Social Learning with Disparate Hypotheses

Konstantinos Ntemos, Virginia Bordignon, Stefan Vlaski, and Ali H. Sayed

Abstract—**In this paper we study the problem of social learning under multiple true hypotheses and *self-interested* agents. In this setup, each agent receives data that might be generated from a different hypothesis (or state) than the data other agents receive. In contrast to the related literature on social learning, which focuses on showing that the network achieves consensus, here we study the case where every agent is self-interested and wants to find the hypothesis that generates its own observations. To this end, we propose a scheme with adaptive combination weights and study the consistency of the agents' learning process. We analyze the asymptotic behavior of agents' beliefs under the proposed social learning algorithm and provide sufficient conditions that enable all agents to correctly identify their true hypotheses. The theoretical analysis is corroborated by numerical simulations.**

Index Terms—**social learning, information diffusion, disparate hypotheses**

I. Introduction

Social learning [1]–[8] refers to the distributed hypothesis testing problem where agents exchange information and aim at *learning* an unknown hypothesis of interest. Every agent has access to its own data, as well as to information provided by its neighbors. Agents update their *beliefs* (probability distributions over the possible states) according to the following two-step procedure. First, they perform a Bayesian update using their own data and then they fuse their own beliefs with the beliefs shared by their neighbors. Most of the existing literature focuses on studying the agents' belief convergence to the true hypothesis or to the hypothesis that better describes all the agents' observation models. In this way, network consensus (i.e., all agents' beliefs converge to the same hypothesis) is achieved.

The setup we study is related to the *conflicting hypothesis* setup considered in [2] where authors showed that agents will converge to the hypothesis that best describes all agents' observation models. In contrast, in this work we are interested in studying the problem where each agent wants to find its *individual* true hypothesis instead of converging to a consensus. There are many reasons for which this problem is interesting, especially when consensus does not describe a system's behavior in an accurate manner. Real-life social networks provide one such one example, where there are generally disparate opinions among the interacting parties. Another example is the scenario of a network of communicating classifiers that use the social learning protocol as in [9], with different subsets of agents aiming at classifying scenes from distinct underlying classes. Likewise, sensor networks where the agents receive observations generated from different sources is another example.

To tackle the problem, we use the idea that agents' cooperative beliefs are driven by agents' *private information*. Our contributions are the following. We propose a scheme with adaptive combination weights that utilizes the agents' private information and helps them in identifying other agents that aim at finding the same hypotheses. In this way, we extend the *log-linear* social learning algorithm [1] to the problem of *multiple* true hypotheses and self-interested agents. For the proposed algorithm we provide sufficient conditions under which all agents in the network manage to find their true hypotheses and demonstrate the agents' learning behavior via computer simulations.

The problem we study is also close to the problem of *multi-task learning over networks* studied in [10]–[14]. In [10], [11], every agent aims at estimating its true parameter vector, which might be different from the target vector of other agents. The authors devise an adaptive combination policy to correctly identify the neighbors with which agents should cooperate to correctly estimate their true parameter vectors. The agents adapt their combination weights based on a mean-square deviation (MSD) criterion and a diffusion least-mean squares (LMS) algorithm. A different approach was followed in [13], where every agent keeps a stand-alone LMS estimate (updated based only on the agent's own signals and not on information from neighbors). At every time instant, the agent performs a binary hypothesis test to decide whether each of its neighbors is searching for the same parameter vector. Related formulations followed in [12], [14]–[19] and references therein.

The aforementioned works focus on parameter estimation tasks. Here, we focus on the distributed hypothesis testing problem where every agent aims at identifying an underlying hypothesis of interest. Thus, our work is closer to the multi-task decision problem [20]. In [20], an LMS-type algorithm is devised. In contrast, here we study the social learning problem, where every agent performs local Bayesian updates before exchanging information with its neighbors. An interesting result of our analysis is the fact that *identifiability* (i.e., the ability of an agent to correctly distinguish among different hypotheses) plays a crucial role on the outcome of the learning process over the network.

A. Notation

We use boldface letters to denote random variables and normal letters to denote their realizations. We denote the KL divergence from distribution L_1 to distribution L_2 by

This work was performed while Dr. K. Ntemos was a post-doctoral scholar at EPFL. Authors V. Bordignon and A. H. Sayed are with the School of Engineering, EPFL, Switzerland, while S. Vlaski is with Imperial College, UK. This work was supported in part by the Swiss National Science Foundation grant 205121-184999. E-mails: kntemos@di.uoa.gr, virginia.bordignon@epfl.ch, s.vlaski@imperial.ac.uk, ali.sayed@epfl.ch.

$D_{KL}(L_1||L_2)$. We use the notation $\xrightarrow{a.s.}$ and $\xrightarrow{P.}$ to denote almost sure convergence and convergence in probability, respectively. Moreover, the notation $\mathbb{1}$ and $|\cdot|$ denote the all-ones vector and the cardinality of a set.

II. System Model

We assume a set $\mathcal{N} = \{1, \dots, N\}$ of agents interacting over a network, which is represented by an undirected graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$, where $\mathcal{E}$ includes bidirectional links between agents. The set of neighbors of an agent k (including k) is denoted by $\mathcal{N}_k$. The set of all possible hypotheses is denoted by $\Theta = \{\theta_1, \dots, \theta_M\}$.

We assume that each agent k has access to observations $\boldsymbol{\zeta}_{k,i} \in \mathcal{Z}_k$ at every time instant $i \geq 1$. Agent k also has access to the likelihood functions $L_k(\zeta_{k,i}|\theta)$, $\theta \in \Theta$. The signals $\boldsymbol{\zeta}_{k,i}$ are independent and identically distributed (i.i.d.) over time. In this work, the sets $\mathcal{Z}_k$ are assumed to be finite. We will use the notation $L_k(\theta)$ instead of $L_k(\boldsymbol{\zeta}_{k,i}|\theta)$ whenever it is clear from the context. Every agent k's true hypothesis $\theta^{(k)}$ is drawn according to some probability $\mathbb{P}(\boldsymbol{\theta}^{(k)})$ initially and remains unchanged throughout the process. Agent k's observations are generated according to the model

$$\boldsymbol{\zeta}_{k,i} \sim L_k\left(\boldsymbol{\zeta}_{k,i}|\boldsymbol{\theta}^{(k)} = \theta^{(k)}\right), \quad \boldsymbol{\theta}^{(k)} \in \Theta. \tag{1}$$

The states $\boldsymbol{\theta}^{(k)}$ are independent across agents, meaning that $\mathbb{P}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\theta}^{(\ell)}) = \mathbb{P}(\boldsymbol{\theta}^{(k)})\mathbb{P}(\boldsymbol{\theta}^{(\ell)})$ for all agents $k \neq \ell$.

Agents' observations are possibly generated by different hypotheses and each agent k aims at finding the realization $\theta^{(k)}$ of its true hypothesis $\boldsymbol{\theta}^{(k)} \in \Theta$ according to which the $\boldsymbol{\zeta}_{k,i}$ are generated. Agents share information with their neighbors in a distributed fashion. This information can be utilized to find the underlying true hypothesis by forming *beliefs*, which are probability distributions over the set of hypothesis Θ. Our algorithm is based on the log-linear social learning rule [1] where the agents update their beliefs, denoted by $\boldsymbol{\nu}_{k,i}$, in the following manner:

$$\boldsymbol{\varphi}_{k,i}(\theta) = \frac{L_k(\boldsymbol{\zeta}_{k,i}|\theta)\boldsymbol{\nu}_{k,i-1}(\theta)}{\sum_{\theta'} L_k(\boldsymbol{\zeta}_{k,i}|\theta')\boldsymbol{\nu}_{k,i-1}(\theta')}, \quad k \in \mathcal{N} \tag{2}$$

$$\boldsymbol{\nu}_{k,i}(\theta) = \frac{\prod_{\ell \in \mathcal{N}_k} (\boldsymbol{\varphi}_{\ell,i}(\theta))^{a_{\ell k}}}{\sum_{\theta'} \prod_{\ell \in \mathcal{N}_k} (\boldsymbol{\varphi}_{\ell,i}(\theta'))^{a_{\ell k}}}, \quad k \in \mathcal{N} \tag{3}$$

where $a_{\ell k}$ denotes the static (time-invariant) *combination weight* assigned by agent k to neighboring agent ℓ, satisfying $0 < a_{\ell k} \leq 1$, for all $\ell \in \mathcal{N}_k$, $a_{\ell k} = 0$ for all $\ell \notin \mathcal{N}_k$ and $\sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1$. Let A denote the *combination matrix*, which consists of all combination weights with $[A]_{\ell k} = a_{\ell k}$. Clearly, A is left-stochastic.

It is known that if agents use the above algorithm, under the assumption of a strongly connected network [21] (information flows from every agent to any other agent in the network and at least one agent has a self-loop, $a_{kk} > 0$), then the network achieves *consensus* [1]–[3], thus ruling out the possibility for agents with different true states to correctly identify them.

In order for agents to find their true state, they should evaluate over time if the received information is beneficial to them or not. This means that they have to decide how much *weight* they should give to the information received from their neighbors in the information aggregation step (3). One way to do so is to dynamically adjust the combination weights according to whether agents believe each one of their neighbors aims at finding the same state or not.

III. Self-Aware Social Learning

The idea is that an agent should gradually assign less weight to neighbors that seek different states. If an agent can identify its true state alone, then it might be better for that agent not to cooperate and just perform stand-alone Bayesian learning. In this way, it will be guaranteed to converge to its true hypothesis without being misled by other agents. However, some agents might not be able to find their true states alone. This happens when for an agent k, its true state $\theta^{(k)}$ is *observationally equivalent* to some other $\theta \neq \theta^{(k)}$. In that case this agent will be unable to find its true state without other agents' help. We define the set of states that are observationally equivalent to $\theta^{(k)}$ as follows.

Definition 1. (**Observationally equivalent states**). The set

$$\Theta_k^\star \triangleq \left\{\theta \in \Theta : L_k(\zeta_{k,i}|\theta) = L_k(\zeta_{k,i}|\theta^{(k)}), \quad \forall \zeta_k \in \mathcal{Z}_k, \forall i\right\} \tag{4}$$

is comprised of all states that are observationally equivalent to $\theta^{(k)}$ for an agent $k \in \mathcal{N}$. ■

Note that $\theta^{(k)}$ is always contained in $\Theta_k^\star$. Before we introduce the adaptive combination weights scheme, we provide a motivating example. In the network example presented in Fig. 1 the set of possible hypotheses is $\Theta = \{\theta_1, \theta_2, \theta_3\}$ and the true hypotheses of agents $1, 2, 3$ are $\theta_2, \theta_2, \theta_3$, respectively. However, agent 1 cannot distinguish between hypotheses θ_1 and θ_2. Since agent 1 communicates with both agents 2 and 3, it may not converge to hypothesis θ_2. However, if agent 1 over time realizes that agent 3's true hypothesis is θ_3 (i.e., it is different than its own true hypothesis), then it can cut off the link with agent 3 and find its true hypothesis with the aid of agent 2 (which can find θ_2 alone as $\Theta_2^\star = \{\theta_2\}$), provided that agent 2 also realizes that its true hypothesis is different from agent 3's true hypothesis and cuts off its link to agent 3 as well. Our goal is to devise an adaptive mechanism that enables agents to discriminate over time which agents aim at finding the same hypothesis with them against other agents.

To begin with, each agent can form a *local belief* about hypothesis $\boldsymbol{\theta}^{(k)}$ based *only on its own* observations $\boldsymbol{\zeta}_{k,1:i} = (\boldsymbol{\zeta}_{k,1}, \dots, \boldsymbol{\zeta}_{k,i})$ until time i. Local beliefs do not contain any (potentially) misleading information from other agents and they are given by

$$\boldsymbol{\pi}_{k,i}(\theta) = \mathbb{P}(\boldsymbol{\theta}^{(k)} = \theta|\boldsymbol{\zeta}_{k,1:i}), \quad \theta \in \Theta \tag{5}$$

where $\boldsymbol{\pi}_{k,i}$ is the posterior belief over $\boldsymbol{\theta}^{(k)}$ given the sequence of private observations of agent k. The local belief $\boldsymbol{\pi}_{k,i}$ can

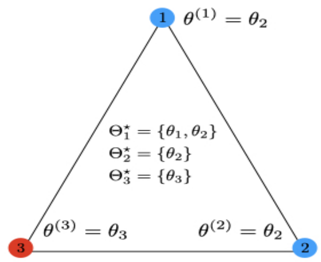

Fig. 1: A network example with three agents. The true state of agents $1, 2$ is θ_2, while the true state of agent 3 is θ_3.

be computed given $\boldsymbol{\pi}_{k,i-1}$ and $\boldsymbol{\zeta}_{k,i}$ recursively according to Bayes' rule:

$$\boldsymbol{\pi}_{k,i}(\theta) = \frac{L_k(\boldsymbol{\zeta}_{k,i}|\theta)\boldsymbol{\pi}_{k,i-1}(\theta)}{\sum_{\theta' \in \Theta} L_k(\boldsymbol{\zeta}_{k,i}|\theta')\boldsymbol{\pi}_{k,i-1}(\theta')}. \tag{6}$$

Now, we can design a scheme based on the local beliefs so that the weights assigned to every neighbor ℓ by agent k evolve according to the probability that the two distinct agents are trying to find the same hypothesis (i.e., $\theta^{(k)} = \theta^{(\ell)}$). Let us denote the event that the two agents have the same hypothesis by

$$\mathcal{S}_{k\ell} \triangleq \{\theta^{(k)} = \theta^{(\ell)}\} = \underset{\theta \in \Theta}{\cup} \mathcal{S}_{k\ell}^{\theta}, \quad k \neq \ell \tag{7}$$

where

$$\mathcal{S}_{k\ell}^{\theta} \triangleq \{\boldsymbol{\theta}^{(k)} = \theta \cap \boldsymbol{\theta}^{(\ell)} = \theta\}, \quad k \neq \ell \tag{8}$$

is the event that both agent k's and agent ℓ's true state is θ. Since the $\mathcal{S}_{k\ell}^{\theta}$ are disjoint events for different θ:

$$\mathbb{P}(\underset{\theta \in \Theta}{\cup} \mathcal{S}_{k\ell}^{\theta}) = \sum_{\theta \in \Theta} \mathbb{P}(\mathcal{S}_{k\ell}^{\theta}). \tag{9}$$

Obviously, the probability that agent k has the same state with itself is 1. Then, the weight each agent k may assign to its neighbor ℓ can be set to

$$\boldsymbol{a}_{\ell k,i} = [\boldsymbol{A}_i]_{\ell k} = \begin{cases} \frac{\mathbb{P}(\mathcal{S}_{k\ell}|\ \boldsymbol{\zeta}_{k,1:i}, \boldsymbol{\zeta}_{\ell,1:i})}{\boldsymbol{\sigma}_{k,i}}, & \text{if } \ell \in \mathcal{N}_k^\star \\ \frac{1}{\boldsymbol{\sigma}_{k,i}}, & \text{if } \ell = k \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $\mathcal{N}_k^\star \triangleq \mathcal{N}_k \setminus \{k\}$ is the set of neighbors of k without including k and $\boldsymbol{\sigma}_{k,i}$ is a normalizing factor to ensure that $\boldsymbol{A}_i$ is left-stochastic.

Construction (10) ensures that agent k incorporates information from agent ℓ in a manner that is proportional to the probability that agents k and ℓ are observing the same state. As agents gain confidence in their true state over time, this allows them to exclude inconsistent information, and collaborate only with agents who observe data that are generated from the same state they are observing. We first show that agents are able to efficiently compute $\mathbb{P}(\mathcal{S}_{k\ell}|\boldsymbol{\zeta}_{k,1:i}, \boldsymbol{\zeta}_{\ell,1:i})$ by simply exchanging their local beliefs (proofs are omitted due to space limitations).

Lemma 1. (Conditional probability of two agents sharing the same hypothesis). *The probability of two distinct agents k, ℓ having the same state conditioned on the joint observations $\boldsymbol{\zeta}_{k,1:i}, \boldsymbol{\zeta}_{\ell,1:i}$ is given by*

$$\mathbb{P}(\mathcal{S}_{k\ell}|\boldsymbol{\zeta}_{k,1:i}, \boldsymbol{\zeta}_{\ell,1:i}) = \sum_{\theta} \boldsymbol{\pi}_{k,i}(\theta)\boldsymbol{\pi}_{\ell,i}(\theta) \tag{11}$$

■

Utilizing Lemma 1, the normalizing factor is given by

$$\boldsymbol{\sigma}_{k,i} = 1 + \sum_{\ell \in \mathcal{N}_k^\star} \sum_{\theta \in \Theta} \boldsymbol{\pi}_{k,i}(\theta)\boldsymbol{\pi}_{\ell,i}(\theta). \tag{12}$$

Note that according to (10), we have $\boldsymbol{a}_{kk,i} > 0$ for all $i \geq 1$ and for all $k \in \mathcal{N}$. In order to account for the information from local beliefs, agents perform two parallel updates in our proposed *Self-Aware Social Learning* (SASL) algorithm. A non-cooperative update, where the local belief $\boldsymbol{\pi}_{k,i}$ is formed by using (6), which is then shared with every neighbor of k; and a social learning update. The novel part introduced in the social learning algorithm (2)-(3) is that the combination step utilizes the adaptive combination weights $\boldsymbol{A}_i$ instead of static weights. More specifically, every agent $k \in \mathcal{N}$ updates its cooperative belief $\boldsymbol{\mu}_{k,i}$ according to the following procedure:

$$\boldsymbol{\psi}_{k,i}(\theta) = \frac{L_k(\boldsymbol{\zeta}_{k,i}|\theta)\boldsymbol{\mu}_{k,i-1}(\theta)}{\sum_{\theta'} L_k(\boldsymbol{\zeta}_{k,i}|\theta')\boldsymbol{\mu}_{k,i-1}(\theta')}, \quad k \in \mathcal{N} \tag{13}$$

$$\boldsymbol{\mu}_{k,i}(\theta) = \frac{\prod_{\ell \in \mathcal{N}_k} \boldsymbol{\psi}_{\ell,i}^{\boldsymbol{a}_{\ell k,i}}(\theta)}{\sum_{\theta'} \prod_{\ell \in \mathcal{N}_k} \boldsymbol{\psi}_{\ell,i}^{\boldsymbol{a}_{\ell k,i}}(\theta')}, \quad k \in \mathcal{N}. \tag{14}$$

For simplicity, and since agents do not have any prior evidence on their true state, we impose the following assumption on the prior local beliefs $\pi_{k,0}(\theta)$ and prior cooperative beliefs $\mu_{k,0}(\theta)$.

Assumption 1. ***(Uniform prior beliefs)***. *The prior beliefs of all agents are uniform*

$$\pi_{k,0}(\theta) = \mu_{k,0}(\theta) = 1/|\Theta|, \quad k \in \mathcal{N}, \theta \in \Theta. \tag{15}$$

■

Moreover, we impose the following technical assumption [22].

Assumption 2. (Likelihood functions with full support). *$L_k(\zeta|\theta) > \alpha$ for some $\alpha > 0$ for all $\zeta \in \mathcal{Z}_k$, and for all $\theta \in \Theta$.* ■

From (11), it follows that the ability of agent k to correctly reject inconsistent information from its neighbors is driven by its ability to reject inconsistent states $\theta \notin \Theta_k^\star$ through $\boldsymbol{\pi}_{k,i}(\theta)$. The following result characterizes the asymptotic behavior of the adaptive combination weights.

Theorem 1. (Limiting behavior of the adaptive combination weights). *The adaptive combination weights exhibit the* *following limiting behavior as $i \to \infty$ for every agent $k \in \mathcal{N}$:*

$$\boldsymbol{a}_{\ell k,i} \xrightarrow{a.s.} \begin{cases} \frac{\eta_{k\ell}}{1+\sum_{\ell' \in \mathcal{N}_k^\star} \eta_{k\ell'}}, & \text{if } \ell \in \mathcal{N}_k^\star \\ 1 - \sum_{\ell'' \in \mathcal{N}_k^\star} \frac{\eta_{k\ell''}}{1+\sum_{\ell' \in \mathcal{N}_k^\star} \eta_{k\ell'}}, & \text{if } \ell = k \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where $\eta_{k\ell} \triangleq \frac{|\Theta_k^\star \cap \Theta_\ell^\star|}{|\Theta_k^\star||\Theta_\ell^\star|}$. ■.

We observe that if an agent k can identify its true hypothesis alone (i.e., $\Theta_k^\star = \{\theta^{(k)}\}$), then it will assign asymptotically positive weights only to the neighbors $\ell \in \mathcal{N}_k^\star$ for which $\theta^{(k)}$ is within their optimal hypothesis set (i.e., $\theta^{(k)} \in \Theta_\ell^\star$). We see here the implications of the identifiability capabilities of the agents. For example, if all agents can identify their true hypothesis alone (i.e., $\Theta_k^\star = \{\theta^{(k)}\}$ for all $k \in \mathcal{N}$), then the network will (asymptotically) decompose into components where every agent exchanges information only with the neighbors that aim at finding the same hypothesis. We observe from Theorem 1 that the combination matrix $\boldsymbol{A}_i$ converges to a limiting matrix A_∞, defined as

$$A_\infty \triangleq \lim_{i\to\infty} \boldsymbol{A}_i. \tag{17}$$

We can show that A_∞ is comprised of $S \in \mathbb{N}$ distinct strongly-connected components $A_{\infty,1}, \ldots, A_{\infty,S}$. Let us define the set $\mathcal{N}_s$, $s \in \{1, \ldots, S\}$ as the set of agents whose combination weights comprise $A_{\infty,s}$. Furthermore, let us define for $\mathcal{N}_s$ the *sub-network confidence* for a state $\theta \in \Theta$ as

$$C_s(\theta) \triangleq -\sum_{k \in \mathcal{N}_s} p_s(k) D_{KL}(L_k(\theta^{(k)}) || L_k(\theta)). \tag{18}$$

where $p_s(k)$ is the kth element of p_s, which is the Perron eigenvector of $A_{\infty,s}$, and let

$$\bar{\Theta}_s^\star \triangleq \left\{ \theta_s^\star \triangleq \arg\max_{\theta\in\Theta} C_s(\theta) \right\}. \tag{19}$$

This set is comprised of the hypotheses that best describe the observation models of the agents belonging to sub-network s weighted by their centrality.

The question of interest is when all agents manage to successfuly identify their true states. Whether an agent is able to learn its true state is dependent on the structure of the sub-network $\mathcal{N}_s$. From Theorem 1 we see that the structure of $\mathcal{N}_s$ depends on the identifiability capabilities of the agents, i.e., on the sets $\Theta_k^\star$ and on the graph topology given by $\mathcal{G}$. The following result provides conditions that guarantee that *every* agent in the network will find its true state.

Theorem 2. ***(Globally consistent learning)**. Under the proposed adaptive combination scheme, every agent $k \in \mathcal{N}$ learns its true state, meaning: $\boldsymbol{\mu}_{k,i}(\theta^{(k)}) \xrightarrow{P.} 1, \quad \forall k \in \mathcal{N}$ if both of the following conditions hold:*

$$\Theta_k^\star \cap \Theta_\ell^\star = \emptyset, \quad \forall k \in \mathcal{N}, \forall \ell \in \mathcal{N}_k^\star \text{ such that } \theta^{(k)} \neq \theta^{(\ell)} \tag{20}$$

$$\bigcap_{\ell \in \mathcal{N}_s} \Theta_\ell^\star = \{\theta^{(k)}\}, \quad \forall s \in \{1, \ldots, S\} \text{ such that } k \in \mathcal{N}_s. \tag{21}$$

■

Condition (20) ensures that for any two neighboring agents k, ℓ with different states both agents can rule out the state of the other agent based on their local beliefs (i.e., $\theta^{(k)} \notin \Theta_\ell^\star$ and $\theta^{(\ell)} \notin \Theta_k^\star$). This condition ensures that in every formed sub-network all agents share the same true state. Then, condition (21) ensures that in every sub-network the agents can collectively identify their true state.

IV. Experiments

In the following experiments we illustrate the agents' belief evolution for a network of $|\mathcal{N}| = 10$ agents. To facilitate the illustration of our results we assume that $|\mathcal{Z}_k| = 10$ for all $k \in \mathcal{N}$ and the set of possible hypotheses is $\Theta = \{\theta_1, \ldots, \theta_{10}\}$. In order to highlight the need for an adaptive combination mechanism in updating the cooperative beliefs, we compare the asymptotic beliefs of our proposed scheme to the traditional cooperative social learning solution (2) – (3) with a static (time-invariant) combination matrix A (every agent assigns uniform combination weights to its neighbors) as well as to non-cooperative learning (where beliefs are given by (6)).

We consider the following scenario. Some agents share the same true hypothesis and some agents face an identification problem. More specifically, the agents' true hypotheses are assigned as follows:

$$\theta^{(k)} = \begin{cases} \theta_1, & \text{if } k \in \{1, \ldots, 5\} \\ \theta_6, & \text{if } k \in \{6, \ldots, 10\}. \end{cases} \tag{22}$$

The agents' likelihood functions are constructed as follows. For agents 1 and 6 their likelihood functions are given by

$$L_k(\zeta_{k,i}^y|\theta_x) = \begin{cases} q_k \in (0,1), & \text{if } y = x, \\ \frac{1-q_k}{|\mathcal{Z}_k|-1}, & \text{otherwise} \end{cases} \tag{23}$$

for $x, y = 1, \ldots, |\mathcal{Z}_k|$. $\zeta_{k,i}^y \in \mathcal{Z}_k$ denotes the y^{th} observation of agent k. We set $q_k = 0.28$ for all $k \in \mathcal{N}$. Thus, agents $1, 6$ can identify their true hypotheses alone (i.e., $\Theta_1^\star = \{\theta_1\}, \Theta_6^\star = \{\theta_6\}$). The remaining agents cannot discriminate among some states and their likelihood functions are given as follows. For agents $2, 3, 4, 5$, $L_k(\zeta_{k,i}^y|\theta_x)$ is given by (23) for $x \geq 6$ and by

$$L_k(\zeta_{k,i}^y|\theta_x) = \frac{1}{|\mathcal{Z}_k|}, \quad \forall y = 1, \ldots, |\mathcal{Z}_k| \tag{24}$$

for $x \leq 5$. For agents $7, 8, 9, 10$, $L_k(\zeta_{k,i}^y|\theta_x)$ is given by (23) for $x \leq 5$ and by (24) for $x \geq 6$. In this case we see that (20) is satisfied and the network decomposes into two strongly connected components ($S = 2$), one consisting of agents 1 to 5 and one consisting of agents 6 to 10 (*see* bottom subfigure in Fig. 2). We can also verify from (23), (24) and for the selected value of q_k that condition (21) holds. As we see in the bottom subfigure in Fig. 2 (third row), all agents' beliefs converge to their true hypotheses, as expected by Theorem 2, while both cooperative and non-cooperative solutions lead to inconsistent learning for some agents as we can see in Fig. 2.

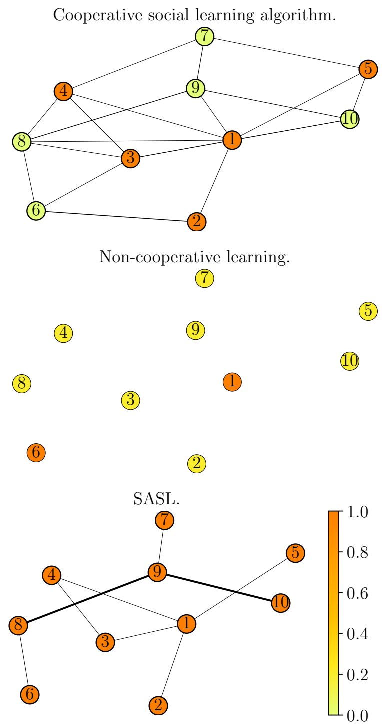

Fig. 2: Agents' steady-state beliefs (on true hypothesis $\theta^{(k)}$ for cooperative social learning (first row), non-cooperative learning (second row) and SASL algorithm (third row). *Colormap*: Agents in dark orange indicate that their beliefs on their true state are close to 1, while light green indicates beliefs close to 0.

V. Conclusions

In this work we investigated the problem of social learning with disparate true hypotheses and self-interested agents. Contrary to previous works that aim at showing that the network achieves consensus, here we investigated the scenario where every agent aims at learning its own true hypothesis. For this purpose, we devised an adaptive combination weights scheme and provided sufficient conditions under which every agent in the network successfully learns its true hypothesis.

References

[1] 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.

[2] 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.

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

[4] V. Bordignon, V. Matta, and A. H. Sayed, "Social learning with partial information sharing," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, pp. 5540–5544, 2020.

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

[7] 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*, vol. 66, no. 9, pp. 4084–4100, 2020.

[8] K. Ntemos, V. Bordignon, S. Vlaski, and A. H. Sayed, "Social learning under inferential attacks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5479–5483, 2021.

[9] V. Bordignon, S. Vlaski, V. Matta, and A. H. Sayed, "Network classifiers based on social learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Canada, pp. 5185–5189, 2021.

[10] J. Chen, C. Richard, and A. H. Sayed, "Adaptive clustering for multitask diffusion networks," in *Proc. European Signal Processing Conference (EUSIPCO)*, Nice, France, pp. 200–204, 2015.

[11] ——, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, 2015.

[12] J. Plata-Chaves, A. Bertrand, M. Moonen, S. Theodoridis, and A. M. Zoubir, "Heterogeneous and multitask wireless sensor networks—algorithms, applications, and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 450–465, 2017.

[13] X. Zhao and A. H. Sayed, "Distributed clustering and learning over networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3285–3300, 2015.

[14] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed, "Multitask learning over graphs: An approach for distributed, streaming machine learning," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 14–25, 2020.

[15] F. K. Teklehaymanot, M. Muma, B. Béjar, P. Binder, A. Zoubir, and M. Vetterli, "Robust diffusion-based unsupervised object labelling in distributed camera networks," in *Proc. AFRICON 2015*, Addis Ababa, Ethiopia, 2015, pp. 1–6, 2015.

[16] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks—part i: Sequential node updating," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5277–5291, 2010.

[17] ——, "Distributed adaptive node-specific signal estimation in fully connected sensor networks—part ii: Simultaneous and asynchronous node updating," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5292–5306, 2010.

[18] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based lms for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3448–3460, 2015.

[19] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based lms for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5382–5397, 2014.

[20] S. Marano and A. H. Sayed, "Decision learning and adaptation over multi-task networks," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2873–2887, 2021.

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

[22] A. Nedić, A. Olshevsky, and C. A. Uribe, "Nonasymptotic convergence rates for cooperative learning over time-varying directed graphs," in *Proc. American Control Conference (ACC)*, Chicago, IL, pp. 5884–5889, 2015.