ASYNCHRONOUS SOCIAL LEARNING

Mert Cemri , Virginia Bordignon† , Mert Kayaalp†, Valentina Shumovskaia†, and Ali H. Sayed†*

ABSTRACT

Social learning algorithms provide a model for the formation and propagation of opinions over social networks. However, most studies focus on the case in which agents share their information synchronously over regular intervals. In this work, we analyze belief convergence and steady-state learning performance for both traditional and adaptive formulations of social learning under asynchronous behavior by the agents, where some of the agents may decide to abstain from sharing any information with the network at some time instants. We also show how to recover the underlying graph topology from observations of the asynchronous network behavior.

Index Terms— Social learning, asynchronous updates, adaptive social learning, graph learning.

1. INTRODUCTION

Social learning strategies model how opinions are formed and shared over social networks [1–13]. In this setting, each user (or *agent*) in the network of users iteratively forms their opinion (or *belief*) regarding a finite set of hypotheses by incorporating private observations and by communicating their beliefs with their neighbors, which are aggregated using a combination rule. The flow of information among agents is represented by a graph of users.

In the real world, it is rarely the case that every agent in the network operates *synchronously*. Instead, agents can be absent at times, e.g., when they do not have a new piece of information to share, or when they are not available to communicate. Twitter is an example of such a network, where users act *asynchronously*, i.e., users share posts at different times, and at different frequencies. This is illustrated in Fig. 1, where we show for three different Twitter users the ratio of days they did not share a tweet over each month (hence their "monthly absence" ratios), over a span of 64 months. Motivated by this observation, we would like to consider a slightly more general formulation for social learning, by proposing an asynchronous version of social learning.

In traditional social learning, each agent updates its belief by performing two steps: **i)** a local Bayesian update using the newly received observation, and **ii)** a combination step, which takes the form of a weighted geometric average of the beliefs received from neighbors. These beliefs are weighted according to a combination matrix A, which reflects the underlying communication graph and the confidence weights associated with each link. In the asynchronous social learning strategy proposed in this work, we introduce a network where, at some time instants, agents may decide not to participate in the process of exchanging information with their neighbors. When some agents are absent, one may imagine that the network topology changes momentarily, as if those agents "turn off" and are not a part of the network at that particular time instant. We therefore model this behavior by considering a random combination matrix, denoted by $\boldsymbol{A}_i$. A related work that considers random combination matrices is [14], in which, at each iteration, agents choose only one neighbor randomly to communicate. Although the analysis in our work will make use of some results from [14], their model is different because we focus on asynchronous communication by agents. In our strategy, inspired by the literature on asynchronous distributed learning [15,16], we allow for fairly general forms of $\boldsymbol{A}_i$, and require only these general properties: **i)** $\boldsymbol{A}_i$ should be independent over time, **ii)** $\boldsymbol{A}_i$ should be left stochastic, and **iii)** the support graph associated with the expected value of $\boldsymbol{A}_i$ should be strongly connected, meaning that, in expectation, information can flow between every two agents in both directions, and that there exists at least one self-loop [17, 18]. Under the aforementioned conditions, we show that consistent truth learning is achieved for traditional social learning [10, 12] and for adaptive social learning [13, 19] over asynchronous graphs. Moreover, inspired by recent works on graph topology recovery in the context of adaptive social learning [20, 21], we also formulate the topology learning problem for asynchronous environments.

Notation: Bold font denotes random variables and normal font denotes deterministic variables. Symbol $\xrightarrow{\text{a.s.}}$ denotes almost sure convergence and $\xrightarrow{\text{p}}$ denotes convergence in probability. Symbol $\langle x, y\rangle$ denotes the inner product $x^\top y$ between column vectors x and y.

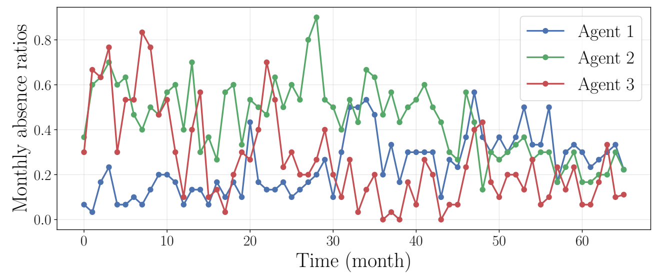

Fig. 1: Absence ratios of some agents on Twitter over time, each time point being a month (there are 64 months, plotted between January 2017 and April 2022).

2. PROBLEM SETUP

Consider a network of K agents whose confidence weights are gathered into a left stochastic combination matrix A. Agents observe some common phenomenon represented by a true discrete-valued state $\theta^\star$. Their goal is to determine the true state from among a set of possible discrete hypotheses, denoted by Θ. Agents receive private observations, denoted by $\boldsymbol{\xi}_{k,i}$, which are identically and independently distributed (i.i.d.) over time, but not necessarily across agents. For each agent k, the observations $\boldsymbol{\xi}_{k,i}$ are assumed to be

M. Cemri is with the Department of Electrical and Electronics Engineering, Bilkent University, 06800, Ankara, Turkey. The author performed the work as a visiting student at the Adaptive Systems Laboratory at EPFL, Switzerland. E-mail: mert.cemri@ug.bilkent.edu.tr

V. Bordignon, M. Kayaalp, V. Shumovskaia and A. H. Sayed are with the School of Engineering, EPFL, CH-1015, Lausanne, Switzerland. E-mails:{virginia.bordignon, mert.kayaalp, valentina.shumovskaia, ali.sayed}@epfl.ch

This work was supported in part by the SNSF grant 205121-184999.

distributed according to some likelihood distribution $L_k(\xi|\theta^\star)$.

To solve the inference problem, we resort to social learning strategies [5, 6, 10, 12, 13], wherein the belief of each agent is updated at each instant by performing adaptation and combination steps in the following manner. The belief vector of agent k at time i is denoted by $\boldsymbol{\mu}_{k,i}$, and each component $\boldsymbol{\mu}_{k,i}(\theta)$ represents the confidence by that agent that $\theta \in \Theta$ is the true state. Under stationary environments, i.e., when the statistical conditions are static over time, traditional social learning strategies [7–10, 12], perform the adaptation step in (1) and the combination step in (2):

$$\boldsymbol{\psi}_{k,i}(\theta) \propto L_k(\boldsymbol{\xi}_{k,i}|\theta)\boldsymbol{\mu}_{k,i-1}(\theta) \tag{1}$$

$$\boldsymbol{\mu}_{k,i}(\theta) \propto \prod_{\ell \in \mathcal{N}_k} \left(\boldsymbol{\psi}_{\ell,i}(\theta)\right)^{a_{\ell k}} \tag{2}$$

where $\propto$ indicates the belief entries are normalized to add up to 1, $a_{\ell k}$ is the ℓk-th element of the combination matrix A, i.e. the combination weight assigned by agent k to neighboring agent ℓ, such that $0 \leq a_{\ell k} \leq 1$, $\sum_{\ell=1}^{K} a_{\ell k} = 1$ and $a_{\ell k} = 0$ if $\ell \notin \mathcal{N}_k$. Here, the symbol $\mathcal{N}_k$ denotes the set of neighbors of agent k. It is known that this strategy enables asymptotic truth learning under mild conditions, i.e., $\boldsymbol{\mu}_{k,i}(\theta^\star) \xrightarrow{\text{a.s.}} 1$. However, when the environment is nonstationary, the above strategy fails to track changes in the underlying true state within a reasonable time and in this case, adaptive social leaning should be considered [13]. In this framework, the adaptation step (1) is replaced by (3):

$$\boldsymbol{\psi}_{k,i}(\theta) \propto L_k(\boldsymbol{\xi}_{k,i}|\theta)^{\delta}\boldsymbol{\mu}_{k,i-1}(\theta)^{1-\delta} \tag{3}$$

where $0 < \delta < 1$ is a step-size parameter that can be used to control the trade off between adaptation speed and learning accuracy: larger δ allows the algorithm to adapt to faster changing regimes in detriment of its steady-state accuracy [13, 19].

The above strategies require agents to operate *synchronously*. However, in many situations, agents do not share information with each other at every instant. To address this issue, we introduce a random combination matrix $\boldsymbol{A}_i$ into the operation of the algorithm.

3. CONVERGENCE OF BELIEFS

In the asynchronous case, we consider a random matrix $\boldsymbol{A}_i = [\boldsymbol{a}_{\ell k,i}]$, where we allow agents to be randomly absent when participating in the combination step in (2). Hence, the combination rule in (2) is replaced by:

$$\boldsymbol{\mu}_{k,i}(\theta) \propto \prod_{\ell \in \mathcal{N}_k} \left(\boldsymbol{\psi}_{\ell,i}(\theta)\right)^{\boldsymbol{a}_{\ell k,i}} \tag{4}$$

We allow for general forms of $\boldsymbol{A}_i$ as long as they satisfy the following conditions. Specific forms for $\boldsymbol{A}_i$ will be discussed in Section 4.

Assumption 1 (Conditions on asynchronous matrices). *$\boldsymbol{A}_i$ is always left stochastic and i.i.d. over time, and $\mathbb{E}[\boldsymbol{A}_i]$ is primitive, i.e., its underlying graph is strongly connected.* ■

Let us examine first the convergence behavior of the asynchronous strategy described by (1) and (4). Later, we consider the adaptive version consisting of (3) and (4).

To begin with, we introduce the vectors of log ratios $\boldsymbol{x}_i$ and $\boldsymbol{\nu}_i$ of dimension $K \times 1$:

$$\boldsymbol{x}_{k,i}(\theta) = \log \frac{L_k(\boldsymbol{\xi}_{k,i}|\theta^\star)}{L_k(\boldsymbol{\xi}_{k,i}|\theta)} \quad \text{(5a)}, \quad \boldsymbol{\nu}_{k,i}(\theta) = \log \frac{\boldsymbol{\mu}_{k,i}(\theta^\star)}{\boldsymbol{\mu}_{k,i}(\theta)} \quad \text{(5b)}$$

For these log-ratios, from now on, we will drop the dependence on θ for brevity, i.e., we will write $\boldsymbol{\nu}_{k,i}$ instead of $\boldsymbol{\nu}_{k,i}(\theta)$. Using these variables, we can rewrite (1) and (4) as a linear recursion in $\boldsymbol{\nu}_i$:

$$\boldsymbol{\nu}_i = \boldsymbol{A}_i^\top \boldsymbol{\nu}_{i-1} + \boldsymbol{A}_i^\top \boldsymbol{x}_i = \Big[\prod_{t=1}^{i} \boldsymbol{A}_t^\top\Big]\nu_0 + \sum_{t=1}^{i}\Big[\prod_{m=t}^{i} \boldsymbol{A}_m^\top\Big]\boldsymbol{x}_t \tag{6}$$

where $\prod_{t=1}^{i} \boldsymbol{A}_t^\top = \boldsymbol{A}_i^\top \boldsymbol{A}_{i-1}^\top \dots \boldsymbol{A}_1^\top$. Now, as in [14], let us define $w_0^{(k)} \triangleq e_k = [0, \dots, 0, 1, 0, \dots, 0]$ where the k-th entry of w_0 is 1. Then, computing the following inner product we get:

$$\begin{aligned}\boldsymbol{\nu}_{k,i} &= \left\langle \boldsymbol{\nu}_i, w_0^{(k)} \right\rangle = \boldsymbol{\nu}_i^\top w_0^{(k)} \\ &= \left\langle \Big[\prod_{t=1}^{i} \boldsymbol{A}_t^\top\Big]\nu_0, w_0^{(k)} \right\rangle + \left\langle \sum_{t=1}^{i}\Big[\prod_{m=t}^{i} \boldsymbol{A}_m^\top\Big]\boldsymbol{x}_t, w_0^{(k)} \right\rangle \\ &= \left\langle \nu_0, \Big[\prod_{t=1}^{i} \boldsymbol{A}_{i-t+1}\Big] w_0^{(k)} \right\rangle + \sum_{t=1}^{i}\left\langle \boldsymbol{x}_t, \Big[\prod_{m=t}^{i} \boldsymbol{A}_{i-m+t}\Big] w_0^{(k)} \right\rangle \end{aligned} \tag{7}$$

Then, let us define:

$$\boldsymbol{w}_t^{(k)} \triangleq \left[\prod_{m=0}^{t-1} \boldsymbol{A}_{i-m}\right] w_0^{(k)} \tag{8}$$

Notice that since $\boldsymbol{A}_i$ is left stochastic for all i, then $\prod_{m=0}^{t-1} \boldsymbol{A}_{i-m}$ is also a left stochastic matrix. It follows that $\boldsymbol{w}_t^{(k)}$ is a probability vector for all agents k and for all t. Substituting $\boldsymbol{w}_t^{(k)}$ into (7) and dividing both sides by i, we have:

$$\frac{1}{i}\boldsymbol{\nu}_{k,i} = \frac{1}{i}\big\langle \nu_0, \boldsymbol{w}_i^{(k)} \big\rangle + \frac{1}{i}\sum_{t=1}^{i} \big\langle \boldsymbol{x}_t, \boldsymbol{w}_{i-t+1}^{(k)} \big\rangle \tag{9}$$

The first term on the right hand side of (9) goes to 0 almost surely since both ν_0 and $\boldsymbol{w}_i^{(k)}$ are a.s. bounded by a finite value. We will show next that $\frac{1}{i}\boldsymbol{\nu}_{k,i} \xrightarrow{\text{a.s.}} \langle \pi, d \rangle$, i.e., the second term converges to $\langle \pi, d \rangle$; where π is the Perron vector of $\mathbb{E}[\boldsymbol{A}_i]$, and d is a vector whose entries correspond to the KL divergences[1]

$$d_k \triangleq D_{KL}\Big(L_k(\theta^\star) || L_k(\theta)\Big) = \mathbb{E}[\boldsymbol{x}_{k,i}] \tag{10}$$

i.e., the KL divergence [22] between $L_k(\xi|\theta^\star)$ and $L_k(\xi|\theta)$. First, we show in the next lemma that the ensemble average of the vectors $\{\boldsymbol{w}_t^{(k)}\}$ converges a.s. to π. The proof is tailored according to the proof in [14, Lemma 1]. The details are omitted here due to space limitations.

Lemma 1 (Convergence to Perron vector). *Under Assumption 1, it holds that for all agents $k = 1, 2, \dots, K$:*

$$\frac{1}{i}\sum_{t=1}^{i} \boldsymbol{w}_t^{(k)} \xrightarrow{\text{a.s.}} \pi \tag{11}$$

■

Before introducing the main convergence result, we present some usual assumptions in order to avoid singular behaviors.

Assumption 2 (Finite KL divergences). *For every agent k, and $\theta \in \Theta$, we have $D_{\mathrm{KL}}(L_k(\theta^\star) \| L_k(\theta)) < +\infty$.*

Assumption 3 (Positive initial beliefs). $\mu_{k,0}(\theta) > 0$ *for all agents $k = 1, 2, \dots, K$ and $\theta \in \Theta$.*

Assumption 4 (Global identifiability). *For each wrong hypothesis $\theta \neq \theta^\star$, there is at least one agent k that has strictly positive KL divergence, $D_{\mathrm{KL}}(L_k(\theta^\star) \| L_k(\theta)) > 0$.* ■

[1] Note that the expectation here is computed with respect to the randomness in the observations $\boldsymbol{\xi}_{k,i}$, while the expectation in Assumption 1 is computed with respect to the randomness in $\boldsymbol{A}_i$. Both sources of randomness are independent.

We are ready to introduce the truth learning result for the traditional social learning algorithm (1) and (4) under asynchronous communication.

Theorem 1 (**Learning under asynchronous social learning**). *Using* (1) *and* (4)*, under Assumptions 1, 2, 3, we have that:*

$$\frac{1}{i}\boldsymbol{\nu}_{k,i} \xrightarrow{\text{a.s.}} \langle \pi, d\rangle \tag{12}$$

Under Assumption 4, expression (12) *implies that all agents learn the truth as i grows with probability one, i.e.,*

$$\boldsymbol{\mu}_{k,i}(\theta^\star) \xrightarrow{\text{a.s.}} 1 \tag{13}$$

Proof sketch. Using Lemma 1, we can compute the inner products of both sides in (11) with d to show that $\frac{1}{i}\sum_{t=1}^{i}\langle d, \boldsymbol{w}_t^{(k)}\rangle \xrightarrow{\text{a.s.}} \langle \pi, d\rangle$. Using this intermediate result and similar arguments as in [14] under the Assumptions 1, 2, 3 we show that (12) holds. Then, under the Assumption 4, we can prove that (13) holds [14, Corollary 2]. ■

Consider next the adaptive social learning strategy, given by steps (2) and (3) in the synchronous case. In this case, the beliefs do not go to zero at the wrong hypothesis, as is the case with traditional social learning. Instead, despite this fact, the agents are still able to achieve consistent truth learning in that the belief is maximized at the location of the true hypothesis, i.e., it holds that

$$\theta^\star = \underset{\theta\in\Theta}{\operatorname{argmax}}\, \boldsymbol{\mu}_{k,i}(\theta) \tag{14}$$

This result is established in [13] for small δ. In the following, we will show that consistent truth learning continues to hold under asynchronous communication, and Assumption 1.

To see this, using the notation in (5a) and (5b), we rewrite (3) and (4) as the following linear recursion:

$$\boldsymbol{\nu}_i^{(\delta)} = (1-\delta)^i\left[\prod_{m=1}^{i}\boldsymbol{A}_m^\top\right]\nu_0 + \delta\sum_{t=1}^{i}(1-\delta)^{t-1}\left[\prod_{m=i-t+1}^{i}\boldsymbol{A}_m^\top\right]\boldsymbol{x}_{i-t+1} \tag{15}$$

Following similar arguments as in [13], we can show that $\boldsymbol{\nu}_i^{(\delta)}$ converges in distribution to a random vector $\widetilde{\boldsymbol{\nu}}^{(\delta)}$ as i goes to infinity. Proofs are omitted due to space constraints.

Lemma 2 (**Steady-state random belief ratio**). *Under Assumptions 1, 2 and 3, $\boldsymbol{\nu}_{k,i}^{(\delta)}$ converges in distribution to $\widetilde{\boldsymbol{\nu}}_k^{(\delta)}$ as i goes to infinity, where $\widetilde{\boldsymbol{\nu}}_k^{(\delta)}$ is the k-th element of the random vector defined by*

$$\widetilde{\boldsymbol{\nu}}^{(\delta)} \triangleq \delta\sum_{t=1}^{\infty}(1-\delta)^{t-1}\left[\prod_{m=1}^{t}\boldsymbol{A}_m^\top\right]\boldsymbol{x}_t \tag{16}$$

■

To characterize the steady-state random variable defined by (16) for any δ is a challenging task. We therefore follow the steps in [13] and focus on the behavior of this random variable as δ goes to zero. In the next theorem, we characterize the small-δ regime and show that the algorithm achieves consistent learning. Proofs are omitted.

Theorem 2 (**Learning under asynchronous adaptive social learning**). *Under Assumptions 1, 2 and 3, we have that*

$$\widetilde{\boldsymbol{\nu}}_k^{(\delta)} \xrightarrow[\delta\to 0]{\text{p}} \langle \pi, d\rangle \tag{17}$$

Furthermore, under Assumption 4, all agents consistently learn the truth, i.e.,

$$\lim_{\delta\to 0}\mathbb{P}\left(\underset{\theta\in\Theta}{\operatorname{argmax}}\,\boldsymbol{\mu}_{k,\infty}(\theta) = \theta^\star\right) = 1 \tag{18}$$

■

4. ASYNCHRONOUS PROTOCOLS

In this section, we discuss the form of the Perron vector π of $\mathbb{E}[\boldsymbol{A}_i]$, which governs the asymptotic convergence rate as revealed by (12). To that end, we introduce some formulations to construct appropriate $\boldsymbol{A}_i$, while satisfying the required conditions listed in Assumption 1. Thus, consider a situation in which some agent ℓ is absent with probability p_ℓ and introduce the Bernoulli random variable $\boldsymbol{\beta}_{\ell,i}$, which is 1 with probability p_ℓ and indicates that agent ℓ is absent at iteration i, i.e., it does not share any information with its neighbors. When agent ℓ is absent at iteration i, we zero out the combination weights $\boldsymbol{a}_{\ell k,i}$ for all k, except when $k=\ell$ (indicating that the agent does not communicate with its neighbors). We assume we start from an initial graph with a constant combination matrix A. There are at least two approaches to update the entries of A when agent ℓ is absent to construct $\boldsymbol{A}_i$. In the first approach, we add the missing combination weights from the other agents to the self loops of agent ℓ. More formally, we set:

$$\boldsymbol{a}_{\ell k,i} = \begin{cases}(1-\boldsymbol{\beta}_{\ell,i})a_{\ell k}, & \text{if } k\neq\ell\\ a_{kk} + \sum_{j\neq k}\boldsymbol{\beta}_{j,i}a_{\ell j}, & \text{if } k=\ell\end{cases} \tag{19}$$

Using (19), we can express $\boldsymbol{A}_i$ in terms of A as follows:

$$\boldsymbol{A}_i = A + \sum_{\ell=1}^{K}\boldsymbol{\beta}_{\ell,i}\sum_{k\neq\ell}a_{\ell k}[e_k e_k^\top - e_\ell e_k^\top] \tag{20}$$

Note that in (19) and (20), since the random variables $\boldsymbol{\beta}_{\ell,i}$ are independent over i, we also have that the $\boldsymbol{A}_i$ are independent over i. In the second approach, instead of adding the missing weights to the self loop of ℓ, we renormalize the columns of the combination matrix to add up to one:

$$\boldsymbol{A}_i = A + \sum_{\ell=1}^{K}\boldsymbol{\beta}_{\ell,i}\sum_{k\neq\ell}a_{\ell k}\left[\sum_{j\neq\ell}\frac{a_{jk}}{\sum_{r\neq\ell}a_{rk}}e_j e_k^\top - e_\ell e_k^\top\right] \tag{21}$$

Furthermore, if we were to assume that $\boldsymbol{A}_i$ is constructed according to (20), for the expected value of $\boldsymbol{A}_i$, we would get:

$$\mathbb{E}[\boldsymbol{A}_i]_{\ell k} = \begin{cases}(1-p_\ell)a_{\ell k}, & \text{if } k\neq\ell\\ a_{kk} + \sum_{j\neq k}p_\ell a_{\ell j}, & \text{if } k=\ell\end{cases} \tag{22}$$

so that

$$\mathbb{E}[\boldsymbol{A}_i] = A + \sum_{\ell=1}^{K}p_\ell\sum_{k\neq\ell}a_{\ell k}[e_k e_k^\top - e_\ell e_k^\top] \tag{23}$$

From (22) we note that, for $p_\ell < 1$, if $a_{\ell k} > 0$ it follows that $\mathbb{E}[\boldsymbol{A}_i]_{\ell k} > 0$. This means that the graph support of A is contained in the graph support of $\mathbb{E}[\boldsymbol{A}_i]$. Thus, if A is strongly connected and has a self loop, so does $\mathbb{E}[\boldsymbol{A}_i]$. Besides, since A is assumed left-stochastic, we can verify from (22) that so is $\mathbb{E}[\boldsymbol{A}_i]$. These facts imply that $\mathbb{E}[\boldsymbol{A}_i]$ is primitive with Perron eigenvector π.

5. LEARNING THE GRAPH TOPOLOGY

Let us now examine the inverse problem, where the aim is to discover the underlying combination matrix A by analyzing the information shared by the users over the network. In particular, in this section we are going to show that by formulating an appropriate optimization problem inspired by [20], we can solve this optimization problem to recover the so-called effective combination matrix of the network under asynchronous communication, $\mathbb{E}[\boldsymbol{A}_i]$. Then, we will show that under some assumptions, we can recover A from $\mathbb{E}[\boldsymbol{A}_i]$.

Here, we introduce $K \times (|\Theta| - 1)$ dimensional matrices of log-ratios, $\boldsymbol{\Lambda}_i, \boldsymbol{\mathcal{L}}_i$ as follows for the true hypothesis $\theta^\star$ and any arbitrary hypothesis $\theta_j \neq \theta^\star$:

$$[\boldsymbol{\Lambda}_i]_{kj} = \log \frac{\boldsymbol{\psi}_{k,i}(\theta^\star)}{\boldsymbol{\psi}_{k,i}(\theta_j)}, \quad [\boldsymbol{\mathcal{L}}_i]_{kj} = \log \frac{L_k(\boldsymbol{\xi}_{k,i}|\theta^\star)}{L_k(\boldsymbol{\xi}_{k,i}|\theta_j)} \tag{24}$$

We can thus write the following recursion by using the update rules (2) and (3) of adaptive social learning given as in [20]:

$$\boldsymbol{\Lambda}_i = (1-\delta)A^\top \boldsymbol{\Lambda}_{i-1} - \delta \boldsymbol{\mathcal{L}}_i \tag{25}$$

In asynchronous adaptive social learning, at each time i, agents communicate with each other according to the stochastic matrix $\boldsymbol{A}_i$. Hence, the recursion in (25) becomes

$$\boldsymbol{\Lambda}_i = (1-\delta)\boldsymbol{A}_i^\top \boldsymbol{\Lambda}_{i-1} - \delta \boldsymbol{\mathcal{L}}_i \tag{26}$$

We work under the assumption that we only observe publicly exchanged beliefs $\boldsymbol{\psi}_{k,i}$ and allow limited knowledge of likelihoods, namely, we assume we know the mean matrix $\mathbb{E}[\boldsymbol{\mathcal{L}}_i] = \bar{\mathcal{L}}$. Hence, the recursion above allows to define the cost function as follows:

$$Q(\widehat{A}; \boldsymbol{\Lambda}_i, \boldsymbol{\Lambda}_{i-1}) = \frac{1}{2}\|\boldsymbol{\Lambda}_i^\top - (1-\delta)\boldsymbol{\Lambda}_{i-1}^\top \widehat{A} - \delta \bar{\mathcal{L}}^\top\|_{\mathrm{F}}^2 \tag{27}$$

We can define the corresponding risk function as:

$$\min_{\widehat{A}} J(\widehat{A}) \triangleq \frac{1}{N}\sum_{i=1}^{N} J_i(\widehat{A}) \tag{28}$$

where

$$J_i(\widehat{A}) \triangleq \mathbb{E}Q(\widehat{A}; \boldsymbol{\Lambda}_i, \boldsymbol{\Lambda}_{i-1}) \tag{29}$$

since $\boldsymbol{\Lambda}_i$ has different statistics for different i. This leads to the stochastic gradient recursion

$$\widehat{\boldsymbol{A}}_i = \widehat{\boldsymbol{A}}_{i-1} + \alpha(1-\delta) \times \boldsymbol{\Lambda}_{i-1}(\boldsymbol{\Lambda}_i^\top - (1-\delta)\boldsymbol{\Lambda}_{i-1}^\top \widehat{\boldsymbol{A}}_{i-1} - \delta \bar{\mathcal{L}}^\top) \tag{30}$$

where α is the learning rate. We state the next relevant properties of the risk function without proof, due to space limitations.

Lemma 3 (Properties of the risk function). *The risk function in (29) denotes a strongly convex function with Lipschitz gradients. Furthermore, $\mathbb{E}[\boldsymbol{A}_i]$ is the unique minimizer of this risk function.* ■

Hence, according to Lemma 3, if we were to apply the algorithm in (30), we would find $\mathbb{E}[\boldsymbol{A}_i]$, which does not depend on i. Then, assuming the formulation in (20) holds, we can recover A from $\mathbb{E}[\boldsymbol{A}_i]$ by estimating the probability of absence of agents from the experimental data (by simply finding the absence ratio of agents) as $\widehat{p}_\ell$ and then doing the following conversion:

$$\frac{1}{1-\widehat{p}_\ell}\mathbb{E}[\boldsymbol{A}_i]_{\ell k} \to A_{\ell k}, \text{ for } \ell \neq k \tag{31}$$

Finally, for all k, we select A_{kk} such that the columns of A add up to 1.

6. EXPERIMENTS

To demonstrate the convergence of beliefs under asynchronous social learning, and to show how the Perron eigenvector π of $\mathbb{E}[\boldsymbol{A}_i]$ changes with the asynchronous behavior of the agents, we first consider a numerical example with a network of 3 agents with $|\Theta| = 2$ hypotheses. We construct the combination matrix A using the Metropolis rule, yielding a primitive matrix that is doubly stochastic, i.e., uniform Perron vector [17]. For this example, we assume that the construction of $\boldsymbol{A}_i$ in (20) holds. Then, we run the asynchronous social learning algorithm (i.e., (1) and (4)) for different probabilities of absence. As in [21], we consider binary observations $\boldsymbol{\xi}_{k,i} \in \{0,1\}$, and we define the likelihood functions for all agents k and hypotheses $\theta \in \Theta$ as:

$$L_k(\xi|\theta) = \mathbb{I}[\xi=0]\rho_{k,\theta} + \mathbb{I}[\xi=1](1-\rho_{k,\theta}) \tag{32}$$

where $\rho_{k,\theta} \in (0,1)$ are randomly generated. In this experiment, illustrated in the left panel of Fig. 2, we observe that all agents learn the truth, in accordance with (13). We also illustrate how the entries of π change with different probabilities of absence of agents, p_k, in the right panel of Fig. 2. This experiment shows that the Perron entries of agents with high probability of absence are smaller. The experiment thus agrees with the intuition that more active agents have on average a larger centrality and exert larger dominance on the asymptotic convergence rate of the convergence, $\langle \pi, d\rangle$.

In Fig. 3, we illustrate how well we can recover the true graph by following the steps explained in Section 5. For this example, we build an Erdős-Rényi graph of 12 nodes (agents) with edge probability $p_{\text{edge}} = 0.2$. Then we build the left stochastic and primitive combination matrix A using the averaging rule [17]. We set the step size of the model to $\delta = 0.1$, with a learning rate $\alpha = 0.5$. The likelihood model and the hypotheses set Θ are the same as in the previous simulation. Next, we run the graph learning algorithm in (30), where $\boldsymbol{A}_i$ is constructed according to (20). From this experiment, we see in the third panels in the top and bottom of Fig. 3 that we learn $\mathbb{E}[\boldsymbol{A}_i]$, which contains information on the "connectivity" of the nodes, i.e., the graph edges. After that step, using the changes explained in (31), we adjust the weights of the edges in the fourth panels of in the top and bottom of Fig. 3, and recover A.

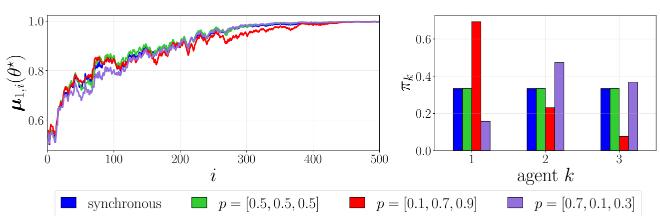

Fig. 2: Belief evolution and Perron vector entries under different probabilities of absence. *Left:* The belief concentrates on the true hypothesis over time for all setups, as suggested by Theorem 1. *Right:* There is an inverse relationship between Perron vector entry of agent k, π_k, and its probability of absence, p_k.

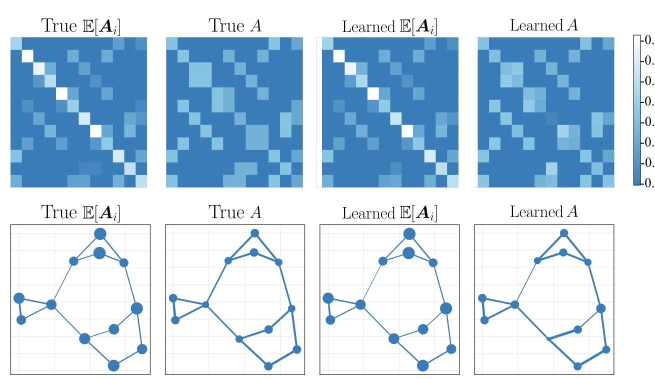

Fig. 3: Learning the graph topology with the descent algorithm in (30) and then recovering it with (31). *Top:* Combination matrices. *Bottom:* Illustrations of the graphs, where the edge width represents the combination weight between each pair of agents, and the node size represents the weight of each self-loop.

7. REFERENCES

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

[2] D. Acemoglu, M. A. Dahleh, I. Lobel, and A. Ozdaglar, "Bayesian learning in social networks," *The Review of Economic Studies*, vol. 78, no. 4, pp. 1201–1236, 2011.

[3] P. M. Djurić and Y. Wang, "Distributed Bayesian learning in multiagent systems: Improving our understanding of its capabilities and limitations," *IEEE Signal Processing Magazine*, vol. 29, no. 2, pp. 65–76, 2012.

[4] C. Chamley, A. Scaglione, and L. Li, "Models for the diffusion of beliefs in social networks: An overview," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 16–29, 2013.

[5] 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.

[6] X. Zhao and A. H. Sayed, "Learning over social networks via diffusion adaptation," in *Proc. Asilomar Conference on Signals, Systems and Computers*, 2012, pp. 709–713.

[7] 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.

[8] A. Lalitha, T. Javidi, and A. D. Sarwate, "Social learning and distributed hypothesis testing," *IEEE Transactions on Information Theory*, vol. 64, no. 9, pp. 6161–6179, 2018.

[9] M. Kayaalp, Y. Inan, E. Telatar, and A. H. Sayed, "On the arithmetic and geometric fusion of beliefs for distributed inference," *arXiv:2204.13741*, April 2022.

[10] P. Molavi, A. Tahbaz-Salehi, and A. Jadbabaie, "A theory of non-Bayesian social learning," *Econometrica*, vol. 86, no. 2, pp. 445–490, 2018.

[11] I. Arieli, Y. Babichenko, and S. Shlomov, "Robust non-Bayesian social learning," in *Proc. ACM Conference on Economics and Computation*, 2019, pp. 549–550.

[12] V. Matta, V. Bordignon, A. Santos, and A. H. Sayed, "Interplay between topology and social learning over weak graphs," *IEEE Open Journal of Signal Processing*, vol. 1, pp. 99–119, 2020.

[13] V. Bordignon, V. Matta, and A. H. Sayed, "Adaptive social learning," *IEEE Transactions on Information Theory*, vol. 67, no. 9, pp. 6053–6081, 2021.

[14] Y. Inan, M. Kayaalp, E. Telatar, and A. H. Sayed, "Social learning under randomized collaborations," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 115–120.

[15] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks—part i: Modeling and stability analysis," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 811–826, 2015.

[16] A. H. Sayed and X. Zhao, "Asynchronous adaptive networks," in *Cooperative and Graph Signal Processing*. Elsevier, 2018, pp. 3–68.

[17] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[18] ——, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[19] V. Bordignon, V. Matta, and A. H. Sayed, "Adaptation in online social learning," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2020, pp. 2170–2174.

[20] V. Shumovskaia, K. Ntemos, S. Vlaski, and A. H. Sayed, "Online graph learning from social interactions," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 1263–1267.

[21] ——, "Explainability and graph learning from social interactions," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 946–959, 2022.

[22] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.