Local non-Bayesian social learning with stubborn agents

Daniel Vial, Vijay Subramanian, *Senior Member, IEEE*

***Abstract*— We study a social learning model in which agents iteratively update their beliefs about the true state of the world using private signals and the beliefs of other agents in a non-Bayesian manner. Some agents are stubborn, meaning they attempt to convince others of an erroneous true state (modeling fake news). We show that while agents learn the true state on short timescales, they "forget" it and believe the erroneous state to be true on longer timescales. Using these results, we devise strategies for seeding stubborn agents so as to disrupt learning, which outperform intuitive heuristics and give novel insights regarding vulnerabilities in social learning.**

I. Introduction

With the rise of social networks, people increasingly receive news through non-traditional sources. One recent study shows that two-thirds of American adults have gotten news through social media [1]. Such news sources are fundamentally different than traditional ones like print media and television, in the sense that social media users read and discuss news on the same platform. As a consequence, users turning to these platforms for news receive information not only from major publications but from others users as well; in the words of [2], a user "with no track record or reputation can in some cases reach as many readers as Fox News, CNN, or the New York Times." This phenomenon famously reared its head during the 2016 United States presidential election when fake news stories were shared tens of millions of times [2], and it has remained a critical issue in the years since [3].

In this paper, we study a mathematical model describing this situation. The model includes a set of agents attempting to learn the true state of the world (e.g. which of two candidates is better suited for office). Each agent iteratively updates its belief (i.e. its distribution over possible states) in a manner similar to the non-Bayesian social learning model of [4] using information from three sources. First, each agent receives noisy observations of the true state, modeling e.g. news stories. Second, each agent observes the beliefs of a subset of other agents, modeling e.g. discussions with other social media users. Third, each agent may observe the beliefs of *stubborn agents* or *bots* who aim to persuade others of an erroneous true state, modeling e.g. users spreading fake news.[1] This process continues iteratively until a finite learning horizon.

Under this model, two competing forces emerge as the learning horizon grows. On the one hand, agents receive more observations of the true state, which help them learn. On the other hand, the beliefs of the bots gradually propagate through the system, suggesting that agents become increasingly exposed to bots and thus less likely to learn. Hence, while the horizon clearly affects the learning outcome, the nature of this effect – namely, whether learning becomes more or less likely as the horizon grows – is less clear.

This effect of the learning horizon has often been ignored in works with models similar to ours. For example, our model is nearly identical to that in the empirical work [6], in which the authors show that polarized opinions can arise when there are two types of bots with diametrically opposed viewpoints. However, the experiments in [6] simply fix a large learning horizon and do not consider the effect of varying it. Models similar to ours have also been treated analytically in e.g. [4], [7]–[9], but these works consider infinite horizons and/or cooperative settings (i.e. no stubborn agents). See Section V for details on these (and other) works.

In the first part of the paper (see Section III), we argue that *the learning horizon plays a prominent role when stubborn agents are present and should not be ignored*. In particular, we show that the learning outcome depends on the relationship between the horizon T_n and a quantity p_n that describes the "density" of bots in the system, where both quantities may vary with the number of agents n. Mathematically, letting $\theta \in (0,1)$ denote the true state and $\theta_{T_n}(i^*)$ the mean of the belief (hereafter, the *estimate*) for a uniformly random agent i^* at the horizon T_n, we show (see Theorem 1)[2]

$$\theta_{T_n}(i^*) \xrightarrow[n\to\infty]{\mathbb{P}} \begin{cases} \theta, & T_n(1-p_n) \xrightarrow[n\to\infty]{} 0 \\ 0, & T_n(1-p_n) \xrightarrow[n\to\infty]{} \infty \end{cases}. \tag{1}$$

Here p_n is smaller when more bots are present and 0 is the erroneous true state promoted by the bots. Hence, in words, (1) says the following: if there are sufficiently few bots, in the sense that $T_n(1-p_n) \to 0$, i^* learns the true state; if there are sufficiently many bots, in the sense that $T_n(1-p_n) \to \infty$, i^* adopts the extreme estimate 0 promoted by the bots.

We are grateful for financial support from NSF via grants ECCS:1603861, ECCS:2038416, CNS:1955777 and CCF:2008130. D. Vial is with the University of Texas, Austin, TX (email: dvial@utexas.edu). V. Subramanian is with the University of Michigan, Ann Arbor, MI (email: vsubram@umich.edu).

[1]The term *stubborn agents* has been used in the literature to describe such agents; the term *bots* is used in reference to automated social media accounts spreading fake news while masquerading as real users [5].

[2]The theorem also addresses the case $\lim_{n\to\infty} T_n(1-p_n) \in (0,\infty)$.

Additionally, we show the belief of i^* converges to a Dirac measure in a certain sense (see Corollary 1).

We note the result in (1) assumes a particular random graph model for the social network connecting agents and bots (a modification of the so-called *directed configuration model*). For such models, *phase transitions* – wherein small changes to model parameters lead to starkly different behaviors – are often observed. In this case, assuming $T_n = (1-p_n)^{-k}$ for some $k > 0$, and also assuming $p_n \to 1$, the learning outcome suddenly drops from θ to 0 as k changes from e.g. 0.99 to 1.01. Put differently, agents initially (at time $(1-p_n)^{-0.99}$) learn the true state, then suddenly (at time $(1-p_n)^{-1.01}$) "forget" the true state and adopt the extreme estimate 0. Hence, we show the horizon can lead to drastically different outcomes. We also note proving (1) involves analyzing hitting probabilities for random walks on random graphs with absorbing states (bots in our setting), which may be of independent interest.

In the second part of the paper (see Section IV), we study a setting in which an adversary chooses how many bots to connect to each agent, subject to a budget constraint. The adversary would like to minimize $\theta_{T_n}(i^*)$ (i.e. to convince agents of the erroneous state 0), but this quantity depends on the graph topology, which is not publicly available for social networks like Twitter. Hence, motivated by (1), we formulate the adversary's problem as minimizing p_n, which only depends on the *degrees* in the graph – e.g. number of followers on Twitter, which is publicly available. We clarify that $\theta_{T_n}(i^*)$ is monotone in p_n only as $n \to \infty$ for the random graph of Section III (see Theorem 1). Thus, we use p_n as a tractable (albeit nonrigorous) surrogate for the true objective function $\theta_{T_n}(i^*)$, and we show empirically that these quantities are closely correlated for real social networks (see Figure 2). Alternatively, given a target $\theta_{T_n}(i^*)$, we can minimize the horizon T_n when this target estimate is reached. However, we view T_n as fixed and thus do not pursue this dual problem.

Minimizing p_n amounts to solving an integer program, which can be done in polynomial time owing to the structure of p_n. However, the computational complexity is $\Omega(n^2)$, which is infeasible for social networks like Twitter. Thus, we propose a randomized approximation algorithm that runs in time $n \log n$ and that produces a constant-fraction approximation of the optimal solution with high probability (see Theorem 2). Moreover, whereas the logic of the optimal solution is somewhat opaque, the form of our approximate solution offers the interpretation that *successful adversaries carefully balance agents' influence and susceptibility to influence*. For a social network like Twitter, this means targeting users with many followers (i.e. influential users) who follow very few users themselves, so that fake news will occupy a greater portion of the targeted users' feeds. While somewhat intuitive, the precise form of the randomized scheme is far from obvious. Furthermore, empirical results show that our scheme disrupts learning to a larger extent than schemes that more obviously balance influence and susceptibility. Thus, we believe our analysis provides new insights into vulnerabilities of news sharing platforms and non-Bayesian social learning models.

The paper is organized as follows. In Section II, we define our learning model. Sections III and IV follow the outline above. We discuss related work in Section V. Proof details are deferred to the full version of the paper, [10]. Preliminary versions of the paper appeared in abstracts [11], [12].

Notational conventions: The following notation is used frequently. For $k \in \mathbb{N}$, we let $[k] = \{1, \dots, k\}$, and for $k, k' \in \mathbb{N}$ we let $[k] + k' = k' + [k] = \{1 + k', \dots, k + k'\}$. All vectors are treated as row vectors. We let e_i denote the vector with 1 in the i-th position and 0 elsewhere. We denote the set of nonnegative integers by $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. We use $1(A)$ for the indicator function, i.e. $1(A) = 1$ if A is true and 0 otherwise. All random variables are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with $\mathbb{E}[\cdot] = \int_\Omega \cdot \, d\mathbb{P}$ denoting expectation, $\xrightarrow{\mathbb{P}}$ denoting convergence in probability, and $a.s.$ meaning $\mathbb{P}-$almost surely.

II. Learning model

We begin by defining the model of social learning studied throughout the paper. The basic ingredients are (1) a true state of the world, (2) a social network connecting two sets of nodes, some who aim to learn the true state and some who wish to persuade others of an erroneous true state, and (3) a learning horizon. We discuss each in turn.

The true state of the world is a constant $\theta \in (0, 1)$. For example, in an election between candidates representing two political parties (say, Party 1 and Party 2), $\theta \approx 0$ and $\theta \approx 1$ means the Party 1 and 2 candidates are superior, respectively. We emphasize that θ is a deterministic constant and depends neither on time, nor on the number of nodes in the system.

A directed graph $G = (A \cup B, E)$ connects disjoint sets of nodes A and B. We refer to elements of A as *regular agents*, or simply *agents*, and elements of B as *stubborn agents* or *bots*. While agents attempt to learn the true state θ, bots aim to disrupt this learning and convince agents that the true state is instead 0. In the election example, agents represent voters who study the two candidates to learn which is superior, while bots are loyal to Party 1 and aim to convince agents that the corresponding candidate is superior (despite possible evidence to the contrary). Edges in the graph represent connections in a social network over which nodes share beleifs in a manner that will be described shortly. An edge $j \to i$ means that i observes j's belief. Let $N_{in}(i) = \{j \in A \cup B : j \to i \in E\}$ and $d_{in}(i) = |N_{in}(i)|$; we assume $N_{in}(i) \neq \emptyset$.

Agents and bots share beliefs until a learning horizon $T \in \mathbb{N}$. We will allow the horizon to depend on the number of agents $n \triangleq |A|$ and will thus denote it by T_n at times. In the election example, T represents the duration of the election, i.e. the number of time units that agents can learn about the candidates and that bots can attempt to convince agents of the superiority of the Party 1 candidate.

Given these basic ingredients, we can define the learning process. At time $t = 0$, agent $i \in A$ has a Beta$(\alpha_0(i), \beta_0(i))$ belief, where $\alpha_0(i) \in (0, \bar{\alpha}]$ and $\beta_0(i) \in (0, \bar{\beta}]$ for some $\bar{\alpha}, \bar{\beta} \in (0, \infty)$ that do not depend on n. For each $t \in [T]$, i receives the signal $s_t(i) \sim$ Bernoulli(θ). In the absence of a network, the Bayesian approach dictates that i update its parameters to $\alpha_t(i) = \alpha_{t-1}(i) + s_t(i)$ and $\beta_t(i) = \beta_{t-1}(i) + (1 - s_t(i))$ and its belief to $\mu_t(i) =$ Beta$(\alpha_t(i), \beta_t(i))$, namely,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2022.3154679, IEEE Transactions on Control of Network Systems

for any (measurable) $\mathcal{A} \subset [0,1]$,

$$\mu_t(i)(\mathcal{A}) \propto \int_{x\in\mathcal{A}} x^{\alpha_t(i)-1}(1-x)^{\beta_t(i)-1}dx.$$

In our running example, $\alpha_t(i)$ and $\beta_t(i)$ represent the number of news stories favorable to respective parties that i has read during the election, plus some prior parameters $\alpha_0(i)$ and $\beta_0(i)$ that account for i's biases from before the election. As t grows, the belief $\mu_t(i)$ converges to a Dirac measure on its mean $\theta_t(i) = \alpha_t(i)/(\alpha_t(i)+\beta_t(i))$; intuitively, i becomes increasingly confident that the true state is the fraction of stories favorable to a certain party.

In the presence of a network, we proceed in the same manner, except the parameters are updated as follows:

$$\alpha_t(i) = (1-\eta)(\alpha_{t-1}(i) + s_t(i)) + \sum_{j\in N_{in}(i)} \frac{\eta\alpha_{t-1}(j)}{d_{in}(i)}, \quad (2)$$

$$\beta_t(i) = (1-\eta)(\beta_{t-1}(i) + 1 - s_t(i)) + \sum_{j\in N_{in}(i)} \frac{\eta\beta_{t-1}(j)}{d_{in}(i)},$$

where $\eta \in (0,1)$. Intuitively, i reads the news and calculates its favorability of the parties as before, then discusses with its neighbors to update its favoribility. Mathematically, i performs a Bayesian parameter update and then averages parameters. [6] uses the same update, whereas agents in [4] do Bayesian *belief* updates and then average *beliefs*. We study the former mainly for tractability. Our update also resembles the deGroot model [13], with the key difference being that we consider sequences of signals (to model a sequence of news stories). See Section V for details on related work.

Finally, we specify bot behavior. For $i \in B$, we set $N_{in}(i) = \{i\}$, $\alpha_0(i) = 0$, $\beta_0(i) = \bar{\beta}$, and $s_t(i) = 0 \ \forall\ t \in [T]$, then iteratively define $\{\alpha_t(i), \beta_t(i)\}_{t=1}^T$ via (2). More explicitly, a simple inductive proof shows

$$\alpha_t(i) = 0, \quad \beta_t(i) = \bar{\beta} + (1-\eta)t \quad \forall\ t \in [T]. \quad (3)$$

In our running example, $\alpha_0(i) = 0$, $\beta_0(i) = \bar{\beta}$, and $s_t(i) = 0$ means i's prior parameters and signals are maximally biased toward Party 1. Furthermore, we can interpret $N_{in}(i) = \{i\}$ as bots being "echo chambers" who only listen to themselves. Finally, note that since all bots $i \in B$ have the same behavior, we assume (without loss of generality) that the outgoing neighbor set of $i \in B$ is $N_{out}(i) = \{i, i'\}$ for some $i' \in A$, i.e. in addition to its self-loop, each bot has a single outgoing neighbor from the agent set.

III. Learning Outcome

To begin our analysis of the learning outcome, we show when all agents are (pathwise) connected to bots, their beliefs converge to those of the bots. Formally, for $p \geq 1$, let

$$W_p(\mu,\nu) = \inf_{(X,Y):X\sim\mu,Y\sim\nu}(\mathbb{E}|X-Y|^p)^{1/p}$$

denote the p-Wasserstein distance for probability measures μ and ν, where $X \sim \mu, Y \sim \nu$ means X and Y have respective marginals μ and ν. For $x \in [0,1]$, let δ_x denote the Dirac measure $\delta_x(\mathcal{A}) = 1(x \in \mathcal{A})$ for measurable $\mathcal{A} \subset [0,1]$. We then have the following (see [10, Appendix V] for a proof).

Proposition 1: Suppose that for any $i \in A$, there exists $l \in \mathbb{N}$ and $(i_\tau)_{\tau=0}^l \in (A \cup B)^{l+1}$ such that $i_0 = i$, $i_{\tau-1} \to i_\tau \in E\ \forall\ \tau \in [l]$, and $i_l \in B$. Then for any $i \in A$ and $p \geq 1$, $\lim_{t\to\infty}\theta_t(i) = \lim_{t\to\infty} W_p(\mu_t(i), \delta_0) = 0\ a.s.$

Hence, for a large enough horizon, estimates and beliefs become arbitrarily close to zero. A natural follow-up question is how such a horizon scales – and in which graph parameters – for a sequence of graphs $\{G_n\}_{n=1}^\infty$. In general, this scaling has a complicated dependence on the specific graphs chosen. To ensure a tractable characterization, we thus restrict attention to a particular random graph model, namely, a *directed configuration model* (DCM) with bots. The DCM constructs a graph with prespecified degrees, which, conditioned on being simple (i.e. having no self-loops or multi-edges), is uniformly distributed among (simple) graphs of those degrees [14, Proposition 7.15]. Thus, our analysis is "average-case" over graphs of given degrees. In Section IV, we will exploit the resulting insights empirically for more general graphs.

Having motivated our study of the DCM, we define it in Section III-A, present our main result for the DCM in Section III-B, and discuss our assumptions in Section III-C.

A. Graph model

To begin, we realize a sequence $\{d_{out}(i), d_{in}^A(i), d_{in}^B(i)\}_{i\in A}$ called the *degree sequence* from some distribution; here we let $A = [n]$. In the construction described next, $i \in A$ will have $d_{out}(i)$ outgoing neighbors (i will be observed by $d_{out}(i)$ other agents), $d_{in}^A(i)$ incoming neighbors from the A (i will observe $d_{in}^A(i)$ agents), and $d_{in}^B(i)$ incoming neighbors from B (i will observe $d_{in}^B(i)$ bots). Here the total in-degree of i is $d_{in}(i) = d_{in}^A(i) + d_{in}^B(i)$ (as used in (3)). We assume

$$d_{out}(i), d_{in}^A(i) \in \mathbb{N}, \quad d_{in}^B(i) \in \mathbb{N}_0 \quad \forall\ i \in A,$$
$$\textstyle\sum_{i\in A} d_{out}(i) = \sum_{i\in A} d_{in}^A(i).$$

In words, the first condition says i is observed by and observes at least one agent, and may observe one or more bots. The second condition says sum out-degree must equal sum in-degree in the agent sub-graph; this will be necessary to construct a graph with the given degrees. Finally, it will be convenient to define the degree vector of $i \in A$ as

$$d(i) = (d_{out}(i), d_{in}^A(i), d_{in}^B(i)). \quad (4)$$

After realizing the degree sequence, we begin the graph construction.[3] First, we attach $d_{out}(i)$ outgoing half-edges, $d_{in}^A(i)$ incoming half-edges labeled A, and $d_{in}^B(i)$ incoming half-edges labeled B, to each $i \in A$; we will refer to these half-edges as *outstubs*, *A-instubs*, and *B-instubs*, respectively. Let O_A denote the set of all agents' outstubs. We then pair each outstub in O_A with an A-instub to form edges between agents in a breadth-first-search fashion that proceeds as follows:

- Sample i^* from A uniformly. For each the $d_{in}^A(i^*)$ A-instubs attached to i^*, sample an outstub uniformly from O_A (resampling if the sampled outstub has already been paired), and connect the instub and outstub to form an edge from some agent to i^*.

[3]This construction is presented more formally in [10, Appendix II-A].

- Let $A_1 = \{i \in A \setminus \{i^*\}$: an outstub of i was paired with an A-instub of $i^*\}$. For each $i \in A_1$, pair the $d_{in}^A(i)$ A-instubs attached to i in the same manner the A-instubs of i^* were paired in the previous step.
- Continue iteratively until all A-instubs have been paired. In particular, during the l-th iteration, we pair all A-instubs attached to A_l, the agents at geodesic distance l from i^*.

The procedure above yields the standard DCM, plus unpaired B-instubs attached to some agents. To pair these instubs, we define $B = n + \left[\sum_{i\in A} d_{in}^B(i)\right]$ to be the set of bots (hence, the node set is $A \cup B = \left[n + \sum_{i\in A} d_{in}^B(i)\right]$). To each $i \in B$ we add a single self-loop and a single unpaired outstub (as described at the end of Section II). This yields $\sum_{i\in A} d_{in}^B(i)$ unpaired outstubs attached to bots. Finally, we pair these outstubs arbitrarily with the $\sum_{i\in A} d_{in}^B(i)$ unpaired B-instubs from above to form edges from bots to agents (the pairing can be arbitrary since all bots behave the same).

We note that the pairing of A-instubs with outstubs from O_A did not prohibit multi-edges, so the set of edges E formed will in general be a multi-set. For this reason, we replace the summation in the $\alpha_t(i)$ update (2) with

$$\sum_{j\in A\cup B} \eta|\{j' \to i' \in E : j' = j, i' = i\}|\alpha_{t-1}(j)/d_{in}(i),$$

and analogously for the $\beta_t(i)$ update, i.e. we weigh the parameters of i's neighbors proportional to the number of edges pointing to i. We also note that if $d_{in}^B(i) = 0\ \forall\ i \in A$, the construction above reduces to the standard DCM.

Our results will require assumptions on the degree sequence $\{d(i)\}_{i\in A}$, where (we recall) $d(i)$ is the degree vector of i (see (4)). First, we define $f_n^*, f_n : \mathbb{N}\times\mathbb{N}\times\mathbb{N}_0 \to [0,1]$ by

$$f_n^*(i,j,k) = \textstyle\sum_{a=1}^n 1(d(a) = (i,j,k))/n,$$
$$f_n(i,j,k) = \sum_{a=1}^n d_{out}(a)1(d(a) = (i,j,k))/\sum_{a'=1}^n d_{out}(a').$$

In words, f_n^* and f_n are the degree distributions of agents sampled uniformly and sampled proportional to out-degree, respectively. Note that, since the first agent i^* added to the graph is sampled uniformly from A, the degrees of i^* are distributed as f_n^*. Furthermore, recall that, to pair A-instubs, we sample outstubs uniformly from O_A, resampling if the sampled outstub is already paired. It follows that, each time we add a new agent to the graph (besides i^*), its degrees are distributed as f_n. We also note that, because the degree sequence is random, these distributions are random as well. From these random distributions, we define the random variables

$$\begin{aligned}\tilde{p}_n^* &= \textstyle\sum_{j\in\mathbb{N},k\in\mathbb{N}_0}(j/(j+k))\sum_{i\in\mathbb{N}} f_n^*(i,j,k),\\ \tilde{p}_n &= \textstyle\sum_{j\in\mathbb{N},k\in\mathbb{N}_0}(j/(j+k))\sum_{i\in\mathbb{N}} f_n(i,j,k),\\ \tilde{q}_n &= \textstyle\sum_{j\in\mathbb{N},k\in\mathbb{N}_0}(j/(j+k)^2)\sum_{i\in\mathbb{N}} f_n(i,j,k).\end{aligned} \tag{5}$$

Following the discussion above, $\tilde{p}_n^*$ is the expected value (conditioned on the degree sequence) of the ratio of A-instubs to total instubs for i^*; $\tilde{p}_n$ is the expected value of this same ratio, but for new agents added to the graph. The interpretation of $\tilde{q}_n$ is similar. At the end of Section III-B, we discuss in more detail why these random variables arise in our analysis.

We now state four assumptions, which we discuss in detail in Section III-C. Two of these require the degree sequence to be well-behaved (with high probability) – specifically, A1 requires certain moments of the degree sequence to be finite, while A3 requires $\{\tilde{p}_n\}_{n\in\mathbb{N}}$ to be close to a deterministic sequence $\{p_n\}_{n\in\mathbb{N}}$. The other assumptions, A2 and A4, impose maximum and minimum rates of growth for the learning horizon T_n. In particular, T_n must be finite for each finite n but grow to infinity with n.

A1 $\lim_{n\to\infty}\mathbb{P}(\Omega_{n,1}) = 1$, where, for some $\nu_1,\nu_2,\nu_3,\gamma > 0$ independent of n such that $\nu_3 > \nu_1$,[4]

$$\begin{aligned}\Omega_{n,1} = &\left\{\left|(\textstyle\sum_{i=1}^n d_{out}(i)/n) - \nu_1\right| < n^{-\gamma}\right\}\\ &\cap\left\{\left|(\textstyle\sum_{i=1}^n d_{out}(i)^2/n) - \nu_2\right| < n^{-\gamma}\right\}\\ &\cap\left\{\left|(\textstyle\sum_{i=1}^n d_{out}(i)d_{in}^A(i)/n) - \nu_3\right| < n^{-\gamma}\right\}.\end{aligned}$$

A2 $\exists\ N \in \mathbb{N}$ and $\zeta \in (0,1/2)$ independent of n s.t. $T_n \le \zeta\log(n)/\log(\nu_3/\nu_1)\ \forall\ n \ge N$.

A3 $\lim_{n\to\infty}\mathbb{P}(\Omega_{n,2}) = 1$, where, for some $p_n \in [0,1]$ s.t. $\lim_{n\to\infty} p_n = p \in [0,1]$, some $0 \le \delta_n = o(1/T_n)$, and some $\xi \in (0,1)$ independent of n,

$$\Omega_{n,2} = \{|p_n - \tilde{p}_n| < \delta_n, \tilde{p}_n^* \ge \tilde{p}_n, \tilde{q}_n < 1-\xi\}.$$

A4 $\lim_{n\to\infty} T_n = \infty$.

B. Main result

We can now present Theorem 1. The theorem states that the estimate at time T_n of a uniformly random agent converges in probability as $n \to \infty$. As discussed in the introduction, the limit depends on the relative asymptotics of the time horizon T_n and the quantity p_n defined in A3. For example, this limit is θ when $T_n(1-p_n) \to 0$; note that $T_n(1-p_n) \to 0$ requires p_n to quickly approach 1 (since $T_n \to \infty$ by A4), which by A3 and (5) suggests the number of bots is small. Hence, i^* learns the true state when there are sufficiently few bots. (The other cases can be interpreted similarly.)

Theorem 1: Assume that G is the DCM and that A1, A2, A3, and A4 hold. Then for $i^* \sim A$ uniformly,

$$\theta_{T_n}(i^*) \xrightarrow[n\to\infty]{\mathbb{P}} \begin{cases}\theta, & T_n(1-p_n)\to 0\\ \frac{\theta(1-e^{-c\eta})}{c\eta}, & T_n(1-p_n)\to c\in(0,\infty)\\ 0, & T_n(1-p_n)\to\infty\end{cases}.$$

Before discussing the proof, we make several observations:

- Suppose p_n is fixed and consider varying T_n. To be concrete, let $p_n = 1-(\log n)^{-1/2}$ and define $T_{n,1} = (\log n)^{1/4}$ and $T_{n,2} = (\log n)^{3/4}$ (note $T_{n,1}, T_{n,2}$ satisfy A2, A4). Then $T_{n,1}(1-p_n) \to 0$ and $T_{n,2}(1-p_n) \to \infty$, so by Theorem 1, the estimate of i^* converges to θ at time $T_{n,1}$ and to 0 at time $T_{n,2}$. In words, i^* initially (at time $(\log n)^{1/4}$) learns the state of the world, then later (at time $(\log n)^{3/4}$) forgets it and adopts the bot estimates.
- Alternatively, suppose T_n is fixed and consider varying p_n. For example, let $p_n = 1 - c/T_n$ for some $c \in (0,\infty)$. Here smaller c implies fewer bots, and Theorem 1 says the limiting estimate of i^* is a decreasing convex function of c. One interpretation is that, if an adversary deploys bots in hopes of driving agent estimates to 0, the marginal benefit

[4]The assumption $\nu_3 > \nu_1$ only eliminates the trivial case of a line graph; see Section III-C for details.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2022.3154679, IEEE Transactions on Control of Network Systems

of deploying additional bots is smaller when c is larger, i.e. the adversary experiences "diminishing returns". It is also worth noting that, since $(1-e^{-c\eta})/(c\eta) \to 1$ as $c \to 0$ and $(1-e^{-c\eta})/(c\eta) \to 0$ as $c \to \infty$, the limiting estimate of i^* is continuous as a function of c.

- If $T_n(1-p_n) \to c \in (0,\infty)$, consider the limiting estimate of i^* as a function of η. By Theorem 1, this estimate tends to θ as $\eta \to 0$ and tends to $(1-e^{-c})/c$ as $\eta \to 1$. This is expected from (2): when $\eta = 0$, agents ignore the network (and thus avoid exposure to biased bot beliefs) and form estimates based only on unbiased signals; when $\eta = 1$, the opposite is true.
- If $p_n \to p < 1$, we must have $T_n(1-p_n) \to \infty$ (since $T_n \to \infty$ by A4), and the estimate of i^* tends to 0 by Theorem 1. Loosely speaking, this says that a necessary condition for learning is that the bots vanish asymptotically (in the sense that $p_n \to 1$).
- In fact, in the case $p_n \not\to 1$, a stronger result holds: the set of agents i for which $\theta_{T_n}(i) \not\to 0$ vanishes relative to n. See [10, Appendix I] for details.

The proof of Theorem 1 is lengthy and deferred to [10, Appendices II and IV], where [10, Appendix II] lays out the structure of the proof. However, we next present a short argument to illustrate the fundamental reason why the three cases of the limiting estimate arise in Theorem 1.

At a high level, these three cases arise as follows. First, when $T_n(1-p_n) \to 0$, the "density" of bots within the T_n-step incoming neighborhood of i^* is small. As a consequence, i^* is not exposed to the biased beliefs of bots by time T_n and is able to learn the true state ($\theta_{T_n}(i^*) \to \theta$). On the other hand, when $T_n(1-p_n) \to \infty$, this "density" is large; i^* is exposed to bot beliefs and thus adopts them. Finally, when $T_n(1-p_n) \to c \in (0,\infty)$, the "density" is moderate; i^* does not fully learn, nor does i^* fully adopt bot beliefs.

This explanation is not at all surprising; what is more subtle is what precisely *density of bots within the T_n-step incoming neighborhood of i^** means. It turns out that the relevant quantity is the probability that a random walker exploring this neighborhood reaches the set of bots. To illustrate this, consider a random walk $\{X_l\}_{l\in\mathbb{N}}$ that begins at $X_0 = i^*$ and, for $l \geq 0$, chooses X_{l+1} uniformly from all incoming neighbors of X_l (agents and bots); note here that the walk follows edges in the direction opposite to their polarity in the graph. For this walk, it is easy to see that, conditioned on the event $X_l \in A$, the event $X_{l+1} \in A$ occurs with probability

$$d^A_{in}(X_l)/(d^A_{in}(X_l) + d^B_{in}(X_l)). \tag{6}$$

Crucially, we sample this walk and construct the graph simultaneously, by choosing which instub of X_{l-1} to follow *before* actually pairing these instubs. Assuming they are later paired with agent outstubs chosen uniformly at random, and hence connected to agents chosen proportional to out-degree, we can average (6) over the out-degree distribution to obtain that $X_{l+1} \in A$ occurs with probability

$$\sum_{a\in A} \frac{d^A_{in}(a)}{d^A_{in}(a)+d^B_{in}(a)} \frac{d_{out}(a)}{\sum_{a'\in A} d_{out}(a')} = \tilde{p}_n. \tag{7}$$

Now since bots have a self-loop and no other incoming edges, they are absorbing states on this walk. It follows that $X_{T_n} \in A$ if and only if $X_l \in A \ \forall\ l \in [T_n]$; by the argument above, this latter event occurs with probability $\tilde{p}_n^{T_n}$. Since $\tilde{p}_n \approx p_n$ by A3, we thus obtain that $X_{T_n} \in A$ with probability

$$\tilde{p}_n^{T_n} \approx p_n^{T_n} \approx e^{-\lim_{n\to\infty} T_n(1-p_n)}.$$

From this final expression, Theorem 1 emerges: when $T_n(1-p_n) \to 0$, the random walker remains in the agent set with probability ≈ 1; this corresponds to i^* avoiding exposure to bot beliefs and learning the true state. Similarly, $T_n(1-p_n) \to \infty$ means the walker is absorbed into the bot set with probability ≈ 1, corresponding to i^* adopting bot estimates. Finally, $T_n(1-p_n) \to c \in (0,\infty)$ means the walker stays in the agent set with probability $\approx e^{-c} \in (0,1)$, corresponding to i^* not fully learning nor fully adopting bot estimates.

We note that the actual proof of Theorem 1 does not precisely follow the foregoing argument. Instead, we locally approximate the graph construction with a certain branching process; we then study random walks on the tree resulting from this branching process.[5] However, the foregoing argument illustrates the basic reason why the three distinct cases of Theorem 1 arise. We also observe that the argument leading to (7) shows why $\tilde{p}_n$ enters into our analysis. The other random variables defined in (5) enter similarly. Specifically, $\tilde{p}^*_n$ arises in almost the same manner, but pertains only to the first step of the walk; this distinction arises since the walk starts at i^*, the degrees of which relate to $\tilde{p}^*_n$. On the other hand, $\tilde{q}_n$ arises when we analyze the variance of agent estimates. This is because analyzing the variance involves studying *two* random walks; by an argument similar to (7), the probability of both walks visiting the same agent is

$$\sum_{a\in A} \frac{d^A_{in}(a)}{(d^A_{in}(a)+d^B_{in}(a))^2} \frac{d_{out}(a)}{\sum_{a'\in A} d_{out}(a')} = \tilde{q}_n.$$

Finally, we note that the proof of Theorem 1 reveals that the variance of each agent's belief vanishes, so beliefs converge to Dirac measures. Combined with the theorem, this yields the following corollary. See [10, Appendix V] for a proof.

Corollary 1: Assume G is the DCM and A1, A2, A3, and A4 hold. Let $L(p_n) = L(\{p_n\}_{n=1}^{\infty}, T_n)$ denote the limit (in probability) of $\theta_{T_n}(i^*)$ from Theorem 1. Then for any $p \geq 1$ and for $i^* \sim A$ uniformly, $W_p(\mu_{T_n}(i^*), \delta_{L(p_n)}) \xrightarrow[n\to\infty]{\mathbb{P}} 0$.

C. Comments on assumptions

We now return to comment on the assumptions needed to prove our results. First, A1 states that certain empirical moments of the degree distribution – namely, for $i^* \sim A$ uniformly, the first two moments of $d_{out}(i^*)$ and the correlation between $d_{out}(i^*)$ and $d^A_{in}(i^*)$ – converge to finite limits. Roughly speaking, this says our graph lies in a sparse regime, where typical node degrees do not grow with the number of

[5]This is necessary because the argument leading to (7) assumes instubs are paired with outstubs chosen uniformly at random, which is not true if resampling of outstubs occurs in the construction from Section III-A.

nodes.[6] We also note $\nu_3 > \nu_1$ in A1 is minor and simply eliminates an uninteresting case. To see this, first note that when $\Omega_{n,1}$ holds, we have (roughly)

$$\nu_3/\nu_1 \approx \textstyle\sum_{i=1}^n d_{out}(i) d_{in}^A(i) / \sum_{i'=1}^n d_{out}(i') \geq 1, \quad (8)$$

where we have used the assumed inequality $d_{in}^A(i) \geq 1 \; \forall \; i \in [n]$. Hence, $\nu_3 < \nu_1$ cannot occur, so assuming $\nu_3 > \nu_1$ only prohibits $\nu_3 = \nu_1$. This remaining case is uninteresting because ν_3/ν_1 is the limiting number of offspring for each node in the branching process we analyze; thus, if $\nu_3 = \nu_1$, the tree resulting from this process is simply a line graph.

Next, A2 states $T_n = O(\log n)$. Together with A1, these assumptions are standard given our analysis approach, which, as discussed previously, locally approximates the graph construction with a branching process. We also note that, with the interpretation of ν_3/ν_1 above, it follows that the number of agents within the T_n-step neighborhood of i^* is roughly

$$(\nu_3/\nu_1)^{T_n} = O((\nu_3/\nu_1)^{\zeta \log_{\nu_3/\nu_1}(n)}) = O(n^\zeta) = o(n).$$

In words, the size of the aforementioned neighborhood vanishes relative to n. This is why our title refers to the learning as "local": only a vanishing fraction of other agents (those within this neighborhood) affect the estimate of i^*.

The remaining statements are needed to establish estimate convergence on the tree resulting from the branching process. A4 states $T_n \to \infty$ with n, which is an obvious requirement for convergence. A3 essentially says that three events occur with high probability. First, $\tilde{p}_n$ should be close to a convergent, deterministic sequence p_n; this is necessary since the asymptotics of p_n play a prominent role in Theorem 1. Second, $\tilde{p}_n^* \geq \tilde{p}_n$ essentially says that bots prefer to attach to agents with higher out-degrees, i.e. more influential agents; this is the behavior one would intuitively expect from bots aiming to disrupt learning. Third, $\tilde{q}_n < 1 - \xi \in (0,1)$ is satisfied if, for example, all agents have total in-degree at least two.

Finally, while we focused on the DCM in this section, our analytical approach is more general. At a high level, the key properties of the DCM we used are that most nodes' $O(\log n)$-step neighborhoods are treelike and "statistically similar," which allows for a branching process coupling. Such couplings exist more generally, though this $O(\log n)$ scaling will be smaller for denser graphs, which makes T_n smaller as well.

IV. Adversarial Setting

We next formalize the adversarial problem introduced in Section I. We begin with some notation. Let $m_n = \sum_{i=1}^n d_{out}(i)$, and (with slight abuse of notation to the previous section), define the function $\tilde{p}_n : \mathbb{N}_0^n \to [0,1]$ by

$$\tilde{p}_n(d) = \sum_{i=1}^n \frac{d_{in}^A(i)}{d_{in}^A(i) + d(i)} \frac{d_{out}(i)}{m_n} \; \forall \; d \in \mathbb{N}_0^n,$$

which is simply $\tilde{p}_n$, as defined in (5), viewed as a function of the bot in-degrees $d(i) \triangleq d_{in}^B(i)$[7]. Given a budget $b_n \in \mathbb{N}$, the adversary's problem is then as follows:

$$\min_{d \in \mathbb{N}_0^n} \tilde{p}_n(d) \; s.t. \; \textstyle\sum_{i=1}^n d(i) \leq b_n. \quad (9)$$

Thus, the adversary's objective function only depends on the agent degrees $\{d_{out}(i), d_{in}^A(i)\}_{i \in [n]}$ (e.g. numbers of followers and followees on Twitter), and not the topology of the agent sub-graph. Consequently, the topology will play no role in this section, i.e. we do not require the DCM assumption. We reiterate that, by Theorem 1, solving (9) is equivalent to minimizing estimates asymptotically for the DCM.[8] For general graph topologies, we treat (9) as a nonrigorous but tractable surrogate for estimate minimization, and we will soon show empirically that this is a reasonable choice.

ALGORITHM 1: Exact solution of (9)

Let $d \in dom(\hat{p}_n)$, compute $\hat{p}_n(d)$
while *not terminated* **do**
  Compute $\hat{p}_n(d - e_i + e_j) \; \forall \; i,j \in [n]$ s.t. $i \neq j$
  Let $(i^*, j^*) \in \arg\min_{(i,j) \in [n]^2 : i \neq j} \hat{p}_n(d - e_i + e_j)$
  if $\hat{p}_n(d) \leq \hat{p}_n(d - e_{i^*} + e_{j^*})$ **then** terminate
  else Set $d = d - e_{i^*} + e_{j^*}$

ALGORITHM 2: Approximate solution of (9)

Compute $d_n^{rel}(i)$ as in (11) and set $d_n^{rand}(i) = 0 \; \forall \; i \in [n]$
for $j = 1$ **to** b_n **do**
  Sample W_j from the distribution $d_n^{rel} / \sum_{k=1}^n d_n^{rel}(k)$, i.e. $\mathbb{P}(W_j = i) = d_n^{rel}(i) / \sum_{k=1}^n d_n^{rel}(k) \; \forall \; i \in [n]$
Set $d_n^{rand}(i) = \sum_{j=1}^{b_n} 1(W_j = i) \; \forall \; i \in [n]$

A. Exact solution

First, we let $dom(\hat{p}_n) = \{d \in \mathbb{N}_0^n : \sum_{i=1}^n d(i) = b_n\}$ and rewrite (9) as $\min_{d \in \mathbb{Z}^n} \hat{p}_n(d)$, where

$$\hat{p}_n(d) = \begin{cases} \tilde{p}_n(d), & d \in dom(\hat{p}_n) \\ \infty, & \text{otherwise} \end{cases}.$$

In words, we incorporated the constraints from (9) into the objective; we also used the (obvious) fact that the solution of (9) satisfies the budget constraint with equality. The new objective $\hat{p}_n$ satisfies a certain discrete convexity property, which implies that d minimizes $\hat{p}_n$ if and only if $\hat{p}_n(d) \leq \hat{p}_n(d + e_i - e_j)$ for any i,j pair. Hence, we can find the minimizer by iteratively replacing d with $d + e_i - e_j$ until the objective stops decreasing. This approach is known as *steepest descent* [15, Section 10.1.1] and is provided in Algorithm 1. In [10, Appendix III-E], we show its runtime is $\Theta(n^2)$ in the best case and $O(n^2 b_n)$ in the general case.

B. Approximation algorithm

Algorithm 1's $\Omega(n^2)$ runtime is prohibitive for massive networks like Twitter, which motivates our approximation scheme. The idea is to first solve the relaxed problem

$$\min_{d \in \mathbb{R}_+^n} \tilde{p}_n(d) \; s.t. \; \textstyle\sum_{i=1}^n d(i) \leq b_n, \quad (10)$$

[6] This is analogous to e.g. an Erdős-Rényi model with edge probability λ/n for constant $\lambda > 0$, where degrees converge to Poisson(λ) random variables.

[7] We suppress the sub- and super-scripts to avoid cumbersome notation.

[8] More precisely, this only holds if the solution of (9) converges in the sense of A3. We are unsure if this holds, but we view it as a minor technical point and leave it as an open problem.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2022.3154679, IEEE Transactions on Control of Network Systems

and then to sample bot locations in proportion to the relaxed solution. More formally, our approximate solution d_n^{rand} is constructed via Algorithm 2. We note that by definition, the budget constraint holds with equality for Algorithm 2. Also, as shown in [10, Appendix III-A], the solution of (10) is

$$d_n^{rel}(i) = d_{in}^A(i)((\sqrt{r(i)}/h^*) - 1)_+ \ \forall \ i \in [n], \tag{11}$$

where $x_+ = x1(x > 0)$, $r(i) = d_{out}(i)/d_{in}^A(i) \ \forall \ i \in [n]$, $h^* = \max_{x \in \mathbb{R}_+} h(x)$, and

$$h(x) = \frac{\sum_{i \in [n]: r(i) \geq x^2} \sqrt{d_{out}(i) d_{in}^A(i)}}{b_n + \sum_{i \in [n]: r(i) \geq x^2} d_{in}^A(i)} \ \forall \ x \in \mathbb{R}_+. \tag{12}$$

This randomized scheme yields useful insights, in contrast to the optimal algorithm. In particular, the randomized and relaxed solutions d_n^{rand} and d_n^{rel} are equal in expectation, and the relaxed solution d_n^{rel} satisfies some intuitive properties:

- $d_n^{rel}(i)$ grows with $r(i) = d_{out}(i)/d_{in}^A(i)$, i.e. the adversary targets agents i with large $d_{out}(i)$ and small $d_{in}^A(i)$ under the relaxed solution. Here large $d_{out}(i)$ means i is *influential* (e.g. i has many Twitter followers), while small $d_{in}^A(i)$ means i is *susceptible to influence* (e.g. i has few Twitter followees, so bot tweets will appear prominently in i's Twitter feed).
- If $r(i) < (h^*)^2$, then $d_n^{rel}(i) = d_n^{rand}(i) = 0$. Hence, if i is sufficiently non-influential, and/or sufficiently non-susceptible, targeting i gives no value to the adversary.
- If $r(i) = r(j) > (h^*)^2$, the relaxed solution yields

$$d_{in}^A(i)/(d_{in}^A(i) + d_n^{rel}(i)) = d_{in}^A(j)/(d_{in}^A(j) + d_n^{rel}(j)).$$

  This can be interpreted as follows: the adversary strives for a similar proportion of fake news in the feeds of users with similar ratios of influence to susceptibility.

In short, our approximate solution strives to balance influence and susceptibility. While somewhat intuitive, the precise manner in which this balance occurs (in particular, the form of (11)-(12)) is far from obvious.

In [10, Appendix III-E] , we show Algorithm 2 has complexity $O(n \log n + b_n)$. In terms of accuracy, we next prove that with high probability, Algorithm 2 is a $(2+\delta)$-approximation algorithm for the constrained problem $\max_{d \in \mathbb{N}_0^n : \sum_d (i) \leq b_n} (1 - \tilde{p}_n(d))$, which is equivalent to (9). More precisely, letting d_n^{opt} be any solution of (9), i.e.

$$d_n^{opt} \in \arg\min_{d \in \mathbb{N}_0^n : \sum_{i=1}^n d(i) \leq b_n} \tilde{p}_n(d), \tag{13}$$

we have the following result.

Theorem 2: Let $\delta > 0$ and $c_\delta = \frac{\delta^2}{4(2+\delta)^2}$. Then

$$\begin{aligned}\mathbb{P}(1 - \tilde{p}_n(d_n^{rand}) \leq (1 - \tilde{p}_n(d_n^{opt}))/(2+\delta)) \\ \leq \exp(-c_\delta m_n (1 - \tilde{p}_n(d_n^{rel}))/\max_{j \in [n]} r(j)).\end{aligned}$$

Proof: As mentioned above, [10, Appendix III-A] shows (11) solves (10) (the proof amounts to verifying *KKT conditions*, see e.g. [16, Section 5.5.3]). Hence, by definition (13),

$$\tilde{p}_n(d_n^{rel}) \leq \tilde{p}_n(d_n^{opt}). \tag{14}$$

We next rewrite $1 - \tilde{p}_n(d_n^{rand})$ in terms of the random vector $W = (W_j)_{j=1}^{b_n}$ from Algorithm 2. Toward this end, let $\bar{r} = \max_{j \in [n]} r(j)$, and for $w = (w_j)_{j=1}^{b_n} \in [n]^{b_n}$ define

$$g_n(w) = \frac{1}{\bar{r}} \sum_{j=1}^{b_n} \frac{d_{out}(w_j)}{d_{in}^A(w_j) + \sum_{k=1}^{b_n} 1(w_k = w_j)}.$$

Then a simple calculation yields

$$g_n(W) = m_n(1 - \tilde{p}_n(d_n^{rand}))/\bar{r},$$

and using Jensen's inequality, one can show

$$\mathbb{E} g_n(W) \geq m_n(1 - \tilde{p}_n(d_n^{rel}))/2\bar{r}. \tag{15}$$

(see [10, Appendix III-B] for details.) Combining (14)-(15),

$$1 - \tilde{p}_n(d_n^{rand}) \leq \frac{1 - \tilde{p}_n(d_n^{opt})}{2+\delta} \Rightarrow g_n(W) \leq \frac{2\mathbb{E} g_n(W)}{2+\delta}.$$

Also, using (15) and recalling $\bar{r} = \max_{j \in [n]} r(j)$, we have

$$c_\delta m_n (1 - \tilde{p}_n(d_n^{rel}))/\max_{j \in [n]} r(j) \leq 2 c_\delta \mathbb{E} g_n(W).$$

By the previous two lines, the following implies the theorem:

$$\mathbb{P}\left(g_n(W) \leq 2\mathbb{E} g_n(W)/(2+\delta)\right) \leq \exp\left(-2 c_\delta \mathbb{E} g_n(W)\right). \tag{16}$$

Such an inequality would follow from a simple Hoeffding bound if $g_n(W)$ was simply $\sum_j W_j$; however, $g_n(W)$ is a much more complicated function. Fortunately, g_n belongs to a special class called *self-bounding functions* [17, Section 3.3], for which concentration inequalities of the form (16) are known. See [10, Appendix III-C] for details. ■

The tail bound in Theorem 2 is opaque, as it relies on $\tilde{p}_n(d_n^{rel})$, which (in general) is difficult to interpret. Under certain assumptions, we can obtain more transparent results. For example, we have the following corollary.

Corollary 2: Let $\bar{r} = \max_{j \in [n]} r(j)$ as above. Assume $\lim_{n \to \infty} b_n = \infty$ and for some $\epsilon > 0$ independent of n,

$$\lim_{n \to \infty} |\{i \in [n] : r(i) \geq \epsilon \bar{r}\}| = \infty. \tag{17}$$

Then $\exists \ \{\delta_n\}_{n \in \mathbb{N}} \subset (0, \infty)$ s.t. $\lim_{n \to \infty} \delta_n = 0$ and

$$\lim_{n \to \infty} \mathbb{P}\left(1 - \tilde{p}_n(d_n^{rand}) \leq (1 - \tilde{p}_n(d_n^{opt}))/(2+\delta_n)\right) = 0.$$

Proof: Since d_n^{rel} solves (10), we can weaken the bound in Theorem 2 by replacing $\tilde{p}_n(d_n^{rel})$ with $\tilde{p}_n(d)$ for any $d \in \mathbb{R}_+^n$ with $\sum_i d(i) \leq b_n$. Thus, the proof chooses a particular d that leads to a more tractable bound, and the assumptions ensure this bound vanishes. See [10, Appendix III-D] for details. ■

In words, the corollary shows our randomized scheme is (asymptotically) a 2-approximation algorithm with probability tending to 1. The assumption (17) only precludes the case where only finitely many of the degree ratios $r(i)$ are comparable to the maximum $\bar{r}$. This restriction arises because our self-bounding concentration analysis in Theorem 2 requires normalization by $\bar{r}$ (see [10, Appendix III-C].)

C. Empirical results

A fundamental assumption in our adversary solutions is that $\tilde{p}_n$ and $\theta_{T_n}(i^*)$ are correlated, in the sense that minimizing $\tilde{p}_n$ also minimizes $\theta_{T_n}(i^*)$. While Theorem 1 states this correlation holds for the random graph model of Section III-A,

it is unclear if this correlation occurs in practice. To conclude this section, we present empirical results suggesting that this indeed occurs. In our experiments, we compare our proposed solutions against some natural heuristics:

- A naive baseline, which uses Algorithm 2 but samples each W_j uniformly from $[n]$.
- Three schemes which similarly use Algorithm 2, along with the observed degrees: sampling W_j proportional to d_{out} (i.e. targeting influential nodes), d^A_{in} (i.e. targeting susceptible nodes), and d_{out}/d^A_{in} (i.e. naively balancing the two).
- Sampling W_j proportional to PageRank(ϵ) [18], where[9]

$$\text{PageRank}(\epsilon) = (\epsilon \mathbf{1}_n/n) \sum_{j=0}^{\infty} (1-\epsilon)^j \left(P_A^{\mathsf{T}}\right)^j,$$

where $\epsilon \in (0,1)$, $\mathbf{1}_n$ is the length-n ones vector, and P_A is the agent sub-graph's column-normalized adjacency matrix, i.e. the matrix with (i,j)-th element

$$P_A(i,j) = 1(i \to j \in E_n)/d^A_{in}(j) \;\forall\; i,j \in [n].$$

PageRank is a commonly-used measure of influence or centrality for graphs in many domains [19] (and a richer such measure than d_{out}).

We compare our proposed solutions with these heuristics using four datasets from [20], described in Table I. We chose these datasets so we could test our proposed solutions on real social networks of two scales: Gnutella and Wiki-Vote have $n < 10^4$, a scale at which the exact solution Algorithm 1 is feasible; Pokec and LiveJournal have $n > 10^6$, a scale that renders Algorithm 1 infeasible but that more closely resembles social networks of interest. For the experiments, we set $\theta = 0.5$ (to maximize signal variance), $\eta = 0.9$ (to emphasize the effect of the network), and $T_n = 101$ (to ensure the code had reasonable runtime). We let $b_n = \lceil |E_n|/400 \rceil$, so that 0.25% of all agent in-edges are connected to bots. For each graph and each of five experimental trials, we chose $\{d^B_{in}(i)\}_{i\in[n]}$ as described above, added bots to the original graph accordingly, and simulated the learning process from Section II.

In Figure 1, we plot the mean and standard deviation (across experimental trials) of $\theta_t(i^*)$ as a function of t. For all datasets, our proposed solutions outperform all heuristics, in the sense that our solutions yield the lowest average $\theta_t(i^*)$ for most values of t. Furthermore, we note the following:

- Across all graphs, our solutions outperform PageRank(ϵ) for all values of ϵ tested. This is quite surprising, because PageRank uses the entire *graph topology*, whereas our solutions only use *degree information*. Also, as ϵ becomes increasingly smaller, PageRank(ϵ) performs increasingly better, but this comes at the cost of higher runtime to estimate PageRank(ϵ).
- Among the heuristics using (at most) degree information, d_{out}/d^A_{in} performs best – but still worse than Algorithm 2 – across all datasets. Put differently, naively balancing influence and susceptibility is not enough; the non-obvious form of Algorithm 2 yields better performance.
- For Gnutella and Wiki-Vote, Algorithm 1 noticeably outperforms Algorithm 2. Though the former is an exact solution and the latter is an approximation, this is still surprising, since it is unclear that these schemes are even optimizing the correct objective for real graphs.

[9]In experiments, we compute the first $\lceil \log(0.99)/\log(1-\epsilon) \rceil$ summands, which guarantees an l_1 error bound of 0.01.

TABLE I: Dataset details

Name	Description	Nodes	Edges
Gnutella	Peer-to-peer network	6,301	20,777
Wiki-Vote	Wiki admin elections	7,115	103,689
Pokec	Slovakian social network	1,632,803	30,622,564
LiveJournal	Blogging platform	4,847,571	68,993,773

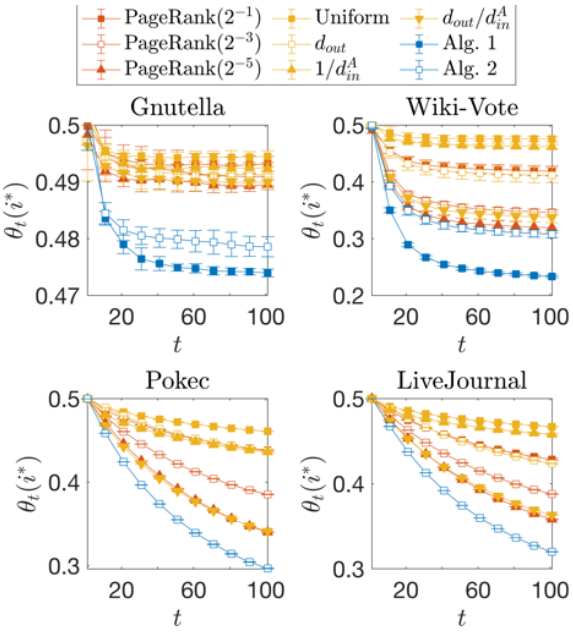


Fig. 1: Estimates when simulating our learning model on real datasets; Algorithms 1 and 2 outperform intuititive heuristics.

While Figure 1 only considers one choice of b_n, we believe our conclusions are robust. In particular, we also tested the cases $b_n = \lceil \tilde{b}|E_n| \rceil$ for each $\tilde{b} \in \{\frac{1}{1600}, \frac{1}{800}, \frac{1}{400}, \frac{1}{200}, \frac{1}{100}\}$, so that between $\approx 0.0625\%$ and $\approx 1\%$ of edges connected to bots (thus, Figure 1 shows the intermediate case $\tilde{b} = \frac{1}{400}$). [10, Appendix III-F] contains a figure analogous to Figure 1 for the other choices of $\tilde{b}$; the plots are qualitatively similar.

We have thus far shown that our solutions outperform heuristics, even those using graph topology. This is quite surprising: our solutions were derived under the fundamental

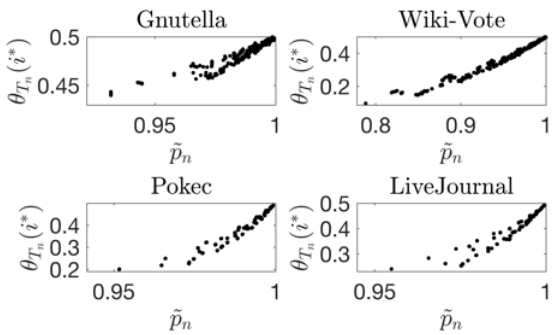


Fig. 2: As suggested by Figure 1, $\theta_{T_n}(i^*)$ and $\tilde{p}_n$ are closely correlated for real social networks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2022.3154679, IEEE Transactions on Control of Network Systems

assumption that *minimizing $\theta_{T_n}(i^*)$ amounts to minimizing $\tilde{p}_n$*, but we only verified this assumption asymptotically for a class of random graphs. Thus, our empirical results suggest that *even for real social networks, this assumption holds*. Indeed, in Figure 2 we show scatter plots of $\theta_{T_n}(i^*)$ against $\tilde{p}_n$ (each dot represents one experimental trial). For all datasets, the two quantities are closely correlated.

V. Related work

As discussed in Section II, (2) resembles the non-Bayesian social learning model from [4], which uses belief update

$$\mu_t(i) = \eta_{ii}\text{BU}(\mu_{t-1}(i), \omega_t(i)) + \textstyle\sum_{j\in N_{in}(i)} \eta_{ij}\mu_{t-1}(j), \quad (18)$$

where $\sum_j \eta_{ij} = 1$, $\omega_t(i)$ is a signal, and BU means Bayesian update. Hence, agents perform Bayesian updates and then average in terms of *beliefs* in [4] but *parameters* in this work. The main advantage of the latter is that beliefs remain Beta distributions, which simplifies our analysis. This simplification, along with weights $\eta/d_{in}(j)$ instead of (18), are needed since we consider a finite horizon and a graph which need not be connected, in contrast to [4]. Another distinction is that agents in [4] need not be able to learn the true state individually (i.e., in the absence of a network). In contrast, agents in our work can learn in isolation (simply by averaging their signals), so the network can either speed up learning or be a detriment. This (potential) detriment is relevant to platforms like Twitter, where users who could have read accurate news in isolation instead risk exposure to bots.

Our parameter update is also studied in [6], which features bots defined in a slightly different manner but in the same spirit. However, [6] only includes theoretical results in the case $B = \emptyset$; the case $B \neq \emptyset$ is studied empirically. This allowed [6] to use a slightly richer model, including a time-varying graph and agent-dependent mixture parameters $\sum_{j\in N_{in}(i)\cup\{i\}} \eta_{ij}$. Notably, the empirical results from [6] fix a learning horizon and do not investigate the effects of different timescales; in particular, the delicate relationship between timescale and bot prevalence from Theorem 1 is not brought to light. Beyond stubborn agents, [21], [22] propose different non-Bayesian updates to cope with Byzantine agents with arbitrary behavior.

From an analytical perspective, our approach of analyzing estimates by studying random walks is similar to the deGroot model [13]. Here the estimate vector $\theta_t = \{\theta_t(i)\}_i$ is updated as $\theta_t = \theta_{t-1}W$ for some column-stochastic matrix W. Hence, $\theta_t = \theta_0 W^t$, so i's belief is determined by the distribution of a t-step random walk from i. This observation has been exploited in the literature; see the surveys [23, Section 3] and [24, Section 4], and the references therein. For example, assuming W is irreducible and aperiodic, and therefore has a well-defined stationary distribution π, [7] establishes conditions for learning using the fact that $\theta_t(i) = \theta_0 W^t e_i^\mathsf{T} \approx \theta_0 \pi^\mathsf{T}\ \forall\ i$ when t is large. Roughly speaking, our model combines deGroot-like averaging with exogenous unbiased signals. As discussed, the averaging in our case exposes agents to biased beliefs (due to bots); the resulting tension between biased and unbiased information is a key feature in our model not present in deGroot's. Ours is arguably a richer model of platforms like Twitter, where there is a similar tension between legitimate news and bots. Beyond the deGroot model, agents in [25] perform Bayesian updates using the prior of a randomly-chosen neighbor, which yields a different connection to random walks; assuming strong connectedness, the authors exploit the fact that the walk visits every agent infinitely often (i.o.) to derive conditions for learning.

Similar to [4], the papers of the previous paragraph typically assume strong connectedness and long learning horizons so as to leverage properties such as stationary distributions and i.o. visits. This is a fundamental distinction from our work. Indeed, even if we disregard stubborn agents, the random walk converges to a stationary distribution, but it does *not* converge within our local learning horizon. This is because, as shown in [26], the DCM we consider has mixing time that exceeds

$$\frac{\log n}{\sum_{i\in[n]} \log(d_{in}^A(i)) \frac{d_{out}(i)}{\sum_{i'\in A} d_{out}(i')}} \gtrapprox \frac{\log n}{\log(\nu_3/\nu_1)},$$

where we used Jensen's inequality and (8). The right side exceeds T_n by A2, i.e. our learning horizon occurs before the underlying random walk mixes. In fact, [26] shows that the random walk on the DCM exhibits *cutoff*, meaning that the T_n-step distribution of this walk can be maximally far from the stationary distribution (i.e. the total variation distance between these distributions can be 1 for certain starting locations of the walk). Hence, not only can we not use this stationary distribution, we cannot even use an approximation of it. Again, this means our analysis cannot leverage global properties typically used when relating estimates to random walks. We circument this using the DCM, which has a well-behaved local structure. We also note that our idea to simultaneously construct the graph and sample the walk is taken from [26].

Some other works have considered social learning with stubborn agents. For example, [8] studies a model in which agents meet and either retain their own estimates, adopt the average of their estimates, or adopt a weighted average; the agent whose estimate has a larger weight is called a "forceful" agent. Here the authors show that all agent estimates converge to a common random variable and study its deviation from the true state. A crucial difference between this work and ours is that [8] assumes even forceful agents occasionally observe other agents' opinions. This yields an underlying Markov chain that is irreducible (unlike ours); the analysis then relies on this chain having a well-defined stationary distribution.

Stubborn agents have also been considered in the consensus setting [27], which asks whether agent estimates converge to a common value, i.e. a consensus. For example, [28] considers a model in which regular agents adopt weighted averages of estimates upon meeting other agents, while stubborn agents always retain their own estimates. This intuitively prohibits a consensus from forming; indeed, it is shown that agent estimates fail to converge, i.e. disagreement can persist indefinitely. Another example is [29], in which an agent's estimate at time $t+1$ is a weighted average of their own estimate at time 0 and their neighbors' estimates at time t. In this model, stubborn agents place all weight on their own estimate from time 0 and thus do not update their estimates. The analysis in [29] is similar to ours as it relates agent estimates to hitting

probabilities of the stubborn agent set, but it differs as the learning horizon is infinite in [29]. Also in the consensus setting, [30] investigates protocols for robust consensus that may lessen the undesirable effects of stubborn agents.

The problem of deploying stubborn agents is studied in [31], [32], though for the voter model. Both assume knowledge of a matrix describing the graph topology (like P_A from Section IV-C), and the optimization requires inverting this matrix at complexity n^3. Our algorithms overcome both of these issues. We also note this inversion is common in more general influence maximization settings.

Without stubborn agents, [33] considers a non-Bayesian update for infinite horizons, where agents treat neighbors' beliefs as independent. Convergence rates are provided in [9], [34], [35] for (2) or similar Bayesian-plus-aggregation updates. An open question is how these models behave with stubborn agents, particularly for [9], [34], [35], where the convergence may be slower than the propagation of stubborn agent bias.

REFERENCES

[1] E. Shearer and J. Gottfried, "News use across social media platforms 2017," *Pew Research Center, Journalism and Media*, 2017.

[2] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–36, 2017.

[3] P. Savodnik, "'You start seeing the dreaded sensitivity label': Is a pro-Trump Twitter army strategically throttling Biden ads?" https://www.vanityfair.com/news/2020/10/is-a-pro-trump-twitter-army-strategically-throttling-biden-ads, 2020.

[4] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, "Non-Bayesian social learning," *Games and Economic Behavior*, vol. 76, no. 1, pp. 210–225, 2012.

[5] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nature communications*, vol. 9, no. 1, p. 4787, 2018.

[6] M. Azzimonti and M. Fernandes, "Social media networks, fake news, and polarization," National Bureau of Economic Research, Tech. Rep., 2018.

[7] B. Golub and M. O. Jackson, "Naive learning in social networks and the wisdom of crowds," *American Economic Journal: Microeconomics*, vol. 2, no. 1, pp. 112–49, 2010.

[8] D. Acemoglu, A. Ozdaglar, and A. ParandehGheibi, "Spread of (mis) information in social networks," *Games and Economic Behavior*, vol. 70, no. 2, pp. 194–227, 2010.

[9] A. Lalitha, A. Sarwate, and T. Javidi, "Social learning and distributed hypothesis testing," in *2014 IEEE International Symposium on Information Theory*. IEEE, 2014, pp. 551–555.

[10] D. Vial and V. Subramanian, "Local non-Bayesian social learning with stubborn agents," *arXiv preprint arXiv:1904.12767*, 2020.

[11] ——, "Local non-Bayesian social learning with stubborn agents," in *Proc. of the 2019 ACM Conference on Economics and Computation*, 2019, pp. 551–552.

[12] ——, "Local non-Bayesian social learning with stubborn agents," in *57th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2019, pp. 902–903.

[13] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[14] R. Van Der Hofstad, *Random graphs and complex networks*. Cambridge university press, 2016, vol. 1.

[15] K. Murota, *Discrete convex analysis*. SIAM, 2003.

[16] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[17] S. Boucheron, G. Lugosi, and P. Massart, *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[19] D. F. Gleich, "PageRank beyond the web," *SIAM Review*, vol. 57, no. 3, pp. 321–363, 2015.

[20] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data.

[21] A. Mitra, J. A. Richards, and S. Sundaram, "A new approach to distributed hypothesis testing and non-Bayesian learning: Improved learning rate and Byzantine-resilience," *IEEE Transactions on Automatic Control*, 2020.

[22] L. Su and N. H. Vaidya, "Non-bayesian learning in the presence of Byzantine agents," in *International symposium on distributed computing*. Springer, 2016, pp. 414–427.

[23] B. Golub and E. Sadler, "Learning in social networks," *Available at SSRN 2919146*, 2017.

[24] D. Acemoglu and A. Ozdaglar, "Opinion dynamics and learning in social networks," *Dynamic Games and Applications*, vol. 1, no. 1, pp. 3–49, 2011.

[25] M. A. Rahimian, S. Shahrampour, and A. Jadbabaie, "Learning without recall by random walks on directed graphs," in *IEEE 54th Annual Conference on Decision and Control*. IEEE, 2015, pp. 5538–5543.

[26] C. Bordenave, P. Caputo, and J. Salez, "Random walk on sparse random digraphs," *Probability Theory and Related Fields*, vol. 170, no. 3-4, pp. 933–960, 2018.

[27] D. Teneketzis and P. Varaiya, "Consensus in distributed estimation," *Advances in Statistical Signal Processing*, vol. 1, p. 361, 1987.

[28] D. Acemoglu, G. Como, F. Fagnani, and A. Ozdaglar, "Opinion fluctuations and persistent disagreement in social networks," in *Proc. of 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 2347–2352.

[29] J. Ghaderi and R. Srikant, "Opinion dynamics in social networks with stubborn agents: Equilibrium and convergence rate," *Automatica*, vol. 50, no. 12, pp. 3209–3215, 2014.

[30] T. Rocket, "Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies," https://pdfs.semanticscholar.org/85ec/19594046bbcfe12137c7c2e3744677129820.pdf, 2018.

[31] E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, and A. Scaglione, "Binary opinion dynamics with stubborn agents," *ACM Transactions on Economics and Computation (TEAC)*, vol. 1, no. 4, pp. 1–30, 2013.

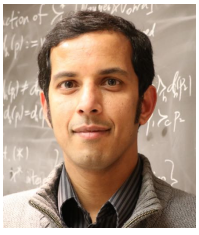
[32] E. Sadler, "Influence campaigns," *Available at SSRN 3371835*, 2020.

[33] J. Anunrojwong and N. Sothanaphan, "Naive Bayesian learning in social networks," in *Proc. of the 2018 ACM Conference on Economics and Computation*. ACM, 2018, pp. 619–636.

[34] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed detection: Finite-time analysis and impact of network topology," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3256–3268, 2015.

[35] A. Nedić, A. Olshevsky, and C. A. Uribe, "Fast convergence rates for distributed non-Bayesian learning," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5538–5553, 2017.

Daniel Vial received the B.S. in Electrical Engineering from the University of Iowa in 2014, and the M.S. and Ph.D. in Electrical Engineering: Systems from the University of Michigan in 2017 and 2020, respectively. Since receiving his Ph.D., he has been a postdoctoral researcher jointly hosted by the University of Texas at Austin and the University of Illinois at Urbana-Champaign. His research interests include reinforcement learning, network science, and applied probability.



Vijay Subramanian received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 1999. He worked at Motorola Inc., at the Hamilton Insitute, Maynooth, Ireland, for many years, and in the EECS Department, Northwestern University, Evanston, IL, USA. From 2014, he is an Associate Professor with the EECS Department at the University of Michigan, Ann Arbor, MI, USA. His research interests are in stochastic analysis, random graphs, multi-agent systems, and game theory (mechanism and information design) with applications to social, and economic and technological networks.