and any trajectory τ_i, we have*

$$\mathbb{Q}\left[\sigma_{\partial i}^t \middle|\middle| \tau_i, s\right] = \prod_{j\in\partial i} \mathbb{Q}\left[\sigma_j^t \middle|\middle| \tau_i^t, s\right]. \tag{4}$$

(Since σ_j^t is unaffected by $\tau_i(t')$ for all $t' > t$, we only need to specify τ_i^t, and not the entire τ_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 $\tau_i^t = g_i^t\left(x_i, \sigma_{\partial i}^{t-1}\right)$. This event provides the connection between the modified process and the original process, and is the inspiration for the following theorem.

Theorem 5.3. *Consider any $i \in V$, $s \in \mathcal{S}$, $t \in \mathbb{N}$, trajectory τ_i and $\sigma_{\partial i}^{t-1}$. For any x_i such that $\mathbb{P}[x_i|s] > 0$, we have*

$$\mathbb{P}\left[\sigma_{\partial i}^{t-1}\middle|s, x_i\right] \mathbf{1}\left(\tau_i^t = g_i^t\left(x_i, \sigma_{\partial i}^{t-1}\right)\right) = \mathbb{Q}\left[\sigma_{\partial i}^{t-1}\middle|\middle|\tau_i, s\right] \mathbf{1}\left(\tau_i^t = g_i^t\left(x_i, \sigma_{\partial i}^{t-1}\right)\right). \tag{5}$$

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 $\tau_i^t = g_i^t\left(x_i, \sigma_{\partial i}^{t-1}\right)$ 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 $\mathbf{1}\left(\tau_i^t = g_i^t\left(x_i, \sigma_{\partial i}^{t-1}\right)\right)$, and the theorem follows. □

Using Eqs. (4) and (5), we can easily write the posterior on s computed by node i at time t, in terms of the probabilities $\mathbb{Q}\left[\cdot||\cdot\right]$:

$$\begin{aligned}\mathbb{P}\left[s|\mathcal{F}_i^t\right] &\propto \mathbb{P}\left[s\right]\mathbb{P}\left[x_i, \sigma_{\partial i}^{t-1}|s\right] \\ &= \mathbb{P}\left[s\right]\mathbb{P}\left[x_i|s\right]\mathbb{P}\left[\sigma_{\partial i}^{t-1}|s, x_i\right] \\ &= \mathbb{P}\left[s\right]\mathbb{P}\left[x_i|s\right]\prod_{j\in\partial i}\mathbb{Q}\left[\sigma_j^{t-1}\middle|\middle|\sigma_i^{t-1}, s\right] \end{aligned} \tag{6}$$

(Recall that σ_i^{t-1} is a deterministic function of $(x_i, \sigma_{\partial i}^{t-1})$. Also, note that if $\mathbb{P}\left[x_i|s\right] = 0$, we simply obtain $\mathbb{P}\left[s|\mathcal{F}_i^t\right] = 0$. Eq. (6) deals with the non-trivial case $\mathbb{P}\left[x_i|s\right] > 0$.)

Remark 5.4. *A naïve (and incorrect) method to estimate the posterior $\mathbb{P}\left[s|\mathcal{F}_i^t\right]$ would be to treat the trajectories of the neighbors and x_i as being independent conditioned on s, leading to the estimate $\tilde{\mathbb{P}}\left[s\middle|\mathcal{F}_i^t\right] \propto \mathbb{P}\left[s\right]\mathbb{P}\left[x_i|s\right]\prod_{j\in\partial i}\mathbb{P}\left[\sigma_j^{t-1}\middle|s\right]$ for posterior beliefs[13]. Eq. (6) 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:

$$g_{i,t}(x_i, \sigma_{\partial i}^{t-1}) = \arg\max_{s\in\mathcal{S}}\mathbb{P}\left[s|\mathcal{F}_i^t\right]\ . \tag{7}$$

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

Remark 5.5. *Suppose, instead, that the action set $\mathcal{A}$ is distinct from $\mathcal{S}$, and the agents have some common utility function $U : \mathcal{A}\times\mathcal{S}\to\mathbb{R}$. Eq. (7) changes to*

$$g_{i,t}(x_i, \sigma_{\partial i}^{t-1}) = \arg\max_{a\in\mathcal{A}}\sum_{s\in\mathcal{S}}\mathbb{P}\left[s|\mathcal{F}_i^t\right]U(a, s)\,, \tag{8}$$

and all results in this section remain unchanged.

The naïve DP in Section 4 admits a similar trivial extension.

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

Theorem 5.6. *For any $i\in V$, $j\in\partial i$, $s\in\mathcal{S}$, $t\in\mathbb{N}$, τ_i^t and σ_j^t, we have*

$$\begin{aligned}\mathbb{Q}\left[\sigma_j^t\middle|\middle|\tau_i^t, s\right] = \sum_{\sigma_1^{t-1}\ldots\sigma_{d-1}^{t-1}}\sum_{x_j}\mathbb{P}\left[x_j|s\right]\mathbf{1}\left[\sigma_j^t = g_j^t\left(x_j, (\tau_i^{t-1}, \sigma_{\partial j\setminus i}^{t-1})\right)\right]\cdot \\ \cdot\prod_{l=1}^{d-1}\mathbb{Q}\left[\sigma_l^{t-1}\middle|\middle|\sigma_j^{t-1}, s\right]\ ,\end{aligned} \tag{9}$$

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

[13]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 [10, 11].

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 $\mathbf{1}\left[\sigma_j^t = \cdots\right]$ with $\mathbb{P}\left[\sigma_j^t = \cdots\right]$, where this probability is over the randomness of the rule. Eq. (6) continues to be valid in this case.

The following proof is similar to the proof of Lemma 2.1 in [17], 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 $G_{j\to i}$ 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 σ_i^t does not exist in the modified process, as i's trajectory is fixed to τ_i.

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

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

We have therefore

$$\mathbb{Q}\left[\sigma_j^t = \lambda^t \middle| \middle| \tau_i^t, s\right] = \sum_{\underline{x}} \mathbb{P}\left[\underline{x}|s\right] \mathbf{1}\left(\lambda^t = F_{j\to i}^t(\underline{x}, \tau_i^t)\right). \tag{10}$$

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

$$\mathbb{P}\left[\underline{x}\,|s\right] = \mathbb{P}\left[x_j|s\right] \mathbb{P}\left[\underline{x}_1|s\right] \mathbb{P}\left[\underline{x}_2|s\right] \ldots \mathbb{P}\left[\underline{x}_{d-1}\middle|s\right]. \tag{11}$$

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

$$\mathbf{1}\left(\lambda^t = F_{j\to i}^t(\underline{x}, \tau_i^t)\right) = \sum_{\sigma_1^{t-1}\ldots\sigma_{d-1}^{t-1}} \mathbf{1}\Big(\lambda^t = g_j^t(x_j, \sigma_{\partial j}^{t-1})\Big) \prod_{l=1}^{d-1} \mathbf{1}\Big(\sigma_l^{t-1} = F_{l\to j}^{t-1}(\underline{x}_l, \lambda^{t-1})\Big). \tag{12}$$

Using Eqs. (11) and (12) in Eq. (10) and separating terms that depend only on $\underline{x}_i$, we get

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

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

5.2 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. $\mathbb{Q}\left[\sigma_j^{t-1} \middle| \middle| \tau_i^{t-1}, s\right]$, for all $s \in \mathcal{S}$, for all $i, j \in V$ such that $j \in \partial i$, and for all τ_i^{t-1} and σ_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. $\mathbb{Q}\left[\sigma_j^t \middle|\middle| \tau_i^t, s\right]$ for all $s, i, j, \sigma_j^t, \tau_i^t$. These can be calculated using Eq. (9), 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. (6) and (7) and the the newly calculated $\mathbb{Q}\left[\sigma_j^t \middle|\middle| \tau_i^t, s\right]$.

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 5.7. *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 3.2.

Proof of Theorem 3.2. 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, ..., $t-1$. 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 $\mathbb{Q}\left[\sigma_i^0 \middle|\middle| \tau_k^0, s\right]$ for all k at distance at most $t-1$ from j, for all $i \in \partial k$, and for all σ_i^0, τ_k^0, s, using Eq. (9).

For any $1 \le t' \le t-1$, step $t-t'$ proceeds as follows. Consider any agent i at distance at most $t' \ge 1$ from j. Suppose that we have already computed $\mathbb{Q}\left[\sigma_l^{t-t'-1} \middle|\middle| \tau_i^{t-t'-1}, s\right]$ 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. (6) and (7) 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 $\mathbb{Q}\left[\sigma_i^{t-t'} \middle|\middle| \tau_k^{t-t'}, s\right]$ for all $i \in \partial k$, for all $\sigma_i^{t-t'}$,$\tau_k^{t-t'}$, s, using Eq. (9). 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 $\mathbb{Q}\left[\sigma_i^{t-t'} \middle|\middle| \tau_k^{t-t'}, s\right]$'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 $t2^{C'td} = 2^{O(td)}$.

□

5.3 Dynamic Cavity Algorithm: Extensions

Our algorithm admits several extensions that we explore in this section: Section 5.3.1 discusses random graphs, Section 5.3.2 relaxes the assumption that the entire graph is common knowledge and Section 5.3.3 allows nodes/edges to be inactive in some rounds.

But first we mention some straightforward generalizations:

It is easy to see that dynamic cavity recursion (Theorem 5.6) does not depend on any special properties of the Bayesian update rule. The decision rule $g_{i,t} : \mathcal{X} \times \mathcal{S}^{|\partial i|t} \to \mathcal{S}$ can be arbitrary.

Thus, if agent i wants to perform a Bayesian update, he can do so (exactly) using our approach even if his neighbor, agent j, is using some other update rule[14].

Remark 5.8. *The dynamic cavity recursion can be used to enable computations of agents even if some of them are using arbitrary update rules (provided the rules are common knowledge).*

Our algorithm is easily modified for the case of a general finite action set $\mathcal{A}$ that need not be the same as $\mathcal{S}$, associated with a payoff function $U : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, as described in Remark 5.5. In fact, the action set and payoff function can each be player dependent ($\mathcal{A}_i$, U_i respectively: Eq. (8) admits a trivial generalization), provided these are common knowledge.

We already mentioned that there is a simple generalization to the case of random tie breaking rules.

Instead of having only undirected edges (corresponding to bidirectional observations), we can allow a subset of the edges of the tree to be directed. In this case, the same algorithm works with suitably defined neighborhood sets ∂i. In other words, our result holds for the class of directed graphs lacking cycles of length greater than two (length two cycles are simply undirected edges).

Agents may receive private signals in rounds later than round 0. This can be incorporated into our computational approach provided that conditioned on s, the private signals are independent for different agents and across time. Let $x_{i,t}$ be the private signal received by agent i just before round t. Then in Eq. (6), $\mathbb{P}[x_i|s]$ is replaced by $\prod_{t'=0}^{t} \mathbb{P}\left[x_{i,t'}\middle|s\right]$, and there is an analogous replacement for $\mathbb{P}[x_j|s]$ in Eq. (9).

The computations of agent i up to round t only depend on a 'ball' of radius t around node i, i.e., the neighborhood of node i up to depth t. For our algorithm to work at node i up to round t, we only need the ball of radius t around node i to be a tree.

It should also be possible to use our dynamic cavity approach to enable efficient Bayesian computations in a *social experimentation* setting (see, e.g., [12, 5]), where (noisy) payoffs are observed in each round by agents and their neighbors. However, we do not pursue this extension here, since it would require us to introduce a new model.

5.3.1 Random graphs

Consider a random graph on n nodes drawn from the configuration model[15] with a given degree distribution. It is well known that such graphs are locally tree-like with high probability(see, e.g. [3]). More formally, for any $t < \infty$, we have

$$\lim_{n\to\infty} \mathbb{P}\left[B_i^t \text{ is a tree.}\right] = 1\,. \tag{13}$$

Since node calculations up to time t depend only on B_i^t, it follows that with high probability (w.h.p.), for an arbitrarily selected node, the tree calculations suffice for any constant number of iterations.

5.3.2 Learning without Knowledge of the Graph

Here we consider the situation where nodes do not know the actual graph G, but know some distribution over possibilities for G. This is potentially a more realistic model; the assumption the

[14]Such settings have been proposed, for instance, in [22], where the network consists of a mixture of Bayesian and non-Bayesian agents.

[15]In the configuration model, one first assigns a degree to each node, draws the appropriate number of 'half-edges' and then chooses a uniformly random pairing between them. One can further specify that a graph constructed thus is 'rejected' if it contains double edges or self-loops; this does not change any of the basic properties, e.g., the local description, of the ensemble.

graph structure is common knowledge may be considered a weakness of the model of Gale and Kariv. We address this issue here, showing that our algorithm can be modified to allow Bayesian estimation in this case as well.

Let $G \equiv G_n$ be a random graph of n nodes constructed according to the configuration model for a given (node perspective degree) distribution. Denote the degree distribution by ρ_V, so that $\rho_V(d) \equiv$ probability that a randomly selected node has degree d.

Now, in this ensemble, the local neighborhood up to distance D of an arbitrary node v with fixed degree d_v converges in distribution as $n \to \infty$ to the following ('local weak convergence', see [3]): Each of the neighbors of node v has a degree drawn independently according to the 'edge perspective' degree distribution ρ_E, defined by:

$$\rho_E(d) = \frac{d\rho_V(d)}{\sum_{d' \in \mathbb{N}} d' \rho_V(d')}$$

Further, each of the neighbors of the neighbors (except v itself) again have a degree drawn independently according to $\rho_E(d)$, and so on up to depth D. Call the resulting distribution over trees $\mathcal{T}_{d_v}^D$.

Now suppose that agents are, in fact, connected in a graph drawn from the ensemble G_n with degree distribution ρ_V, independent of the state of the world s and the private signals $\{x_i\}$. Suppose that each node u knows the distribution ρ_V and its own degree d_u, but does not know anything else about G_n.[16] Further, suppose that this is common knowledge. Now in the limit $n \to \infty$, an exact Bayesian calculation for a node v up to time t depends on ρ_V via $\mathcal{T}_{d_v}^t$. Since nodes know only their own degree, there are only Δ different 'types' of nodes, where Δ is the size of the support of $\rho_E(d)$. There is one type for each degree. This actually makes computations slightly simpler than in an arbitrary known graph.

Fix state s. Take an arbitrary agent i. Replace it with an 'inert' agent following the vote trajectory τ_i. Now fix some ∂i (ensure $\rho_V(|\partial i|) > 0$). Choose arbitrary $j \in \partial i$. Define $\mathbb{Q}\left[\sigma_j^t = \omega_j^t \middle| \middle| \tau_i^t, s\right]$ as the probability of seeing trajectory $\sigma_j^t = \omega_j^t$ at node j in this setting. This probability is over the graph realization (given ∂i) and over the private signals. Note here that $\mathbb{Q}\left[\sigma_j^t = \omega_j^t \middle| \middle| \tau_i^t, s\right]$ *is the same for any* i, ∂i *and* $j \in \partial i$.

Eqs. (4), (6) and (7) continue to hold w.h.p.[17] for the same reasons as before.

The dynamic cavity recursion, earlier given by Eq. (9), becomes

$$\mathbb{Q}\left[\sigma_j^t \middle| \middle| \tau_i, s\right] = \sum_{d \in \mathbb{N}} \rho_E(d) \sum_{\sigma_1^{t-1} \ldots \sigma_{d-1}^{t-1}} \sum_{x_j} \mathbb{P}\left[x_j | s\right] \mathbf{1}\left[\sigma_j^t = g_j^t\left(x_j, (\tau_i^{t-1}, \sigma_{\partial j \backslash i}^{t-1})\right)\right] \cdot \cdot \prod_{l=1}^{d-1} \mathbb{Q}\left[\sigma_l^{t-1} \middle| \middle| \sigma_j^{t-1}, s\right] . \quad (14)$$

in the limit $n \to \infty$, due to local weak convergence (see above). We have written the recursion assuming the neighbors of j are named according to $\partial j \backslash i = \{1, 2, \ldots, d-1\}$.

We comment that there is a straightforward generalization to the case of a multi-type configuration model with a finite number of types. Nodes may or may not be aware of the type of each of their neighbors (both cases can be handled). For instance, here is a simple example with two types: There are 'red' agents and 'blue' agents, and each 'red' agent is connected to 3 'blue' agents,

[16] Other 'knowledge' assumptions can be similarly handled, for instance where a node knows its own degree, the degree of its neighbors and ρ_V.

[17] We need the ball of radius t around i to be a tree.

whereas each 'blue' agent is connected to either 5 or 6 'red' agents with equal likelihood. In this case the degree distribution itself ensures that nodes know the type of their neighbors as being the opposite of their own type. Multi-type configuration models are of interest since they allow for a rich variety 'social connection' patterns.

Remark 5.9. *Our algorithm suggests a heuristic update rule that can apply to general graphs. Consider any random graph model with some degree distribution that is common knowledge. The model can include many short loops but suppose agents perform their computations as though their neighborhood (beyond immediate neighbors) is a tree drawn according to the edge perspective degree distribution, and this is common knowledge. The computations and updates employed are then exactly as described above.*

We discuss this further in Section 8.2.

5.3.3 Observing random subsets of neighbors

We may not interact with each of our friends every day. Suppose that for each edge e, there is a probability p_e that the edge will be 'active' in any particular iteration, independent of everything else. Let $a_e(t) \in \{*, \mathtt{a}\}$, be an indicator variable for whether edge e was active at time t ($\mathtt{a}$ denotes 'active'). Now, the observation by node i of node j belongs to an extended set that includes an additional symbol $*$ corresponding to the edge being inactive. Thus, there are $(|\mathcal{S}|+1)^{t+1}$ possible observed trajectories up to time t. Our algorithm can be easily adapted for this case. The modified 'inert' agent process involves fixing state of the world s, trajectory τ_i and also $(a_{ij}(t))_{j\in\partial i}$ for all times t. The form of posterior on the state of the world, Eq. (6), remains unchanged. The cavity recursion Eq. (9) now includes a summation over the possibilities for $(a_1^{t-1}, \ldots, a_{d-1}^{t-1})$. The overall complexity remains $2^{O(td)}$.

The case where node v becomes inactive with some probability p_v in an iteration, independent of everything else, can also be handled similarly. A suitable formulation can also be obtained when both the above situations are combined, so that both nodes and edges may be inactive in an iteration.

6 Proofs of convergence results

6.1 Directed trees: Proof of Proposition 3.5

Lemma 6.1. *Consider the setting of Proposition 3.5. The error probability at any node i at time t is bounded as*

$$\mathbb{P}\left[\sigma_i(t) \neq s\right] \leq \delta_t\,,$$

where $\delta_0 \equiv \delta$ and we have a recursive definition

$$\delta_{t'} \equiv \mathbb{P}\left[\text{Binomial}(d, \delta_{t'-1}) \geq d/2\right]\,. \tag{15}$$

Proof of Lemma 6.1. We proceed by induction on time t. Clearly, the error probability is bounded above by δ_0 at $t=0$. Let $\mathcal{S} = \mathcal{X} = \{-1, +1\}$ for convenience (we already assumed s and x_i are binary). Note that, by symmetry, the error probability $\mathbb{P}\left[\sigma_i(t) \neq s\right]$ does not depend on s or on i. Suppose, $\mathbb{P}\left[\sigma_i(t) \neq s\right] \leq \delta_t$. Consider a node j making a decision at time $t+1$. Let the children of j be $1, 2, \ldots, d$. Define $\widetilde{\sigma}_j$, the opinion of the majority of the children, by

$$\widetilde{\sigma}_j(t+1) = \text{sgn}\left(\sum_{l=1}^{d} \sigma_l(t)\right),$$

where sgn(0) is arbitrarily assigned the value -1 or $+1$. The 'error-or-not' variables $[\sigma_l(t) \neq s]$ are independent identically distributed (i.i.d.), with $\mathbb{P}[\sigma_l(t) \neq s] \leq \delta_t$ by the induction hypothesis. Hence,

$$\mathbb{P}[\widetilde{\sigma}_j(t+1) \neq s] \leq \mathbb{P}[\text{Binomial}(d, \delta_t) \geq d/2] = \delta_{t+1}\,. \tag{16}$$

Since the agent j is Bayesian, she in fact uses the information $(x_j, \sigma_1^t, \ldots, \sigma_d^t)$ to compute a MAP estimate $\sigma_j(t+1)$ of the true state of the world. Clearly, $\mathbb{P}[\sigma_j(t+1) \neq s] \leq \mathbb{P}[\widetilde{\sigma}_j(t+1) \neq s]$. Using Eq. (16), it follows that $\mathbb{P}[\sigma_j(t+1) \neq s] \leq \delta_{t+1}$. Induction completes the proof. □

Proof of Proposition 3.5. For a symmetric tie breaking rule and $\delta < 1/2$, it is straightforward to establish that $\delta_t \equiv \mathbb{P}[\sigma_i(t) \neq s]$ is monotonic decreasing in t, and converges to 0. It follows (by an argument similar to the one used in the proof of theorem 3.7 below) that we have doubly exponential convergence to the true state of the world:

$$-\log \mathbb{P}[\sigma_i(t) \neq s] \in \Omega\left((d/2)^t\right).$$

implying that $O(\log\log(1/\epsilon))$ rounds suffice to reduce the error probability to below ϵ. □

6.2 Majority dynamics: Proof of Theorem 3.7

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

Definition 6.2. *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.*

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

Ties are broken by flipping an unbiased coin.

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

Lemma 6.3. *Consider the setting in Theorem 3.7. Let i and j be adjacent nodes in the tree. Then for all $(\widehat{\sigma}_i^{t-1}, \widehat{\sigma}_j^{t-1}) \in \{-1, +1\}^{2t}$*

$$\mathbb{P}\left[\widehat{\sigma}_i(t) = -1 | \widehat{\sigma}_i^{t-1}, \widehat{\sigma}_j^{t-1}, s = +1\right] \leq \delta_t \tag{17}$$

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

$$\delta_t \equiv \mathbb{P}[\text{Binomial}(d-1, \delta_{t-1}) \geq d/2 - 1] \tag{18}$$

Proof. We proceed by induction. Clearly Eq. (17) holds for $t = 0$. Suppose Eq. (17) holds for some t. We want to show

$$\mathbb{P}\left[\widehat{\sigma}_i(t+1) = -1 \,|\, \widehat{\sigma}_i^t, \widehat{\sigma}_j^t, s = +1\right] \leq \delta_{t+1}\,, \tag{19}$$

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

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

$$\mathbb{P}\left[\widehat{\sigma}_i(t+1) = -1 \,|\, \widehat{\sigma}_i^t, \widehat{\sigma}_j^t, \widehat{\sigma}_{l_1}^{t-1}, \ldots, \widehat{\sigma}_{l_{d-1}}^{t-1}, s=+1\right] \,\le\, \delta_{t+1}\,, \tag{20}$$

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

We reason as follows. Fix the state of the world s and the trajectories $\widehat{\sigma}_i^t$ and $\widehat{\sigma}_j^t$. Now this induces correlations between the trajectories of the neighbors $l_1, \ldots, l_{d-1}$, caused by the requirement of consistency with the majority rule at node i, but *only up to time* $t-1$. If we further fix $\widehat{\sigma}_{l_m}^{t-1}$, then $\widehat{\sigma}_{l_m}(t)$ (and $\widehat{\sigma}_{l_m}$ at all future times) is conditionally independent of $\left(\widehat{\sigma}_{l_{m'}}^t\right)_{m' \neq m}$. Thus, we have[18]

$$\mathbb{P}\left[\widehat{\sigma}_{l_m}(t) = -1 | \, \xi, s=+1\right] \,=\, \mathbb{P}\left[\widehat{\sigma}_{l_m}(t) = -1 | \, \widehat{\sigma}_{l_m}^{t-1}, \widehat{\sigma}_i^{t-1}, s=+1\right]\,,$$

and therefore, using the induction hypothesis

$$\mathbb{P}\left[\widehat{\sigma}_{l_m}(t) = -1 | \, \xi, s=+1\right] \le \delta_t \tag{21}$$

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

$$\widehat{\sigma}_i(t+1) = \mathrm{sgn}(\widehat{\sigma}_j(t) + \widehat{\sigma}_{l_1}(t) + \ldots + \widehat{\sigma}_{l_{d-1}}(t))\,,$$

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

$$\mathbb{P}\left[\widehat{\sigma}_i(t+1) = -1 \,|\, \xi, s=+1\right] \,\le\, \mathbb{P}\left[\mathrm{Binomial}(d-1, \delta_t) \ge d/2 - 1\right]$$

from Eq. (21) and conditional independence of $\widehat{\sigma}_{l_1}(t), \ldots, \widehat{\sigma}_{l_{d-1}}(t)$. Thus, we obtain Eq. (20). Eq. (19) follows by summing over $\widehat{\sigma}_{l_1}^{t-1}, \widehat{\sigma}_{l_2}^{t-1}, \ldots, \widehat{\sigma}_{l_{d-1}}^{t-1}$. □

Proof of Theorem 3.7. By applying the multiplicative version of the Chernoff bound[19] to Eq. (18) we have that

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

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

$$\delta_{t+1} \le (2e\delta_t (d-1)/(d-2))^{\frac{1}{2}(d-2)}. \tag{22}$$

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

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

[18] A alternate argument can be constructed using the modified process with an inert agent, mirroring the reasoning used in the proof of Theorem 5.6.

[19] $\mathbb{P}\left[X \ge (1+\eta)\mathbb{E}\left[X\right]\right] \le \left(\frac{\exp \eta}{(1+\eta)^{1+\eta}}\right)^{\mathbb{E}[X]}$. We substitute $\mathbb{E}\left[X\right] = \delta_t(d-1)$ 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 2: $d = 3$, $\mathbb{P}[x_i \neq s] = 0.15$

and so

$$-\log \delta_t \in \Omega\left(\left(\tfrac{1}{2}(d-2)\right)^t\right), \tag{23}$$

as long as

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

□

Theorem 3.7 is non-trivial for $d \geq 5$. The upper limit of the 'noise' δ for which it establishes rapid convergence approaches $(2e)^{-1}$ as d grows large (see also the discussion below for large d).

Convergence for large d

We present now a short informal discussion on the limit $d \to \infty$. We can, in fact, use Lemma 6.1 to show convergence is doubly exponential for $\delta < 1/2 - c/d$ for some $c < \infty$ that does not depend on d.

Here is a sketch of the argument. Suppose $\delta = 1/2 - c_1/d$. Then, for all $d > d_1$ where $d_1 < \infty$, there exists $c_2 < \infty$ such that $\mathbb{P}[\text{Binomial}(d-1, \delta) \geq d/2 - 1] < 1/2 - c_2/\sqrt{d}$. This can be seen, for instance, by coupling with the Binomial$(d-1, 1/2)$ process and using an appropriate local central limit theorem (e.g., see [17, Theorem 4.4]). Thus, $\delta_1 < 1/2 - c_2/\sqrt{d}$. Further, c_2 can be made arbitrarily large by choosing large enough c_1. Next, with a simple application of the Azuma's inequality, we arrive at $\delta_2 < c_3$ (where $c_3 \to 0$ as $c_2 \to \infty$). Now, for small enough c_3, we use the Chernoff bound analysis in the proof of Theorem 3.7 and obtain doubly exponential convergence.

7 Further numerical results and discussion on Conjecture 3.6

Table 2, together with Table 1 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 3.6. Table 3 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 4).

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 3.6. 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 $j \in V$ leaves $\sigma_i(t)$ unchanged.

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 3: 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.

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 4: Error probabilities for agents using majority updates with $\mathbb{P}[x_i \neq s] = 0.3$, for regular trees of different degrees d.

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 [17]. However, it turns out that iterative Bayesian learning is not always monotonic[20]! 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 3.6, is even harder.

Despite non-monotonicity, it is tempting to hope for a direct proof of Conjecture 3.6, by showing inductively (in time) that iterative Bayesian learning is always at least as good 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 3.6 difficult.

8 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 networks where exact Bayesian calculations can be efficiently

[20] Elchanan Mossel and Omer Tamuz, private communication.

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

8.1 Relation to Aaronson's work on the complexity of agreement

The work most closely related in spirit to the present one is that of Aaronson on the complexity of agreement [1]. In that work, as in this one, the author started out aiming to establish complexity theoretic 'hardness': "communication complexity might provide a fundamental reason for why ... people could agree to disagree ... this was our conjecture when we began studying the topic", but instead discovered an efficient procedure to achieve the objective in question under some conditions. We briefly describe Aaronson's work next and compare it to our own.

Aaronson investigated the question of whether Aumann's classic theorem on agreement [4] is supported by an efficient procedure by which agents with a common prior can reach agreement. In that setting, the 'communication protocol' itself is unspecified, and the objective is to formulate an efficient communication protocol, along with an efficient computational procedure to implement this protocol, so as to facilitate agreement between agents. The scaling variables chosen are the number of bits n of private information, and the inverse 'error probability' $1/\epsilon$. Aaronson shows that agreement can be achieved after a 'conversation' of reasonable length (that does not depend on n) in the case of two interacting agents, and also in the case of more than two agents on a strongly connected network. Further, he shows that for two agents, the computational effort required to adequately 'simulate' this conversation is again independent of n ([1] does not establish a computational bound for networks of three or more agents). Given a desired error probability ϵ, the conversation length required grows as $\text{poly}(1/\epsilon)$, whereas the bound on computational effort grows as $\exp(\text{poly}(1/\epsilon))$.

There are several evident differences with the current work. First, the 'communication protocol' is specified implicitly by the model itself in our problem. Thus, the single objective for us is to minimize computational effort. Second, our scaling regime is very different, in that we let the number of nodes n grow large (whereas Aaronson focusses on the two agent case), but demand that private signals of agents belong to a finite set. In other words, we study the effect of large network size on computational difficulty, whereas Aaronson focusses on the effect of a large amount of private information[21]. In terms of dependence of computational effort on 'error probability', our bound of $\text{polylog}(1/\epsilon)$ on computational effort (assuming Conjecture 3.6) is doubly exponentially smaller than the bound of $\exp(\text{poly}(1/\epsilon))$ obtained by Aaronson as evidence of 'efficient computability'[22].

In the spirit of Aaronson's approach [1] to simulating a conversation between Bayesian agents using limited computations, it may be possible to develop a Monte Carlo sampling based version of the naïve dynamic program (Section 4) that is much more efficient. The chief additional difficulty would be in handling the effects of network structure, since Aaronson only deals with the case of two agents. This is an interesting direction of future study.

8.2 The Bayesian calculation on trees as a heuristic on general graphs

Our algorithm for trees, which allows exact posterior calculations, can also be used as a heuristic on general loopy graphs (hereafter 'our heuristic'), for instance in the case described in Remark 5.9. Our heuristic involves 'belief' updates that are similar in some ways (cf. Remark 5.4 and the footnote there) to previously suggested thumb rules for learning when agents are subject to 'persuasion bias', e.g., [10, 11, 14]. DeMarzo, Vayanos, and Zwiebel [11] present a detailed argument suggesting

[21] Interestingly the bounds obtained are independent of n in both works, for n large.

[22] Of course, a direct comparison is somewhat unfair since the problems being addressed are quite different.

the prevalence of persuasion bias in society, meaning that agents tend to ignore repetition in the information they receive. There are multiple possible reasons for the repetition of information, including: (i) the same neighbors' actions may be repeatedly observed, (ii) the neighbors' actions are affected by past actions of the agent herself, (iii) there may be loops in the network, causing dependence in the information received from different neighbors.

Simple heuristics such as majority rules do not correct for (ii), the effect of ones own action on ones neighbors. Our heuristic takes both (i) and (ii) into account, but neglects the effects of loops. In particular, it allows exact posterior computations on trees. Arguably, repetition of information due to loops in the network may be the aspect that agents find most difficult to incorporate into their computations. With this is mind, it seems that our heuristic or similar heuristic update rules may be of interest to investigate on general graphs. They could capture the behavior of agents who are rational in every respect except in accounting for loops in the network. Thus, our work may suggest new directions to pursue in the context of understanding "... how the theory changes if the bounded rationality takes a more general form (perhaps with full rationality being a limiting case)" (Golub and Jackson [14]).

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

References

[1] S. Aaronson. The complexity of agreement. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 634–643. ACM, 2005.

[2] D. Acemoglu, M. A. Dahleh, I. Lobel, and A. Ozdaglar. Bayesian learning in social networks. *The Review of Economic Studies*, 2011.

[3] D. Aldous and J. M. Steele. The Objective Method: Probabilistic Combinatorial Optimization and Local Weak Convergence. In H. Kesten, editor, *Probability on discrete structures*, pages 1–72. Springer Verlag, 2003.

[4] R. Aumann. Agreeing to disagree. *The Annals of Statistics*, 4(6):1236–1239, 1976.

[5] V. Bala and S. Goyal. Learning from neighbours. *Review of Economic Studies*, 65(3):595–621, July 1998.

[6] O. Bandiera and I. Rasul. Social networks and technology adoption in northern mozambique*. *The Economic Journal*, 116(514):869–902, 2006.

[7] A. V. Banerjee. A simple model of herd behavior. *The Quarterly Journal of Economics*, 107(3):797–817, 1992.

[8] D. H. Bikhchandani, S. and I. Welch. A theory of fads, fashion, custom, and cultural change as informational cascade. *Journal of Political Economy*, 100(5):992–1026, 1992.

[9] T. Conley and C. Udry. Social learning through networks: The adoption of new agricultural technologies in ghana. *American Journal of Agricultural Economics*, 83(3):668–673, 2001.

[10] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[11] P. M. DeMarzo, D. Vayanos, and J. Zwiebel. Persuasion bias, social influence, and unidimensional opinions. *Quarterly Journal of Economics*, 118(3):909 – 968, 2003.

[12] G. Ellison and D. Fudenberg. Rules of thumb for social learning. *Journal of Political Economy*, 110(1):93–126, 1995.

[13] D. Gale and S. Kariv. Bayesian learning in social networks. *Games and Economic Behavior*, 45(2):329–346, November 2003.

[14] B. Golub and M. O. Jackson. Nave learning in social networks and the wisdom of crowds. *American Economic Journal: Microeconomics*, 2(1):112–49, 2010.

[15] S. Goyal. *Connections: An Introduction to the Economics of Networks*. Princeton University Press, 2007.

[16] S. Istrail. Statistical mechanics, three-dimensionality and np-completeness: I. universality of intracatability for the partition function of the ising model across non-planar surfaces (extended abstract). In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, STOC '00, pages 87–96, New York, NY, USA, 2000. ACM.

[17] Y. Kanoria and A. Montanari. Majority dynamics on trees and the dynamic cavity method. Preprint. To appear in Annals of Applied Probability., 2011.

[18] Y. Kanoria and A. Montanari. Subexponential convergence for information aggregation on regular trees. 2011.

[19] H. Kohler. Learning in social networks and contraceptive choice. *Demography*, 34(3):369–383, 1997.

[20] L. Ménager. Consensus, communication and knowledge: an extension with bayesian agents. *Mathematical Social Sciences*, 51(3):274–279, 2006.

[21] M. Mueller-Frank. A Comment on 'Bayesian Learning in Social Networks'. *SSRN eLibrary*, 2011.

[22] M. Mueller-Frank. Mixed Communication Structures: Opinion Formation in Social Networks with Bayesian and Non-Bayesian Agents. *SSRN eLibrary*, 2011.

[23] R. Parikh and P. Krasucki. Communication, consensus, and knowledge* 1. *Journal of Economic Theory*, 52(1):178–189, 1990.

[24] M. Raynal. Fault-tolerant agreement in synchronous message-passing systems. *Synthesis Lectures on Distributed Computing Theory*, 1(1):1–189, 2010.

[25] D. Rosenberg, E. Solan, and N. Vieille. Informational externalities and emergence of consensus. *Games and Economic Behavior*, 66(2):979 – 994, 2009. Special Section In Honor of David Gale.

[26] J. Sebenius and J. Geanakoplos. Don't bet on it: Contingent agreements with asymmetric information. *Journal of the American Statistical Association*, 78(382):424–426, 1983.

[27] L. Smith and P. Sørensen. Pathological outcomes of observational learning. *Econometrica*, 68(2):371–398, 2000.