SOCIAL LEARNING WITH ADAPTIVE MODELS

Marco Carpentiero[⋆] *Virginia Bordignon*[†] *Vincenzo Matta*[⋆] *Ali H. Sayed*[†]

[⋆] DIEM, University of Salerno, Fisciano (SA), Italy
[†] EPFL, School of Engineering, CH-1015 Lausanne, Switzerland

ABSTRACT

In social learning, a network of agents assigns probability scores (*beliefs*) to some hypotheses of interest, based on the observation of streaming data. First, each agent updates *locally* its belief with the information extracted from the current data through a suitable likelihood model. Then, these beliefs are diffused across the network, and the agents aggregate the beliefs received from their neighbors by means of a pooling rule. This work studies social learning in the context of *fully online* problems, where the true hypothesis *and* the likelihood models can drift over time. Traditional social learning fails to address both cases. To overcome this limitation, we propose the doubly adaptive social learning (A^2SL) strategy, which infuses traditional social learning with the necessary adaptation capabilities to face drifts in the hypotheses and/or models. The A^2SL strategy achieves this goal by employing two adaptation stages, and we show that all agents learn well (i.e., they end up placing full belief mass on the correct hypothesis) in the regime of small adaptation parameters.

***Index Terms*—** Social learning, adaptation and learning, belief and opinion formation, online learning, model drift.

1. INTRODUCTION AND RELATED WORK

Social learning is a popular paradigm for collaborative opinion formation, where a group of agents assign probability scores (*beliefs*) to some hypotheses of interest, based on private streaming data and the beliefs exchanged with their neighbors [1–7]. Traditional social learning algorithms have been extensively studied in the literature, and have been shown to offer provable learning guarantees: under reasonable technical conditions, each agent ends up placing all the probability mass on the true underlying hypothesis that gives rise to the data [7–13].

An increasing number of applications is focused on *online* settings, which require the social learning strategy to be able to react promptly to the two inherent sources of non-stationarity of the inferential problem: *drifts in the true hypothesis and in the likelihood models*. Traditional social learning does not exhibit any adaptation in both domains and is therefore unreliable in online settings.

The issue of drifting hypotheses can be managed by the recently proposed *adaptive* social learning (ASL) paradigm [14, 15], which infuses traditional social learning with adaptation by means of an *adaptive update step*. However, the ASL strategy requires that the likelihood models are known beforehand.

The issue of unknown models has been recently addressed in the context of social *machine* learning (SML) [16, 17] and social learning with uncertain models [18]. In these works, the agents use *training data* to learn the decision models (e.g., the likelihood ratios), which are then employed to perform social learning over a stream of *prediction data*. However, while these approaches remove the need for prior knowledge of likelihood models, they still do not provide adaptation to model drifts, since training is performed *offline*.

In this work we propose a novel, *doubly adaptive* social learning strategy, nicknamed A^2SL, which enables adaptation in the domain of the *hypotheses and model drifts*, and is therefore suited to *fully online* applications. Once turned on, the A^2SL algorithm can run virtually forever, with no need of resets or re-tuning stages, since it automatically adapts to variations in the training or prediction data. We address the realistic setting where there exists no hard separation between the time epochs where training and prediction are performed. In our model, these phases stay concurrently active, and at each time instant new training or prediction data can be observed, in an asynchronous manner.

The A^2SL strategy instills adaptation by using two adaptation stages: one stage is a stochastic gradient descent (SGD) algorithm, which features a *constant* step-size as an adaptation parameter to perform online model training. The other stage is an adaptive belief update, ruled by another adaptation parameter to balance old knowledge (stored in the past beliefs) and new knowledge (extracted from the current prediction data). We will show that the A^2SL strategy learns consistently, in the sense that the probability that (at any agent) the maximum belief mass concentrates on the correct hypothesis converges to 1 as the adaptation parameters go to zero. We corroborate the theoretical results by showing the effectiveness of the A^2SL strategy on a distributed classification problem using real data.

Notation. We denote random variables with bold font. The operator col$\{\cdot\}$ stacks its column-vector entries in a single column. For a nonnegative function $f(y)$ with positive argument y, the notation $f(y) = O(y)$ means that $f(y) \leq c\,y$ for all $y \leq y_0$, for some positive values c and y_0. The symbols $\mathbb{E}$ and $\mathbb{P}$ denote expectation and probability, respectively.

2. BACKGROUND

Consider H hypotheses belonging to the set $\Theta = \{\theta_1, \theta_2, \ldots, \theta_H\}$. Each agent $k = 1, \ldots, K$, at each time $t = 1, 2, \ldots$, observes some *prediction data* $\boldsymbol{x}_{k,t} \in \mathbb{R}^{M_k}$, and is equipped with a likelihood model linking the hypotheses to the data:

$$\ell_k(x_k|\theta), \qquad \text{for } x_k \in \mathbb{R}^{M_k} \text{ and } \theta \in \Theta. \tag{1}$$

More precisely, $\ell_k(x_k|\theta)$ is a likelihood function when regarded as a function of θ for a fixed x_k. For a fixed θ, it represents the generative model of the data x_k corresponding to that θ, e.g., it can be a probability mass function or a probability density function.

The goal of the agents is to assign a probability score to each hypothesis. The scores assigned by agent k at time t form the *belief vector* $\boldsymbol{\mu}_{k,t} = [\boldsymbol{\mu}_{k,t}(\theta_1), \ldots, \boldsymbol{\mu}_{k,t}(\theta_H)]$, with $\boldsymbol{\mu}_{k,t}(\theta) \geq 0$ and

This work was supported in part by grant 205121-184999 from the Swiss National Science Foundation (SNSF).

$\sum_{\theta\in\Theta}\boldsymbol{\mu}_{k,t}(\theta)=1$. In traditional social learning, the beliefs are constructed with the following recursion, initialized by some deterministic belief vectors $\mu_{k,0}$ [10–13]:

$$\boldsymbol{\psi}_{k,t}(\theta)=\frac{\boldsymbol{\mu}_{k,t-1}(\theta)\ell_k(\boldsymbol{x}_{k,t}|\theta)}{\sum_{\theta'\in\Theta}\boldsymbol{\mu}_{k,t-1}(\theta')\ell_k(\boldsymbol{x}_{k,t}|\theta')}\propto\boldsymbol{\mu}_{k,t-1}(\theta)e^{d_k(\boldsymbol{x}_{k,t};\theta)} \tag{2a}$$

$$\boldsymbol{\mu}_{k,t}(\theta)\propto\prod_{j=1}^{K}[\boldsymbol{\psi}_{j,t}(\theta)]^{a_{jk}} \tag{2b}$$

where the symbol $\propto$ hides the constant necessary to make $\boldsymbol{\psi}_{k,t}=[\boldsymbol{\psi}_{k,t}(\theta_1),\ldots,\boldsymbol{\psi}_{k,t}(\theta_H)]$ and $\boldsymbol{\mu}_{k,t}$ probability vectors. The RHS of (2a) is obtained by dividing $\ell_k(\boldsymbol{x}_{k,t}|\theta)$ and $\ell_k(\boldsymbol{x}_{k,t}|\theta')$ by $\ell_k(\boldsymbol{x}_{k,t}|\theta_H)$ and introducing the *decision statistic or model*:[1]

$$d_k(x_k;\theta)\triangleq\log\frac{\ell_k(x_k|\theta)}{\ell_k(x_k|\theta_H)}. \tag{3}$$

Without loss of generality, we divided by $\ell_k(x_k|\theta_H)$, but we can use any hypothesis as "pivot". Note that $d_k(x_k;\theta_H)=0$ by definition.

Step (2a) produces an *intermediate belief* $\boldsymbol{\psi}_{k,t}(\theta)$ by performing a *Bayesian update* based on the new local observation $\boldsymbol{x}_{k,t}$. In step (2b), each agent implements a pooling rule to combine the intermediate beliefs of the other agents. Specifically, agent k computes a weighted geometric average (scaled to obtain a valid probability vector) where the intermediate belief of agent j is raised to a weight a_{jk}. The weights are conveniently arranged into a combination matrix $A=[a_{jk}]$ that must be left-stochastic, which means that $a_{jk}\geq 0$ and $\sum_{j=1}^{K}a_{jk}=1$ [19, 20].

When $a_{jk}=0$, agent k does not receive information from agent j. Conversely, agent k aggregates the beliefs received from the agents j for which $a_{jk}>0$, which are called *neighbors* (of k). Therefore, the combination matrix describes through a weighted graph the communication structure that links the agents in the network. In our treatment, we consider the following assumption, which is standard in social learning theory.

Assumption 1 (Primitive Combination Matrix [19–21]). *We assume that the $K\times K$ left-stochastic matrix A is primitive, which means that it is irreducible (i.e., in the graph associated with A, for all j,k there is a path starting at j and ending at k) and has a single eigenvalue on the unit circle. From the Perron-Frobenius theorem, irreducibility implies that A has an eigenvector $v=[v_1,\ldots,v_K]^\top$ (called Perron eigenvector) that satisfies the following conditions:*

$$Av=v,\qquad v_k>0\ \text{ for all } k,\qquad \sum_{k=1}^{K}v_k=1. \tag{4}$$

Moreover, since A is primitive, we also have the convergence:

$$\lim_{t\to\infty}A^t=[\underbrace{v,\ldots,v}_{K\text{ times}}]. \tag{5}$$

□

3. ADAPTIVE MODEL LEARNING

To implement the update (2a), the agents should know exactly the decision statistics. This assumption is unrealistic, especially in dynamic environments where the models can drift over time. To overcome this limitation, in this work we allow agents to *learn and track* the decision statistics by exploiting the clues contained in a stream of *training samples*, according to the supervised classification paradigm [20]. Specifically, consider a collection of independent and identically distributed (iid) training samples $(\widehat{\boldsymbol{x}}_{k,t},\widehat{\boldsymbol{\theta}}_{k,t})$, a.k.a. (feature, label) pairs. We assume that the labels $\widehat{\boldsymbol{\theta}}_{k,t}$ in the training set are uniformly distributed across the hypotheses, whereas the features $\widehat{\boldsymbol{x}}_{k,t}$ corresponding to a label $\widehat{\boldsymbol{\theta}}_{k,t}=\theta$ are drawn from the generative model $\ell_k(x_k|\theta)$. Thus, the posterior probability that rules the training data is, for $x_k\in\mathbb{R}^{M_k}$ and $\theta\in\Theta$:

$$p_k(\theta|x_k)=\frac{\ell_k(x_k|\theta)}{\sum_{\theta'\in\Theta}\ell_k(x_k|\theta')}=\frac{e^{d_k(x_k;\theta)}}{\sum_{\theta'\in\Theta}e^{d_k(x_k;\theta')}}, \tag{6}$$

where in the last equality we use (3). The exact decision model $d_k(x_k;\theta)$ is unknown, and must be learned from the training set. In supervised classification, one looks for an approximate posterior:

$$\widehat{p}_k(\theta|x_k)=\frac{e^{\widehat{d}_k(x_k;\theta)}}{\sum_{\theta'\in\Theta}e^{\widehat{d}_k(x_k;\theta')}}, \tag{7}$$

where the approximate decision model $\widehat{d}_k(x_k;\theta)$ is chosen from some admissible family of functions. In this work we consider the standard (multiclass) logistic regression model [20]:

$$\widehat{d}_k(x_k;\theta)=h(x_k)\,w_k(\theta),\quad h(x_k)\triangleq(\mathrm{col}\{x_k,1\})^\top, \tag{8}$$

where $h(x_k)$ is the augmented *row* vector that adds to $x_k^\top$ a dummy entry,[2] and $w_k(\theta)$ is a parameter vector of dimension $(M_k+1)\times 1$. We set $w_k(\theta_H)=0$ to enforce the condition $\widehat{d}_k(x_k;\theta_H)=0$. It is convenient to aggregate the vectors $w_k(\theta)$, for $\theta\neq\theta_H$, into

$$w_k=\mathrm{col}\Big\{w_k(\theta_1),\ldots,w_k(\theta_{H-1})\Big\}\in\mathbb{R}^{(M_k+1)(H-1)}. \tag{9}$$

To select a decision model, we must choose a parameter w_k by optimizing some performance metric. One common choice is to minimize the regularized cross-entropy cost function [20]:

$$J_k(w_k)=\mathbb{E}\Big[\underbrace{-\log\widehat{p}_k\left(\widehat{\boldsymbol{\theta}}_{k,t}|\widehat{\boldsymbol{x}}_{k,t}\right)+\frac{\rho}{2}\|w_k\|^2}_{\substack{\triangleq Q\left(w_k;\widehat{\boldsymbol{x}}_{k,t},\widehat{\boldsymbol{\theta}}_{k,t}\right)\\ \text{regularized log-loss function}}}\Big]. \tag{10}$$

where $\rho>0$ is the regularization parameter. Note that the approximate posterior in (7) depends implicitly on the vector w_k. It can be shown that $J_k(w_k)$ admits a unique minimizer $w_k^\star$ [19, 20]. Unfortunately, in the considered setting, $w_k^\star$ cannot be computed exactly. In fact, the cost function $J_k(w_k)$ is unknown, since the distribution of the (feature, label) pairs in the training set is itself unknown. Moreover, we want to solve the optimization problem in an adaptive manner, because if the models governing the data in the training set change over time, we want to track the drifts. One common choice to achieve this goal is the *stochastic gradient descent* algorithm with *constant* step-size [19, 20], which is defined by the recursion:

$$\boldsymbol{w}_{k,t}=\boldsymbol{w}_{k,t-1}-\eta\,\nabla Q\left(\boldsymbol{w}_{k,t-1};\widehat{\boldsymbol{x}}_{k,t},\widehat{\boldsymbol{\theta}}_{k,t}\right), \tag{11}$$

where the step-size $\eta>0$ scales the gradient (computed with respect to the first argument) of the log-loss function. It is possible to

[1] For nonsingular problems, the likelihood ratio is almost-surely nonzero.

[2] This is a standard choice to incorporate an offset term, i.e., the last entry of $w_k(\theta)$, into the regression model (8).

show that the SGD algorithm approximates well the minimizer $w_k^\star$ for sufficiently large t and small η, in particular [19, 20]:

$$\limsup_{t\to\infty} \mathbb{E}\left[\|\boldsymbol{w}_{k,t} - w_k^\star\|^2\right] = O(\eta). \tag{12}$$

In the following, we set conventionally $\boldsymbol{w}_{k,t}(\theta_H) = w_k^\star(\theta_H) = 0$.

4. DOUBLY ADAPTIVE STRATEGY

The A^{2}SL strategy that we propose in this work consists of the following four steps that are iteratively performed by each agent k at each time instant $t \geq 1$ (with initial vectors $w_{k,0}$ and $\mu_{k,0}$):

$$\boldsymbol{w}_{k,t} = \boldsymbol{w}_{k,t-1} - \eta\, \nabla Q\left(\boldsymbol{w}_{k,t-1}; \widehat{\boldsymbol{x}}_{k,t}, \widehat{\boldsymbol{\theta}}_{k,t}\right) \boldsymbol{\alpha}_{k,t}^{\mathrm{tr}} \tag{13a}$$

$$\widehat{\boldsymbol{d}}_{k,t}(\boldsymbol{x}_{k,t};\theta) = h(\boldsymbol{x}_{k,t})\, \boldsymbol{w}_{k,t}(\theta) \tag{13b}$$

$$\boldsymbol{\psi}_{k,t}(\theta) \propto \boldsymbol{\mu}_{k,t-1}^{1-\delta}(\theta) \exp\left\{\widehat{\boldsymbol{d}}_{k,t}(\boldsymbol{x}_{k,t};\theta)\, \boldsymbol{\alpha}_{k,t}^{\mathrm{pr}}\right\} \tag{13c}$$

$$\boldsymbol{\mu}_{k,t}(\theta) \propto \prod_{j=1}^{K} [\boldsymbol{\psi}_{j,t}(\theta)]^{a_{jk}} \tag{13d}$$

– *Model Parameter Update*. In (13a), each agent k performs an SGD iteration to update the parameter vector $\boldsymbol{w}_{k,t}$ after observing the training sample $(\widehat{\boldsymbol{x}}_{k,t}, \widehat{\boldsymbol{\theta}}_{k,t})$. In comparison to (11), we have added the scalar $\boldsymbol{\alpha}_{k,t}^{\mathrm{tr}}$, which is assumed to be a Bernoulli random variable taking value 1 with probability $q_k^{\mathrm{tr}} > 0$. When $\boldsymbol{\alpha}_{k,t}^{\mathrm{tr}} = 0$, no training sample at time t is observed and the parameter vector is not updated. The insertion of this variable is important because we are using a common time axis for the training and prediction phases, and it is not realistic to assume that these phases are synchronous. In other words, at a given time instant, we can observe a new training sample, and/or a new prediction sample, or no samples at all.
– *Model Computation*. In (13b), each agent k adjusts the current decision statistic by using the updated vector $\boldsymbol{w}_{k,t}$.
– *Adaptive Bayesian Update*. In (13c), the agents update their intermediate beliefs according to the adaptive rule proposed in [14]. Agent k uses an *adaptation parameter* $\delta \in (0,1)$ to discount its own belief, in order to give more importance to new observations and enable reaction to drifts. For the same reasons discussed in relation to the training samples, we assume that agent k observes the prediction sample $\boldsymbol{x}_{k,t}$ with probability $q_k^{\mathrm{pr}} > 0$. When no fresh prediction sample is observed, the likelihood is not informative. To capture this behavior, in (13c) we introduce as a multiplicative factor the Bernoulli random variable $\boldsymbol{\alpha}_{k,t}^{\mathrm{pr}}$ (with success probability q_k^{pr}).
– *Combination Step*. In (13d), the agents update their beliefs combining the intermediate updates received from their neighbors.

The characterization of the A^{2}SL strategy will be carried out under the following standard procedure adopted in the theory of adaptation and learning [14, 19, 20]. Consider an arbitrary time instant t_0, with some drift occurring in the data and/or models from $t_0 + 1$ onward. Given a realization of the training and prediction data until t_0, the belief vectors μ_{k,t_0} and estimated parameters w_{k,t_0} store the knowledge accumulated by the agents until t_0, and are the only quantities necessary for the algorithm (13a)–(13d) to carry on. The aim of the theoretical analysis is to characterize the system evolution over a stationary interval starting from $t_0 + 1$, and to establish how much time is necessary to adapt to the new conditions (*transient analysis*) and which learning performance is achieved as time progresses (*steady-state analysis*). For space limitations, in this work we report only the results pertaining to the steady-state analysis. For convenience, we set $t_0 = 0$.

Assumption 2 (Data Properties). *For each agent k, the random observations $\boldsymbol{x}_{k,t}$ are iid over time, have finite second moment, and are generated from the model $\ell_k(x|\theta_0)$, where θ_0 is the actual hypothesis in force from $t > 0$. Prediction and training data are independent, and the Bernoulli random variables $\boldsymbol{\alpha}_{k,t}^{\mathrm{tr}}$ and $\boldsymbol{\alpha}_{k,t}^{\mathrm{pr}}$ are iid over time and across the agents, they are mutually independent, as well as independent of the training and prediction data.* □

Let each agent k at time t make its decision by choosing the hypothesis that maximizes the belief vector $\boldsymbol{\mu}_{k,t}$. Then, the instantaneous error probability at agent k is defined as:

$$p_{k,t} \triangleq \mathbb{P}\left[\theta_0 \neq \arg\max_{\theta\in\Theta} \boldsymbol{\mu}_{k,t}(\theta)\right] = \mathbb{P}\Big[\exists \theta \neq \theta_0 : \boldsymbol{\beta}_{k,t}(\theta) \leq 0\Big], \tag{14}$$

where we introduce the log belief ratio $\boldsymbol{\beta}_{k,t}(\theta) \triangleq \log \frac{\boldsymbol{\mu}_{k,t}(\theta_0)}{\boldsymbol{\mu}_{k,t}(\theta)}$. In Theorem 1, we establish that $p_{k,t}$ can be kept small, for sufficiently large t and small δ and η, under a suitable *identifiability* condition.

To introduce the identifiability condition, consider the decision statistic $d_k^\star(x_k;\theta)$ and the corresponding approximate posterior $p_k^\star(\theta|x_k)$ constructed with the optimal parameter $w_k^\star$ minimizing the cost function $J_k(w_k)$ in (10). From (7) and (8), we can write:

$$\begin{aligned}\log \frac{p_k^\star(\theta_0|x_k)}{p_k^\star(\theta|x_k)} &= d_k^\star(x_k;\theta_0) - d_k^\star(x_k;\theta) \\ &= h(x_k)\Big(w_k^\star(\theta_0) - w_k^\star(\theta)\Big).\end{aligned} \tag{15}$$

If this log ratio is positive, the estimated posterior leads to a correct decision. Assume that the log ratio is positive *on average*, namely,

$$\mathbb{E}\left[d_k^\star(\boldsymbol{x}_{k,t};\theta_0) - d_k^\star(\boldsymbol{x}_{k,t};\theta)\right] = \mathbb{E}[h(\boldsymbol{x}_{k,t})]\Big(w_k^\star(\theta_0) - w_k^\star(\theta)\Big) > 0. \tag{16}$$

To capture the significance of (16), consider the single-agent (SA) version of (2a)–(2b) (i.e., ignore the combination step and set $\boldsymbol{\psi}_{k,t}^{\mathrm{SA}} = \boldsymbol{\mu}_{k,t}^{\mathrm{SA}}$) and use $d_k^\star(x_k;\theta)$ in place of $d_k(x_k;\theta)$, obtaining:

$$\boldsymbol{\mu}_{k,t}^{\mathrm{SA}}(\theta) \propto \boldsymbol{\mu}_{k,t-1}^{\mathrm{SA}}(\theta) e^{d_k^\star(\boldsymbol{x}_{k,t};\theta)} = \mu_{k,0}^{\mathrm{SA}}(\theta) e^{\sum_{m=1}^{t} d_k^\star(\boldsymbol{x}_{k,m};\theta)}. \tag{17}$$

Computing the log belief ratio, by the law of large numbers we get:

$$\lim_{t\to\infty} \frac{1}{t} \log \frac{\boldsymbol{\mu}_{k,t}^{\mathrm{SA}}(\theta_0)}{\boldsymbol{\mu}_{k,t}^{\mathrm{SA}}(\theta)} = \mathbb{E}\left[d_k^\star(\boldsymbol{x}_{k,t};\theta_0) - d_k^\star(\boldsymbol{x}_{k,t};\theta)\right], \tag{18}$$

with probability 1. This shows that, under (16), the true hypothesis can be perfectly identified by agent k over an infinite stream of data. For this reason, we refer to (16) as a *local identifiability* condition.

In our *social* learning framework, where the agents learn cooperatively, local identifiability at each individual agent can be replaced by the following less stringent condition.

Assumption 3 (Global Identifiability). *We say that global identifiability holds when, for all pairs (θ, θ_0) with $\theta \neq \theta_0$:*

$$\beta_{\mathrm{net}}(\theta) \triangleq \sum_{k=1}^{K} v_k\, q_k^{\mathrm{pr}}\, \mathbb{E}[h(\boldsymbol{x}_{k,t})]\Big(w_k^\star(\theta_0) - w_k^\star(\theta)\Big) > 0. \tag{19}$$

□

A useful interpretation of condition (19) can be obtained by introducing the aggregate decision statistic across the agents:

$$\sum_{k=1}^{K} v_k\, \boldsymbol{\alpha}_{k,t}^{\mathrm{pr}}\, d_k^\star(\boldsymbol{x}_{k,t};\theta), \tag{20}$$

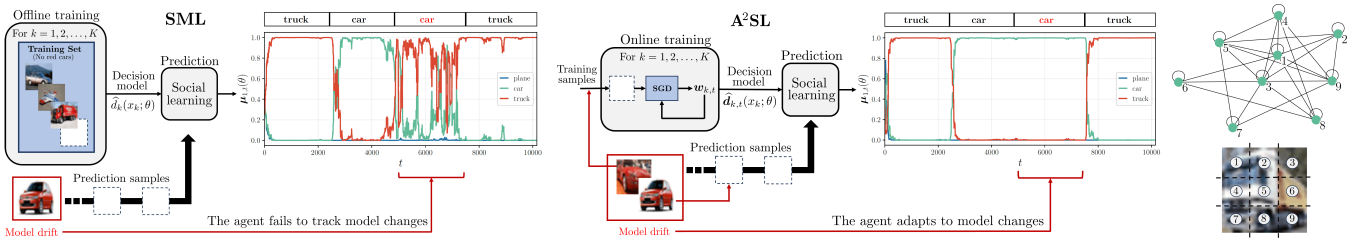

Fig. 1. Social learning problem over the CIFAR-10 data set [22]. *(Left):* SML [16]. *(Middle):* A^{2}SL. *(Right-Top):* Network topology. *(Right-Bottom):* Image patches assigned to agents $1, \ldots, 9$.

where i) agent k is active at time t only if a prediction sample is observed ($\boldsymbol{\alpha}_{k,t}^{\rm pr} = 1$); and ii) the agents' decision statistics are scaled by the Perron eigenvector entries. Condition (19) can now be interpreted as the counterpart of (16) where the local decision statistic $d_k^\star(\boldsymbol{x}_{k,t};\theta)$ is replaced by the aggregate statistic.

The condition $\beta_{\rm net}(\theta) > 0$ is less stringent than local identifiability because it does not require that the individual terms of the summation in (19) are positive for all k. Note that higher network centrality (i.e., higher Perron eigenvector entries v_k) and more frequent data acquisition (i.e., higher probabilities $q_k^{\rm pr}$) enhance the role of agent k in the identifiability condition.

Theorem 1 (Consistency of A^{2}SL). *Assume that $\mu_{k,0}(\theta) > 0$ for all k and θ, and that Assumptions 1 and 2 hold. Then, the scaled log belief ratio $\delta\boldsymbol{\beta}_{k,t}(\theta)$ is close to $\beta_{\rm net}(\theta)$ for large t and sufficiently small δ and η, in the following precise sense:*

$$\limsup_{t\to\infty} \mathbb{P}\Big[\big|\delta\boldsymbol{\beta}_{k,t}(\theta) - \beta_{\rm net}(\theta)\big| \geq \varepsilon\Big] \leq O(\delta) + O(\eta), \qquad (21)$$

for all $\varepsilon > 0$. Moreover, under Assumption 3, Eq. (21) implies that each agent detects the correct hypothesis θ_0 with negligible error for large t and sufficiently small δ and η, in the following precise sense:

$$\limsup_{t\to\infty} p_{k,t} = O(\delta) + O(\eta). \qquad (22)$$

Sketch of proof. From (13c) and (13d), $\boldsymbol{\beta}_{k,t}(\theta)$ can be written as:

$$\sum_{j=1}^{K} a_{jk}\Big((1-\delta)\boldsymbol{\beta}_{j,t-1}(\theta) + \boldsymbol{\alpha}_{j,t}^{\rm pr}\, h(\boldsymbol{x}_{j,t})\big(\boldsymbol{w}_{j,t}(\theta_0) - \boldsymbol{w}_{j,t}(\theta)\big)\Big). \qquad (23)$$

Iterating recursion (23), we obtain:

$$\boldsymbol{\beta}_{k,t}(\theta) = (1-\delta)^t \sum_{j=1}^{K} \big[A^t\big]_{jk} \beta_{j,0}(\theta) + \sum_{m=0}^{t-1}\sum_{j=1}^{K} (1-\delta)^m \big[A^{m+1}\big]_{jk} \times \boldsymbol{\alpha}_{j,t-m}^{\rm pr}\, h(\boldsymbol{x}_{j,t-m})\Big(\boldsymbol{w}_{j,t-m}(\theta_0) - \boldsymbol{w}_{j,t-m}(\theta)\Big). \qquad (24)$$

The first summation on the RHS of (24) is a transient term that dies out as $t \to \infty$. To provide a sketch of the proof, we observe that: i) the columns of the matrix power A^{m+1} converge to v in view of Assumption 1; ii) for small δ, it can be shown that the product $\boldsymbol{\alpha}_{j,t-m}^{\rm pr}\, h(\boldsymbol{x}_{j,t-m})$ can be asymptotically replaced by the expected value $q_k^{\rm pr}\, \mathbb{E}[h(\boldsymbol{x}_{j,t-m})]$, up to a mean-square-error in the order of $O(\delta)$; iii) the estimated parameter $\boldsymbol{w}_{j,t}$ approaches the exact parameter $w_j^\star$ up to a mean-square-error in the order of $O(\eta)$ — see (12); and iv) by applying Chebyshev's inequality, one can relate the error probability to the variance of the log belief ratios. Combining these facts one can rigorously prove (21). Using then the condition $\beta_{\rm net}(\theta) > 0$ and choosing ε appropriately, the claim in (22) comes from (21). ■

5. REAL-DATA EXAMPLE AND CONCLUSION

We extracted real-world images of cars, airplanes, and trucks from the CIFAR-10 data set [22]. Each class is represented by 1616 images. The training samples are purposely biased to contain only non-red cars. We employ the transformer in [23] as a feature extractor, to map the images into feature vectors of dimension 384×1.

For the social learning problem, we consider a network of $K = 9$ agents, connected according to the topology displayed in the figure (top-right). Given an image of car, airplane, or truck, each agent observes only a patch[3] thereof (right-bottom), and must decide from which class the image was generated. The agents employ social learning to blend their partial views on the same image. We assume that both the true hypothesis and the likelihood models change over time. We enforce the second type of drift by introducing training and prediction samples of red car images. Note the new training samples are immaterial to the SML strategy, which has been trained offline.

In the left panel of Fig. 1, we consider the SML strategy, the most advanced adaptive social learning strategy that handles unknown models [16]. With this strategy, each agent learns *offline* (we use a batch SGD algorithm) the parameter vectors $w_k(\theta)$. Since the agents are trained offline, they are not able to track model drifts. In the considered example, at time $t = 5000$ the agents observe patches from red cars. As shown by the belief evolution in the left panel (we report the belief of agent 1, with similar behavior observed for the other agents) red cars are misclassified as trucks. This is reasonable, since the agents were trained on the initial data set that does not contain red cars, while it contains red trucks.

In the middle panel of Fig. 1, we consider the A^{2}SL strategy with adaptation parameters $\delta = 0.005$ and $\eta = 0.1$. We see that the beliefs converge exponentially fast to the truth any time that drifts occur in the true hypothesis or in the models. Differently from the SML approach, when the model drift is encountered, the agents update correctly their decision models by leveraging new training samples that contain red cars. Just after a few iterations required to adapt to the changes, the agents correctly track the true hypothesis even in the presence of the model drift.

In summary, Theorem 1 reveals that, for sufficiently small adaptation parameters δ and η, the A^{2}SL strategy learns the truth regardless of the possible drift sources. Ongoing work includes the analysis of the transient phase to relate δ and η to the adaptation time, which would allow to capture the learning/adaptation trade-off in terms of the relation between error probability and adaptation time.

[3] Patches are resized to match the dimension of the transformer's input and extract the features.

6. REFERENCES

[1] C. Chamley, *Rational Herds: Economic Models of Social Learning*. Cambridge University Press, 2004.

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

[3] 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.

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

[5] V. Krishnamurthy and H. V. Poor, "Social learning and Bayesian games in multiagent signal processing: How do local and global decision makers interact?," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 43–57, 2013.

[6] E. Mossel and O. Tamuz, "Opinion exchange dynamics," *Probability Surveys*, vol. 14, pp. 155–204, 2017.

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

[8] 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.

[9] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed detection: Finite-Time analysis and impact of network topology," *IEEE Trans. Autom. Control*, vol. 61, no. 11, pp. 3256–3268, 2016.

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

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

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

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

[14] V. Bordignon, V. Matta, and A. H. Sayed, "Adaptive social learning," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 6053–6081, 2021.

[15] P. Hu, V. Bordignon, S. Vlaski, and A. H. Sayed, "Optimal aggregation strategies for social learning over graphs," *IEEE Trans. Inf. Theory*, vol. 69, no. 9, pp. 6048–6070, 2023.

[16] V. Bordignon, S. Vlaski, V. Matta and A. H. Sayed, "Learning from heterogeneous data based on social interactions over graphs," *IEEE Trans. on Inf. Theory*, vol. 69, no. 5, pp. 3347–3371, 2023.

[17] P. Hu, V. Bordignon, M. Kayaalp, and A. H. Sayed, "Performance of social machine learning under limited data," in *Proc. IEEE ICASSP*, Rhodes Island, Greece, June 2023, pp. 1–5.

[18] J. Z. Hare, C. A. Uribe, L. Kaplan, and A. Jadbabaie, "Non-Bayesian social learning with uncertain models,", *IEEE Trans. on Signal Process.*, vol. 68, pp. 4178–4193, 2020.

[19] A. H. Sayed, "Adaptation, Learning, and Optimization over Networks," *Found. Trends Mach. Learn.*, vol. 7, no. 4-5, pp. 311–801, 2014.

[20] A. H. Sayed, *Inference and Learning from Data*, 3 vols., Cambridge University Press, 2022.

[21] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2012.

[22] A. Krizhevsky, V. Nair and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," 2009 [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html.

[23] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," available online as arXiv:2010.11929 [cs.CV].