Maximizing Truth Learning in a Social Network is NP-hard

Filip Úradník*
Charles University
Prague, Czech Republic
uradnik@kam.mff.cuni.cz

Amanda Wang*
Princeton University
Princeton, NJ, United States
aw4309@princeton.edu

Jie Gao
Rutgers University
Piscataway, NJ, United States
jg1555@rutgers.edu

ABSTRACT

Sequential learning models situations where agents predict a ground truth in sequence, by using their private, noisy measurements, and the predictions of agents who came earlier in the sequence. We study sequential learning in a social network, where agents only see the actions of the previous agents in their own neighborhood. The fraction of agents who predict the ground truth correctly depends heavily on both the network topology and the ordering in which the predictions are made. A natural question is to find an ordering, with a given network, to maximize the (expected) number of agents who predict the ground truth correctly. In this paper, we show that it is in fact NP-hard to answer this question for a general network, with both the Bayesian learning model and a simple majority rule model. Finally, we show that even approximating the answer is hard.

KEYWORDS

Social Networks, Social Learning, Sequential Learning, Bayesian Learning, Complexity, Majority Vote

ACM Reference Format:
Filip Úradník, Amanda Wang, and Jie Gao. 2025. Maximizing Truth Learning in a Social Network is NP-hard. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Information acquisition, opinion formation and decision making are deeply embedded in a social context. There are many situations in which people make decisions using information that carries uncertainties, and such decisions are easily influenced by the decisions of others. It is therefore of paramount importance to understand how to effectively use information of inherent uncertainty, while considering how social exchanges can reduce such uncertainties.

Broadly, there are two primary families of models that capture decision making processes in a social network. The first family considers opinion dynamics [2, 20], where all agents have individual opinions that are repeatedly updated based on information exchange with others. Opinion dynamics studies the evolution of an opinion landscape over time and asks whether, and how quickly, a consensus can be obtained. Recent work also studies the lack of consensus (i.e., polarization) and asks whether this can be modeled and explained with natural factors [25]. The second family considers sequential decision making processes [10, 18], where agents make one-shot decisions. It is also assumed that there is an unknown ground truth state which all agents wish to learn, whereas opinion dynamics often omits such an assumption. Therefore, this second setting is termed sequential social 'learning'.

Our Model. We consider the classical sequential learning model, where n agents sequentially predict an unknown ground truth θ [14, 19]. Each agent has an independent private measurement of θ. We consider the 'bounded belief' setting, where each agents' private measurement has the same probability p, a constant away from 1, of being correct [1, 24]. In addition, each agent has access to the predictions of agents earlier in the sequence. Ideally, agents can use the information extracted from the earlier predictions to improve their own prediction. However, a well-known problem that arises is *information cascade* or *herding* [5–7, 24, 26], where sufficiently many wrong predictions early on can trigger all subsequent agents to ignore their own private signals and 'follow the herd'. Notably, this can occur even for fully rational agents, in the absence of behavioral factors like peer pressure. So herding arises not by fault of the agents, but as a result of the sequential structure of the setting. Indeed, whenever a large enough part of the crowd discards its private information, whether rationally or not, the crowd as a whole is unable to learn the ground truth.

Motivated by the problem of information cascades, a lot of follow-up work examines how to restore truth learning in crowds. One approach is to limit the visibility of agents [1, 21, 23] so that incorrect predictions do not propagate. A natural setup is to consider a social network, rather than an unstructured crowd, in which an agent can only see the actions of its neighbors earlier in the sequence [3, 4, 17]. It can be shown that certain network structures, coupled with a good agent decision ordering, can in fact achieve a strong learning result called asymptotic network-wide truth learning [17]: that all n agents, except $o(n)$ of them, successfully predict the ground truth as n goes to infinity. In other words, the average network learning rate, or the average success probability over all agents, approaches 1.

One natural open problem from [17] asks whether given a social network, we can either find a good decision ordering or decide if a good ordering exists. The authors in [17] presented sufficient conditions and impossibility results, but the big picture is still largely unclear. In general, there are two factors that prohibit truth learning. If the network is too sparse or a constant fraction of agents make decisions using only their private signals, then their success probability is bounded by a constant away from 1, and network-wide truth learning is already doomed. On the other hand, if the network is too dense or almost every agent is well connected with agents earlier in the ordering, then herding happens. Again in this case,

*These authors contributed equally.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

truth learning is not possible. Regarding the decision problem, [17] conjectures that deciding if a network admits an ordering enabling asymptotic truth learning is NP-hard.

Our Results. This work focuses on the problem of deciding if the best possible average learning rate in a network exceeds a given threshold $1 - \varepsilon$. We prove that this problem is NP-hard both for networks of fully rational agents and those with bounded rationality. Intuitively, we expect this problem to be hard—naively, there are exponentially many decision orderings to check—but proving it formally is highly non-trivial. At a cursory glance, the two barriers for truth learning suggest that a successful network should avoid very sparse and very dense structures. So a natural approach may be to relate network learning and the maximum independent set or maximum clique decision problems. However, it remains unclear how to characterize a network's learning rate by the sizes of its max independent set or its max clique. Thus, it is not obvious how to reduce either problem to the network learning problem.

We instead use a reduction from 3-SAT, the canonical NP-hard decision problem asking if a given 3-CNF formula is satisfiable. We construct a network from an input 3-CNF instance and map decision orderings on the network to boolean variable assignments. In our construction, satisfied clauses correspond to subgraphs with higher learning rates than unsatisfied ones. Therefore, the larger the number of satisfied clauses, the higher the network learning rate. We prove that all orderings corresponding to a satisfying assignment achieve a strictly higher network learning rate than those corresponding to non-satisfying assignments. Thus, deciding if the optimal learning rate exceeds a well-chosen threshold immediately implies an answer to 3-SAT. While the high-level idea is clean, the details are fairly technical due to the dependence between predictions of neighboring agents. We defer the most technical parts of the proof to the Appendix.

Next, we focus on the approximation hardness of this problem. We construct a reduction to Max 3-SAT, which asks for the maximum number of satisfiable clauses under any variable assignment in a 3-SAT instance. It is known [12] that computing a solution that satisfies more than $\frac{7}{8}M^*$ clauses is NP-hard, where M^* is the maximum number of clauses that can be satisfied. Using the ideas from the 3-SAT reduction, we prove that it is impossible to find an efficient approximation up to an arbitrary constant, unless P = NP. Specifically, this also allows us to strengthen our previous claim about NP-hardness—that finding the optimal learning rate is NP-hard for any fixed prior $p \in (\sqrt{7/8}, 1)$.

As noted above, we present hardness for sequential social learning both when agents are fully rational and when they have bounded rationality. Fully rational agents can be realistic, for instance, in modelling financial traders, while agents with bounded rationality may be more faithful models for voters or users on a social platform. More concretely, fully rational agents predict via Bayesian inference, whereas agents with bounded rationality use a simpler heuristic, such as majority vote over all available signals.

From a practical perspective, Bayesian inference is computationally expensive to implement in simulations, and all prior work on this topic [3, 4, 17] used bounded rationality in experiments, such as majority vote. However, while simpler for implementation, majority vote is actually harder for analysis. With the Bayesian model, the success rate of an agent is at least as high as the highest success rate of its earlier neighbors, since a node can never do worse than copying the action of an earlier neighbor. This monotonicity can be helpful for constructing a good ordering. However, with majority vote, this nice property no longer holds, so the validity of the hardness reduction needs to be reconsidered for the majority vote model. We modify the construction and restore the same hardness claims for the majority vote model.

Related Literature. A number of recent works consider models with repeated observations and information exchange, asking if the agents successfully learn the ground truth (see e.g., [14, 19]). One major model choice is how an agent aggregates information from available signals in the network. The most natural choice is the Bayesian model, in which agents compute the posterior probability for θ using all available information and any common knowledge, such as the network topology. A recent line of work showed that computing an agent's prediction via Bayesian inference with repeated opinion exchange is PSPACE-hard [11, 13]. It is one of the few works, to the best of our knowledge, that formally characterizes the complexity of decision making in a social network. From this perspective, our work adds to the relatively scarce literature that combinatorial complexity arises in a sequential, one-shot setup with Bayesian or non-Bayesian inferences, where decision orderings need to be carefully decided with respect to the network topology.

Lastly, we remark that computing and approximating posterior probabilities in general Bayesian networks is known to be hard [8, 9, 15]. Our setting considers a restricted variant of a Bayesian network—for example, all private signals have the same probability of error. Furthermore, we are interested in the average accuracy of the agents in the network, not their individual posterior probabilities. Thus, to the best of our knowledge, there is no straightforward way to translate the existing complexity results for a general Bayesian network to our setting.

2 PROBLEM STATEMENT

This work studies two popular models of opinion exchange on networks. The overarching goal is for a network of truthful, rational agents to learn a binary piece of information, which we call the state of the world or *ground truth*. We can also think of this state as an optimal binary action (buying or selling a stock, voting for a political party's candidate, etc.). The agents are arranged on a directed graph $G = (V, E)$ and broadcast which of the two states they believe is more probable to their neighbors.

More explicitly, we encode the ground truth in $\theta \in \{0, 1\}$, distributed according to $\mathrm{Ber}(q)$, Bernoulli distribution with probability of q of taking 1 and $1 - q$ of taking 0. Every agent $v \in V$ initially receives an independent *private signal* $s_v \in \{0, 1\}$ correlated with the ground truth. They then announce a prediction $a_v \in \{0, 1\}$ of the ground truth along outgoing edges, so out-neighbors of v may use a_v to improve their own predictions. Importantly, the probabilities p and q, as well as the graph G are all common knowledge. This is captured in the following formal definition of a network.

Definition 2.1 (Social Network). A *social network* is $\mathcal{N} := (G, q, p)$, where

(1) $G = (V, E)$ is a directed graph with agents as vertices,

(2) $q \in (0, 1)$ is the prior probability of $\theta = 1$,
(3) $p \in (\frac{1}{2}, 1)$ is the accuracy of agents' private signals $s_v \in \{0, 1\}$, such that

$$\Pr[s_v = 1 \mid \theta = 1] = \Pr[s_v = 0 \mid \theta = 0] = p, \quad \forall v \in V.$$

We further denote $n := |V|$.

We consider a classic asynchronous *sequential* model [10], in which agents announce their predictions in a *decision ordering*, given by a one-to-one mapping $\sigma : V \longrightarrow [n]$. We denote the set of all possible orderings by Σ_n. At every time step i, agent $v = \sigma^{-1}(i)$ makes an announcement $a_v \in \{0, 1\}$. The announcement depends on the agent's private measurement s_v, along with the *previous announcements of in-neighbors*, which we denote as a tuple N_v, defined as

$$N_v := (a_u \mid u \in V \wedge uv \in E \wedge \sigma(u) < \sigma(v)).$$

We call the tuple $X_v = (s_v) \cup N_v$ the *inputs* of node v. This setup of limiting visibility to an agent's neighborhood has been studied in a number of recent papers [3, 4, 17].

When making announcements, agents follow an *aggregation rule*, which is a function $\mu : (X_v, G, \sigma) \mapsto a_v$. Broadly speaking, aggregation rules can either be Bayesian or non-Bayesian. In the *Bayesian* model, agents are fully rational and make predictions according to their posterior probability for θ, given their inputs and knowledge of the network topology G. In particular, agents take into account the correlation between their inputs resulting from the network topology and from the current ordering σ.

$$\mu^B(X_v, G, \sigma) := \begin{cases} 1 & \text{if } \Pr_{G,\sigma}[\theta = 1 \mid X_v] > \frac{1}{2}, \\ 0 & \text{if } \Pr_{G,\sigma}[\theta = 0 \mid X_v] > \frac{1}{2}, \\ \mathrm{Ber}\left(\frac{1}{2}\right) & \text{otherwise.} \end{cases}$$

We also consider a non-Bayesian model, in which agents have bounded rationality and instead use simpler heuristic rules. This is perhaps a more practical model, as computing posterior probabilities in arbitrary networks can become computationally expensive. In particular, we examine the *majority dynamics* model, in which agents simply follow the majority among their inputs [4, 16, 22]. Since this model does not require agents to take into account correlations between their inputs derived from the network topology or the ordering, we omit G and σ as inputs to μ^M:

$$\mu^M(X_v) := \begin{cases} 1 & \text{if } \frac{1}{|X_v|} \sum_{x \in X_v} x > \frac{1}{2}, \\ 0 & \text{if } \frac{1}{|X_v|} \sum_{x \in X_v} x < \frac{1}{2}, \\ s_v & \text{otherwise.} \end{cases}$$

Finally, we quantify how successful the network is in predicting the ground truth by defining the following notion of a learning rate.

Definition 2.2. The *cumulative learning rate* (CLR) of a network $\mathcal{N}$ under the ordering σ and an aggregation rule μ is

$$\mathcal{L}(\mathcal{N}, \sigma, \mu) := \mathop{\mathbb{E}}_{\theta,s}\left[\sum_{v \in V} \mathbb{1}_{\{a_v = \theta\}}\right] = \sum_{v \in V} \Pr_{\theta,s}[a_v = \theta],$$

where the equality follows from linearity of expectation. Further, the *learning rate* (LR) of a network $\mathcal{N}$ under the ordering σ is simply

$$\overline{\mathcal{L}}(\mathcal{N}, \sigma, \mu) := \tfrac{1}{n}\mathcal{L}(\mathcal{N}, \sigma, \mu).$$

We are mainly interested in the *optimal* learning rate of a network, defined as follows.

Definition 2.3 (Optimal LRs). The *optimal cumulative learning rate* of a network $\mathcal{N}$ is

$$\mathcal{L}^*(\mathcal{N}, \mu) := \max_{\sigma \in \Sigma_n} \mathcal{L}(\mathcal{N}, \sigma, \mu),$$

and the *optimal learning rate* of a network $\mathcal{N}$ is

$$\overline{\mathcal{L}}^*(\mathcal{N}, \mu) := \max_{\sigma \in \Sigma_n} \overline{\mathcal{L}}(\mathcal{N}, \sigma, \mu).$$

Note that when p and q are clear from the context, we use the learning rate notation with only the graph, for example $\overline{\mathcal{L}}^*(G, \mu) = \overline{\mathcal{L}}^*((G, p, q), \mu)$. We can now present a formal definition of our main focus, the Opt Network Learning optimization problem, and Network Learning, its decision version.

Definition 2.4 (Opt Network Learning). Suppose μ is a fixed aggregation rule. Given a network $\mathcal{N}$, the Opt Network Learning problem is to maximize $\overline{\mathcal{L}}(\mathcal{N}, \sigma, \mu)$, over $\sigma \in \Sigma_n$.

Definition 2.5 (Network Learning). Suppose μ is a fixed aggregation rule. Given a network $\mathcal{N}$ and a constant threshold $\varepsilon \in (0, 1)$, the Network Learning decision problem asks whether

$$(\exists \sigma \in \Sigma_n) \quad \overline{\mathcal{L}}(\mathcal{N}, \sigma, \mu) \geq 1 - \varepsilon.$$

Note that Definition 2.5 can be formulated equivalently by asking whether an optimal ordering σ^* which maximizes the network learning rate achieves LR at least $1 - \varepsilon$.

In Sections 3 and 4, we focus on the decision problem, offering a proof that it is NP-hard for $\mu = \mu^B$ and $\mu = \mu^M$. Finally, in Section 5, we use insights from the NP-hardness proofs to show Opt Network Learning is hard to even approximate. Surprisingly, this gives us a stronger NP-hardness statement, showing that Network Learning is NP-hard even if we arbitrarily fix the agents' accuracy $p \in (\frac{1}{2}, 1)$.

3 PROOF OF NP-HARDNESS FOR THE BAYESIAN MODEL

We now state one of the main results of this paper—the hardness of Network Learning.

Theorem 3.1. *Network Learning with the Bayesian learning rule $\mu = \mu^B$ is NP-hard.*

3.1 Proof Idea

We perform a reduction from 3-SAT to Network Learning. Specifically, we assume in our reduction that all formulas have exactly 3 distinct literals in each clause, and never a literal along with its negation. For a given formula φ, we construct a network $\mathcal{N}$ and an $\varepsilon > 0$, such that

$$\overline{\mathcal{L}}^*(\mathcal{N}, \mu^B) \geq 1 - \varepsilon \iff \varphi \text{ is satisfiable.}$$

The network $\mathcal{N}$ consists of a directed graph G, the ground truth prior q, and the prior of the agents' private signals p. Our reduction requires setting p and G based on the formula, but it allows us to set $q = \frac{1}{2}$, regardless of φ.

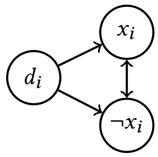

Figure 1: The cell for variable x_i.

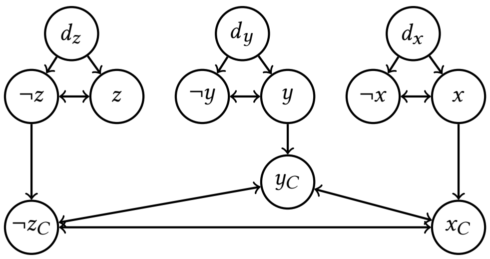

Figure 2: The graph G_φ for $\varphi = C = (x \vee y \vee \neg z)$.

3.2 Graph Construction & Notation

First, we construct the directed graph G. Our construction consists of N *variable cells*, and M *clause gadgets*, where N and M are the number of variables and clauses in φ, respectively. We define the cell and gadget first, and then define the full graph in Definition 3.4.

Definition 3.2 (Variable cell). Let x be a variable of a formula φ. A *variable cell* of x is a directed graph $\mathbb{C}(x) = (V_x, E_x)$, where

(1) $V_x = \{x, \neg x, d_x\}$, and
(2) $E_x = \{(d_x, x), (d_x, \neg x), (x, \neg x), (\neg x, x)\}$.

Figure 1 depicts a cell for some variable x_i.

Definition 3.3 (Clause gadget). Let $C = j \vee k \vee \ell$ be a clause of a 3-CNF formula φ, where $j \neq k \neq \ell$ are some literals. Then the *clause gadget* is $\mathbb{G}(C) = (V_C, E_C)$, where

(1) $V_C = \{j, k, \ell\}$, and
(2) $E_C = \{(x, y) \mid x, y \in \{j, k, \ell\} \wedge x \neq y\}$.

Definition 3.4 (Formula graph). Let $\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_M$ be a CNF formula of variables $\chi = \{x_1, \ldots, x_N\}$, where each clause C_i is the disjunction of exactly three literals. Then G_φ is a disjoint union of the graphs $\mathbb{C}(x_i)$ for all $x_i \in \chi$, and $\mathbb{G}(C_i)$ for all clauses $C_i \in \varphi$. Additionally, there is an edge to each of the vertices in every $\mathbb{G}(C_i)$ from the corresponding literal nodes in the respective variable cell.

Note that that there are no incoming edges to any cell, except from within the same cell, so the learning rate of each cell is determined only by the ordering of its own vertices. Also, no two clause gadgets share any vertices, so the ordering of vertices within a gadget only affects that gadget. For an illustration of the formula graph construction, refer to Figure 2, where we give a sample formula graph for the formula $\varphi = x \vee y \vee \neg z$.

Ordering-assignment relation. To determine satisfiability of φ from the learning rate of G_φ, we map vertex orderings to variable assignments. We then show that orderings achieving higher learning rates correspond to assignments with more satisfied clauses. For a more detailed description of both the mapping and its properties, see Section 3.4.

For clarity, we introduce some more notation. Let ℓ be a literal of some variable $x \in \chi$, meaning $\ell = x$ or $\ell = \neg x$. We say that cell $\mathbb{C}(x)$ is in one of two states: it is "on" under a decision ordering σ if $\sigma(\neg x) < \sigma(x)$; otherwise, cell $\mathbb{C}(x)$ is "off". We say that the literal ℓ is "on" if $\ell = x$ and $\mathbb{C}(x)$ is on, or $\ell = \neg x$ and $\mathbb{C}(x)$ is off; otherwise, literal ℓ is "off". For brevity, we further denote $\mathbb{C}(\ell) := \mathbb{C}(x)$, regardless of whether $\ell = x$ or $\ell = \neg x$.

3.3 Gadget Learning Rates

This section lists the learning rates of the cells and clause gadgets under Bayesian aggregation. We assume WLOG for this section that $\theta = 1$, and compute all probabilities in this section conditioned on $\theta = 1$. Since we always take θ to be uniform on $\{0, 1\}$, this yields the same values as taking the probability over θ as well. We note that our computations were verified using Wolfram Mathematica.

We begin by examining the learning rate of an arbitrary cell under a pair of orderings in which the cell is either "on" or "off". The following lemma shows that cells achieve the same learning rate under either of these orderings, and that this learning rate is the best possible over all orderings.

Lemma 3.5 (Bayesian Cell LR). *Let $x \in \chi$. Let $q = \frac{1}{2}$, and $p > \frac{1}{2}$ be given. Then*

$$\mathcal{L}^*(\mathbb{C}(x)) = \tfrac{5}{2}p + \tfrac{3}{2}p^2 - p^3.$$

In particular, if under an optimal ordering σ^* a literal ℓ is on, then its corresponding literal node has learning rate $\mathcal{L}(\ell, \sigma^*, \mu^B) = \frac{p}{2} + \frac{3}{2}p^2 - p^3$; otherwise $\mathcal{L}(\ell, \sigma^*, \mu^B) = p$.

Proof. First, observe that for the optimal ordering σ^*, it is always beneficial to put the dummy node d_x *before* the nodes x and $\neg x$. This is because d_x has no incoming edges, so it cannot acquire more information by going later, and it has edges going to x and $\neg x$, which can only increase their chances of getting the correct answer. Hence, we can see that $\mathcal{L}^*(d_x) = p$.

The case of the remaining two nodes is symmetric, so WLOG, let us assume that $\sigma^*(x) < \sigma^*(\neg x)$ (so the cell $\mathbb{C}(x)$ is "off"). The node x then receives the action of d_x, which is i.i.d. from its private information. So node x chooses its action correctly either if both s_x and a_{d_x} are correct, where a_{d_x} is the action chosen by node d_x, or if exactly one of the two are correct and x tiebreaks correctly. The first outcome occurs with probability p^2, and the second with probability $2p(1-p)\frac{1}{2}$. Thus, the learning rate of node x is

$$\mathcal{L}^*(x) = p^2 + p - p^2 = p.$$

Finally, the node $\neg x$ receives its private signal, $s_{\neg x}$ and the actions a_{d_x} and a_x. Notice that these three pieces of information are *not* independent, since x was influenced by d_x. We perform case analysis on the relative likelihood

$$\Lambda := \frac{\Pr[\theta = 1 \mid X_{\neg x}]}{\Pr[\theta = 0 \mid X_{\neg x}]} = \frac{\Pr[X_{\neg x} \mid \theta = 1]}{\Pr[X_{\neg x} \mid \theta = 0]},$$

where the equality follows from Bayes' theorem and from the fact that the prior $q = \frac{1}{2}$. Recall that $X_{\neg x} = \{s_{\neg x}, a_x, a_{d_x}\}$, and that $s_{\neg x}$

is independent of the other two. Thus, Λ can be expressed as

$$\Lambda = \frac{\Pr[s_{\neg x} \mid \theta = 1]}{\Pr[s_{\neg x} \mid \theta = 0]} \cdot \frac{\Pr\left[a_{d_x} \mid \theta = 1\right]}{\Pr\left[a_{d_x} \mid \theta = 0\right]} \cdot \frac{\Pr\left[a_x \mid \theta = 1, a_{d_x}\right]}{\Pr\left[a_x \mid \theta = 0, a_{d_x}\right]}.$$

We compute Λ for each case of $(s_{\neg x}, a_x, a_{d_x})$:

(1) $(1, 1, 1)$:

$$\Lambda = \frac{p}{1-p} \cdot \frac{p}{1-p} \cdot \frac{p+(1-p)/2}{(1-p)+p/2} = \frac{p^2}{(1-p)^2} \cdot \frac{1/2+p/2}{1-p/2} > 1.$$

(2) $(1, 1, 0)$:

$$\Lambda = \frac{p}{1-p} \cdot \frac{1-p}{p} \cdot \frac{p/2}{(1-p)/2} = \frac{p}{(1-p)} > 1.$$

(3) $(1, 0, 1)$:

$$\Lambda = \frac{p}{1-p} \cdot \frac{p}{1-p} \cdot \frac{(1-p)/2}{p/2} = \frac{p}{(1-p)} > 1.$$

(4) $(1, 0, 0)$:

$$\Lambda = \frac{p}{1-p} \cdot \frac{1-p}{p} \cdot \frac{(1-p)+p/2}{p+(1-p)/2} = \frac{1-p/2}{1/2+p/2} < 1.$$

The inequalities hold for $\frac{1}{2} < p < 1$. The remaining cases (that is $(0, 1, 1)$, $(0, 1, 0)$, $(0, 0, 1)$, $(0, 0, 0)$) follow from symmetry. It is now clear that if $\theta = 1$, $a_{\neg x}$ is correct in cases 1, 2, 3, 5. We now compute $\mathcal{L}^*(\neg x)$, which is the probability of cases 1, 2, 3, or 5 occurring.

$$\begin{aligned}\mathcal{L}^*(\neg x) &= \Pr[X_{\neg x} \in \{(1,1,1), (1,0,1), (1,1,0), (0,1,1)\}] \\ &= p(p + (1-p)\tfrac{p}{2})) + (1-p)p(p + (1-p)\tfrac{1}{2}) \\ &= \tfrac{p}{2} + \tfrac{3}{2}p^2 - p^3.\end{aligned}$$

The cumulative learning rate of the clause gadget is then the sum of the above,

$$\mathcal{L}^*(\mathbb{C}(x)) = \tfrac{5}{2}p + \tfrac{3}{2}p^2 - p^3.$$

□

To summarize, there exist two orderings for each cell which yield the same cell learning rate. The two orderings place d_x first, and correspond to either the "on" state (when $\sigma(x) < \sigma(\neg x)$) or the "off" state ($\sigma(x) > \sigma(\neg x)$). For now, we will defer the question of which of the two is optimal in the overall network ordering. Addressing this first requires examining the learning rates of the clause gadgets. Since the graph contains edges from cells to clause gadgets, there is an optimal ordering which orders all cell nodes before all clauses gadgets. We begin by examining the learning rate for arbitrary clause gadgets.

Lemma 3.6. *Let $C = \alpha \vee \beta \vee \gamma$ be a clause of φ. Let σ^* be an optimal ordering on $\mathcal{N}$. Let $p' := \frac{p}{2} + \frac{3}{2}p^2 - p^3$. Then if under σ^*, exactly i cells of α, β, γ are "on", then $\mathcal{L}(\mathbb{G}(C), \sigma^*) = \mathcal{L}_i$, where*

$$\mathcal{L}_0 := p(2p^4 - 5p^3 + 5p + 1).$$

$$\mathcal{L}_1 := p^4(2p' - 1) + p^3(2 - 4p') + p^2(1 - 2p') + 4pp' + p',$$

$$\mathcal{L}_2 := 4p^3(p' - 1)p' + p^2(-6(p')^2 + 4p' + 1) + 2pp' + p'(p' + 1),$$

$$\mathcal{L}_3 := p'(p^2(2(p')^2 - 3p' + 1) - p(2(p')^2 + p' - 3) + 2p' + 1).$$

Furthermore, if $D \in \varphi$ is a satisfied clause under the ordering σ^*, meaning at least one of the literals of D is "on" under σ^*, then

$$\mathcal{L}_3 \geq \mathcal{L}(\mathbb{G}(D), \sigma^*) \geq \mathcal{L}_1 \geq \mathcal{L}_0.$$

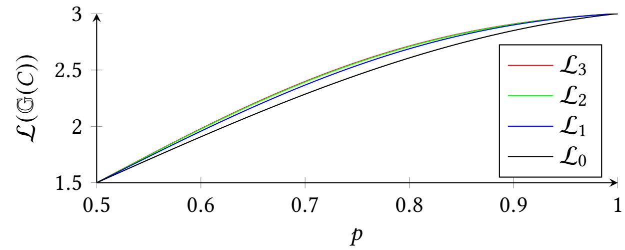

Figure 3: The relationship between learning rates of $\mathbb{G}(C)$, depending on the number of "on" literals, as a function of p.

Proof Idea. The full proof is long and technical. Here we offer the main idea, which is similar to that of Lemma 3.5.

The actions of the nodes in $\mathbb{G}(C)$ are exactly determined by

(1) the ordering of the nodes in $\mathbb{G}(C)$,
(2) the private signals of the nodes in $\mathbb{G}(C)$,
(3) the actions taken by α, β, γ, as well as α, β, γ being on or off.

We compute the learning rate in each case. We then compute the expected value over the private signals and the actions of α, β, γ, the probabilities of which are given by Lemma 3.5.

The resulting cumulative learning rate of $\mathbb{G}(C)$ depends only on the states of α, β, γ. Among these are the learning rates $\mathcal{L}_1$ and $\mathcal{L}_3$, which are the learning rates of the clause gadget when one or all of the literals are on, respectively.

The case of $\mathcal{L}_1$ actually corresponds to three sub-cases, depending on whether the node corresponding to the "on" literal is first, second, or third in the ordering. Notice that swapping the ordering of the three vertices inside the clause gadget does not affect the learning rate of other vertices in the network. Thus, since σ^* is an optimal ordering, it maximizes the learning rate for this clause, and as such, $\mathcal{L}_1$ is the maximum over the three subcases.

The full proof can be found in Appendix A. □

See Figure 3 for the relationship between the different values as a function of p. Notice that $\mathcal{L}_3, \mathcal{L}_2, \mathcal{L}_1$ and $\mathcal{L}_0$ converge as p approaches 1 and as it approaches $\frac{1}{2}$, regardless of the graph topology and ordering. This is exactly what we expect: if $p = 1$, the agents have perfect information from their private signals alone, while if $p = \frac{1}{2}$, then the private signals give the agents no extra information, and thus their the LR approaches $\frac{1}{2}$.

3.4 Optimal Ordering & Restrictions on p

We can now determine the optimal ordering by examining the learning rates derived in the previous section. First, we define an *induced ordering* below:

Definition 3.7. Let $\mathcal{A} : \chi \longrightarrow \{0, 1\}$ be any assignment of values to variables. Define the (partial) *ordering $\sigma(\mathcal{A})$ induced by $\mathcal{A}$* as follows: if $\mathcal{A}(x_i) = 1$, then cell $\mathbb{C}(x_i)$ is on; otherwise, $\mathbb{C}(x_i)$ is off.

In particular, the above definition gives a bijection between assignments and partial ordering over variables. We further say that a total ordering σ *respects* an assignment $\mathcal{A}$ (denoted by $\sigma \sim \mathcal{A}$) if it contains $\sigma(\mathcal{A})$ as a partial ordering over variable nodes. We also write $\sigma^*(\mathcal{A}) := \arg\max_{\sigma \sim \mathcal{A}} \mathcal{L}(\mathcal{N}, \sigma, \mu^B)$.

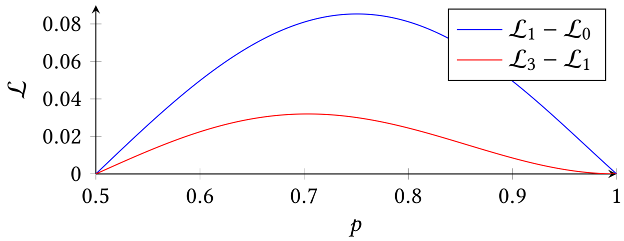

Figure 4: The plot of the numerator and the denominator of the condition on $M-1$ from Lemma 3.6. The denominator approaches zero noticably faster, suggesting the fraction approaches infinity as p goes to 1.

Definition 3.8. Let $\mathcal{A} : \chi \longrightarrow \{0,1\}$ be an assignment of values to variables maximizing the number of satisfied clauses. Then $\mathcal{A}$ is a *maximal assignment.*

Lemma 3.9 (Optimal Ordering). *Let $\mathcal{A}^*$ be a maximal assignment. Let $p(M) := (3M-4)/(3M-3)$ be a threshold probability determined by M, the number of clauses. Then for all $p \geq p(M)$, $\sigma^*(\mathcal{A}^*)$ is an optimal ordering.*

Proof. Note that an assignment-induced ordering only specifies whether each cell is on or off. From Lemma 3.5, an optimal cell ordering exists both if the cell is on or off, and yields the same learning rate in either case. Therefore, the optimal ordering is determined by comparing clause learning rates.

Also note that any total ordering must respect some assignment. So we argue the optimality of $\sigma^*(\mathcal{A}^*)$ by comparing it to $\sigma^*(\mathcal{A}')$ for any non-maximal $\mathcal{A}'$. Let S^* be the number of satisfied clauses under $\mathcal{A}^*$ and $S' < S^*$ that under $\mathcal{A}'$. By Lemma 3.6, for any clause satisfied under an arbitrary assignment $\mathcal{A}$, the corresponding clause gadget under an ordering $\sigma \sim \mathcal{A}$ achieves learning rate lower bounded by $\mathcal{L}_1$. Further, by Lemma 3.6, $\mathcal{L}_1 > \mathcal{L}_0$, so having more satisfied clauses in an assignment can never decrease the CLR under the induced ordering. However, note also that $\mathcal{L}_3 > \mathcal{L}_1$ (by Lemma 3.6). Hence, in the most extreme case, all S' satisfied clauses under $\mathcal{A}'$ are satisfied with three true literals, while all S^* satisfied clauses under $\mathcal{A}^*$ are satisfied with one true literal.

Consider that extreme case, and further impose the worst-case choices of S' and S^* by setting $S^* = M$ and $S' = M-1$. Then over all clause gadgets, $\sigma^*(\mathcal{A}^*)$ gives a CLR of $M\mathcal{L}_1$, $\sigma^*(\mathcal{A}')$ gives a CLR of $(M-1)\mathcal{L}_3 + \mathcal{L}_0$. We can now solve for conditions on p such that $\sigma^*(\mathcal{A}^*)$ achieves a higher network learning rate:

$$M\mathcal{L}_1 > (M-1)\mathcal{L}_3 + \mathcal{L}_0$$
$$\frac{\mathcal{L}_1 - \mathcal{L}_0}{\mathcal{L}_3 - \mathcal{L}_1} > M - 1.$$

The numerator and denominator is plotted in Figure 4 as a function of p. Since the left-hand side can be lower bounded by $\frac{1}{3-3p}$ for all $p \in (0.5, 1)$, it is sufficient to have $\frac{1}{3-3p} \geq M-1$. We can thus define $p(M) := \frac{3M-4}{3M-3}$, which respects the condition. Note that $p(M)$ is well-defined for any $M \geq 2$, and lies in the interval $(\frac{1}{2}, 1)$. □

To recap, given any instance φ of 3-SAT with N variables and M clauses, we can construct a network G_φ with ground truth prior $q = 1/2$ and signal accuracy $p = p(M)$. In particular, whenever φ is satisfiable under some maximal assignment $\mathcal{A}^*$, there is an optimal decision ordering σ^* which respects the ordering induced by $\mathcal{A}^*$, and which achieves a network CLR of at least $N(2p + 3p^2 - 2p^3) + M\mathcal{L}_1$. Otherwise, if φ is non-satisfiable under any maximal assignment $\mathcal{A}^*$, then the optimal decision ordering σ^* respecting the ordering induced by $\mathcal{A}^*$ achieves a network CLR of no more than $N(2p + 3p^2 - 2p^3) + (M-1)\mathcal{L}_3 + \mathcal{L}_0$. Choosing $p = p(M)$ allowed us to show the optimality of σ^*, as well as to separate the learning rates in networks corresponding to satisfiable and non-satisfiable formulas. All that remains to complete the reduction is to pick an appropriate choice of ε, such that a formula graph G_φ achieves expected network learning rate greater than ε under an optimal ordering iff φ is satisfiable.

3.5 Picking the Epsilon

We will simply pick ε to lie exactly halfway between the satisfiable and non-satisfiable network learning rates. Recalling that each variable and clause gadget contains 3 vertices, we can compute the learning rates below. For networks corresponding to satisfiable formulas, we have $\frac{N\mathcal{L}_{\text{cell}} + M\mathcal{L}_1}{3(N+M)}$, and for networks corresponding to non-satisfiable formulas, we have $\frac{N\mathcal{L}_{\text{cell}} + (M-1)\mathcal{L}_3 + \mathcal{L}_0}{3(N+M)}$. This gives

$$\varepsilon = \frac{1}{2}\left(\frac{2N\mathcal{L}_{\text{cell}} + M\mathcal{L}_1 + (M-1)\mathcal{L}_3 + \mathcal{L}_0}{3(N+M)}\right),$$

thus completing the reduction.

3.6 Generalization for a constant range of p

Finally, we remark that, while this construction proves the Theorem, it uses p approaching 1 as the size of φ increases. A natural question, then, is whether it is hard to compute the maximum learning rate for other values of p, especially when p is a fixed constant. In fact, Section 5 shows a stronger version of Theorem 3.1, stated as follows.

Theorem 3.10. *Network Learning with Bayesian learning rule $\mu = \mu^B$ and a fixed $p \in (\sqrt{7/8}, 1)$ is NP-hard.*

This follows as a direct corollary of Theorem 5.1.

4 PROOF OF NP-HARDNESS FOR MAJORITY DYNAMICS

In this section, we adapt the statement of Theorem 3.1 to the Majority dynamics setting.

Theorem 4.1. *Network Learning with the majority vote rule $\mu = \mu^M$ is NP-hard.*

The proof is again a reduction from 3-SAT. The main idea is identical to that of Theorem 3.1, so we offer here only the main points and construction, with emphasis on differences from the proof of Theorem 3.1. We offer the full proof in Appendix B.

4.1 Adapted Graph Construction

Unfortunately, if we wanted to directly apply the previous construction, we would find that the worst-case satisfied ordering reaches a *lower* learning rate than the best-case unsatisfied ordering (as discussed in Section 3.4). Then there is no ε which separates the

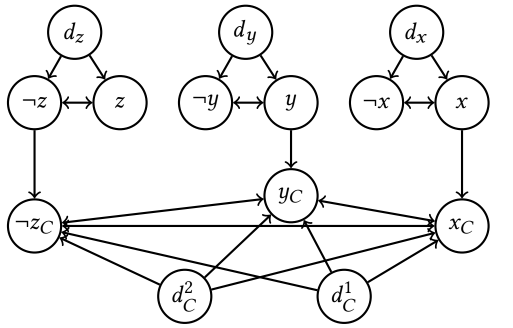

Figure 5: The graph G_φ for $\varphi = C = (x \vee y \vee \neg z)$.

learning rates corresponding to satisfied and unsatisfied assignments. We thus adapt the construction from Section 3.2, modifying the clause gadget. We keep the variable cell unchanged.

Definition 3.2 (Variable cell). Let x be a variable of a formula φ. A *variable cell* of x is a directed graph $\mathbb{C}(x) = (V_x, E_x)$, where

(1) $V_x = \{x, \neg x, d_x\}$, and
(2) $E_x = \{(d_x, x), (d_x, \neg x), (x, \neg x), (\neg x, x)\}$.

Intuitively, we need to give the nodes in the clause gadgets more input, so they are better equipped to use the input from the "on" cells. To achieve this, we add two dummy nodes to the clause gadget.

Definition 4.2 (Clause gadget). Let $C = j \vee k \vee \ell$ be a clause of a 3-CNF formula φ, where $j \neq k \neq \ell$ are some literals. Then the *clause gadget* is $\mathbb{G}(C) = (V_C, E_C)$, where

(1) $V_C = \{j, k, \ell, d^1, d^2\}$,
(2) $E_C = \{(x, y) \mid x, y \in \{j, k, \ell\}, x \neq y\} \cup \{(d^i, x) \mid i \in \{1, 2\}, x \in \{j, k, \ell\}\}$.

We now define the formula graph using the same definition, only with the new clause gadget.

Definition 3.4 (Formula graph). Let $\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_M$ be a CNF formula of variables $\chi = \{x_1, \ldots, x_N\}$, where each clause C_i is the disjunction of exactly three literals. Then G_φ is a disjoint union of the graphs $\mathbb{C}(x_i)$ for all $x_i \in \chi$, and $\mathbb{G}(C_i)$ for all clauses $C_i \in \varphi$. Additionally, there is an edge to each of the vertices in every $\mathbb{G}(C_i)$ from the corresponding literal nodes in the respective variable cell.

See Figure 5 for an illustration of this construction for the simple formula $\varphi = x \vee y \vee z$. Note that we also use the "on"/"off" states of a variable/literal, as defined for the Bayesian proof (see Section 3.2).

Next, we compute the learning rates of the variable cell, and of the clause gadgets, depending on whether its literals are on or off.

Lemma 4.3 (Majority Dynamics Cell LR). *Let $x \in \chi$. Let $q = \frac{1}{2}$, and p be given. Then*

$$\mathcal{L}^*(\mathbb{C}(x)) = 2p + 3p^2 - 2p^3.$$

Lemma 4.4 (Majority Dynamics Gadget Learning Rate). *Let C be a clause. Suppose that σ^* is an optimal learning rate. Then, in the gadget for C, σ^* places the cells first, then the dummy nodes, and finally the three literal nodes. Further,*

(1) if one literal is on, then $\mathcal{L}(\mathbb{G}(C), \sigma^*)$ is

$$\mathcal{L}_1 := p\left(2 + 2p + 6p^2 + 11p^3 + 4p^4 - 51p^5 - 6p^6 + 21p^7 + 115p^8 - 136p^9 + 13p^{10} + 36p^{11} - 12p^{12}\right).$$

(2) if one literal is on, then $\mathcal{L}(\mathbb{G}(C), \sigma^*)$ is

$$\mathcal{L}_2 := p\left(12p^8 - 54p^7 + 76p^6 - 14p^5 - 40p^4 + 9p^3 + 12p^2 + 2p + 2\right).$$

(3) if one literal is on, then $\mathcal{L}(\mathbb{G}(C), \sigma^*)$ is

$$\mathcal{L}_3 := p\left(2 + 2p + 3p^2 + 14p^3 + 22p^4 - 66p^5 - 69p^6 + 310p^7 - 688p^8 + 710p^9 + 756p^{10} - 2581p^{11} + 2304p^{12} - 558p^{13} - 372p^{14} + 264p^{15} - 48p^{16}\right).$$

Furthermore, for a clause D satisfied under σ^*, it holds

$$\mathcal{L}_3 \geq \mathcal{L}(\mathbb{G}(D), \sigma^*) \geq \mathcal{L}_1 \geq \mathcal{L}_0.$$

Finally, we can now determine the optimal ordering, compute its learning rate, and show that there is an ε such that the learning rate is above ε if and only if the induced ordering is satisfied.

Lemma 4.5 (Optimal Ordering). *Let $\mathcal{A}^*$ be a maximal assignment. Let $p(M) < 1$ be a threshold probability determined by M, the number of clauses. Then for all $p \geq p(M)$, the decision ordering σ^* which places all dummy nodes first, then all variable nodes respecting the partial ordering induced by $\mathcal{A}^*$, and finally all literal nodes in the clause gadgets, maximizes the network learning rate.*

The proof of this Lemma is very similar that of Lemma 3.9, and is included in Appendix B.

We now apply the same reasoning as in Section 3.5. If $p = p(M)$, then the worst-case learning rate of a satisfying assignment is $\frac{N\mathcal{L}_{\text{cell}} + M\mathcal{L}_1}{3N+5M}$, strictly higher than the best-case non-satisfying LR, $\frac{N\mathcal{L}_{\text{cell}} + (M-1)\mathcal{L}_3 + \mathcal{L}_0}{3N+5M}$. We can thus define the threshold (mind the new number of vertices—G_φ now has 5 for each clause):

$$\varepsilon = \frac{1}{2}\left(\frac{2N\mathcal{L}_{\text{cell}} + M\mathcal{L}_1 + (M-1)\mathcal{L}_3 + \mathcal{L}_0}{3N + 5M}\right).$$

This concludes the proof.

5 APPROXIMATING THE OPTIMAL LR

In this section, we extend the ideas from the 3-SAT reduction to show that even approximating a solution to Network Learning is hard. More precisely, we show hardness for the optimization version of the problem, defined in Section 2 as Opt Network Learning. We give a proof for the Bayesian learning rule; we believe for the proof for majority vote to be similar.

Theorem 5.1. *Opt Network Learning with the Bayesian inference $\mu = \mu^B$ is APX-hard.*

Proof. We perform a PTAS reduction to Max 3-SAT, where again each clause has exactly three literals (also referred to as Max E3-SAT), which is known to be APX-hard. In particular, we show that an approximation scheme of $\overline{\mathcal{L}}^*(\mathcal{N})$ implies an approximation scheme of Max 3-SAT.

Given an instance of Max 3-SAT, we construct the graph G_φ from Definition 3.4. Let $\mathcal{A}^*$ be a maximal assignment, defined in Definition 3.8, satisfying M^* clauses. To prove the theorem, it suffices to show that for every $\delta \in (0,1)$, there is an $\alpha \in (0,1)$ such that if an ordering σ achieves a LR which is α-close to the optimum, then the induced assignment $\mathcal{A}(\sigma)$ satisfies δM^* clauses. Abusing notation, for an assignment $\mathcal{A}$ we denote $\mathcal{L}(\mathcal{A}) := \mathcal{L}(\mathcal{N}, \sigma^*(\mathcal{A}))$.

Note that for any 3-CNF formula, assigning truth values independently and uniformly at random satisfies $\frac{7}{8}$ of the clauses in expectation, implying that $M^* \geq \frac{7}{8}M$ [12]. From Lemma 3.6, it follows that unsatisfied clauses receive optimal CLR $\mathcal{L}_0$, and satisfied clauses receive at least $\mathcal{L}_1$ and at most $\mathcal{L}_3$. It then follows that

$$\mathcal{L}(\mathcal{A}^*) \geq M^*\mathcal{L}_1 + (M - M^*)\mathcal{L}_0 + N\mathcal{L}_{\text{cell}}.$$

Let $\mathcal{A}'$ be an assignment which achieves $\overline{\mathcal{L}}(\mathcal{A}') \geq \overline{\mathcal{L}}^*(\mathcal{N}) - \varepsilon \geq \overline{\mathcal{L}}(\mathcal{A}^*) - \varepsilon$ for some $\varepsilon > 0$ specified later. Multiplying by the number of vertices in G_φ, we get

$$\mathcal{L}(\mathcal{A}') \geq \mathcal{L}(\mathcal{A}^*) - \varepsilon(3M + 3N) \geq \mathcal{L}(\mathcal{A}^*) - 6\varepsilon M,$$

where the second inequality holds since WLOG $N \leq M$, by duplication of clauses. Re-arranging, we have that the number of satisfied clauses under $\mathcal{A}'$ is

$$M' \geq \frac{\mathcal{L}_1 - \mathcal{L}_0}{\mathcal{L}_3 - \mathcal{L}_0} M^* - \frac{6\varepsilon M}{\mathcal{L}_3 - \mathcal{L}_0} \geq M^* \left(\frac{\mathcal{L}_1 - \mathcal{L}_0}{\mathcal{L}_3 - \mathcal{L}_0} - \frac{48\varepsilon}{7(\mathcal{L}_3 - \mathcal{L}_0)} \right),$$

where the second inequality holds since $M^* \geq \frac{7}{8}M$. If we now set the final expression equal to δM^*, we get an equality in ε and p. Solving for ε, we get

$$\varepsilon = \frac{7}{48}((\mathcal{L}_1 - \mathcal{L}_0) - \delta(\mathcal{L}_3 - \mathcal{L}_0)). \tag{1}$$

This is a function of p and δ. We only require that this $\varepsilon > 0$. Using that $p \in (\frac{1}{2}, 1)$, this gives us

$$\delta < \frac{4p^4 - 8p^3 - 4p^2 + 8p + 2}{4p^8 - 16p^7 + 13p^6 + 17p^5 - 12p^4 - 23p^3 + 5p^2 + 12p + 2}.$$

On $p \in (\frac{1}{2}, 1)$, the RHS can be strictly lower-bounded by p^2. It thus suffices to set $p \geq \sqrt{\delta}$.

This proves that an approximation to Opt Network Learning within an additive bound ε for any $p \geq \sqrt{\delta}$ implies an approximation to Max 3-SAT within a factor of δ. Converting ε to a multiplicative bound, since $\overline{\mathcal{L}}^* \in (0,1)$, we get

$$\overline{\mathcal{L}}^* - \varepsilon = (1 - \frac{\varepsilon}{\overline{\mathcal{L}}^*})\overline{\mathcal{L}}^* \leq (1 - \varepsilon)\overline{\mathcal{L}}^*.$$

Thus we also have a multiplicative bound $\alpha = (1 - \varepsilon)$ for each δ, which is what we wanted to prove. □

Corollary 5.2. *For any fixed constant $p \in (\sqrt{7/8}, 1)$, an α-approximation of Opt Network Learning with Bayesian inference rule is NP-hard for every $\alpha > \alpha(p)$, defined as*

$$\alpha(p) := 1 - \frac{7}{48}((\mathcal{L}_1 - \mathcal{L}_0) - \frac{7}{8}(\mathcal{L}_3 - \mathcal{L}_0)).$$

Proof. This follows from Theorem 5.1, since approximating Max 3-SAT is NP-hard for $\frac{7}{8} + \xi$, where $\xi > 0$ [12]. The condition of $\alpha(p)$ is achieved by substituting $\delta = \frac{7}{8}$ to Equation (1). □

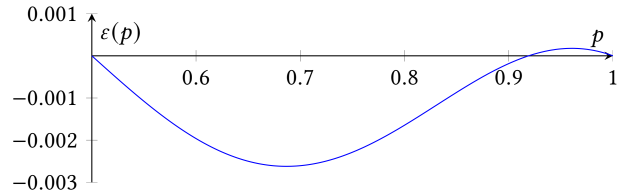

Figure 6: The value of ε below which approximation is hard (i.e., $\delta = \frac{7}{8}$ [12]). The requirement of $\varepsilon > 0$ gives us $p \geq \sqrt{7/8}$.

6 CONCLUSION

In this paper, we tackle the *complexity* of judging how well-equipped a given network is for social truth learning in the setting of sequential decision-making by agents with bounded belief. We then show that it is NP-hard to decide whether a large proportion of the network successfully learns a binary ground truth, both when agents are fully rational and when agents have bounded rationality. Finally, we show that it is actually hard to even approximate the learning rate of a fully rational network.

Future Work

There are many open directions for future work. A natural one is whether this problem belongs to NP. Namely, it remains open whether there exist efficiently verifiable characterizations of networks achieving high learning rates. We conjecture that the answer is no, but we have so far been unable to prove it.

Yet another interesting direction is connecting Network Learning and other combinatorial problems. Network Learning seems to be somewhat connected to finding big independent sets, in which information can be aggregated, along with big enough cliques in which information cascades may happen. Clarifying the balance between these two contributing factors could improve our understanding of both network learning and its relation to the rest of combinatorics.

Finally, it would be fascinating to look for classes of networks for which Network Learning is easy. A full characterization is impossible by our results, but identifying families of graphs achieving high network learning rates would be useful.

ACKNOWLEDGMENTS

Part of this work was carried out while Amanda Wang and Filip Úradník were participants in the DIMACS REU program at Rutgers University. Amanda Wang was supported by the NSF award CNS-2150186, the REU supplement to NSF 2208663, and by Princeton University's Office of Undergraduate Research Undergraduate Fund for Academic Conferences through the Hewlett Foundation Fund. Filip Úradník was supported by CoSP, a project funded by European Union's Horizon 2020 research and innovation programme, grant agreement No. 823748, by the Czech Science Foundation Grant 25-15714S, and by the Charles University Grant Agency (GAUK 206523). Jie Gao would like to acknowledge funding through NSF IIS-2229876, DMS-2220271, DMS-2311064, CCF-2208663, CCF-2118953. The authors would also want to thank Kevin Lu, Júlia Križanová and Rhett Olson for their useful discussion and insightful comments.

REFERENCES

[1] Daron Acemoglu, Munther A Dahleh, Ilan Lobel, and Asuman Ozdaglar. 2011. Bayesian Learning in Social Networks. *Rev. Econ. Stud.* 78, 4 (March 2011), 1201–1236.

[2] Daron Acemoglu and Asuman Ozdaglar. 2011. Opinion dynamics and learning in social networks. *Dyn. Games Appl.* 1, 1 (March 2011), 3–49.

[3] Itai Arieli, Fedor Sandomirskiy, and Rann Smorodinsky. 2021. On Social Networks That Support Learning. In *Proceedings of the 22nd ACM Conference on Economics and Computation* (Budapest, Hungary) *(EC '21)*. 95–96.

[4] Gal Bahar, Itai Arieli, Rann Smorodinsky, and Moshe Tennenholtz. 2020. Multi-issue social learning. *Math. Soc. Sci.* 104 (March 2020), 29–39.

[5] A V Banerjee. 1992. A simple model of herd behavior. *Q. J. Econ.* 107, 3 (Aug. 1992), 797–817.

[6] Sushil Bikhchandani, David Hirshleifer, and Ivo Welch. 1992. A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades. *J. Polit. Econ.* 100, 5 (Oct. 1992), 992–1026.

[7] Christophe Chamley. 2004. *Rational Herds: Economic Models of Social Learning*. Cambridge University Press.

[8] Gregory F Cooper. 1990. The computational complexity of probabilistic inference using bayesian belief networks. *Artif. Intell.* 42, 2-3 (March 1990), 393–405.

[9] Paul Dagum and Michael Luby. 1993. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artif. Intell.* 60, 1 (March 1993), 141–153.

[10] Benjamin Golub and Evan Sadler. 2016. Learning in social networks. In *The Oxford Handbook of the Economics of Networks*. Oxford University Press, 504–542.

[11] Jakub Hązła, Ali Jadbabaie, Elchanan Mossel, and Mohammad Ali Rahimian. 2019. Reasoning in Bayesian opinion exchange networks is PSPACE-hard. In *Conference on Learning Theory*. PMLR, 1614–1648. https://proceedings.mlr.press/v99/hazla19a.html

[12] Johan Håstad. 2001. Some optimal inapproximability results. *J. ACM* 48, 4 (July 2001), 798–859.

[13] Jan Hązła, Ali Jadbabaie, Elchanan Mossel, and M Amin Rahimian. 2021. Bayesian Decision Making in Groups is Hard. *Oper. Res.* 69, 2 (March 2021), 632–654.

[14] Ali Jadbabaie, Pooya Molavi, Alvaro Sandroni, and Alireza Tahbaz-Salehi. 2012. Non-Bayesian social learning. *Games Econ. Behav.* 76, 1 (Sept. 2012), 210–225.

[15] Johan Kwisthout. 2018. Approximate inference in Bayesian networks: Parameterized complexity results. *Int. J. Approx. Reason.* 93 (Feb. 2018), 119–131.

[16] Kevin N Laland. 2004. Social learning strategies. *Learn. Behav.* 32, 1 (Feb. 2004), 4–14.

[17] Kevin Lu, Jordan Chong, Matt Lu, and Jie Gao. 2024. Enabling Asymptotic Truth Learning in a Social Network. In *Proceedings of the 20th Conference on Web and Internet Economics (WINE'24)*.

[18] Markus Mobius and Tanya Rosenblat. 2014. Social Learning in Economics. *Annu. Rev. Econom.* 6, 1 (Aug. 2014), 827–847.

[19] Elchanan Mossel, Joe Neeman, and Omer Tamuz. 2014. Majority dynamics and aggregation of information in social networks. *Auton. Agent. Multi. Agent. Syst.* 28, 3 (May 2014), 408–429.

[20] Elchanan Mossel and Omer Tamuz. 2017. Opinion exchange dynamics. *Probab. Surv.* 14, none (Jan. 2017), 155–204.

[21] Daniel Sgroi. 2002. Optimizing Information in the Herd: Guinea Pigs, Profits, and Welfare. *Games Econ. Behav.* 39, 1 (April 2002), 137–166.

[22] Yoav Shoham and Moshe Tennenholtz. 1992. Emergent conventions in multi-agent systems: initial experimental results and observations (preliminary report). In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*. 225–231.

[23] Lones Smith. 1991. *Essays on dynamic models of equilibrium and learning*. Ph.D. Dissertation. University of Chicago.

[24] Lones Smith and Peter Sørensen. 2000. Pathological Outcomes of Observational Learning. *Econometrica* 68, 2 (March 2000), 371–398.

[25] Haotian Wang, Feng Luo, and Jie Gao. 2022. Co-evolution of opinion and social tie dynamics towards structural balance. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 3362–3388.

[26] Ivo Welch. 1992. Sequential sales, learning, and cascades. *J. Finance* 47, 2 (June 1992), 695–732.