# Efficient Bayesian Social Learning

Yashodhan Kanoria[*] and Omer Tamuz[†]

May 25, 2011

**Abstract**

Most results concerning agents on social networks have been achieved either in 'herd behavior' models, where the interaction is not bidirectional, or in models where agents use rules of thumb or are boundedly rational. Bayesian models of bidirectional interaction have posed two problems, intimately related to each other: (1) they have proved notoriously difficult to analyze; and (2) in general, the calculations required of interacting Bayesian agents are seemingly 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 extend our results in various directions, including some loopy graphs. Our results indicate computational efficiency of iterative Bayesian social learning in a wide range of situations, contrary to widely held beliefs.

## 1 Introduction

Understanding the interaction of Bayesian agents on a social network has been a goal of theoretical economics for the past few decades (cf., Goyal [12]). Much progress has been made on models that are not Bayesian, such as those of Ellison and Fudenberg [9], Bala and Goyal [4], and Golub and Jackson [11], or models where the interaction is not bidirectional, such as the herd behavior models of Banerjee [5], Bikhchandani, Hirshleifer and Welch [6], Smith and Sørenson [18] and Acemoglu et al [2]. While Gale and Kariv [10], in the spirit of Aumann's Agreement Theorem [3], show an agreement theorem for a fully Bayesian model[1], it seems that very little further progress has been made towards understanding the dynamics of interacting Bayesian agents.

A key difficulty is the seemingly intractable nature of the computations made by the agents. Ellison and Fudenberg [9] describe a model of agents on a social network who use rules of thumb to make decisions, rather then use rational strategies. They cite a number of reasons for this choice, amongst them that the complexity of the calculations required of a Bayesian agent makes it both unrealistic to expect the agent to be Bayesian, and makes its behavior difficult to analyze. Likewise Bala and Goyal [4] choose to study boundedly rational agents to *"keep the model mathematically tractable."*

[*]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

[1]A gap in the proof of Gale and Kariv's agreement theorem was recently pointed out [15]. However, recent works [17, 16] establish similar results in more general settings.

While the bounded rationality approach has led to impressive results, it has two disadvantages, as compared with a fully Bayesian one: first, it 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. [17]) often opt for the more difficult but fully Bayesian model.

Another approach to Bayesian interaction is that taken by herd behavior models (cf. Banerjee [5], Bikhchandani, Hirshleifer and Welch [6]) that do feature fully Bayesian agents who learn from each other. However, the interaction is *not bidirectional*: each agent acts only once, taking into account the actions of her predecessors. These models avoid the difficulty of taking into account the effect of ones own actions on the behavior of others.

We consider a model that features repeated bidirectional interaction between fully Bayesian agents connected by a social network, and describe a novel algorithm for the agents' calculation. When the social network graph is a tree or nearly a tree, this algorithm has running time that is exponentially faster than previously known algorithms.

The model we consider (or rather, a slightly more general version of it) was introduced by Gale and Kariv [10]. 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 - her neighbors. The agents must choose between a set of possible actions, the relative merit of which depends on the $s$. The agents iteratively learn by observing their neighbors' actions, and picking an action that is myopically optimal, given the information known to them. 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 dynamic programming algorithm[2] 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 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 $2^{O(\min(n,d^t)}$. Obviously, this grows very rapidly. In the words of Gale and Kariv [10], "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.

Our main contribution is to show that when the graph of social ties is locally a tree, or close to one, and when private signals are independent conditioned on $s$, then the computational outlook is much brighter than previously thought. We develop a sophisticated dynamic program for locally tree-like graphs that reduces the computational effort to $2^{O(\min(n,td))}$.

The restriction of the discussion to tree or tree-like social networks certainly excludes many natural settings that tend to exhibit highly clustered social ties graphs. However, in some cases artificially constructed networks have no or few loops by design; these include some highly hierarchical or compartmentalized organization, as well as some physical communication networks where redundancy is expensive, since the minimal connected graph is a tree. Furthermore, the fact that a range of networks do not present a major computational hurdle for fully Bayesian calculations is itself somewhat surprising.

We conjecture, and show supporting numerical evidence, that on regular trees of degree at least five, the number of iterations needed to calculate the correct answer with probability $1-\epsilon$ is $O(\log\log(1/\epsilon))$. In fact, we rigorously establish this for the 'majority dynamics' update rule, in which agents adopt the opinion of the majority of their neighbors in the previous round[3]. Thus, our

[2]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.

[3]This result should be of independent interest. Majority dynamics is a reasonable model of social learning with

conjecture follows if iterative Bayesian learning learns at least as fast as majority, as suggested by intuition and numerical evidence, which we present. Assuming this conjecture, the computational effort required drops from quasi-polynomial in $1/\epsilon$ (using the naive dynamic program) to polynomial in $\log(1/\epsilon)$ (i.e., polylogarithmic in $1/\epsilon$).

An additional difficulty of the Gale and Kariv model is that it requires the individuals to exactly know the structure of the graph. A possible solution to this is a modification 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 the natural configuration model of random graphs (see below for full explanation) and show that our computational approach extends to this case.

We extend our approach to certain loopy graphs: We allow a bounded number of 'hub' nodes who are each observed by many nodes leading to several short loops in the connectivity graph. We show that our algorithm can be suitably modified for this case. Also, graphs with a low 'density' of short loops can be handled. 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 this can be handled when 'inactive' edges/nodes occur independently of each other and in time.

The key technique used in this paper is the dynamic cavity method, introduced by Kanoria and Montanari [14] in their study of 'recursive majority' updates on trees, which was also motivated by social learning. A dynamical version of the cavity method of Statistical Physics, this technique was used to analyze majority dynamics on trees, and appears promising for the analysis of iterative tree processes in general. In this work, we use this technique for the first time to give an algorithm for efficient computation by nodes. This is in contrast to the case of majority updates, where the update rule is computationally trivial. Our algorithmic approach leveraging the dynamic cavity method may be applicable to a range of iterative update situations on locally treelike graphs.

This work can be viewed as a step towards charting the boundaries between domains where Bayesian calculations are feasible and those where they are hard. An interesting complementary result would be a formal proof of computational hardness of a more general Bayesian model. To the best of our knowledge none are currently known.

### 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 simple 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 including some loopy graphs.

We prove our convergence results in Section 6. Section 7 discusses a conjecture regarding convergence (Conjecture 3.6) and presents further 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 [10]. We first state our model and then discuss some of the differences with the model of [10]. Some simplifications are made simply for ease of presentation, and we discuss this in Section 2.1 below.

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

bounded rationality. It is also relevant in other contexts like consensus in distributed systems.

$i$ 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. The network $G$ and the prior distribution of $s$ are assumed to be common knowledge.

Each agent $i$ receives a private signal $x_i \in \mathcal{X}$, where $\mathcal{X}$ is finite. Private signals are independent conditioned on $s$. The distributions $\mathbb{P}[x_i|s]$ are also assumed to be common knowledge.

In each discrete time period (or round) $t$ the agents choose must choose an action in $\mathcal{S}$, or 'vote[4]' on the state of the world. I.e., at each $t = 0, 1, \ldots$, each agent $i$ chooses a vote $\sigma_i(t) \in \mathcal{S}$. 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 state of the world that she currently believes is most likely, given the Bayesian posterior distribution she computes. For simplicity, we assume deterministic rules for breaking ties. The decision rules employed by agents are assumed to be common knowledge.

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 $\sigma_i^t \equiv (\sigma_i(0), \sigma_i(1), \ldots, \sigma_i(t))$ 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 $t$. Note that this does **not** include her neighbors' votes at time $t$.

The vote $\sigma_i(t)$ is chosen as $\arg\max_{s \in \mathcal{S}} \mathbb{P}\left[s \middle| \mathcal{F}_i^t\right]$. For simplicity, 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 $g_i(t) : \mathcal{X} \times |\mathcal{S}|^{t|\partial i|}$, 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. For each state of the world $s$, action $\sigma$ has utility one when the state of the world is $s = \sigma$, and zero otherwise. Thus, 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 from $\arg\max_{s \in \mathcal{S}} \mathbb{P}\left[s \middle| \mathcal{F}_i^t\right]$. Thus, $\{\sigma_i(t)\}$ and $\{\mathcal{F}_i^t\}$ form a weak perfect Bayesian equilibrium (cf. [10, Definition 1]) with this utility function. 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 utility function $U : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. Section 5 includes a precise description of why this is the case.

[4]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 $s$.

The natural objection that to such a model of behavior is that the agents should want to maximize the discounted sum of their utilities, instead of making the myopic optimal choice. Gale and Kariv [10] 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. However, this appears to be somewhat contrived. It would, in fact, be ideal if we could deal with the fully rational model with agents maximizing the discounted sum of payoffs (as considered in [17]). The myopic model can be thought of as a stepping stone towards the fully rational model, being a special case corresponding to discount factor 0.

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, 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 fixed by the model itself and hence communication cost cannot be reduced.

## 2.2 Comparison with other models

The model presented above is a special case of the Gale-Kariv model [10], which we refer to as the GK model henceforth.

In the GK model there is no explicit 'state of the world' $s$. Rather, the agents have a set of possible actions $\mathcal{A}$, and the utility of the actions is a general function $U : \mathcal{A} \times \Omega \to \mathbb{R}$ of the private signals $\omega = (\omega_1, \dots, \omega_n) \in \Omega$ of all the agents[5].

We specialize the GK model as follows:

- We reduce $\omega$ to $((x_i)_{i\in V}, s)$, where $s$ belongs to a given (finite) set $\mathcal{S}$, and 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 $s$ and $x_i$ belong to *finite* sets $\mathcal{S}$ and $\mathcal{X}$ respectively.

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 [5] Bikhchandani, Hirshleifer and Welch [6], Smith and Sørensen [18]). It is also the basis of the model of boundedly-rational agents on social networks studied by Bala and Goyal [4]. Nevertheless it is

[5]In this subsection, $\Omega$ is used to denote the set of possible private signals.

important to note that our first and second assumptions represents an important specialization of the GK model. The third assumption above is standard in a computational framework.

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

### 3.1 Asymptotic/Landau notation

We make use of the following notation:

- 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 exists $C < \infty$, $z_0$ such that $f_1(z) \leq Cf_2(z)$ 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 exists $C > 0$, $z_0$ such that $f_1(z) \geq Cf_2(z)$ 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 $\min(n, td)$). We remark further that $O(\cdot)$ and $\Omega(\cdot)$ are the only asymptotic notations that we use. For instance, $\omega$ is *not* used as an asymptotic notation.

### 3.2 Efficient computation

To the best of our knowledge, the literature (e.g., [10, 17, 16]) does not contain an explicit description of an algorithm to compute the actions chosen by agents in our model. However, it seems that the following 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 $2^{O(\min(n,(d-1)^t))}$, 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[6].

[6]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 $2^{O(\min(n,td))}$.*

The algorithm we use employs a technique called the dynamic cavity method [14], previously used only in analytical contexts. A full description of the algorithm and analysis leading to Theorem 3.2 is described in Section 5.

We extend our algorithm and computational bounds to locally 'almost' tree-like graphs, graphs with a few 'hub' nodes, random graphs, and so on. These and other generalizations are described in detail in Section 5.3. The algorithm can be modified to deal with arbitrary loopy graphs, but as the number of loops increases, the computational effort approaches that of the naive DP (cf. Proposition 3.1).

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 quite small, usually for two reasons: (1) In many settings, agents appear to converge to the 'right' answer in a very small number of iterations [10]. In Section 3.3 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.[7] (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, the issue of which candidate is better becomes almost irrelevant once ballots have been cast, and iterative updates essentially cease at that point.

## 3.3 Convergence

Since the agents gain information at each round, and since they are Bayesians, then the probability that they choose the correct action is non-decreasing with $t$, the number of rounds. We say that *convergence* occurs if this probability converges to one, or alternatively if the probability that an agent takes a wrong action converges to zero.

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

$$\mathbb{P}[\sigma_i(t) \neq s] = \exp\big(-\Omega(b^t)\big) \tag{1}$$

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 polylogarithmic in $(1/\epsilon)$ suffices to achieve error probability $\mathbb{P}[\sigma_i(t) \neq s] \leq \epsilon$.*

**Remark 3.4.** *If computational effort grows only polylogarithmically in an approximation parameter, 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. Nevertheless, we provide the following evidence for doubly exponential convergence on trees: we study a simple case with two possible states of the world and two possible

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

| Round | Bayesian | Majority |
|---|---|---|
| 0 | 0.15 | 0.15 |
| 1 | $2.7 \cdot 10^{-2}$ | $2.7 \cdot 10^{-2}$ |
| 2 | $7.6 \cdot 10^{-4}$ | $1.7 \cdot 10^{-3}$ |
| 3 | $2.8 \cdot 10^{-7}$ | $8.4 \cdot 10^{-6}$ |
| 4 | $1.4 \cdot 10^{-12}$ | $2.5 \cdot 10^{-10}$ |

Table 1: Error probability on 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.

private signal values 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.

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.

**Directed trees**

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 an undirected tree) because the trajectories of the $d$ children are uncorrelated, given $s$.

We assume a binary state of the world $s$ and independent binary signals that are each incorrect with probability $\delta$.

**Proposition 3.5.** *Consider a directed $d$-ary tree, a binary state of the world, and conditionally independent binary private signals that are each wrong with probability $\delta$. For any $\delta < 1/2$ and a symmetric tie breaking rule (e.g., "follow your private signal", or uniformly random tie breaking), 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.

**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'. 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 is natural since the iterative Bayesian update rule chooses the vote in each round that (myopically) minimizes the error probability.

**Conjecture 3.6.** *On any regular tree with independent identically distributed private signals, the error probability under iterative Bayesian learning is no larger than the error probability under majority dynamics (cf. Definition 6.2) after the same number of iterations.*

We use $\widehat{\sigma}_i(t)$ to denote votes under the majority dynamics.

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

**Theorem 3.7.** *Assume binary $s$ with uniform prior. Agents' initial votes $\widehat{\sigma}_i(0)$ are correct with probability $1-\delta$, and independent conditioned on $s$. Let $i$ be any node in an (undirected) $d$ regular tree for $d \geq 5$. Then, under the majority dynamics,*

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

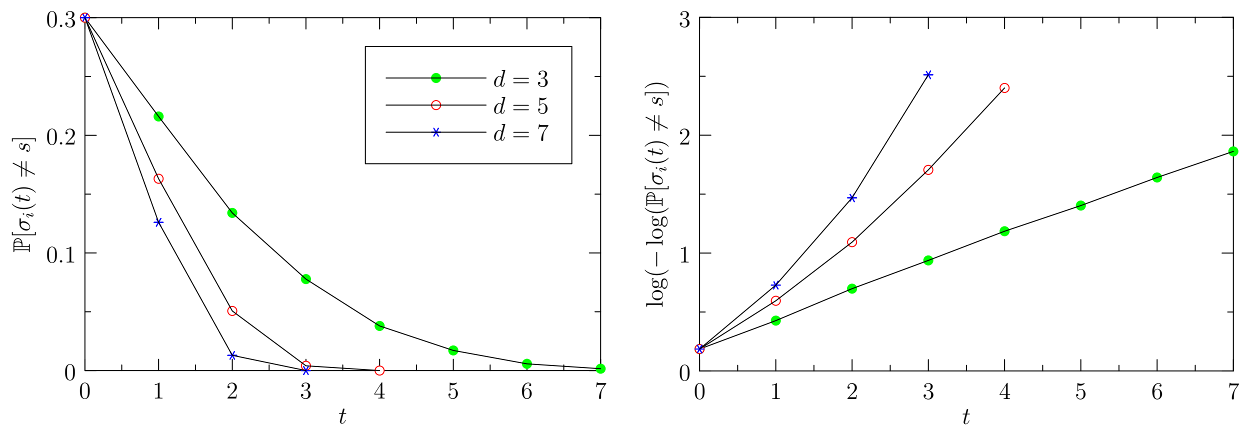

Figure 1: Error probability decay on regular trees for iterative Bayesian learning, with $\mathbb{P}[x_i \neq s] = 0.3$ (cf. Section 7). The data used to generate this figure is displayed in Table 3.

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

Thus, if Conjecture 3.6 holds:

- We also 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 under iterative Bayesian learning.
- Combining with Theorem 3.2 (cf. 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 simple 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 [13], it would be natural to conjecture that the number of iterations 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 simple 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 the error probabilities 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 most powerful supercomputers available today. With our approach, we are able to compute the decision function at iteration 3 and even at iteration 4.

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 number of iterations.

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 this Bayesian calculation is that even the brute-force solution for 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 initial 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 initial configurations are ruled out by a signal coming from a neighbor, an agent must "simulate" that neighbor's behavior, and so each agent must calculate 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 $i$, $\sigma_i$, 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 $\underline{y}$ 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 $I_i^t(x_i, \sigma_{\partial i}^t)$. To calculate $\sigma_i^t(\underline{x})$, one need only note that

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

and

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

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

It is easy to verify that using this the calculation of each $\sigma_i^t(\underline{x})$ takes $O(tn|\mathcal{X}|^n)$. One can do better than perform each of these separately, but in any case the result is exponential in $n$, so we derive a rough upper bound of $2^{O(n)}$ for this method. Since we are in particular interested in graphs of maximum degree $d$, we note that up to time $t$ an agent need only perform this for agents at distance at most $t$, and so this bound becomes $2^{O((d-1)^t)}$ for large graphs, i.e., graphs for which $n > (d-1)^t$ for relevant values of $t$.

## 5 The Dynamic Cavity Algorithm on Trees

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 $j$'s connected component in the graph $G$ with the edge $(i, j)$ removed. That is, $V_{j \to i}$ is $j$'s subtree when $G$ is rooted in $i$.

## 5.1 The Dynamic Cavity Method

We consider a modified process where agent $i$ is replaced by a *zombie* which takes fixed actions $\tau_i = (\tau_i(0), \tau_i(1), \ldots)$, and *the true state of the world is assume to be some fixed $s$.* Furthermore, this 'fixing' goes unnoticed by the agents (except $i$, who is a zombie anyway) who carry on their calculations, assuming $i$ is her regular Bayesian self, and the state of the world is 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 inserting 'zombies' 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 $V_{j\to i}$ are independent. This is formalized in the following claim, which is immediate to see:

**Claim 5.2.**

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

(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 zombie 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.** *For all $i$, $t$ and $\tau_i$*

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

*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 $\tau_i^t = g_i^t\left(x_i, \sigma_{\partial i}^{t-1}\right)$ 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. (3) and (4), we can easily write the posterior on $s$ computed by node $i$ at time $t$, in terms of the probabilities $\mathbb{Q}[\cdot||\cdot]$:

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

(Note that $\sigma_i^{t-1}$ is a deterministic function of $(x_i, \sigma_{\partial i}^{t-1})$.)

**Remark 5.4.** *A naive (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 $\tilde{\mathbb{P}}\left[s|\mathcal{F}_i^t\right] \propto \mathbb{P}[s]\,\mathbb{P}[x_i|s] \prod_{j\in\partial i} \mathbb{P}\left[\sigma_j^{t-1} \middle| s\right]$. 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:

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

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

**Remark 5.5.** *Suppose that the action set $\mathcal{A}$ was distinct from $\mathcal{S}$, and the agents had some common utility function $U : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. Eq.* (6) *would change to*

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

*All results in this section would remain unchanged. The simple DP in Section 4 admits a similar trivial extension.*

We are finally 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 $j \in \partial i$ and $t \in \mathbb{N}$*

$$\begin{aligned} &\mathbb{Q}\left[\sigma_j^t \middle|\middle| \tau_i, s\right] = \\ &\sum_{\sigma_1^{t-1} \ldots \sigma_{d-1}^{t-1}} \sum_{x_j} \mathbb{P}\left[x_j|s\right] \mathbf{1}\left[\sigma_j^t = g_j^t\left(x_j, (\tau_i^{t-1}, \sigma_{\partial j \setminus i}^{t-1})\right)\right] \cdot \\ &\qquad \cdot \prod_{l=1}^{d-1} \mathbb{Q}\left[\sigma_l^{t-1} \middle|\middle| \sigma_j^{t-1}, s\right] . \end{aligned} \tag{8}$$

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

We mention without proof that the recursion easily generalizes to the case of a *random* tie-breaking rule, provided the rule is common knowledge; 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. (5) continues to be valid in this case.

The following proof is similar to the proof of Lemma 2.1 in [14], 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 $V_{j \to i}$ 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 $\sigma_i^t$ 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 distance $t$ from $j$ (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 $\sigma_j^t$ is a function - deterministic, by our assumption - of $\underline{x}$ and $\tau_i^t$. We shall denote this function by $F_{j \to i}$ 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

$$\mathbb{Q}\left[\sigma_j^t = \lambda^t \middle|\middle| \tau_i^t, s\right] = \sum_{\underline{x}} \mathbb{P}\left[\underline{x}|s\right] \mathbf{1}\big(\lambda^t = F_{j \to i}^t(\underline{x}, \tau_i^t)\big) . \tag{9}$$

We now analyze each of the terms appearing in this sum. Since the initialization is i.i.d., we have

$$\mathbb{P}\left[\underline{x}\,|s\right] = \mathbb{P}\left[x_j|s\right]\mathbb{P}\left[\underline{x}_1|s\right]\mathbb{P}\left[\underline{x}_2|s\right]\dots\mathbb{P}\left[\underline{x}_{d-1}\big|s\right]\ . \tag{10}$$

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

$$\begin{aligned}\mathbf{1}\big(\lambda^t = F^t_{j\to i}(\underline{x}, \tau^t_i)\big) = &\sum_{\sigma^{t-1}_1\dots\sigma^{t-1}_{d-1}} \mathbf{1}\Big(\lambda^t = g^t_j(x_j, \sigma^{t-1}_{\partial j})\Big)\\ &\cdot\prod_{l=1}^{d-1}\mathbf{1}\Big(\sigma^{t-1}_l = F^{t-1}_{l\to j}(\underline{x}_l, \lambda^{t-1})\Big)\,.\end{aligned} \tag{11}$$

Using Eqs. (10) and (11) in Eq. (9) and separating terms that depend only on $\underline{x}_i$, we get

$$\begin{aligned}&\mathbb{Q}\left[\sigma^t_j = \lambda^t\middle|\middle|\tau^t_i, s\right] =\\ &\sum_{\sigma^{t-1}_1\dots\sigma^{t-1}_{d-1}}\sum_{x_j}\mathbb{P}\left[x_j|s\right]\mathbf{1}\left(\lambda^t = g^t_j(x_j, \sigma^{t-1}_{\partial j}\right)\cdot\\ &\qquad\cdot\prod_{l=1}^{d-1}\sum_{\underline{x}_l}\mathbb{P}\left[\underline{x}_l|s\right]\ \mathbf{1}\left(\sigma^{t-1}_l = F^{t-1}_{l\to j}(\underline{x}_l, \lambda^{t-1})\right)\,.\end{aligned}$$

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

## 5.2 The Agents' Calculations

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

1. $\mathbb{Q}\left[\sigma^{t-1}_j\middle|\middle|\tau^{t-1}_i, s\right]$, for all $s$ and for all $i, j \in V$ such that $j \in \partial i$, and for all $\tau^{t-1}_i$ and $\sigma^{t-1}_j$.
2. $g^t_i(x_i, \sigma^{t-1}_{\partial i})$ for all $i$, $x_i$ and $\sigma^{t-1}_{\partial i}$.

Note that these can be calculated without making any observations - only knowledge of the graph is needed.

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

1. $\mathbb{Q}\left[\sigma^t_j\middle|\middle|\tau^t_i, s\right]$ for all $s, i, j, \sigma^t_j, \tau^t_i$. These can be calculated using Eq. (8), given the quantities from the previous iteration.
2. $g^{t+1}_i(x_i, \sigma^t_{\partial i})$ for all $i$, $x_i$ and $\sigma^t_{\partial i}$. These can be calculated using Eqs. (5) and (6) and the the newly calculated $\mathbb{Q}\left[\sigma^t_j\middle|\middle|\tau^t_i, s\right]$.

Since agent $j$ calculates $g^{t+1}_i$ for all $i$, then she in particular calculates $g^{t+1}_j$. Therefore, she can use this to calculate her next action, once she observes her neighbors' 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.* The simple DP requires computational effort of $2^{O(n)}$(cf., Proposition 3.1). Thus it suffices to show that using the dynamic cavity approach, we can complete the computations with effort $2^{O(td)}$.

Consider an agent $j$, whose actions we want to compute up to round $t$, i.e., we want 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 $g_i^0(x_i) = \arg\max_{s\in\mathcal{S}} \mathbb{P}\left[s|x_i\right]$ 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 $i\in\partial k$, for all $\sigma_i^0$,$\tau_k^0$, $s$, using Eq. (8).

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. (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 $\mathbb{Q}\left[\sigma_i^{t-t'}\middle|\middle|\tau_k^{t-t'}, s\right]$ for all $i\in\partial k$, for all $\sigma_i^{t-t'}$,$\tau_k^{t-t'}$, $s$, using Eq. (8). 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 $C = C(d) < \infty$. 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 graphs with loops and 'hubs', Section 5.3.2 discusses random graphs, Section 5.3.3 relaxes the assumption that the entire graph is common knowledge and Section 5.3.4 allows nodes/edges to be inactive in some rounds.

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 update rule $g_{i,t}(\cdot)$ 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[8].

**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 'well specified' and 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. (7) admits a trivial generalization), provided these are common knowledge.

[8]Such settings have been proposed, for instance, in [16], where the network consists of a mixture of Bayesian and non-Bayesian agents.

We already mentioned that there is a simple generalization to the case of random tie breaking rules (that are common knowledge).

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 $\partial i$. In other words, our result holds for the class of directed graphs lacking cycles of length greater than two (which correspond to 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. (5), $\mathbb{P}[x_i|s]$ is replaced by $\prod_{t'=0}^{t} \mathbb{P}\left[x_{i,t'}|s\right]$, and there is an analogous replacement for $\mathbb{P}[x_j|s]$ in Eq. (8).

**Remark 5.9.** *It should be possible to use our dynamic cavity approach to enable efficient Bayesian computations in a* social experimentation *setting (see, e.g., [9, 4]), where payoffs are observed in each round by agents and their neighbors.*

We do not pursue this extension here, to avoid complicating the discussion by introducing a new model.

### 5.3.1 Loops and Hub nodes

A class of graphs that are not trees, but for which this dynamic cavity method can be easily extended is that of trees with 'hub' nodes in addition.

Consider then a graph that is not a tree, but can be transformed into a tree by the removal of some small set of nodes $V_{\text{loop}} \subset V$. Then the same calculations above can still be performed, with a time penalty of $|\mathcal{X}|^{|V_{\text{loop}}|}$; the calculation in Eq. (8) is simply repeated for each possible set of private signals of the hub nodes, and the probabilities in Eq. (5) are arrived at by averaging the $|\mathcal{X}|^{|V_{\text{loop}}|}$ different possible cases. In fact, one may not even need to enumerate over all nodes in $V_{\text{loop}}$, since at iteration $t$ only those inside $B_i^t$ (the ball of radius $t$ around $i$) effect the outcome of $i$'s calculations. Hence the complexity of calculations up to iteration $t$ is now $|\mathcal{X}|^{n_t} 2^{O(td)}$, where $n_t = \max(|B_i^t \cap V_{\text{loop}}|)_{i \in V}$.

**Remark 5.10.** *Thus, our computational framework can be extended to arbitrary graphs. However, as described above, the computational effort required grows exponentially in the number of nodes $n_t$ one has to enumerate over.*

If we also allow directed edges in this model, then we can extend it to include nodes of unlimited in-degree, i.e., some nodes may be observed by a unbounded number of others. These are agents who are observed by any number (perhaps an infinity) of other agents, in the spirit of Bala and Goyal's 'royal family' [4]. We call such nodes 'hubs' for obvious reasons. For instance, a popular blogger or a newspaper might constitute such a hub. Here too the same computational guarantees holds.

### 5.3.2 Random graphs

Consider a random graph on $n$ nodes drawn from the configuration model with a given degree distribution[9]. It is well known that such graphs are locally tree-like with high probability(see, e.g.

[9]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.

[8]). More formally, for any $t < \infty$ we have

$$\lim_{n\to\infty} \mathbb{P}\left[B_i^t \text{ is a tree.}\right] = 1\,. \tag{12}$$

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.[10] Thus, our computational approach works for random graphs w.h.p.

### 5.3.3 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. Moreover, the assumption that agents are assumed to know the entire graph structure is 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 $\rho_V$, 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: Each of the neighbors of node $v$ has a degree drawn independently according to the 'edge perspective' degree distribution $\rho_E$, 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 $T_{d_v}^D$.

Now suppose that agents are, in fact, connected in a graph drawn from the ensemble $G_n$ with degree distribution $\rho_V$. Suppose that each node $u$ knows the distribution $\rho_V$ and its own degree $d_u$, but does not know anything else about $G_n$.[11] 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 $\rho_V$ via $T_{d_v}^t$. Since nodes know only their own degree, there are only $\Delta$ different 'types' of nodes, where $\Delta$ 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 node $i$. Make it a 'zombie' following the vote trajectory $\tau_i$. Now fix some $\partial i$ (ensure $\rho_V(|\partial i|) > 0$). Choose arbitrary $j \in \partial i$. Define $\mathbb{Q}\left[\sigma_j^t = \omega_j^t \middle\| \tau_i^t, s\right]$ 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 $\partial i$) and over the private signals. Note here that $\mathbb{Q}\left[\sigma_j^t = \omega_j^t \middle\| \tau_i^t, s\right]$ *is the same for any $i$, $\partial i$ and $j \in \partial i$.*

Eqs. (3), (5) and (6) continue to hold w.h.p.[12] for the same reasons as before.

[10]In fact, as mentioned earlier, nodes with a small number of loops in the vicinity can also do their calculations without trouble.

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

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

The dynamic cavity recursion, earlier given by Eq. (8), becomes

$$\begin{aligned}\mathbb{Q}\left[\sigma_j^t \middle\| \tau_i, s\right] = \sum_{d\in\mathbb{N}} \rho_E(d) \sum_{\sigma_1^{t-1}\dots\sigma_{d-1}^{t-1}} \sum_{x_j} \mathbb{P}\left[x_j|s\right] \cdot \\ \cdot\, \mathbf{1}\left[\sigma_j^t = g_j^t\left(x_j, (\tau_i^{t-1}, \sigma_{\partial j\setminus i}^{t-1})\right)\right] \cdot \\ \cdot \prod_{l=1}^{d-1} \mathbb{Q}\left[\sigma_l^{t-1} \middle\| \sigma_j^{t-1}, s\right] . \qquad (13)\end{aligned}$$

We have written the recursion assuming the neighbors of $j$ are named according to $\partial j \setminus i = \{1, 2, \dots, d-1\}$. Again, this holds w.h.p. with respect to $n$.

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, 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.

### 5.3.4 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 'zombie' process involves fixing state of the world $s$, trajectory $\tau_i$ and also $(a_{ij}(t))_{j\in\partial i}$ for all times $t$. The form of posterior on the state of the world, Eq. (5), remains unchanged. The cavity recursion Eq. (8) now includes a summation over the possibilities for $(a_1^{t-1}, \dots, a_{d-1}^{t-1})$. 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

**Lemma 6.1.** *On an infinite directed $d$-ary tree, the error probability (at any node) at time $t$ is bounded above by $\delta_t$, where $\delta_0 \equiv \delta$ and we have a recursive definition*

$$\delta_t \equiv \mathbb{P}\left[\text{Binomial}(d, \delta_{t-1}) \geq d/2\right] . \qquad (14)$$

*Proof of Lemma 6.1.* We proceed by induction on time $t$. Clearly, the error probability is bounded above by $\delta_0$ at $t = 0$. 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 $1, 2, \dots, d$. Define $\widetilde{\sigma}_j$, the opinion of the majority of the children, by

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

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,

$$\mathbb{P}[\widetilde{\sigma}_j(t+1) \neq s] \leq \mathbb{P}[\mathrm{Binomial}(d, \delta_t) \geq d/2] = \delta_{t+1}\,. \tag{15}$$

Since the agent $j$ is Bayesian, she in fact uses the information $(x_j, \sigma_1^t, \dots, \sigma_d^t)$ 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. (15), it follows that $\mathbb{P}[\sigma_j(t+1) \neq s] \leq \delta_{t+1}$. Induction completes the proof. $\square$

*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 $\epsilon$. $\square$

## 6.2 Majority dynamics: Proof of Theorem 3.7

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

**Definition 6.2.** *Under the majority dynamics, each node $i \in V$ chooses his vote in round $t+1$ according to the majority of the votes of his neighbors in round $t$, i.e.*

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

*Ties are broken by flipping an unbiased coin.*

As before, $s \in \{-1, +1\}$ is drawn from a 50-50 prior and nodes receive 'private signals' $\widehat{\sigma}_i(0)$ that are correct with probability $1-\delta$, and independent conditioned on $s$.

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.

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

$$\mathbb{P}\left[\widehat{\sigma}_i(t) = -1 | \widehat{\sigma}_i^{t-1}, \widehat{\sigma}_j^{t-1}, s = +1\right] \leq \delta_t \tag{16}$$

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

$$\delta_t \equiv \mathbb{P}[\mathrm{Binomial}(d-1, \delta_{t-1}) \geq d/2 - 1] \tag{17}$$

*Proof.* We proceed by induction. Clearly Eq. (16) holds for $t = 0$. Suppose Eq. (16) holds for some $t$. We want to show

$$\mathbb{P}\left[\widehat{\sigma}_i(t+1) = -1 \,|\, \widehat{\sigma}_i^t, \widehat{\sigma}_j^t, s = +1\right] \leq \delta_{t+1}\,, \tag{18}$$

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,

$$\mathbb{P}\left[\widehat{\sigma}_i(t+1) = -1 \,|\, \widehat{\sigma}_i^t, \widehat{\sigma}_j^t, \widehat{\sigma}_{l_1}^{t-1}, \ldots, \widehat{\sigma}_{l_{d-1}}^{t-1}, s = +1\right] \\ \leq \delta_{t+1}\,, \tag{19}$$

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, 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 $\left(\widehat{\sigma}_{l_{m'}}^t\right)_{m' \neq m}$. [13] Thus, we have

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

and therefore, using the induction hypothesis

$$\mathbb{P}\left[\widehat{\sigma}_{l_m}(t) = -1 |\, \xi, s = +1\right] \leq \delta_t \tag{20}$$

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. We have

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

from Eq. (20) and conditional independence of $\widehat{\sigma}_{l_1}(t), \ldots, \widehat{\sigma}_{l_{d-1}}(t)$. This yields Eq. (19). Eq. (18) 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[14] to Eq. (17) 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

$$\delta_{t+1} \leq (2e\delta_t(d-1)/(d-2))^{\frac{1}{2}(d-2)}. \tag{21}$$

[13]A more formal proof can be constructed by mirroring the reasoning used in the proof of Theorem 5.6.

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

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

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

and so

$$-\log \delta_t \in \Omega\left(\left(\tfrac{1}{2}(d-2)\right)^t\right), \tag{22}$$

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 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 $\mathbb{P}\left[\text{Binomial}(d-1, \delta) \geq d/2 - 1\right] < 1/2 - c_2/\sqrt{d}$. 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 [14, 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 (though we omit to present the majority results here).

The running time to generate these tables, on a standard desktop machine was less than a minute. 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. One would 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 [14].

| Round | Bayesian | Majority |
|---|---|---|
| 0 | 0.15 | 0.15 |
| 1 | $6.1 \cdot 10^{-2}$ | $6.1 \cdot 10^{-2}$ |
| 2 | $1.5 \cdot 10^{-2}$ | $3.0 \cdot 10^{-2}$ |
| 3 | $3.0 \cdot 10^{-3}$ | $1.6 \cdot 10^{-2}$ |
| 4 | $3.4 \cdot 10^{-4}$ | $9.2 \cdot 10^{-3}$ |
| 5 | $2.7 \cdot 10^{-5}$ | $5.5 \cdot 10^{-3}$ |
| 6 | $2.2 \cdot 10^{-6}$ | $3.4 \cdot 10^{-3}$ |
| 7 | $1.4 \cdot 10^{-7}$ | $3.4 \cdot 10^{-3}$ |

Table 2: $d = 3$, $\mathbb{P}[x_i \neq s] = 0.15$

| Round | $d = 3$ | $d = 5$ | $d = 7$ |
|---|---|---|---|
| 0 | 0.30 | 0.30 | 0.30 |
| 1 | 0.22 | 0.16 | 0.13 |
| 2 | 0.13 | $5.1 \cdot 10^{-2}$ | $1.3 \cdot 10^{-2}$ |
| 3 | $7.8 \cdot 10^{-2}$ | $4.1 \cdot 10^{-3}$ | $4.4 \cdot 10^{-6}$ |
| 4 | $3.8 \cdot 10^{-2}$ | $1.6 \cdot 10^{-5}$ | |
| 5 | $1.7 \cdot 10^{-2}$ | | |
| 6 | $5.7 \cdot 10^{-3}$ | | |
| 7 | $1.5 \cdot 10^{-3}$ | | |

Table 3: Error probabilities with $\mathbb{P}[x_i \neq s] = 0.3$, for regular tree of different degrees $d$. This data is displayed in Figure 1.

However, it turns out that iterative Bayesian learning is not always monotonic[15]! 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 have presented a new algorithmic approach that casts doubt on the conventional wisdom 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 multiple short loops, that do occur frequently in practice. A significant open question suggested by our results is: What is the 'computational boundary' between networks where exact Bayesian calculations can be efficiently performed, and networks where this is not possible?

Our work suggests another interesting direction of further study. It has been suggested that agents updating their individual beliefs on a loopy network are subject to 'persuasion bias' (see, e.g.,

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

the work of DeMarzo, Vayanos, and Zwiebel [7]), tending to ignore repetitions in information due to loops in the neighborhood. Indeed, persuasion bias is built in to any bounded rationality model where agents update using some kind of weighted average of information received from different neighbors (e.g., Golub and Jackson [11]). Our dynamic cavity based update rule allows an agent to combine information received from neighbors so as to compute exact Bayesian posteriors on a locally treelike graphs. The same update rule can be used as a model of agent behavior on general loopy graphs: It constitutes, then, the rational limit among the class of heuristic update rules that ignore loops in the network, and is thus a natural candidate for study.

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 here, 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 [3] 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). Given a desired error probability $\epsilon$, 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[16]. 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'[17].

**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, 2005.

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

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

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

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

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

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

[6] 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.

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

[8] A. Dembo and A. Montanari. Ising models on locally tree-like graphs. submitted.

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

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

[11] 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.

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

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

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

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

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

[17] 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.

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