Non-Bayesian social learning with observation reuse and soft switching

Md. Zulfiqar Ali Bhotto; Tay, Wee Peng

2018

Md. Zulfiqar Ali Bhotto., & Tay, W. P. (2018). Non-Bayesian social learning with observation reuse and soft switching. ACM Transactions on Sensor Networks, 14(2), 14-.
doi:10.1145/3199513

<https://hdl.handle.net/10356/102641>

<https://doi.org/10.1145/3199513>

© 2018 ACM. All rights reserved. This paper was published in ACM Transactions on Sensor Networks and is made available with permission of ACM.

Downloaded on 26 Apr 2023 20:42:14 SGT

Non-Bayesian Social Learning with Observation Reuse and Soft Switching

MD. ZULFIQUAR ALI BHOTTO and WEE PENG TAY, Nanyang Technological University

We propose a non-Bayesian social learning update rule for agents in a network, which minimizes the sum of the Kullback Leibler divergence between the true distribution generating the agents' local observations and the agents' beliefs (parametrized by a hypothesis set), and a weighted varentropy related term. The varentropy related term allows us to control the rate of convergence of our update rule, which also reuses some of the most recent observations of each agent to speed up convergence. Under mild technical conditions, we show that the belief of each agent concentrates on the optimal hypothesis set, and we derive a bound for the convergence rate. Furthermore, in order to overcome the performance degradation due to misinforming agents, who use a corrupted likelihood functions in their belief updates, we propose to use multiple social networks that update their beliefs independently, and a convex combination mechanism among the beliefs of all the networks. Simulations with applications to location identification and group recommendation demonstrate that our proposed methods offer improvements over two other current state-of-the-art non-Bayesian social learning algorithms.

CCS Concepts: • **Information systems** → **Collaborative and social computing systems and tools**;

Additional Key Words and Phrases: Non-Bayesian social learning, misinforming agent, belief, convex combination, varentropy, multiple networks

ACM Reference format:

Md. Zulfiqar Ali Bhotto and Wee Peng Tay. 2018. Non-Bayesian Social Learning with Observation Reuse and Soft Switching. *ACM Trans. Sensor Netw.* 1, 1, Article 1 (January 2018), 21 pages.
<https://doi.org/10.1145/3199513>

1 INTRODUCTION

In social learning, a group of agents or sensors forms a network in which agents collaborate through information exchange to detect or identify a common phenomenon of interest. Social learning can be classified as either Bayesian or non-Bayesian. In Bayesian learning [1, 14, 17, 33, 35], each agent uses a Bayesian rule to update its belief based on its local observation's likelihood and neighboring agents' beliefs. This often leads to intractability in analyzing belief convergence, and may be impractical in some applications as full knowledge of the network topology and other agents' observation distributions are required. In [29], Bayesian learning is interpreted as a proximal stochastic gradient descent algorithm that uses a distributed, online variant of Nesterov's dual averaging method [24]. In non-Bayesian social learning [3, 15, 19, 23, 30, 36], each agent uses

This research is supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE2014-T2-1-028, and the NTU-NXP Intelligent Transport System Test-Bed Living Lab Fund S15-1105-RF-LLF from the Economic Development Board, Singapore.

Author's addresses: School of Electrical and Electronic Engineering Nanyang Technological University, Singapore.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

1550-4859/2018/1-ART1 \$15.00

<https://doi.org/10.1145/3199513>

a typically simple combination protocol that does not require global topology information to update its belief [11]. As a result, non-Bayesian social learning can often be implemented with low complexity and its convergence analysis also becomes relatively simpler. The use of non-Bayesian learning is also popular in a wide range of applications, including localization, detection, inference, and clustering [5, 23].

1.1 Related Work

Many non-Bayesian social learning algorithms for undirected graphs have been developed in the literature [15, 23, 36]. In [36], an algorithm for social learning via diffusion (SLD) is developed in which each agent shares its intermediate belief with its neighbors. The SLD algorithm converges almost surely [36]. In the social learning algorithm in [15], each agent shares its final beliefs which for fixed topology graph converges almost surely. Asymptotic convergence rate and information heterogeneity issues of the algorithm in [15] are addressed in [16]. Both the algorithms in [15, 36] uses arithmetic mean as in [26]. The social learning algorithm in [23] is a recent one that offers improved performance over the algorithms in [15, 36] and it also converges almost surely. We call the social learning algorithm in [23] as the least Kullback-Leibler (LKL) divergence algorithm since it minimizes the KL divergence [32] between two probability distributions. The LKL algorithm uses geometric mean as was used in the estimation algorithm in [4]. Convergence speed is one of the main advantageous features of the LKL algorithm in [23] besides its exact convergence rate analysis. Fast convergence speed is always desired for real time adaptive signal processing and it plays an important role in the development of adaptive filtering algorithms [10, 28].

Data reusing has been successfully used to improve convergence for adaptive filtering algorithms in the literature [8, 25]. Using a similar approach, we attempt in this paper to further improve the convergence speed of the LKL social learning algorithm [23]. In addition, we address in this paper the performance degradation of the social learning algorithm when a network is infected by misinforming agents, which are agents using corrupted likelihood functions to update their local beliefs. Robust performance always plays a major role in real time adaptive signal processing as evident in numerous literature of adaptive filtering algorithms [2, 7]. However, in non-Bayesian social learning, combating the performance degradation of the LKL algorithm [23] due to misinforming agents is not yet addressed in the literature. Robustness with respect to connectivity failure among nodes is also an important criterion for assessing non-Bayesian social learning algorithms. Uncertainty about the connectivity in a social network is discussed in [12, 13]. In this paper, for better comparison with the results in [23, 27], we test the robustness of our algorithms by adding noise to the communication links.

1.2 Our Contributions

In this paper, we propose an Observation Reusing Varentropy constrained (ORV) social learning algorithm that reuses the M most recent observations to update the local belief of each agent. In addition, the ORV algorithm uses the varentropy to control the rate of convergence. Convergence analysis shows that the proposed ORV algorithm converges to the optimal hypothesis set (cf. Definition 2) almost surely, and it achieves faster convergence on the order of M^2 over the LKL algorithm [23]. Simulation results with applications to location identification and group recommendation demonstrate the superior performance of the proposed ORV algorithm over the LKL algorithm [23] and the SLD algorithm in [27, 36].

In addition, we propose a multiple-network ORV algorithm (MORV) that performs soft switching between multiple networks in order to mitigate the effect of misinforming agents. Misinforming agents result in the network belief to converge to a false hypothesis [23]. The MORV algorithm

uses a convex combination of an m number of ORV algorithms deployed in m different networks. The convex combination is based on the ORV algorithm with a centralized scheme that performs soft switching among the social networks automatically from the infected social network to the best performing non-infected social network. The proposed soft switching does not interfere with the learning of each network and the soft switching is performed by a decision maker who is honest. Some examples of scenarios where multiple social networks perform a shared common job are political parties in a democratic system, social welfare groups, and multiple input and multiple output communication systems in intelligent transportation systems. Simulation results demonstrate that the proposed MORV approach is resilient to misinforming agents.

The rest of this paper is organized as follows. In Section 2, we discuss non-Bayesian social learning. In Section 3, we present our ORV algorithm. Convergence analysis of the proposed ORV algorithm is presented in Section 4. In Section 5, we propose the MORV algorithm. Simulation results are presented in Section 6 by using both the simulated and empirical data. Finally we draw conclusions in Section 7.

2 NON-BAYESIAN SOCIAL LEARNING

We consider a social network comprising of n agents in the set $\mathcal{N} \triangleq \{1, 2, \dots, n\}$. Each $i \in \mathcal{N}$ observes random variables $s_{i,k} \in \mathcal{S}_i$ as outcomes of a random process at time instants $k = 1, 2, \dots$. We assume that the $s_{i,k}$, for all $i \in \mathcal{N}$ and $k = 1, \dots$, are independent and for each $i \in \mathcal{N}$, $\mathcal{S}_i$ is a finite set with $\{s_{i,k} : k = 1, \dots, K\}$ identically distributed with an unknown probability distribution $f_i(\cdot)$.

Since the probability distributions of agents' observations are unknown to the agents, following [23], we assume that each agent $i \in \mathcal{N}$ has access to a set of likelihood functions $\{l_i(\cdot|\theta) : \theta \in \Theta\}$, where $\Theta = \{\theta_1, \theta_2, \dots, \theta_P\}$ is a finite set, which we call the hypothesis set. The agents collaborate with each other in order to find a subset of Θ that minimizes

$$\sum_{i \in \mathcal{N}} D(f_i \parallel l_i(\cdot|\theta)), \quad (1)$$

where $D(\cdot \parallel \cdot)$ is the KL divergence. Note that it is not required there exists a $\theta \in \Theta$ for which $l_i(\cdot|\theta) = f_i(\cdot)$. We assume that the network is a connected graph. We further assume that it is not possible for a single agent to learn the solution to (1) individually.

Let $\mu_{i,k}(\theta)$ be the belief of agent i that hypothesis θ is a solution to (1) at time instant k . In the SLD algorithm in [27, 36], agent i updates its belief in two steps: for each $\theta \in \Theta$,

$$\begin{aligned} \phi_{i,k+1}(\theta) &= \frac{\mu_{i,k}(\theta) l_i(s_{i,k+1}|\theta)}{\sum_{p=1}^P \mu_{i,k}(\theta_p) l_i(s_{i,k+1}|\theta_p)}, \\ \mu_{i,k+1}(\theta) &= \sum_{j \in \mathcal{N}_i} A_{ij} \phi_{j,k+1}(\theta), \end{aligned} \quad (2)$$

where $\phi_{i,k+1}(\theta)$ is the intermediate belief of agent i , $\mathcal{N}_i$ is the neighborhood of agent i (which includes agent i itself), $s_{i,k+1}$ is the observed signal of agent i at time $k + 1$, and A_{ij} is called the belief combiner. In [15], agent i updates its belief using the following two steps:

$$\begin{aligned} \phi_{i,k+1}(\theta) &= \frac{\mu_{i,k}(\theta) l_i(s_{i,k+1}|\theta)}{\sum_{p=1}^P \mu_{i,k}(\theta_p) l_i(s_{i,k+1}|\theta_p)}, \\ \mu_{i,k+1}(\theta) &= A_{ii} \phi_{i,k+1}(\theta) + \sum_{j \in \mathcal{N}_i \setminus \{i\}} A_{ij} \mu_{j,k}(\theta), \end{aligned} \quad (3)$$

which is extended for time varying graphs in [19]. Both algorithms in (2) and (3) have been theoretically proven to converge to the optimal hypothesis, i.e., $\mu_{k \rightarrow \infty}(\theta) \rightarrow 1$ when $l_i(\cdot|\theta) = f_i(\cdot)$ almost surely [15, 36] for all i . However, it is demonstrated in [36] through simulation results that the social learning algorithm in (2) performs better than the one in (3) in terms of convergence speed for the Metropolis combination rule [20], given as

$$A_{ij} = \begin{cases} \frac{1}{\max(d_i, d_j)}, & \text{for } j \in \mathcal{N}_i \setminus i, \\ 1 - \sum_{j \in \mathcal{N}_i \setminus i} A_{ij}, & \text{for } i = j, \end{cases} \quad (4)$$

where d_i is the degree of node i .

Let $\ell_i(\cdot) = (l_i(\cdot|\theta_j))_{j=1}^P$ be the vector consisting of the likelihood functions of agent i . To minimize (1) is equivalent to minimizing $J(\pi) = \sum_{i \in \mathcal{N}} J_i(\pi)$, with respect to $\pi \in \mathcal{P}(\Theta)$, where $\mathcal{P}(\Theta)$ is the set of all probability mass functions on Θ and

$$J_i(\pi) = -\langle \mathbb{E}_{f_i} [\log \ell_i(s_{i,k+1})], \pi \rangle. \quad (5)$$

Here, $\mathbb{E}_{f_i}$ denotes expectation over $s_{i,k+1}$ with respect to f_i . Since f_i is unknown, [23] proposes to use a stochastic mirror-descent method [6] to minimize $J(\pi)$, with

$$\nabla J_i(\pi) \approx -\log \ell_i(s_{i,k+1}), \quad (6)$$

where $s_{i,k+1}$ is now the observed outcome of agent i at time $k+1$. This leads [23] to propose an update rule for each agent i to be

$$\mu_{i,k+1} = \arg \min_{\pi \in \mathcal{P}(\Theta)} \left\{ \sum_{j=1}^n [A_k]_{ij} D(\pi \parallel \mu_{j,k}) - \langle \log \ell_i(s_{i,k+1}), \pi \rangle \right\}, \quad (7)$$

where A_k is a doubly stochastic matrix, and $[A_k]_{ij}$ is its (i, j) -th element. Solving (7), we obtain

$$\mu_{i,k+1}(\theta) = \frac{\prod_{j=1}^n \mu_{j,k}(\theta)^{[A_k]_{ij}} l_i(s_{i,k+1}|\theta)}{\sum_{p=1}^P \prod_{j=1}^n \mu_{j,k}(\theta_p)^{[A_k]_{ij}} l_i(s_{i,k+1}|\theta_p)}. \quad (8)$$

Note that all the aforementioned update rules are iterative in nature, and require that agents' observations remain stationary over a period of time in order for the agents' local beliefs to converge. The rate of convergence is therefore critical for a social network as social phenomena and observations may change in a short period of time. In the next section, we discuss the use of varentropy and observation reuse to improve the speed of convergence of the social learning algorithm (8).

3 VARENTROPY AND OBSERVATION REUSE

In this section, we propose the use of a varentropy related constraint and observation reuse to achieve a faster convergence. Fix a $\theta \in \Theta$. For $\epsilon \in (0, 1)$, let $\beta(n, \epsilon)$ be the smallest Type II error in the hypothesis test between $f = \prod_{i=1}^n f_i$ and $\prod_{i=1}^n l_i(\cdot|\theta)$ using the sensor observations $(s_{i,k})_{i=1}^n$, subject to the Type I error being not more than ϵ . Then from Theorem 3.1 of [31], we have

$$\log \beta(n, \epsilon) = -\sum_{i \in \mathcal{N}} D(f_i \parallel l_i(\cdot|\theta)) - \sqrt{nV(\theta)} \Phi^{-1}(\epsilon) - \frac{1}{2} \log n + O(1), \quad (9)$$

where Φ is the standard normal cumulative distribution function, and

$$V(\theta) = \frac{1}{n} \sum_{i \in \mathcal{N}} \mathbb{E}_{f_i} \left[\left| \log \frac{f_i(s_{i,k})}{\ell_i(l_i(s_{i,k})|\theta)} - D(f_i \parallel l_i(\cdot|\theta)) \right|^2 \right] \quad (10)$$

$$\leq \frac{1}{n} \sum_{i \in \mathcal{N}} \mathbb{E}_{f_i} \left[\log^2 \frac{f_i(s_{i,k})}{\ell_i(l_i(s_{i,k})|\theta)} \right]. \quad (11)$$

The term $V(\theta)$ is known as the varentropy. From (9), we see that (1) is the Type II error decay rate in the asymptotic regime $n \rightarrow \infty$. By minimizing (1), we are seeking a $\theta \in \Theta$ that is “closest” to $f = \prod_{i=1}^n f_i$ in the hypothesis testing framework. However, in the finite population regime where the number of agents n is small, $V(\theta)$ has a significant impact on the Type II error. Therefore, to better control the convergence of agent beliefs, we propose to include the upper bound (11) in our optimization objective. After removing terms not dependent on θ , our objective is to minimize

$$\sum_{i \in \mathcal{N}} (-\mathbb{E}_{f_i} [\log l_i(\cdot|\theta)] + \beta \mathbb{E}_{f_i} [\log^2 l_i(\cdot|\theta)]), \quad (12)$$

where β is a positive weight. Minimizing the cost function (12) is equivalent to minimizing the sum over all $i \in \mathcal{N}$ of the following local cost functions:

$$J_i(\pi) = -\langle \mathbb{E}_{f_i} [\log \ell_i(s_{i,k+1})], \pi \rangle + \beta \langle \mathbb{E}_{f_i} [\log^2 \ell_i(s_{i,k+1})], \pi \rangle. \quad (13)$$

As in (6), we use the stochastic mirror descent method by using the gradient of (13) to construct the cost function

$$\mu_{i,k+1} = \arg \min_{\pi} \left\{ \sum_{j=1}^n [A_k]_{ij} D(\pi \parallel \mu_{j,k}) + M \langle \nabla J_i(\pi), \pi \rangle \right\}, \quad (14)$$

where we approximate

$$\nabla J_i(\pi) \approx -\frac{1}{M} \sum_{m=1}^M \log \ell_i(s_{i,k+m-2}) + \frac{\beta}{M} \sum_{m=1}^M \log^2 \ell_i(s_{i,k+m-2}). \quad (15)$$

The proposed ORV algorithm is obtained by solving the above cost function as

$$\begin{aligned} \chi_{i,k+1}(\theta) &= \exp \left(-\beta \sum_{m=1}^M \log^2(\ell_i(s_{i,k+2-M}|\theta)) \right), \\ \omega_{i,k+1}(\theta) &= \prod_{m=1}^M \ell_i(s_{i,k+2-m}|\theta), \\ \mu_{i,k+1}(\theta) &= \frac{\prod_{j=1}^n \mu_{j,k}^{[A_k]_{ij}}(\theta) \omega_{i,k+1}(\theta) \chi_{i,k+1}(\theta)}{\sum_{p=1}^P \prod_{j=1}^n \mu_{j,k}^{[A_k]_{ij}}(\theta_p) \omega_{i,k+1}(\theta_p) \chi_{i,k+1}(\theta_p)}, \end{aligned} \quad (16)$$

for all agents i and $\theta = \theta_1, \theta_2, \dots, \theta_P$. Our simulation results in Section 6 demonstrate that the ORV social learning algorithm in (16) converges faster than the LKL algorithm in (8). This is because (i) we explicitly include the rate of convergence in the form of the varentropy into our optimization, and (ii) we use M sample to obtain a better approximation of the cost function gradient in (15). However, a larger M results in less adaptability of the algorithm to changes in the underlying distribution $f = \prod_{i=1}^n f_i$. For the choice of $\beta = 0$ and $M = 1$, the ORV algorithm becomes the LKL algorithm in (8). On the other hand, for the choice of $\beta = 0$ and $M > 1$, the ORV algorithm becomes the ORLKL algorithm in [9].

4 CONVERGENCE ANALYSIS FOR ORV

In this section, we present the convergence analyses results of the proposed ORV algorithm in (16). We redefine some of the definitions given in [23], namely group confidence $C_M(\theta)$ and the optimal hypothesis set Θ^* , which are required for the convergence analysis. The group confidence of the agents in the network is defined as follows.

DEFINITION 1. *The group confidence of the agents in the network is given by $C_M(\theta) = -M \sum_{i=1}^n (D(f_i(\cdot) \parallel l_i(\cdot|\theta)) + \beta \mathbb{E}_{f_i} [\log^2(l_i(\cdot|\theta))])$ for all $\theta \in \Theta$.*

The optimal hypothesis set denoted by Θ^* is defined as follows.

DEFINITION 2. *The optimal hypothesis set is defined as $\Theta^* = \arg \max_{\theta \in \Theta} C_M(\theta)$ and the confidence of the optimal hypothesis set is given as C_M^* i.e., $C_M^* = C_M(\theta^*)$ for any $\theta^* \in \Theta^*$.*

In addition, we recall the assumptions made in [23] for the convergence analysis of the update formula in (8):

ASSUMPTION 1. *The hypothesis set Θ^* is a strict subset of Θ . In other words, the set of optimal hypotheses is strictly smaller than the set of hypotheses.*

ASSUMPTION 2. *For each agent, there is a non-empty subset Θ_i^* of the optimal hypothesis set Θ^* on which the prior belief is strictly positive and $\hat{\Theta}^* = \cap_{i=1}^n \Theta_i^*$ is nonempty.*

ASSUMPTION 3. *The support of the true distribution of the observations lies in the support of the likelihood functions for all θ , i.e., if $f_i(s_i) > 0$, then we have an $\alpha \in (0, 1)$ such that $l_i(s_i|\theta) > \alpha$ for all $\theta \in \Theta$.*

Under the above assumptions we have the following theorem about the convergence of the update formula in (16)

PROPOSITION 1. *For the proposed update formula in (16) under assumptions 1-3, the belief of each agent concentrates almost surely on the optimal hypothesis set Θ^* asymptotically as $k \rightarrow \infty$.*

PROOF. The proof is a straightforward extension of the proof given in [23] for the LKL algorithm upon using the definition of group confidence. We have given the proof in Appendix A for convenience. $\square$

We also obtain the following theorem about the convergence rate of the formula in (16).

PROPOSITION 2. *For each agent i , the update formula in (16) under Assumptions 1-3 achieves*

$$\mathbb{P} \left(\mu_{i,k}(\theta_v) \geq \exp \left(-\frac{k}{2} \gamma_{M,2} + \gamma_{i,1} \right) \right) \leq \rho$$

in iteration

$$k \geq \left\lceil \frac{8(\log(\alpha_s))^2 \log(1/\rho)}{M^2 \gamma_{1,2}^2} + 1 \right\rceil,$$

where

$$\gamma_{i,1} \triangleq \max_{\theta_v \notin \Theta^*} \max_{\theta_w \in \Theta^*} \frac{\sqrt{2}}{1-\lambda} \sum_{i=1}^n |C_{i,M}(\theta_v) - C_{i,M}(\theta_w)| + C_{i,M}(\theta_v) - C_{i,M}(\theta_w),$$

with $\lambda \in (0, 1)$, $\log(\alpha_s) = \log(\alpha)(\beta \log(\alpha) - 1)$,

$$\gamma_{M,2} \triangleq \frac{1}{n} \min_{\theta_v \notin \Theta^*} (C_M^* - C_M(\theta_v)),$$

and $\gamma_{1,2} = \frac{1}{M} \gamma_{M,2}$.

PROOF. The proof is a straightforward extension of the convergence rate proof of the LKL algorithm given in [23] using the definition of group confidence. Hence we omit the proof here. $\square$

Proposition 2 shows that the ORV algorithm with $\beta = 0$ would achieve an order of M^2 faster convergence over the LKL algorithm as reported in [9]. The numerator of the convergence rate of the ORV algorithm has an extra term $(\beta \log(\alpha) - 1)^2$ as compared to the numerator of the convergence rate of the LKL algorithm in [23]. The denominator term of the ORV algorithm, i.e., $\gamma_{1,2}$ has an extra positive term corresponding to the varentropy related term in the group confidence C_M as compared to the denominator term of the LKL algorithm. Some remarks about the effect of parameter β in (16) and some considerations about its choice.

REMARK 1. First consider the case $\beta = 1$ in $\chi_{i,k+1}(\theta)$ given with (16). As can be seen, scalar $\chi_{i,k+1}(\theta)$ puts nonnegative exponential weights on $\omega_{i,k+1}(\theta)$ to update the prior belief in (16). Since we would get $\chi_{i,k+1}(\theta_v) \rightarrow 0$ for all $\theta_v \notin \Theta^*$ and $\chi_{i,k+1}(\theta_w) \rightarrow 1$ for all $\theta_w \in \hat{\Theta}^*$, scalar $\chi_{i,k+1}(\theta)$ gives the prior belief $\mu_{\cdot,k}(\theta_v)$ in (16) an extra push towards a zero value while letting $\omega_{i,k+1}(\theta_w)$ to determine the belief $\mu_{i,k+1}(\theta_w)$ alone. As a result, the ORV algorithm would converge at a faster rate as compared to the ORLKL algorithm.

REMARK 2. Now consider the case where $0 < \beta \ll 1$ in $\chi_{i,k+1}(\theta)$ given with (16). In this case parameter β would cause $\chi_{i,k+1}(\theta) \rightarrow 1$ for all $\theta \in \Theta$ and hence, it would not interfere the information $\omega_{i,k+1}(\theta)$ that updates the prior belief in (16). Consequently, $\omega_{i,k+1}(\theta)$ would update the prior belief $\mu_{\cdot,k}(\theta)$ in (16) towards the optimal belief. As a result, the ORV algorithm would converge at a similar rate as compared to the ORLKL algorithm.

REMARK 3. Consider the case where $1 \ll \beta < \infty$. In this case, $\chi_{i,k+1}(\theta) \rightarrow 0$ for all $\theta \in \Theta$, which results in $\chi_{i,k+1}(\theta)\omega_{i,k+1}(\theta) \rightarrow 0$ in (16) for all θ . As a result, there will be no update or positive prior belief in (16). In such cases, the ORV algorithm will not converge due to the information loss caused by a very large β .

For the case where $0 < \beta \ll 1$, the upper bound of the varentropy related term (i.e., the second term) in the definition of group confidence (cf. Definition 1) can be neglected. In this case, the order of magnitude of k becomes the same as given in [9]. For the case when $1 \ll \beta < \infty$, the KL divergence term in the definition of group confidence (cf. Definition 1) can be neglected. The order of magnitude of k can be obtained from Proposition 2 with $\log(\alpha_s) \approx \beta \log^2(\alpha)$. However, convergence in this case does not imply that we obtain the true hypothesis since the KL divergence term in Definition 1 becomes relatively negligible. For the values of β at which the KL divergence term in Definition 1 remains dominant, the convergence rate as given in Proposition 2 implies the convergence rate towards the true hypothesis.

In the next section, we discuss the proposed soft switching method that performs a soft switching from the infected network to the non-infected network and thereby avoids the effect of misinforming agents. An agent is defined as a misinforming agent when it modifies or corrupts its own observation intentionally in a manner such that a certain false hypothesis θ can appear to be the optimal hypothesis in the network. A misinforming agent is implemented by substituting its likelihood function $l_i(\cdot|\theta)$ corresponding to the ground truth by the likelihood function $\tilde{l}_i(\cdot|\theta)$ corresponding to a false θ .

5 SOCIAL LEARNING WITH SOFT SWITCHING

In this section, we consider the case where some agents in the network are misinforming agents. A misinforming agent i is an agent that uses a corrupted likelihood function $\tilde{l}_i(\cdot|\cdot)$ in place of $l_i(\cdot|\cdot)$ to update its local belief. The likelihood function $\tilde{l}_i(\cdot|\cdot)$ is such that there exists some $\theta \in \Theta$ with

$\tilde{l}_i(\cdot|\theta) \neq l_i(\cdot|\theta)$. We propose to use a convex combination of m social networks $\mathcal{N}_\rho \triangleq \{1, \dots, n_\rho\}$ for $\rho = 1, \dots, m$. We assume an honest decision maker who knows the true likelihood functions of some selected agents from the m networks, with at least one agent from each network. We also assume that the optimal hypothesis set (cf. Definition 2) has only one element.

In order for the honest decision maker's belief to converge to the optimal hypothesis, we assume that all networks can be infected by misinforming agents but at any time instant, we have at least one non-infected network. This assumption though restrictive, is plausible in the case where we have access to many different social networks formed by distinct interest groups. The diversity of the different available networks makes it unlikely that all networks are misinformed at the same time. Our proposed convex combination parameter acts as a soft switch that attempts to select the belief of the current best non-infected network. In the case where all networks are infected, our method attempts to find a network whose belief is closest to the optimal hypothesis in terms of the KL divergence, as demonstrated in the simulation results in Section 6.1.

At time instant $k + 1$, the following information is supplied to the decision maker:

- All the selected agents supply their observations to the decision maker.
- One agent $\check{r}_\rho$ in each network $\rho \in \{1, \dots, m\}$ sends its belief updated according to (16) to the decision maker.

For each $\rho \in \{1, \dots, m\}$, the decision maker then finds

$$\check{\theta}_{\rho,k+1} = \arg \max_{\theta \in \Theta} \mu_{\check{r}_\rho,k+1}(\theta).$$

The decision maker now considers only the *reduced* hypothesis set $\check{\Theta}_{k+1} \triangleq \{\check{\theta}_{1,k+1}, \dots, \check{\theta}_{m,k+1}\}$ (note that the elements in this set may not be distinct).

The decision maker aims to find the hypothesis $\check{\theta} \in \check{\Theta}_{k+1}$ that is more likely to be true given the collected observations from all the selected agents. To do this, it iteratively updates a decision vector δ_{k+1} at time instant $k + 1$ by solving the minimization problem:

$$\delta_{k+1} = \arg \min_{\pi \in \mathcal{P}(\check{\Theta}_{k+1})} \left\{ D(\pi || \delta_k) - \sum_{i=1}^{n_h} \sum_{m=1}^M \mathbb{E}_\pi [\log(\check{\ell}_i(s_{i,k+2-m}))] + \beta \sum_{i=1}^{n_h} \sum_{m=1}^M \mathbb{E}_\pi [\log^2(\check{\ell}_i(s_{i,k+2-m}))] \right\}, \quad (17)$$

where $\check{\ell}_i(\cdot) = (l_i(\cdot|\theta))_{\theta \in \check{\Theta}_{k+1}}$, and n_h is the total number of selected agents from all networks. Similar to (16), the solution of the problem in (17) can be obtained as

$$\begin{aligned} \chi_{k+1}(\check{\theta}_{\rho,k+1}) &= \exp \left(-\beta \sum_{i=1}^{n_h} \sum_{m=1}^M \log^2(l_i(s_{i,k+2-m}|\check{\theta}_{\rho,k+1})) \right), \\ \omega_{k+1}(\check{\theta}_{\rho,k+1}) &= \prod_{i=1}^{n_h} \prod_{m=1}^M l_i(s_{i,k+2-m}|\check{\theta}_{\rho,k+1}), \\ \delta_{k+1}(\check{\theta}_{\rho,k+1}) &= \frac{\delta_k(\check{\theta}_{\rho,k}) \omega_{k+1}(\check{\theta}_{\rho,k+1}) \chi_{k+1}(\check{\theta}_{\rho,k+1})}{\sum_{p=1}^m \delta_k(\check{\theta}_{p,k}) \omega_{k+1}(\check{\theta}_{p,k+1}) \chi_{k+1}(\check{\theta}_{p,k+1})}, \end{aligned} \quad (18)$$

for each $\check{\theta}_{\rho,k+1} \in \check{\Theta}_{k+1}$. Let $R_{k+1} = \arg \max_{\rho} \delta_{k+1}(\check{\theta}_{\rho,k+1})$. The decision maker then chooses its belief to be $\mu_{f,k+1} = \mu_{\check{r}_{\rho^*},k+1}$, where $\rho^* = \arg \max_{\rho \in R_{k+1}} \|\mu_{\check{r}_\rho,k+1}\|_\infty$. We call the above procedure the MORV algorithm.

6 SIMULATION RESULTS

In this section, we compare the performance of proposed ORV algorithm in (16) with the SLD algorithm in (2) [27, 36], LKL algorithm in (8) [23], and the ORLKL algorithm in [9] (cf. discussion before Section 4) in location identification and group recommendation applications.

6.1 Location Identification Application

In the location identification problem, a group of n agents randomly distributed in an area attempts to locate the position of a source in a two dimensional grid through mutual cooperation. Each agent i has a sensor that receives a signal $s_{i,k}$ from the source location at coordinates θ_s . The source signal $s_{i,k}$ is assumed to have a normal distribution, i.e., $s_{i,k} \sim \mathcal{N}(d_i(\theta_s), \sqrt{0.1d_i(\theta_s)})$ where $d_i(\theta_s)$ is the distance between agent i and the source θ_s . The hypothesis set Θ in this application becomes the set of grid points. The likelihood functions $l_i(\cdot|\theta)$ are also based on the sensor model with $l_i(\cdot|\theta) \sim \mathcal{N}(d_i(\theta), \sqrt{d_i(\theta)})$ where $d_i(\theta)$ is the distance between agent i and the hypothesis θ . The location identification problem was also addressed in the Bayesian framework using a particle filter with inertial sensors in [34]. However, in our current simulation, we do not assume that inertial sensors are available.

We suppose that there are three networks, $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$ as illustrated by using the yellow, blue, and green colors, respectively in Fig. 1. There are $P = 441$ uniformly distributed grid points. The number of agents in each network in Fig. 1 is $n = 35$ and their positions are distributed randomly with a uniform distribution in each experiment. The graph connecting the agents in each network was also generated randomly in each experiment. The source location $\theta_p^{(t_r)}$ is chosen randomly at each trial to be one of the grid points. We compute the average KL divergence at time step k as

$$v_k = \frac{1}{T_r} \sum_{t_r=1}^{T_r} \frac{1}{n} \sum_{i=1}^n D(p_{\delta}^{(t_r)} \parallel \mu_{i,k}^{(t_r)}), \quad (19)$$

where $p_{\delta}^{(t_r)}$ is the probability distribution with mass one at $\theta_p^{(t_r)}$, and $\mu_{i,k}^{(t_r)}$ is agent i 's belief at time step k for trial number t_r . We also compute the average belief as

$$\xi_k = \frac{1}{T_r} \sum_{t_r=1}^{T_r} \frac{1}{n} \sum_{i=1}^n \mu_{i,k}^{(t_r)}(\theta_p) \quad (20)$$

to compare the performance of the social learning algorithms. The belief vector in each trial was initialized with uniform distribution in all algorithms for all agents in all trials. We average over $T_r = 500$ trials. In the ideal case where agents identify the true source location, we have $v_k \rightarrow 0$ and $\xi_k \rightarrow 1$ as $k \rightarrow \infty$.

In Experiment 1, we examine the effect of parameter β on the performance of the proposed ORV algorithm in (16) for the network $\mathcal{N}_1$ in Fig. 1 for a fixed value of $M = 1$, while letting $\beta = 0.25, 0.5, 1, 3$, and 70 . In addition we compare the performance of the ORV algorithm with the LKL and SLD algorithms. The observation signals were contaminated with a white Gaussian noise signal with standard deviation 0.01 . We assume in this case there is no measurement noise in the communication links among agents and also there are no misinforming agents. The average KL divergence v_k and average belief ξ_k obtained by using the ORV, LKL, and SLD algorithms are plotted in dB scale in Fig. 2a and Fig. 2b, respectively. It can be observed that the proposed ORV algorithm converges faster than both the SLD and LKL algorithms for values of β that are not unreasonably large. For large values of β , e.g., $\beta = 70$, the ORV algorithm fails to converge, which verifies Remark 3.

Fig. 1. Sensor Networks.

In Experiment 2, we examine the effect of parameter M on the performance of the ORV algorithm for a fixed value of $\beta = 1$, while $M = 1, 2, 3$. We use $M = 3$ for the ORLKL algorithm. The average KL divergence v_k and average belief ξ_k obtained by using the ORV, LKL, ORLKL, and SLD algorithms are plotted in dB scale in Fig. 3a and Fig. 3b, respectively. We see that the proposed ORV algorithm converges faster than SLD, LKL and ORLKL for its all choices of M . It is also seen that the convergence speed of the ORV algorithm increases as M increases for a fixed value of β .

In Experiment 3, we examine the performance of the ORV algorithm for a fixed $M = 1$ and different values of β in the presence of communication noise among agents. We used a white Gaussian noise signal with standard deviation 10^{-2} to contaminate the beliefs in the communication links. As can be seen in Figs. 4a-4b, the proposed ORV algorithm converges faster than both the SLD and LKL algorithms for its all choices of β . In addition the convergence speed of the ORV algorithm increases as β increases from 0 to 3 for a fixed value of M . However, it can be seen from

(a) Average KL-Divergence in dB for $M = 1$ and different choices of β in the ORV algorithm.

(b) Average belief for $M = 1$ and different choices of β in the ORV algorithm.

Fig. 2. Experiment 1. Varying β with $M = 1$.

(a) Average KL-Divergence in dB for $\beta = 1$ and different choices of M in the ORV algorithm.

(b) Average Belief for $\beta = 1$ and different choices of M in the ORV algorithm.

Fig. 3. Experiment 2. Varying M while $\beta = 1$.

Fig. 4a that the ORV algorithm shows an increase in the steady state error when β becomes large i.e., $\beta = 3$.

In Experiment 4, we examine the performance of the ORV algorithm for different values of M in the presence of communication noise among agents. We used a white Gaussian noise signal with standard deviation 10^{-2} to contaminate the beliefs in the communication links. As can be seen from Figs. 5a-5b, the proposed ORV algorithm converges faster than the SLD and LKL algorithms for its all choices of M . The ORV algorithm with $M = 2, 3$ converges faster than the ORLKL algorithm with $M = 3$. The ORV algorithm with $M = 1$ converges at a similar rate as compared to the ORLKL algorithm with $M = 3$ due to the use of a smaller $\beta = 0.25$. Upon setting $\beta \geq 0.3$ we found that ORV algorithm with $M = 1$ converges faster than the ORLKL algorithm. The convergence speed of the ORV algorithm increases as M increases for a fixed value of β .

We used MATLAB R2015b on a computer with the following specifications: Intel(R) Xeon(R) CPU E5-1603 2.80 GHz processor, 8.00 GB RAM, and 64 bit Windows OS. The average CPU time

(a) Average KL-Divergence in dB for $M = 1$ and different choices of β in the ORV algorithm.

(b) Average belief for $M = 1$ and different choices of β in the ORV algorithm.

Fig. 4. Experiment 3. Varying β with $M = 1$ in presence of communication noise.

(a) Average KL-Divergence in dB for $\beta = 0.25$ and different choices of M in the ORV algorithm.

(b) Average Belief for $\beta = 0.25$ and different choices of M in the ORV algorithm.

Fig. 5. Experiment 4. Varying M while $\beta = 1$ in presence of communication noise.

Table 1. Average CPU time per agent, millisecond

Exp.	SLD	LKL	ORV, $\beta = 0.25$	ORV, $\beta = 0.5$	ORV, $\beta = 1$	ORV, $\beta = 3$
1	0.144	0.304	0.196	0.188	0.180	0.176
Exp.	SLD	LKL	ORLKL	ORV, $M = 1$	ORV, $M = 2$	ORV, $M = 3$
2	0.143	0.267	0.198	0.178	0.178	0.183
Exp.	SLD	LKL	ORV, $\beta = 0.25$	ORV, $\beta = 0.5$	ORV, $\beta = 1$	ORV, $\beta = 3$
3	0.143	0.585	0.571	0.569	0.570	0.571
Exp.	SLD	LKL	ORLKL	ORV, $M = 1$	ORV, $M = 2$	ORV, $M = 3$
4	0.144	0.567	0.546	0.577	0.579	0.585

per time sample and per agent for all the algorithms used are given in Table 1. It can be observed that the average CPU time per sample index required by the ORV algorithm is similar to those

required by the other algorithms. The SLD algorithm uses the arithmetic mean to fuse an agent's neighboring beliefs, which requires less computation. Hence, its average CPU time used is the lowest.

In Experiment 5, we evaluate the performance of the proposed switching MORV procedure. As a comparison, we also run the SLD and LKL algorithms on the whole network formed by the multiple social networks connected together via the decision maker. We call these the MSLD and MLKL procedure for convenience. In addition, we also use the centralized fusion (CF) method as a comparison benchmark. In our CF method, the decision maker uses all the hypothesis in Θ to identify the true location by using its collected observations of all the honest agents. We consider three cases in this experiment. In case (a), one network ($\mathcal{N}_3$) is infected by 15 randomly selected agents as misinforming agents in that network. In case (b), two networks ($\mathcal{N}_2$ and $\mathcal{N}_3$) are infected by 15 randomly selected misinforming agents in each network. In case (c), all networks are infected with 15 randomly selected misinforming agents in each network. Network structures are generated independently in all cases as was done in Experiments 1–4. The network structure used for networks 2 and 3 in case (a) are illustrated in Fig. 1. Note that the CF method will always converge to the true hypothesis even if all networks are infected with misinforming agents as it uses only honest agents, whereas none of the other methods will converge to the true hypothesis in case (c). However, the convergence rate of CF in cases (a) and (b) is expected to be slower.

In all cases, the misinforming agents used likelihood functions corresponding to a source at coordinate location $[0, 0]$ to replace the likelihood function of the true hypothesis. For the update formula in (18) we used $n_h = 5$ agents where the decision maker selects at least one agent from each of the three networks. In Fig. 6, we truncated the average KL divergence to 7 dB when it goes above 7 dB for better clarity in the plots. The average KL divergence ν_k obtained by using the ORV, MSLD, MLKL, CF method, and the MORV algorithm in cases (a), (b), and (c) are illustrated in Figs. 6a–6c, respectively.

It can be seen from Figs. 6a–6b that MORV, MLKL, and MSLD all successfully converges to the true hypothesis. Note that the ν_k plot corresponding to the MORV, MSLD, and MLKL algorithms is the belief of one selected agent $\check{r}_\rho$ as discussed in Section 5, and this selected agent changes from trial to trial if there are multiple noninfected networks. We see that the decision maker in MORV follows the best performing noninfected networks in Fig. 6a. The CF method offers similar performance as compared to the MSLD and MLKL algorithms, but since its belief is based only on the consensus of $n_h = 5$ agents its convergence rate is less than MORV. From Fig. 6c, it can be seen that the CF method has a better performance than the MORV, MSLD, and MLKL algorithms. This is because since all the networks are infected, the decision maker does not have the true hypothesis in its reduced hypothesis set. As a result, the MORV, MSLD, and MLKL algorithms did not converge.

We note that although MORV requires running ORV on multiple networks, and incurs additional computation cost (cf. Table 2) in the soft switching process at the decision maker, it is able to obtain

Table 2. Average CPU time, millisecond

	Case (a)	Case (b)	Case (c)
MSLD	12.1961	11.9881	12.291
MLKL	22.4735	22.3043	22.6041
MORV	13.7999	13.6549	13.8042

a more accurate belief than running social learning algorithms independently in each network.

Fig. 6. Experiment 5. Average KL divergence in dB with $M = 1$ and $\beta = 1$ for the ORV algorithms.

Thus, if multiple networks are available, MORV is a better approach than performing independent learning on each network. The same conclusion holds true for the MLKL and MSLD algorithms.

In Experiment 6, we investigate the performance of the MORV algorithm in the presence of communication noise among agents. We used white Gaussian noise with standard deviations 0.01, 0.005, and 0.001 for the networks $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$, respectively in order to contaminate the agent beliefs in the communication links. We considered the first two cases in this experiment. As can be seen in Figs. 7a-7b, the MORV algorithm successfully avoids the infected network beliefs. The MORV algorithm follows the best performing the ORV algorithm and achieves a faster convergence and lower average KL divergence than CF, MSLD and MLKL.

For the MORV algorithm, during the transient phase where the infected and noninfected networks have not yet converged close to their own optimal beliefs, the set Θ_{k+1} in (18) can vary significantly with k . This causes the δ_{k+1} sequence in (18) to fluctuate, and can lead to multiple switchings. To avoid this, one can apply the MORV algorithm only after a delay to allow the individual networks time to converge.

Fig. 7. Experiment 6. Average KL divergence in dB with $M = 2$ and $\beta = 0.25$ for the ORV algorithms.

6.2 Group Recommendation Application

Fig. 8. Friendship network.

In this application, a friendship network of n friends receives observation signals $s_k \in \{1, 2, 3, 4, 5\}$ which are ratings about pubs/restaurants from the source circle (i.e., another group of q users). The job of the friendship network is to give recommendation to the source group about a pubs/restaurant that fits the observations at best. In social learning context, the set of pubs/restaurant becomes the hypothesis set Θ with P elements, n becomes the number of agents in the social network. Upon receiving the source signal samples (ratings), the social network with n agents would find the pubs/restaurant that fits all the different preferences of the q users forming a friend circle at best i.e., they all can meet in a single pub/restaurant.

We consider a friendship network shown in Fig. 8 which is taken from [18] with $n = 10$ agents. We considered all the $P = 16$ restaurants in [18] as a set of hypothesis. The likelihood functions $l_i(s|\theta)$ were computed by using the voter-id probability distribution computation algorithm reported in [18]. In this algorithm, we used the rating data given in [18] for the 10 agents over 16 restaurants. We tested the system by using $q = 10$ source samples from 10 friends taken from the dataset given in [18] where the ratings ranges from 3 to 4. We used these ratings periodically to excite the system. Since we have $l_i(1|\theta_{14}) = l_i(2|\theta_{14}) = l_i(5|\theta_{14}) = 0$ for all i , we computed the average KL

divergence v_k by considering $\theta_p = \theta_{14}$ in $p_\delta(\theta)$. We do the same while plotting the average belief ξ_k . The average KL divergence v_k obtained by using the SLD, LKL, ORLKL, and ORV algorithms as illustrated in Fig. 9a. Likewise, the average belief ξ_k plots by using the SLD, LKL, ORLKL, and ORV algorithms are illustrated in Fig. 9b.

(a) Average KL-Divergence in dB for $M = 1$ and different choices of β in the ORV algorithm.

(b) Average belief for $M = 1$ and different choices of β in the ORV algorithm.

Fig. 9. Experiment 7. Comparison of ORV with SLD, LKL, and ORLKL.

As can be seen in Figs. 9a-9b, all algorithms recommend the correct restaurant while the ORV algorithm converges faster than the other algorithms for all choices of β for a fixed M . The convergence rate of the ORV algorithm increases for an increased value of the parameter β . Since we excited the system with the same input sequences in each 10 iterations the convergence had occurred in steps where in each step the most recent old belief acted like a new initial belief.

We again consider the same friendship network shown in Fig. 8 with $n = 10$ agents in order to examine the MORV algorithm. We computed the average KL divergence v_k by using $\theta_p = \theta_{14}$ in $p_\delta(\theta)$. We do the same while plotting the average belief ξ_k . We then updated the voter-id probability

(a) Average KL-Divergence in dB for a fixed $M = 1$ and $\beta = 0.25$ in the ORV algorithm.

(b) Average belief for a fixed $M = 1$ and $\beta = 0.25$ in the ORV algorithm.

Fig. 10. Experiment 8. Performance of MORV.

distribution at a different time instant with simulated ratings where one agent turns into is a misinforming agent. The information of the misinforming agent causes its updated likelihood function $l_i(\cdot|\theta_{14})$ to be the same as original likelihood function $l_i(\cdot|\theta_1)$. For the MORV algorithm, we used 1 trusted agents from the original network (original voter-id probability distribution) and 1 from the updated network (updated voter-id probability distribution) for the decision maker. The average KL divergence plots for the ORV algorithm in the original network is denoted as ORV, 1 and the updated network as ORV, 2. The average KL divergence plots obtained by using the SLD, LKL, ORV, and MORV algorithms are illustrated in Fig. 10a. The average belief plots obtained by using the SLD, LKL, ORV, and MORV algorithms are illustrated in Fig. 10b. As can be seen in Figs. 10a-10b, the MORV algorithm follows the non-infected network and hence was able to avoid the recommendation of the misinforming agent infected network.

7 CONCLUSION

We have developed a social learning algorithm based on observation reuse and varentropy to achieve faster convergence than social learning based only on KL divergence. We showed that our algorithm converges to the optimal hypothesis set almost surely and provided a bound for its convergence rate. Simulations demonstrated that our algorithm has better convergence performance than two other existing social learning approaches. We also developed a soft switching algorithm that allows a decision maker to learn from several social networks, some of which are infected with misinforming agents. Simulations suggest that our approach is effective at overcoming the performance degradation suffered by methods that learn only from a single network. In our soft switching approach, we require selected agents to send their observations to a decision maker. For future work, it would be of interest to consider incorporating privacy mechanisms in this process so that data privacy of the selected agents are protected, but still enabling the decision maker to perform effective soft switching amongst the multiple social networks.

A PROOF OF PROPOSITION 1

The steps of this proof are similar to the steps of the proof of Theorem 1 given in [23]. We reproduce the proof after adapting to the structure of our ORV algorithm. Let $\theta_v \in \Theta$ be a hypothesis for which $\theta_v \notin \Theta^*$ and $\theta_w \in \Theta$ be a hypothesis for which $\theta_w \in \hat{\Theta}^*$. Using hypotheses θ_v and θ_w we define $H_M(\theta_v, \theta_w)$ whose i -th coordinate is given by

$$H_{i,M}(\theta_v, \theta_w) = M \left(D(f_i(\cdot) \parallel l_i(\cdot|\theta_v)) - D(f_i(\cdot) \parallel l_i(\cdot|\theta_w)) - \beta \mathbb{E}_{f_i} [\log^2(l_i(\cdot|\theta_w))] + \beta \mathbb{E}_{f_i} [\log^2(l_i(\cdot|\theta_v))] \right), \quad (21)$$

vector $\mathcal{L}_k^{\theta_v, \theta_w}$ whose i -th coordinate is given by [23]

$$\mathcal{L}_{i,k}^{\theta_v, \theta_w} = \log \frac{l_i(s_{i,k}|\theta_v)}{l_i(s_{i,k}|\theta_w)} + \beta \log^2(l_i(s_{i,k}|\theta_w)) - \beta \log^2(l_i(s_{i,k}|\theta_v)), \quad (22)$$

and

$$\psi_{i,k+1}(\theta_v, \theta_w) = \log \frac{\mu_{i,k+1}(\theta_v)}{\mu_{i,k+1}(\theta_w)} \quad (23)$$

for $i = 1, 2, \dots, n$. Using (16) in (23), we get after some manipulation

$$\begin{aligned} \psi_{i,k+1}(\theta_v, \theta_w) &= \sum_{j=1}^n [A_k]_{ij} \psi_{j,k}(\theta_v, \theta_w) + \sum_{m=1}^M \log \frac{l_i(s_{i,k+2-m}|\theta_v)}{l_i(s_{i,k+2-m}|\theta_w)} - \beta \log^2(l_i(s_{i,k+2-m}|\theta_v)) \\ &\quad + \beta \log^2(l_i(s_{i,k+2-m}|\theta_w)) \end{aligned}$$

for $i = 1, 2, \dots, n$. Stacking up $\psi_{i,k}$ for all agent i , we get a vector

$$\boldsymbol{\psi}_{k+1}(\theta_v, \theta_w) = A_k \boldsymbol{\psi}_k(\theta_v, \theta_w) + \sum_{m=1}^M \mathcal{L}_{k+2-m}^{\theta_v, \theta_w} \quad (24)$$

Using the initial value $\boldsymbol{\psi}_0(\theta_v, \theta_w)$ we can express (24) as

$$\boldsymbol{\psi}_{k+1}(\theta_v, \theta_w) = A_{k:0} \boldsymbol{\psi}_0(\theta_v, \theta_w) + \sum_{t=1}^k A_{k:t} \mathcal{L}_{M,t}^{\theta_v, \theta_w} + \mathcal{L}_{M,k+1}^{\theta_v, \theta_w} \quad (25)$$

where

$$A_{k:t} = A_k A_{k-1} \cdots A_t \quad \text{and} \quad \mathcal{L}_{M,k+1}^{\theta_v, \theta_w} = \sum_{m=1}^M \mathcal{L}_{k+2-m}^{\theta_v, \theta_w}. \quad (26)$$

By adding and then subtracting $\sum_{t=1}^k \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' H_M(\theta_v, \theta_w)$ in (25), we get

$$\begin{aligned} \boldsymbol{\psi}_{k+1}(\theta_v, \theta_w) &= A_{k:0} \boldsymbol{\psi}_0(\theta_v, \theta_w) + \sum_{t=1}^k \left(A_{k:t} \mathcal{L}_{M,t}^{\theta_v, \theta_w} + \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' H_M(\theta_v, \theta_w) \right) + \mathcal{L}_{M,k+1}^{\theta_v, \theta_w} \\ &\quad - \frac{k}{n} \sum_{i=1}^n H_{i,M}(\theta_v, \theta_w) \mathbf{1}_n \end{aligned} \quad (27)$$

Using the definition of group confidence

$$C_M(\theta) = -M \sum_{i=1}^n (D(f_i(\cdot) \parallel l_i(\cdot|\theta)) + \beta \mathbb{E}_{f_i} [\log^2(l_i(\cdot|\theta))])$$

we get from (21)

$$\sum_{i=1}^n H_{i,M}(\theta_v, \theta_w) = C_M(\theta_w) - C_M(\theta_v) = C_M^* - C_M(\theta_v)$$

and as a result (27) can be expressed as

$$\begin{aligned} \boldsymbol{\psi}_{k+1}(\theta_v, \theta_w) &= A_{k:0} \boldsymbol{\psi}_0(\theta_v, \theta_w) + \sum_{t=1}^k \left(A_{k:t} \mathcal{L}_{M,t}^{\theta_v, \theta_w} + \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' H_M(\theta_v, \theta_w) \right) + \mathcal{L}_{M,k+1}^{\theta_v, \theta_w} \\ &\quad - \frac{k}{n} (C_M^* - C_M(\theta_v)) \mathbf{1}_n \end{aligned} \quad (28)$$

Dividing both sides of (28) by k and taking the limit as $k \rightarrow \infty$ almost surely, we have

$$\begin{aligned} \lim_{k \rightarrow \infty} \frac{1}{k} \boldsymbol{\psi}_{k+1}(\theta_v, \theta_w) &= \lim_{k \rightarrow \infty} \frac{1}{k} A_{k:0} \boldsymbol{\psi}_0(\theta_v, \theta_w) + \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{t=1}^k \left(A_{k:t} \mathcal{L}_{M,t}^{\theta_v, \theta_w} + \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' H_M(\theta_v, \theta_w) \right) \\ &\quad + \lim_{k \rightarrow \infty} \frac{1}{k} \mathcal{L}_{M,k+1}^{\theta_v, \theta_w} - \frac{1}{n} (C_M^* - C_M(\theta_v)) \mathbf{1}_n \end{aligned} \quad (29)$$

The first term on the right hand side of (29) goes to zero deterministically and the third term also goes to zero since each term in $\mathcal{L}_{M,k+1}^{\theta_v, \theta_w}$ in (26) is bounded [23]. Eq. (29) therefore simplifies to

$$\begin{aligned} \lim_{k \rightarrow \infty} \frac{1}{k} \boldsymbol{\psi}_{k+1}(\theta_v, \theta_w) &= \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{t=1}^k \left(A_{k:t} \mathcal{L}_{M,t}^{\theta_v, \theta_w} + \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' \mathbf{H}_M(\theta_v, \theta_w) \right) - \frac{1}{n} (C_M^* - C_M(\theta_v)) \mathbf{1}_n \\ &= \lim_{k \rightarrow \infty} \frac{1}{k} \left(\sum_{t=1}^k \left(A_{k:t} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' \right) \mathcal{L}_{M,t}^{\theta_v, \theta_w} + \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' \left(\mathcal{L}_{M,t}^{\theta_v, \theta_w} + \mathbf{H}_M(\theta_v, \theta_w) \right) \right) \\ &\quad - \frac{1}{n} (C_M^* - C_M(\theta_v)) \mathbf{1}_n \end{aligned} \quad (30)$$

Using the relation [21–23]

$$\left| [A_{k:t}]_{ij} - \frac{1}{n} \right| \leq \sqrt{2} \lambda^{k-t} \quad (31)$$

where $\lambda \in (0, 1)$ [23] the first term on the right hand side of (30) goes to zero almost surely. Therefore, Eq. (30) becomes

$$\begin{aligned} \lim_{k \rightarrow \infty} \frac{1}{k} \boldsymbol{\psi}_{k+1}(\theta_v, \theta_w) &= \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{t=1}^k \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' \left(\mathcal{L}_{M,t}^{\theta_v, \theta_w} + \mathbf{H}_M(\theta_v, \theta_w) \right) - \frac{1}{n} (C_M^* - C_M(\theta_v)) \mathbf{1}_n \\ &= \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{t=1}^k \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' \left(\sum_{m=1}^M \mathcal{L}_{t+1-m}^{\theta_v, \theta_w} + \mathbf{H}_M(\theta_v, \theta_w) \right) - \frac{1}{n} (C_M^* - C_M(\theta_v)) \mathbf{1}_n \\ &= \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{t=1}^k \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' \sum_{m=1}^M \left(\mathcal{L}_{t+1-m}^{\theta_v, \theta_w} + \mathbf{H}(\theta_v, \theta_w) \right) - \frac{1}{n} (C_M^* - C_M(\theta_v)) \mathbf{1}_n \\ &= \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{m=1}^M \sum_{t=1}^k \frac{1}{n} \mathbf{1}_n \mathbf{1}_n' \left(\mathcal{L}_{t+1-m}^{\theta_v, \theta_w} + \mathbf{H}(\theta_v, \theta_w) \right) - \frac{1}{n} (C_M^* - C_M(\theta_v)) \mathbf{1}_n \end{aligned} \quad (32)$$

Taking expectation of the both side of the i -th coordinate of $\mathcal{L}_{t+1-m}^{\theta_v, \theta_w}$ as defined in (22) for any m , we get

$$\begin{aligned} \mathbb{E} \left[\mathcal{L}_{t+1-m}^{\theta_v, \theta_w} \right] &= \sum_{k=1}^K f_i(s_{i,k}) \log \frac{f_i(s_{i,k})}{l_i(s_{i,k} | \theta_w)} - \sum_{k=1}^K f_i(s_{i,k}) \log \frac{f_i(s_{i,k})}{l_i(s_{i,k} | \theta_v)} + \beta \sum_{k=1}^K f_i(s_{i,k}) \log^2(s_{i,k} | \theta_w) \\ &\quad - \beta \sum_{k=1}^K f_i(s_{i,k}) \log^2(s_{i,k} | \theta_v) = -H_i(\theta_v, \theta_w) \end{aligned} \quad (33)$$

Repeating (33) for all i , we get

$$\mathbb{E} \left[\mathcal{L}_{t+1-m}^{\theta_v, \theta_w} \right] = -\mathbf{H}(\theta_v, \theta_w) \quad (34)$$

for any m . Recall Kolmogorov's strong law of large numbers that states for a sequence of independent random variables $\{X_t\}$ with variances such that $\sum_{t=1}^{\infty} \text{Var}(X_t)/t^2 < \infty$, we obtain

$$\frac{1}{k} \sum_{t=1}^k X_t - \frac{1}{k} \sum_{t=1}^k \mathbb{E}[X_t] \rightarrow 0$$

almost surely. Using $X_t = \frac{1}{n} \mathbf{1}'_n \mathcal{L}_{t+1-m}^{\theta_v, \theta_w}$ which is a sequence of independent random variables for all t with $\sup_{t \geq 0} \text{Var}(X_t) < \infty$ in the above relation and then applying (34), we get

$$\frac{1}{k} \sum_{t=1}^k \frac{1}{n} \mathbf{1}'_n \mathcal{L}_{t+1-m}^{\theta_v, \theta_w} \rightarrow -\frac{1}{k} \sum_{t=1}^k \frac{1}{n} \mathbf{1}'_n \mathbf{H}(\theta_v, \theta_w). \quad (35)$$

We can now use (35) in (32) and obtain

$$\lim_{k \rightarrow \infty} \frac{1}{k} \psi_{k+1}(\theta_v, \theta_w) = -\frac{1}{n} (C_M^* - C_M(\theta_v)) \mathbf{1}_n$$

As can be seen, taking the limit at infinity, we would get

$$\lim_{k \rightarrow \infty} \psi_{k+1}(\theta_v, \theta_w) = -\infty \quad (36)$$

almost surely. Using the relation $\mu_{i,k}(\theta_v) \leq \exp(\psi_{i,k}(\theta_v, \theta_w))$ for all i , we get $\mu_{i,k}(\theta_v) \rightarrow 0$. Hence the update rule in (16) would concentrate on the optimal set Θ^* asymptotically as $k \rightarrow \infty$.

ACKNOWLEDGMENTS

This research is supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE2014-T2-1-028.

REFERENCES

- [1] Daron Acemoglu, Munther A. Dahleh, Ilan Lobel, and Asuman Ozdaglar. 2011. Bayesian Learning in Social Networks. *Review of Economic Studies* 78, 4 (March 2011), 1201–1236.
- [2] Sara Al-Sayed, Abdelhak M. Zoubir, and Ali H. Sayed. 2016. Robust Adaptation in Impulsive Noise. *IEEE Trans. Signal Process.* 64, 11 (June 2016), 2851–2865.
- [3] Robert J. Aumann. 1976. Ageing to disagree. *Annals of Statistics* 4, 6 (1976), 1236–1239.
- [4] Saptarshi Bandyopadhyay and Soon-Jo Chung. 2014. Distributed estimation using Bayesian consensus filtering. In *Proc. American Control Conference*. American Automatic Control Council, Portland, USA, 634–641.
- [5] A.R. Barron and T. M. Cover. 1991. Minimum complexity density estimation. *IEEE Trans. Inform. Theory* 37, 4 (July 1991), 1034–1054.
- [6] Amer Beck and Marc Teboulle. 2003. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operation Research Letters* 31, (2003), 167–175.
- [7] Md. Zulfiqar Ali Bhotto and Andreas Antoniou. 2012. Robust set-membership affine-projection adaptive-filtering algorithm. *IEEE Trans. Signal Process.* 60, 1 (Jan. 2012), 73–81.
- [8] Md. Zulfiqar Ali Bhotto and Andreas Antoniou. 2013. A family of shrinkage adaptive filtering algorithms. *IEEE Trans. Signal Process.* 61, 7 (April 2013), 1689–1697.
- [9] Md. Zulfiqar Ali Bhotto and Wee Peng Tay. May 2017. Non-Bayesian social learning using observation reuse and soft switching. In *Proc. IEEE Int. Conf. on Commun.* IEEE Communications Society, Paris, France. 10.1109/ICC.2017.7996912
- [10] P. S. R. Diniz. 2008. *Adaptive Filtering: Algorithms and Practical Implementation* (3rd ed.). Springer, New York, USA.
- [11] L. G. Epstein, J. Noor, and A. Sandroni. 2010. Non-Bayesian learning. *The BE Journal of Theoretical Economics* 10, 1 (2010), 1–20.
- [12] Luoyi Fu, Xinzhe Fu, Zhiying Xu, Qianyang Peng, Xinbing Wang, and Songwu Lu. 2017. Determining source-destination connectivity in uncertain networks: Modeling and Solutions. *IEEE/ACM Trans. Networking* 25, 6 (Dec. 2017), 3237–3252.
- [13] Luoyi Fu, Xinbing Wang, and P. R. Kumar. 2017. Are we connected? Optimal determination of source-destination connectivity in random networks. *IEEE/ACM Trans. Networking* 25, 2 (April 2017), 751–764.
- [14] Jack Ho, Wee Peng Tay, Tony Q. S. Quek, and Edwin K. P. Chong. 2015. Robust decentralized detection and social learning in tandem networks. *IEEE Trans. Signal Process.* 63, 19 (Oct. 2015), 5019 – 5032.
- [15] Ali Jadbabaie, Pooya Molavi, Alvaro Sandroni, and Alireza Tahbaz-Salehi. Sep. 2012. Non-Bayesian social learning. *Game Econ. Behavior* 76, 1 (Sep. 2012), 210–225.
- [16] Ali Jadbabaie, Pooya Molavi, and Alireza Tahbaz-Salehi. 2013. Information heterogeneity and the speed of learning in social networks. *Columbia Business School Research Paper* 13, 28 (2013), 1–38. <http://dx.doi.org/10.2139/ssrn.2266979>
- [17] Y. Kanoria and O. Tamuz. 2013. Tractable Bayesian Social Learning on Trees. *IEEE J. Sel. Areas Commun.* 31, 4 (April 2013), 756–765.

- [18] Ofrit Lesser, Lihi Naamani-Dery, Meir Kalech, and Yuval Elovici. 2017. Group decision support for leisure activities using voting and social networks. *Group Decision and Negotiation* 26, 3 (May 2017), 473–494.
- [19] Q. Liu, A. Feng, L. Wang, and X. Wang. 2014. Social learning with time varying weights. *Journal of Systems Science and Complexity* 27, 3 (2014), 81–593.
- [20] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. Equations of state calculations by fast machines. *J. Chem. Phys.* 21, 6 (1953), 1087–1092.
- [21] Angelia Nedic and Alex Olshevsky. 2015. Distributed optimization over time-varying directed graphs. *IEEE Trans. Autom. Control* 60, 3 (March 2015), 601–615.
- [22] Angelia Nedic, Alex Olshevsky, Asuman Ozdaglar, and John N. Tsitsiklis. 2009. On distributed averaging algorithms and quantization effects. *IEEE Trans. Autom. Control* 54, 11 (Nov. 2009), 2506–2517.
- [23] Angelia Nedic, Alex Olshevsky, and Cesar A. Uribe. 2017. Fast convergence rates for distributed non-Bayesian learning. *IEEE Trans. Autom. Control* 62, 11 (Nov. 2017), 5538–5553.
- [24] Yurii Nesterov. 2009. Primal-dual subgradient methods for convex problems. *Mathematical programming* 120, 1 (2009), 221–259.
- [25] K. Ozeki and T. Umeda. 1984. An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties. *Electron. Commun. Japan* 67-A (1984), 19–27.
- [26] Kamiar Rahnama Rad and Alireza Tahbaz-Salehi. 2010. Distributed parameter estimation in networks. In *Proc. IEEE Conference Decision and Control*. IEEE Control Systems Society, Georgia, USA, 5050–5055.
- [27] Hawraa Salami, Bicheng Ying, and Ali H. Sayed. 2017. Social learning over weakly-connected graphs. *IEEE Trans. Signal and Information Process. over Networks* 3, 2 (June 2017), 222–238.
- [28] Ali H. Sayed. 2003. *Fundamentals of Adaptive Filtering*. Wiley, New Jersey, USA.
- [29] Shahin Shahrapour and Ali Jadbabaie. 2013. Exponentially fast parameter estimation in networks using distributed dual averaging. (Sep 2013). arXiv:1309.2350v1[cs.LG]
- [30] Shahin Shahrapour, Alexander Rakhlin, and Ali Jadbabaie. 2015. Finite-time analysis of the distributed detection problem. (Dec 2015). arXiv:1512.09311v1[cs.SY]
- [31] V. Strassen. 2009. Asymptotic estimates in Shannon’s information theory. (2009), 689–723 pages. www.math.cornell.edu/~pmlut/strassen.pdf
- [32] Masashi Sugiyama, Song Liu, Marthinus Christoffel du Plessis, Masao Yamanaka, Makoto Yamada, Taiji Suzuki, and Takafumi Kanamori. 2013. Direct divergence approximation between probability distributions and its applications in machine learning. *Journal of Computing Science and Engineering* 7, 2 (2013), 99–111.
- [33] Wee Peng Tay. 2015. Whose Opinion to follow in Multihypothesis Social Learning? A Large Deviations Perspective. *IEEE J. Sel. Topics Signal Process.* 9, 2 (March 2015), 344 – 359.
- [34] Xudong Wu, Ruofei Shen, Luoyi Fu, Xiaohua Tian, Peng Liu, and Xinbing Wang. 2017. iBILL: Using iBeacon and inertial sensors for accurate indoor localization in large open areas. *IEEE Access* 5, # (Aug. 2017), 14589–14599. 0.1109/ACCESS.2017.2726088
- [35] Zhenliang Zhang, E.K.P. Chong, A. Pezeshki, W. Moran, and S.D. Howard. 2013. Learning in Hierarchical Social Networks. *Selected Topics in Signal Processing, IEEE Journal of* 7, 2 (April 2013), 305–317.
- [36] Xiaochuan Zhao and Ali H. Sayed. Nov. 2012. Learning over social networks via diffusion adaptation. In *Proc. Asilomar Conf. on Signals, Systems and Computers*. IEEE Signal Processing Society, California, USA, 709–713.