2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Sharing beliefs to learn Nash equilibria

Barbara Franci and Filippo Fabiani

***Abstract*— We consider finite games where the agents only share their beliefs on the possible equilibrium configuration. Specifically, the agents experience the strategies of their opponents only as realized parameters, thereby updating and sharing beliefs on the possible configurations iteratively. We show that combining non-bayes updates with best-response dynamics allows the agents to learn the Nash equilibrium, i.e., the belief distribution over the set of parameters has a peak on the true configuration. Convergence results of the learning mechanism are provided in two cases: the agents learn the equilibrium configuration as a whole, or the agents learn those strategies of the opponents that constitute such an equilibrium.**

I. Introduction

In several game-theoretic scenarios the agents share their decision variables and the entire Nash equilibrium (NE) learning process is based on the fact that such decisions can be accessed by the other agents (full-decision information). To reduce the amount of shared information, it has been assumed that only part of the neighbors' decision variables can be communicated, while the rest need to be inferred via some communication rounds (partial-decision information). In this second case, the seeking algorithm entails two learning dynamics, one to retrieve the missing information via a consensus mechanism and one towards the NE [1]–[3].

Both in the full- and partial-decision information setups, a key assumption establishes that the agents share the exact decision variable. In other words, the communication is deterministic and the problem can be solved with standard consensus dynamics. However, this is not realistic. First, untrustworthy agents might want to share wrong information, especially in a competitive scenario. Secondly, there might be some noise so that, even if the exact information is shared, it can not be retrieved or received by the other agents. In these cases, one can still assume that the agents have some insight on the opponents' possible strategies and that they can form an opinion, i.e., a belief, on the outcome of such strategy.

In these situations where the exact parameter is unknown, researchers often turn to (non-)bayesian learning to retrieve at least a probability distribution on the set of parameters [4]–[6]. Loosely speaking, Bayesian learning is used to update the probability of a certain parameter being used as more information becomes available [7], [8]. Since this requires significant computational efforts, particularly in a distributed setup [6], [9], non-bayesian mechanisms are then considered also for large networks [4], [9].

B. Franci is with the Faculty of Science and Engineering, Department of Advanced Computing Sciences, Maastricht University, 6200 MD Maastricht, The Netherlands (b.franci@maastrichtuniversity.nl). F. Fabiani is with the IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100, Lucca, Italy (filippo.fabiani@imtlucca.it)

In this paper we consider the case where the agents can not observe the true decision variable of the other agents, but rather a realization of those decisions, i.e., a configuration of parameters. The agents experience in fact a realized reward, and know the likelihood of such reward corresponding to a certain configuration. In this situation, learning a NE is tangled with updating the beliefs on the possible configurations and with learning the equilibrium one with high probability. To this aim, we consider a finite game where the strategies of the other players are perceived as parameters. The goal of the agents is to learn the NE configuration by learning also the probability of seeing these parameters realized. The learning algorithm is inspired by [9], where a non-bayesian learning step is paired with a best-response iteration that allows the players to reach an NE. In [9], however, the agents share their strategies and the uncertainty is provided by an external parameter that affects all the payoff functions of all the agents. In this work, instead, the decision variables themselves are perceived as parameters and learned, together with the equilibrium configuration. Our contributions can then be summarized as follows:

- We derive a iterative algorithm allowing the agents to build a probability distribution on the possible configurations with the highest peak on the NE one (§III);
- We improve the results in [9] by accounting for the centrality of the agents, i.e., considering how influential the agents are in the communication network (§III);
- With a slight modification, the algorithm can be used not only to retrieve the NE configuration as a whole, but also to learn the strategies of the opponents that correspond to the equilibrium strategy (§IV).

Compared to [9], we allow the agents to have their own stepsize (instead of the same for everyone involved in the game), include the influence of the agents by considering the centrality, and propose a weaker identifiability assumption that holds for any pair of parameters and not only with respect to (w.r.t.) the true one. Finally, our learned parameter is not an external factor (e.g., nature) but it corresponds to the unknown strategies of the opponents or configuration thereof.

Notation: A matrix $W = [w_{ij}] \in \mathbb{R}^{N\times N}$ is doubly stochastic if both the sum over the rows and columns is equal to one, i.e., $\sum_{i=1}^{N} w_{ij} = 1$ for all j and $\sum_{j=1}^{N} w_{ij} = 1$ for all i. Given a set S, $|S|$ indicates the cardinality of the set. We denote with $\Delta(S)$ the simplex obtained as the convex hull of the points in the finite set S, i.e., $\Delta(S) = \{\sum_{i=1}^{|S|} a_i s_i : \sum_{i=1}^{|S|} a_i = 1, a_i \geq 0 \text{ for all } i = 1,\dots,|S|\}$. Given two probability distributions P and P', the Kullback-Leibler (KL) divergence is $D_{KL}(P\|P') = \mathbb{E}_P[\log(\frac{P}{P'})]$. The

Copyright ©2024 EUCA

KL divergence is zero if and only if $P = P'$ with probability 1. We use the acronym *a.s.* for almost surely.

II. Problem setup

We consider N agents, indexed by the set $\mathcal{N} = \{1, \dots, N\}$, taking part to a finite game, i.e., each $i \in \mathcal{N}$ makes a decision $x_i \in \mathcal{X}_i$ with $\mathcal{X}_i$ finite subset of $\mathbb{R}^{n_i}$. Each agent aims at maximizing a reward function $u_i : \mathcal{X}_i \times \mathcal{X}_{-i} \to \mathbb{R}$, $\mathcal{X}_{-i} = \prod_{j \in \mathcal{N} \setminus \{i\}} \mathcal{X}_j$, so that the game at hand is defined by the following collection of optimization problems:

$$\forall i \in \mathcal{N} : \max_{x_i \in \mathcal{X}_i} u_i(x_i, \boldsymbol{x}_{-i}), \tag{1}$$

emphasizing the mutual dependency of each u_i from the opponents' decisions, $\boldsymbol{x}_{-i} = \mathrm{col}((x_j)_{j \in \mathcal{N} \setminus \{i\}}) \in \mathcal{X}_{-i}$.

As a solution notion for the underlying game, we adopt the well-known NE, which coincides with a collective strategy $\boldsymbol{x}^* = \mathrm{col}((x_i^*)_{i \in \mathcal{N}})$ so that, for all $i \in \mathcal{N}$,

$$u_i(x_i^*, \boldsymbol{x}_{-i}^*) \geq u_i(z_i, \boldsymbol{x}_{-i}^*), \ \forall z_i \in \mathcal{X}_i. \tag{2}$$

We consider a scenario in which the agents do not have access to the decision variables of the other participants, which are hence seen as *parameters*. This framework may correspond, for instance, to the case in which the agents are not willing to communicate their exact decision directly due to, e.g., privacy reasons. In particular, we assume that to each agent corresponds a set $\Theta_i = \{\theta_i^1, \dots, \theta_i^{M_i}\} \subset \mathbb{R}^{n_i}$ of possible values that such parameters can assume. These sets of parameters are known to the other agents $j \in \mathcal{N} \setminus \{i\}$. Specifically, one can think of Θ_i as the set of possible parameters that any other agent $j \in \mathcal{N} \setminus \{i\}$ can see realized as a possible action of agent i. Note that, in general, $M_i \neq |\mathcal{X}_i|$, i.e., the actual number of possible decision variables of agent i. All the opponents see the same Θ_i for the possible parameters of agent i. We call *configuration* the vector $\boldsymbol{\theta} = \mathrm{col}((\theta_i)_{i \in \mathcal{N}}) \in \boldsymbol{\Theta} = \prod_{i \in \mathcal{N}} \Theta_i$ collecting the parameters of all the agents.

The agents have then access to the set of configurations $\boldsymbol{\Theta}$, since they know sets Θ_i, $i \in \mathcal{N}$, and aim at learning the "true" one, i.e., the configuration corresponding to the NE. Along the line of (2), the agents then want to learn a configuration $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}$ such that, for all $i \in \mathcal{N}$,

$$u_i(x_i^*, \boldsymbol{\theta}_{-i}^*) \geq u_i(z_i, \boldsymbol{\theta}_{-i}^*), \ \forall z_i \in \mathcal{X}_i, \tag{3}$$

and $\boldsymbol{\theta}^* = \mathrm{col}(x_i^*, \boldsymbol{\theta}_{-i}^*)$, for all $i \in \mathcal{N}$.

Assumption 1: A NE exists and it is unique. □

We assume that $\boldsymbol{\theta}^*$ in (3) corresponds to the NE $\boldsymbol{x}^*$ in (2), i.e., $\boldsymbol{\theta}^* = \boldsymbol{x}^*$. Moreover, $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}$, that is, the true configuration lies in the parameter set that all the agents can access. In words, this means that the agents have a possibility to learn the equilibrium strategy of the other participants, hence the equilibrium configuration once the individual preference (1) is also taken into account.

We conclude this section by describing the communication network that allows the agents to share some information, i.e., their belief on the possible configurations, with their neighbors. Specifically, we assume that the agents are connected over a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{E}$ is the set of edges, namely for a given pair of agents $i, j \in \mathcal{N}$, $(i, j) \in \mathcal{E}$ if agent i communicates with agent j. We indicate the set of neighbors of agent i in the graph $\mathcal{G}$ as $\mathcal{N}_i = \{j \in \mathcal{N} : (i, j) \in \mathcal{E}\}$. The connections on the graph are collected in the weighted adjacency matrix $W = [w_{ij}]_{i,j \in \mathcal{N}}$, where $w_{ij} > 0$ if $(i, j) \in \mathcal{E}$ and $w_{ij} = 0$ otherwise. To following assumption guarantees some connectivity properties to $\mathcal{G}$ [9].

Assumption 2: The graph $\mathcal{G}$ is connected and the adjacency matrix W is doubly stochastic. □

III. Learning the equilibrium configuration

Since the agents can access only the parameters and not the true decision variables of the other agents, they experience a realized reward

$$y_i(x_i, \boldsymbol{\theta}_{-i}^c) = u_i(x_i, \boldsymbol{\theta}_{-i}^c) + \varepsilon(x_i, \boldsymbol{\theta}_{-i}^c) \tag{4}$$

after the parameters are realized in certain configuration $c \in \mathcal{C} = \{1, \dots, M\}, M = \prod_{i \in \mathcal{N}} M_i$. The term $\varepsilon(x_i, \boldsymbol{\theta}_{-i}^c)$ represents a noise term that might depend on the configuration $\boldsymbol{\theta}^c = \mathrm{col}(x_i, \boldsymbol{\theta}_{-i}^c)$. In words, once a certain configuration is realized, the agents experience the corresponding reward that depends on agent i's choice and on the realized parameters of the other agents. Let $f_i(y_i \mid x_i, \ \boldsymbol{\theta}_{-i}^c)$ be the likelihood function of the realized reward y_i, where to ease the notation we have removed the dependency of y_i on x_i and $\boldsymbol{\theta}_{-i}^c$.

To measure the informative content of the parameters, we consider the KL divergence among the likelihoods of the realized rewards, and define for each agent $i \in \mathcal{N}$ the set of reward equivalent configurations w.r.t. the NE one:

$$\bar{\Theta}_i(x_i^*) = \{\boldsymbol{\theta}^c \in \boldsymbol{\Theta} \mid D_{KL}(f_i(y_i^* \mid x_i^*, \ \boldsymbol{\theta}_{-i}^*) \| f_i(y_i^* \mid x_i^*, \ \boldsymbol{\theta}_{-i}^c)) = 0\},$$

where y^* is the reward realized at a NE configuration. Specifically, these sets collect the configurations that are more likely to give the same reward as the NE configuration.

The parameters in these sets are locally indistinguishable to agent i. Therefore, to learn the true parameter, we assume that at least one agent can identify the true parameter.

Assumption 3: For every $\boldsymbol{\theta}^{c_1} \neq \boldsymbol{\theta}^{c_2}$, $c_1, c_2 \in \mathcal{C}$, there exists at least one agent $i \in \mathcal{N}$ for which $D_{KL}(f_i(y_i \mid x_i, \boldsymbol{\theta}^{c_1}) \| f_i(y_i \mid x_i, \boldsymbol{\theta}^{c_2})) > 0$ for all $x_i \in \mathcal{X}_i$. □

Assumption 3 does not require that one of the agents can distinguish $\boldsymbol{\theta}^*$ from all other $\boldsymbol{\theta}^c$, $c \in \mathcal{C}$, but rather that for any two configurations there is at least one agent that can distinguish them. Assumption 3 is also a sufficient condition for the global identifiability of $\boldsymbol{\theta}^*$ [10]. In fact, it holds that:

$$\bigcap_{i \in \mathcal{N}} \bar{\Theta}_i(x_i^*) = \{\boldsymbol{\theta}^*\}.$$

The following example is meant to introduce the quantities above, and will be also used to run numerical experiments.

Example 1: Consider a three players repeated game described by the following tables, where Player 1 (P1) plays rows, Player 2 (P2) plays columns and Player 3 (P3) picks the tables. The goal is to maximize the obtainable reward:

P3 \ P2		L	R
P1	T	0,0,0	5,7,3
	B	10,10,10	6,6,11

P3 \ P2		L	R
P1	T	3,7,5	9,6,6
	B	5,7,5	4,4,4

The set of possible configurations coincides with $\boldsymbol{\Theta} = \{(T,L,1),(T,R,1),(B,L,1),(B,R,1),(T,L,2),(T,R,2),(B,L,2),(B,R,2)\}$ and the only NE is $(B,L,1)$. With some abuse of notation, we let coincide here the parameters with the decision variables (although they might differ in practice). For the sake of this example, assume that the likelihood gives the same probability to all rewards greater or equal than 6, e.g., when the agents' reward is fairly high. Then, the sets of reward equivalent configurations are:

$$\bar{\Theta}_1(B) = \{(B,L,1),(B,R,1)\},$$
$$\bar{\Theta}_2(L) = \{(B,L,1),(T,L,2),(B,L,2)\},$$
$$\bar{\Theta}_3(1) = \{(B,L,1),(B,R,1)\},$$

and their intersection contains the NE configuration only. □

To learn the true configuration, the agents keep a private belief $\mu_i \in \Delta(\boldsymbol{\Theta})$ and a posterior belief $b_i \in \Delta(\boldsymbol{\Theta})$ based on the realized reward. Given the private belief, the agents then compute the expected reward to update their decisions:

$$u_i(x_i,\mu_i) = \sum_{c=1}^{M} u_i(x_i,\boldsymbol{\theta}^c_{-i})\mu_i(\boldsymbol{\theta}^c). \tag{5}$$

The expected reward can be used successively to update the local strategy, following a standard best-response paradigm:

$$\mathrm{BR}_i(\mu_i) = \mathrm{argmax}_{x_i\in\mathcal{X}_i}\ u_i(x_i,\mu_i). \tag{6}$$

Algorithm 1 reports the main steps of this first procedure derived, including the update of the belief and the decisions.

A. Convergence results

To obtain convergence of the beliefs, updated according to Algorithm 1, we start by assuming that the realized rewards have bounded information content:

Assumption 4: For each $i \in \mathcal{N}$, the likelihood function $f_i(y_i \mid x_i,\ \boldsymbol{\theta}^c_{-i})$ is continuous in x_i for all $c \in \mathcal{C}$. Moreover, there exist $L > 0$ such that

$$\max_{i\in\mathcal{N}}\max_{\boldsymbol{\theta}^{c_1}_{-i},\boldsymbol{\theta}^{c_2}_{-i}\in\boldsymbol{\Theta}_{-i}}\max_{x_i\in\mathcal{X}_i}\sup_{y_i}\left|\log\frac{f_i(y_i\mid x_i,\ \boldsymbol{\theta}^{c_1}_{-i})}{f_i(y_i\mid x_i,\ \boldsymbol{\theta}^{c_2}_{-i})}\right| \le L. \ \square$$

Concerning the non-bayesian update of the posterior beliefs, we assume next that the exponent is vanishing with the number of iterations:

Assumption 5: For every agent $i \in \mathcal{N}$, the stepsize sequence $\{\alpha_i^{(t)}\}_{t\in\mathbb{N}}$ is such that $0 < \alpha_i^{(t)} < 1$ for all $t \ge 0$, $\sum_{t=1}^{\infty}\alpha_i^{(t)} = \infty$ and $\sum_{t=1}^{\infty}(\alpha_i^{(t)})^2 = 0$. □

According to [9], we thus have the following result.

Algorithm 1: Configuration Learning Algorithm

Initialization: $x_i^{(0)} \in \mathcal{X}_i$ for $i \in \mathcal{N}$ and $\mu_i^{(0)} = \frac{1}{M}\mathbf{1}_M$ for $i,\ j \in \mathcal{N}$, $\boldsymbol{\theta}^c \in \boldsymbol{\Theta}$

Iteration t: For configuration $\boldsymbol{\theta}^c \in \boldsymbol{\Theta}$, Agent i

1) Updates the posterior belief:

$$b_i^{(t)}(\boldsymbol{\theta}^c) = \frac{f_i(y_i^{(t)}\mid x_i^{(t)},\ \boldsymbol{\theta}^c_{-i})^{\alpha_i^{(t)}}\mu_i^{(t)}(\boldsymbol{\theta}^c)}{\sum_{c\in\mathcal{C}} f_i(y_i^{(t)}\mid x_i^{(t)},\ \boldsymbol{\theta}^c)^{\alpha_i^{(t)}}\mu_i^{(t)}(\boldsymbol{\theta}^c)}$$

2) Receives $b_\ell^{(t)}(\boldsymbol{\theta}^c)$ from $\ell \in \mathcal{N}_i$ and updates the private belief:

$$\mu_i^{(t+1)}(\boldsymbol{\theta}^c) = \frac{\exp(\sum_{\ell=1}^{N} w_{i\ell}\log b_\ell^{(t)}(\boldsymbol{\theta}^c))}{\sum_{c\in\mathcal{C}}\exp(\sum_{\ell=1}^{N} w_{i\ell}\log b_\ell^{(t)}(\boldsymbol{\theta}^c))}$$

3) Updates strategy:

$$x_i^{(t+1)} \in \mathrm{BR}_i(\mu_i^{(t+1)})$$

Lemma 1: Let Assumption 1 and 5 hold true. Then, for all $c \in \mathcal{C}$, it holds that:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\mu_i^{(t)}(\boldsymbol{\theta}^c)}{\mu_i^{(t)}(\boldsymbol{\theta}^*)} \to \nu^c \text{ a.s., as } t\to\infty,$$

where $\nu^c \ge 0$ is a nonnegative random variable. □

Lemma 1 states that the average belief ratio converges to a random variable a.s.. We note that despite $\boldsymbol{\theta}^*$ is used for the convergence analysis (here and later on) the agents do not ned to know it in advance. In fact, in Algorithm 1 all the parameters follow the same updating rule. Next, we show that the agents reach consensus on a common belief, which however may differ from the true one:

Theorem 1: Let Assumptions 1, 2, 4, 5 hold. Then, the belief sequence $\{\mu_i^{(t)}\}_{t\in\mathbb{N}}$, $i \in \mathcal{N}$ generated by Algorithm 1 converges a.s. to a common belief μ of the form

$$\mu(\boldsymbol{\theta}^c) = \frac{\nu^c}{\sum_{c=1}^{M}\nu^c} \text{ for all } c \in \mathcal{C}$$

where ν^c is as in Lemma 1. □

Proof: The proof follows the same steps as that of [9, Th. 1], by adapting to the agent-wise stepsize $\alpha_i^{(t)}$ and by assuming that, w.l.o.g., $\boldsymbol{\theta}^c = \boldsymbol{\theta}^*$ for $c = 1$. ■

Since the agents beliefs converge to a common belief, we now revise the NE definition, given such common belief μ.

Definition 1: A NE with common belief μ is a collective strategy $\boldsymbol{x}^*(\mu) = \mathrm{col}((x_i^*(\mu))_{i\in\mathcal{N}})$ such that, for all $i \in \mathcal{N}$,

$$u_i(x_i^*(\mu),\mu)) \le u_i(z_i,\mu),\ \forall z_i \in \mathcal{X}_i,$$

where $u_i(\cdot,\mu)$ is defined as in (5). □

Assumption 6: Given a common belief μ, the collective strategy $\boldsymbol{x}^{(t)}$ generated by the best response in (6) converges a.s. to $\boldsymbol{x}^*(\mu)$, as $t\to\infty$. □

Remark 1: The assumption above simply states that, if the collective decision do not converge to a NE with constant

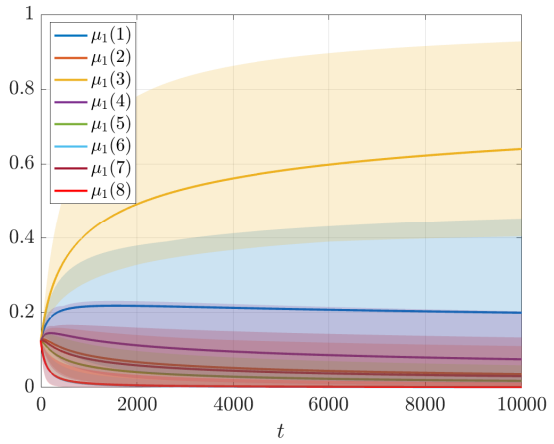

Fig. 1. Private belief of Player 1 over the iterations. Each solid line represents the mean value associated to each component of μ_1, while the shaded area of the same colour the resulting variance, both obtained by running 100 numerical experiments of the game in Example 1.

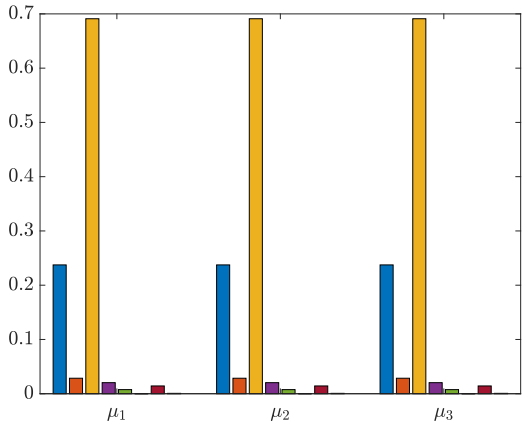

Fig. 2. Private beliefs of the three players after 10^4 iterations of Algorithm 1. The highest peaks correspond to the NE configuration.

belief, i.e., the common belief, then convergence will also fail when the beliefs are updated at each iteration [8]. □

Next, we introduce the notion of network divergence:

Definition 2 (Network divergence): Given a common belief μ, for all $c \in \mathcal{C}$ the network divergence is defined as

$$Z(\theta^*, \theta^c) = \frac{1}{N}\sum_{i=1}^{N} D_{KL}(f_i(y_i \mid x_i^*(\mu),\ \boldsymbol{\theta}_{-i}^*) \| f_i(y_i \mid x_i^*(\mu),\ \boldsymbol{\theta}_{-i}^c)). \ \square$$

Then, we state our first main result establishing the convergence of the common belief to the true parameter:

Theorem 2: Let Assumptions 1-6 hold true. Then, we have the following: for each $i \in \mathcal{N}$,

1) $\lim_{T\to\infty} \frac{1}{\sum_{t=1}^{T} \alpha_i^{(t)}} \log \frac{\mu_i^{(T+1)}(\boldsymbol{\theta}^*)}{\mu_i^{(T+1)}(\boldsymbol{\theta}^c)} = Z(\boldsymbol{\theta}^*, \boldsymbol{\theta}^c)$;
2) $\mu_i^{(t)}(\boldsymbol{\theta}^*) \to 1$ as $t \to \infty$. □

Proof: The proof follows similarly to that of [9, Th. 2] by replacing the stepsize with the agent-wise ones. ■

Remark 2: As a by-product of learning the true parameter, derived from its beliefs converging to 1, we obtain also convergence to the NE configuration of the game in (1). □

Example 1 (Cont'd): We corroborate Theorem 2 by running Algorithm 1 on the three players game. The configurations are numbered from left to right, top to bottom, hence the NE $(B, L, 1)$ corresponds to configuration $c = 3$. Specifically, we conduct 100 numerical instances of Example 1 by randomly choosing each $x_i^{(0)}$ and likelihood $f_i(y_i \mid x_i,\ \boldsymbol{\theta}_{-i}^c)$ associated to the eight possible realized costs, in each configuration, for every player $i \in \{1, 2, 3\}$. The stepsize, instead, is identical for each agent and is set to $1/(t+100)$, while all the agents are allowed to communicate with each other. In Fig. 1, we show how the private belief of Player 1 changes through the iterations (Players 2 and 3 feature a similar behaviour), while Fig. 2 shows the final private belief for all the players. The configuration with the highest belief correctly coincides with the NE one. □

B. Learning the NE configuration with influential agents

If one is interested in considering the influence an agent has within the network, Assumption 2 can be replaced with a stronger one relying on the centrality of the network [10].

Assumption 7: The graph $\mathcal{G}$ is strongly connected. □

Then, the following holds true:

*Lemma 2 (*Fact 1, [10]*):* If W is irreducible, then its stationary distribution $\boldsymbol{\pi} = \mathrm{col}((\pi_i)_{i\in\mathcal{N}})$ is the normalized left eigenvector associate to the eigenvalue 1, i.e., $\boldsymbol{\pi} = \boldsymbol{\pi}W$. All components of $\boldsymbol{\pi}$ are strictly positive and if W is aperiodic, it holds that $\lim_{t\to\infty} W^t(i, j) = \pi_j$ for all $i, j \in \mathcal{N}$. □

Definition 3 (Weighted Network divergence): Given a common belief μ, for all $c \in \mathcal{C}$ the weighted network divergence is defined as

$$Z_\pi(\boldsymbol{\theta}^*, \boldsymbol{\theta}^c) = \sum_{i=1}^{N} \pi_i D_{KL}(f_i(y_i \mid x_i^*(\mu),\ \boldsymbol{\theta}_{-i}^*) \| f_i(y_i \mid x_i^*(\mu),\ \boldsymbol{\theta}_{-i}^c)),$$

where the subscript π indicates that the centrality is taken into account. □

Armed with these new assumptions and definitions, we can state a additional convergence results for our Algorithm 1:

Lemma 3: Let Assumptions 1 and 5 hold true. Then, for all $c \in \mathcal{C}$, it holds that

$$\sum_{i=1}^{N} \pi_i \frac{\mu_i^{(t)}(\boldsymbol{\theta}^c)}{\mu_i^{(t)}(\boldsymbol{\theta}^*)} \to \nu^c \text{ a.s., as } t \to \infty,$$

where $\nu^c \geq 0$ is a nonnegative random variable. □

Proof: The proof follows the same steps as the one of [9, Lemma 2] by taking the average with weights π_i. ■

Theorem 3: Let Assumptions 1, 4, 5 and 7 hold true. Then, the belief sequence $\{\mu_i^{(t)}\}_{t\geq 0}$, $i \in \mathcal{N}$ generated by Algorithm 1 converges a.s. to a common belief μ of the form

$$\mu(\boldsymbol{\theta}^c) = \frac{\nu^c}{\sum_{c=1}^{M} \nu^c} \text{ for all } c \in \mathcal{C}$$

where ν^c is as in Lemma 3. □

Proof: The proof follows the same steps as the one for [9, Th. 1], by noting that $\sum_{i\in\mathcal{N}}\pi_i = 1$ and using Lemmas 2 and 3 for the weighted average. ■

Theorem 4: Let Assumptions 1, 3-7 hold true. Then, we have the following: for each $i \in \mathcal{N}$,

1) $\lim_{T\to\infty} \frac{1}{\sum_{t=1}^{T}\alpha_i^{(t)}} \log \frac{\mu_i^{(T+1)}(\boldsymbol{\theta}^*)}{\mu_i^{(T+1)}(\boldsymbol{\theta}^c)} = Z_\pi(\boldsymbol{\theta}^*, \boldsymbol{\theta}^c)$;
2) $\mu_i^{(t)}(\boldsymbol{\theta}^*) \to 1$ as $t \to \infty$.

Proof: The proof follows by combining [9, Th. 2] and [10, Th. 1], in particular using the fact that $\lim_{t\to\infty} W^t(i,j) = \pi_j$ for all $i, j \in \mathcal{N}$. ■

IV. Learning the strategies of the other agents

We now focus on the scenario in which the agents infer the NE configuration by learning the single strategies of their opponents. Although the realized rewards still relate to the configurations as in (4), we consider here the effect of opponent $j \in \mathcal{N}\backslash\{i\}$ on such reward, i.e., the likelihood read as $f_{ij}(y_i \mid x_i,\ \theta_j^{k_j})$. Note that we put particular emphasis on the fact that the parameter $\theta_j^{k_j} \in \Theta_j$, $k_j \in M_j$ was played by agent j, i.e., we do not focus on the configuration but on what the single agents play and on how this affect agent i.

Also in this case, we define the set of reward equivalent parameters w.r.t. the optimal one, for each i and $j \in \mathcal{N}_i$:

$$\bar{\Theta}_{ij}(x_i^*) = \{\theta_j \in \Theta_j : \\ D_{KL}(f_{ij}(y_i \mid x_i^*,\ \theta_j^*) \| f_{ij}(y_i \mid x_i^*,\ \theta_j)) = 0\}.$$

The parameters in these sets are locally indistinguishable to agent i. To learn the true parameter, we shall therefore assume that the latter is *globally* identifiable, for each agent.

Assumption 8: For every $\theta_j \neq \theta_j^*$, $j \in \mathcal{N}$, there exists at least one agent i for which $D_{KL}(f_{ij}(y_i \mid x_i^*,\ \theta_j^*) \| f_{ij}(y_i \mid x_i^*,\ \theta_j)) > 0$ for all $x_i \in \mathcal{X}_i$. □

Thus, for all $j \in \mathcal{N}$, $\bigcap_{i\in\mathcal{N}} \bar{\Theta}_{ij}(x_i) = \{\theta_j^*\}$, i.e., even if the single agents can not locally distinguish the true parameter for the strategy of the j-th one, globally it is identifiable.

Example 2: Consider the game in Example 1 where Player 1 (P1) plays rows, Player 2 (P2) plays columns and Player 3 (P3) picks the tables, maximizing the obtainable reward. Here, the sets of reward equivalent parameters, according to the same likelihood used in Example 1, are

$$\bar{\Theta}_{12}(B) = \{L, R\} \qquad \bar{\Theta}_{13}(B) = \{1\}$$
$$\bar{\Theta}_{21}(L) = \{B, T\} \qquad \bar{\Theta}_{23}(L) = \{1, 2\}$$
$$\bar{\Theta}_{31}(1) = \{B\} \qquad \bar{\Theta}_{32}(1) = \{L, R\}$$

In this case, it is clear why Assumption 8 is needed: if either P1 or P3 cannot distinguish $\theta_2^* = L$, the equilibrium configuration might not be found. □

Then, to learn the true parameter, the agents keep a private belief $\mu_{i,j} \in \Delta(\Theta_j)$ and a posterior belief $b_{i,j} \in \Delta(\Theta_j)$ on the parameters of agent j, based on the realized reward.

Given the private belief, the agents can compute the expected reward to update their decision variables:

$$u_i(x_i, \mu_{i,-i}) = \sum_{c=1}^{M} u_i(x_i, \theta_{-i}^c)\mu_{i,-i}(\theta_{-i}^c), \tag{7}$$

Algorithm 2: Strategy Learning Algorithm

Initialization: $x_i^{(0)} \in \mathcal{X}_i$ for $i \in \mathcal{N}$ and $\mu_{i,j}^0 = \frac{1}{M_j}\mathbf{1}_{M_j}$ for $i, j \in \mathcal{N}$

Iteration t: For parameter $\theta_j^{k_j} \in \Theta_j$, Agent i

1) Updates the posterior belief:
$$b_{i,j}^{(t)}(\theta_j^{k_j}) = \frac{f_{i,j}(y_i^{(t)} \mid x_i^{(t)},\ \theta_j^{k_j})^{\alpha_i^{(t)}}\mu_{i,j}^{(t)}(\theta_j^{k_j})}{\sum_{\theta_j\in\Theta_j} f_{i,j}(y_i^{(t)} \mid x_i^{(t)},\ \theta_j)^{\alpha_i^{(t)}}\mu_{i,j}^{(t)}(\theta_j)}$$
2) Receives information $b_{\ell,j}^{(t)}(\theta_j^{k_j})$ from $\ell, j \in \mathcal{N}_i$ and updates the private belief:
$$\mu_{i,j}^{(t+1)}(\theta_j^{k_j}) = \frac{\exp(\sum_{\ell=1}^{N} w_{i\ell}\log b_{\ell,j}^{(t)}(\theta_j^{k_j}))}{\sum_{\theta_j\in\Theta_j}\exp(\sum_{\ell=1}^{N} w_{i\ell}\log b_{\ell,j}^{(t)}(\theta_j))}$$
3) Updates strategy:
$$x_i^{(t+1)} \in \mathrm{BR}_i(\mu_{i,j}^{(t+1)})$$

where $\mu_{i,-i}(\theta_{-i}^c) = \prod_{j\neq i, c\in\mathcal{C}_{k_j}} \mu_{i,j}(\theta_j^c)$ and $\mathcal{C}_{k_j} = \{c \in \mathcal{C} : \boldsymbol{\theta}^c = [\theta_1^c, \ldots, \theta_{j-1}^c,\ \theta_j^{k_j}, \theta_{j+1}^c, \ldots, \theta_N^c]\}$ is the set of configurations where agent j communicates the parameter $\theta_j^{k_j} \in \Theta_j$. In words, $\mu_{i,-i}$ is the collective belief of agent i on the parameters of the other agents composing the configuration $\boldsymbol{\theta}^c$, i.e., it is the probability of the configuration $\boldsymbol{\theta}^c$ obtained as the product of the θ_j^c, $j \in \mathcal{N}$ composing it. Note that θ_j^c, $j \in \mathcal{N}$, are independent as the communication noise is independent on the actions chosen by the agents. The belief and the decisions are then updated according to the set of instructions summarized in Algorithm 2, where the best-response is computed considering the reward in (7).

A. Convergence

Results similar to §III-A hold with some modifications of the assumptions required. Let us start with the likelihood.

Assumption 9: For each $i \in \mathcal{N}$, the likelihood function $f_i(y_i \mid x_i,\ \theta_j^{k_j})$ is continuous in x_i for all $\theta_j^{k_j} \in \Theta_j$, $j \in \mathcal{N}\setminus\{i\}$. Moreover, there exists $L > 0$ such that

$$\max_{i\in\mathcal{N}} \max_{\theta_j^{k_1}, \theta_j^{k_2}\in\Theta_j} \max_{x_i\in\mathcal{X}_i} \sup_{y_i} \left|\log \frac{f_{ij}(y_i \mid x_i,\ \theta_j^{k_1})}{f_{ij}(y_i \mid x_i,\ \theta_j^{k_2})}\right| \leq L. \ \square$$

Then, we can state the equivalent of Lemma 1 and Theorem 1 also for this different case:

Lemma 4: Let Assumption 1 and 5 hold true. Then, for all $\theta_j^{k_j} \in \Theta_j$, $j \in \mathcal{N}\setminus\{i\}$, it holds that

$$\frac{1}{N}\sum_{i=1}^{N} \frac{\mu_{ij}^{(t)}(\theta_j^{k_j})}{\mu_{ij}^{(t)}(\theta_j^*)} \to \nu_j^{k_j} \text{ a.s., as } t \to \infty,$$

where $\nu_j^{k_j} \geq 0$ is a nonnegative random variable. □

Theorem 5: Let Assumptions 1, 2, 5, 9 hold true. Then, the belief sequence $\{\mu_{ij}^{(t)}\}_{t\in\mathbb{N}}$, $i \in \mathcal{N}$ generated by Algo-

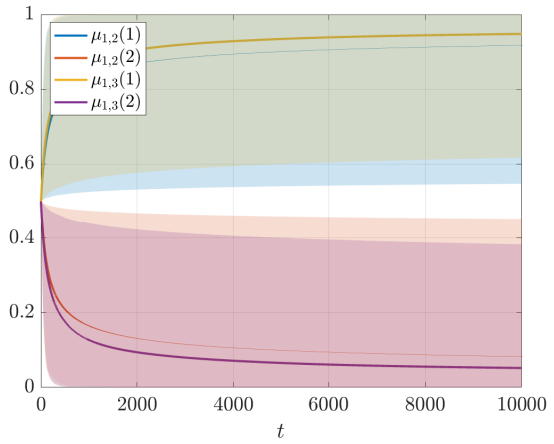

Fig. 3. Private belief of Player 1 on the strategies played by agents 2 and 3 over the iterations. Each solid line represents the mean value associated to each component of $\mu_{1,j}$, $j \in \{2,3\}$, while the shaded area of the same colour the resulting variance, both obtained by running 100 numerical experiments of the game in Example 2.

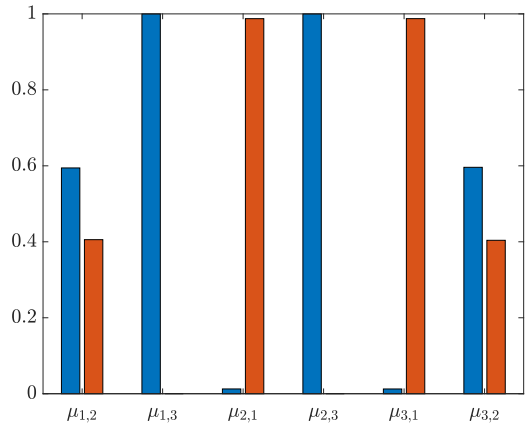

Fig. 4. Private beliefs of the three players after 10^4 iterations of Algorithm 2. The highest peaks correspond to the NE configuration.

rithm 2 converges a.s. to a common belief μ_j of the form

$$\mu_j(\theta_j^{k_j}) = \frac{\nu_j^{k_j}}{\sum_{k_j=1}^{M_j} \nu_j^{k_j}} \text{ for all } \theta_j^{k_j} \in \Theta_j, j \in \mathcal{N} \setminus \{i\}$$

where $\nu_j^{k_j}$ is defined as in Lemma 4. □

Proof: The proof follows the same steps as [9, Th. 1] and by assuming w.l.o.g. that $\theta_j^{k_j} = \theta_j^*$ for $k_j = 1$. ■

Theorem 5 ensures consensus of the beliefs on the strategies of agent j, however, it does not guarantee that the agents have learnt the true parameter. For the latter to happen, we first need to revise the definition of network divergence and adapt it to the agent-wise setup considered in this section.

Definition 4 (Agent-wise network divergence): For all $\theta_j^{k_j}$, $j \in \mathcal{N}$, the network divergence is defined as

$$Z_j(\theta_j^*, \theta_j^{k_j}) = \frac{1}{N} \sum_{i=1}^{N} D_{KL}(f_{ij}(y_i \mid x_i^*(\mu), \theta_j^*) \| f_{ij}(y_i \mid x_i^*(\mu), \theta_j^{k_j})),$$

with $\mu = \text{col}((\mu_j)_{j \in \mathcal{N}})$ common belief as in Theorem 5. □

Also in this case, if the best-response dynamics in Algorithm 2 converges once fixed the common beliefs (Assumption 6), we shall also have convergence to the true parameter:

Theorem 6: Let Assumptions 1, 2, 5, 6, 8 and 9 hold true. Then, the following hold: for each $i \in \mathcal{N}$,

1) $\lim_{T\to\infty} \frac{1}{\sum_{t=1}^{T} \alpha_i^{(t)}} \log \frac{\mu_{ij}^{(T+1)}(\theta_j^*)}{\mu_{ij}^{(T+1)}(\theta_j^{k_j})} = Z_j(\theta_j^*, \theta_j^{k_j})$;
2) $\mu_{i,j}^{(t)}(\theta_j^*) \to 1$, for all $j \in \mathcal{N} \setminus \{i\}$, as $t \to \infty$. □

Proof: The proof follows the same steps as that of [9, Th. 2] by simply extending the dimension of the problem to the new set of parameters. ■

Example 2 (Cont'd): We verify the results of this section by running Algorithm 2 on the three players game in Example 2, according to the methodology employed to generate Figs. 1–2. In Fig. 3 we report the private belief of Player 1 and its update over the iterations the iterations ($\mu_{2,j}^{(t)}$ and $\mu_{3,j}^{(t)}$ for Player 2 and 3, respectively, show a similar behaviour), while Figure 4 show the final (i.e., after 10^4 iterations) private belief for all the players. The strategies with the highest beliefs correctly identify the NE configuration. ■

V. Conclusions

In this paper we have shown that the NE configuration can be learnt through a non-bayesian learning. Future work will concentrate to consider games in fully continuous settings and different decision dynamics besides the best-response one, thus investigating whether the learning processes can be extended to more involved, yet realistic, cases of interest.

References

[1] B. Franci and S. Grammatico, "Stochastic generalized Nash equilibrium seeking under partial-decision information," *Automatica*, vol. 137, p. 110101, 2022.

[2] L. Pavel, "Distributed GNE seeking under partial-decision information over networks via a doubly-augmented operator splitting approach," *IEEE Transactions on Automatic Control*, vol. 65, no. 4, pp. 1584–1597, 2019.

[3] M. Bianchi, G. Belgioioso, and S. Grammatico, "Fast generalized Nash equilibrium seeking under partial-decision information," *Automatica*, vol. 136, p. 110080, 2022.

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

[5] S. Shahrampour, M. A. Rahimian, and A. Jadbabaie, "Switching to learn," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2918–2923.

[6] C. Eksin, P. Molavi, A. Ribeiro, and A. Jadbabaie, "Learning in linear games over networks," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2012, pp. 434–440.

[7] M. Wu, S. Amin, and A. Ozdaglar, "Multi-agent bayesian learning with adaptive strategies: Convergence and stability," *arXiv preprint arXiv:2010.09128*, 2020.

[8] ——, "Multi-agent bayesian learning with best response dynamics: Convergence and stability," *arXiv preprint arXiv:2109.00719*, 2021.

[9] S. Huang, J. Lei, and Y. Hong, "Distributed non-bayesian learning for games with incomplete information," *arXiv preprint arXiv:2303.07212*, 2023.

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