Distributed Learning based on Entropy-Driven Game Dynamics

Bruno Gaujal

joint work with Pierre Coucheney and Panayotis Mertikopoulos

Inria

Aug., 2014

Model

Shared resource systems (network, processors) can often be modeled by a finite game where:

- the structure of the game is unknown to each player (e.g. the set of players),
- players only observe the payoff of their chosen action,
- Observations (of the payoffs) may be corrupted by noise.

Learning Algorithm

In the following we consider a finite potential game.
Our goal is to design a distributed algorithm (executed by each user) that learns the Nash equilibria. Each user computes its strategy according to a sequential process that should satisfies the following desired properties:

- only uses in-game information (stateless),
- does not require time-coordination between the users (asynchronous),
- tolerates outdated measurements on the observations of the payoffs (delay-oblivious),
- tolerates random perturbations on the observations (robust),
- converges fast even if the number of users is very large (scalable).

Notations

- k is for players,
- $\alpha, \beta \in \mathcal{A}$ are for actions,
- $x \in \mathcal{X}$ is a mixed strategy profile,
- $u_{k\alpha}(x)$ is the (expected) payoff of k when playing α and facing x.

Notations

- k is for players,
- $\alpha, \beta \in \mathcal{A}$ are for actions,
- $x \in \mathcal{X}$ is a mixed strategy profile,
- $u_{k\alpha}(x)$ is the (expected) payoff of k when playing α and facing x.

Potential Games (Monderer and Shapley, 1996)

There exists a multilinear function $U : \mathcal{X} \to \mathbb{R}$ such that

$$u_{k\alpha}(x) - u_{k\beta}(x) = U(\alpha; x_{-k}) - U(\beta; x_{-k})$$

Notations

- k is for players,
- $\alpha, \beta \in \mathcal{A}$ are for actions,
- $x \in \mathcal{X}$ is a mixed strategy profile,
- $u_{k\alpha}(x)$ is the (expected) payoff of k when playing α and facing x.

Potential Games (Monderer and Shapley, 1996)

There exists a multilinear function $U : \mathcal{X} \to \mathbb{R}$ such that

$$u_{k\alpha}(x) - u_{k\beta}(x) = U(\alpha; x_{-k}) - U(\beta; x_{-k})$$

- Congestion games are potential games.
- Mechanism design can turn a game (e.g. general routing game with additive costs) into a potential game.

Basic Convergence Results in Potential Games

When facing a vector of payoffs $u \in \mathbb{R}^{\mathcal{A}}$ coming from a potential game, if players successively

- play best response $\Rightarrow$ (asymptotic) convergence to Nash equilibrium,

Basic Convergence Results in Potential Games

When facing a vector of payoffs $u \in \mathbb{R}^{\mathcal{A}}$ coming from a potential game, if players successively

- play best response $\Rightarrow$ (asymptotic) convergence to Nash equilibrium,
- play according to the Gibbs map $G : \mathbb{R}^{\mathcal{A}} \to \mathcal{X}$

$$G_\alpha(u) = \frac{e^{\gamma u_\alpha}}{\sum_\beta e^{\gamma u_\beta}}$$

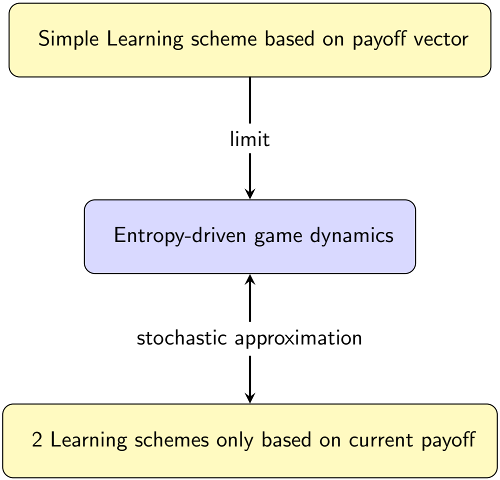
then convergence with high probability (when $\gamma \to \infty$) to a NE that globally maximizes the potential.

Basic Convergence Results in Potential Games II

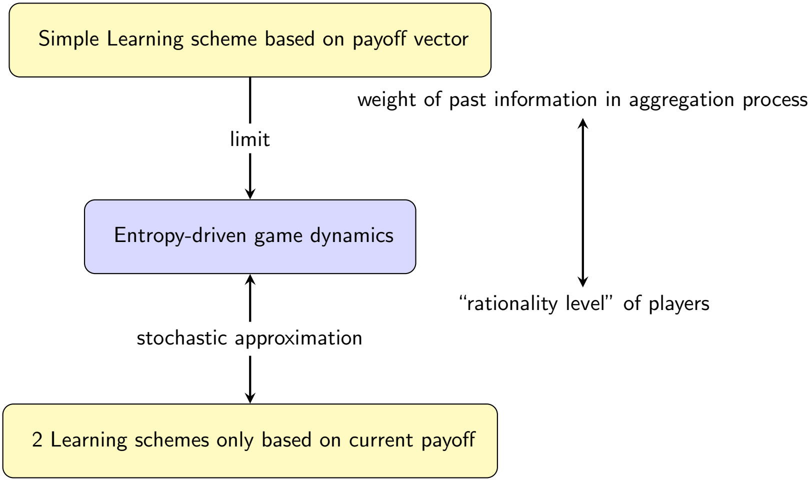
But in both cases,

- each players needs to know its payoff vector. Stateless: no
- Convergence may fail in the presence of noise. Robust: no
- Convergence may fail if players do not play one at a time. asynchronous: no
- Convergence may also fail with delayed observations. Delay-oblivious: no
- Convergence is slow when the number of players is large. Scalable: no

Outline of the Talk



Outline of the Talk



A General Learning Process

Let us consider a single agent repeatedly observing a stream of payoffs $u(t)$.

A General Learning Process

Let us consider a single agent repeatedly observing a stream of payoffs $u(t)$.

Two stages:

1. Score phase.

2. Choice phase.

A General Learning Process

Let us consider a single agent repeatedly observing a stream of payoffs $u(t)$.

Two stages:

1. Score phase.
   - $y_\alpha(t)$ is the score of $\alpha \in \mathcal{A}$,
     i.e. the agent assessment of α up to time t.
   - The score phase describes how $y_\alpha(t)$ is updated
     using the payoffs $u_\alpha(s)$, $s \leq t$.
2. Choice phase.

A General Learning Process

Let us consider a single agent repeatedly observing a stream of payoffs $u(t)$.

Two stages:

1. Score phase.
   - $y_\alpha(t)$ is the score of $\alpha \in \mathcal{A}$,
     i.e. the agent assessment of α up to time t.
   - The score phase describes how $y_\alpha(t)$ is updated
     using the payoffs $u_\alpha(s)$, $s \leq t$.
2. Choice phase.
   - Specifies the choice map $Q : \mathbb{R}^{\mathcal{A}} \to \mathcal{X}$ which prescribes the agent's mixed strategy $x \in \mathcal{X}$ given his assessment of actions (the vector y).

Score Stage

The score is a discounted aggregation of the payoffs:

$$y_\alpha(t) = \int_0^t e^{T(t-s)} u_\alpha(x(s)) ds,$$

Score Stage

The score is a discounted aggregation of the payoffs:

$$y_\alpha(t) = \int_0^t e^{T(t-s)} u_\alpha(x(s)) ds,$$

In differential form:

$$\dot{y}_\alpha(t) = u_\alpha(t) - T y_\alpha(t)$$

Score Stage

The score is a discounted aggregation of the payoffs:

$$y_\alpha(t) = \int_0^t e^{T(t-s)} u_\alpha(x(s)) ds,$$

In differential form:

$$\dot{y}_\alpha(t) = u_\alpha(t) - T y_\alpha(t)$$

and T is the discount factor of the learning scheme.

- $T > 0$: exponentially more weight to recent observations.
- $T = 0$: no discounting: the score is the aggregated payoff
- $T < 0$: reinforcing past observations.

Choice Stage: Smoothed Best-Response

At each decision time t, the agent chooses a mixed action $x \in \mathcal{X}$ according to the choice map $Q : \mathbb{R}^{\mathcal{A}} \to \mathcal{X}$

$$x(t) = Q(y(t)).$$

Choice Stage: Smoothed Best-Response

At each decision time t, the agent chooses a mixed action $x \in \mathcal{X}$ according to the choice map $Q : \mathbb{R}^{\mathcal{A}} \to \mathcal{X}$

$$x(t) = Q(y(t)).$$

We assume $Q(y)$ to be a smoothed best-response, i.e. the unique solution of

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_\beta x_\beta y_\beta - h(x), \\ \text{subject to} \quad & x \geq 0, \textstyle\sum_\beta x_\beta = 1, \end{aligned}$$

where the entropy function h is smooth, strictly convex on $\mathcal{X}$ such that

$$|\mathrm{d}h(x)| \to \infty \text{ whenever } x \to \mathrm{bd}(\mathcal{X}).$$

Entropy-Driven Learning Dynamics

If the agent's payoffs are coming from a finite game, the continuous-time learning dynamics is

Score Dynamics

$$\dot{y}_{k\alpha} = u_{k\alpha}(x) - T y_{k\alpha},$$

$$x_k = Q_k(y_k),$$

where

- Q_k is the choice map of the driving entropy of player k, $h_k : \mathcal{X}_k \to \mathbb{R}$,
- T is the discounting parameter.

Entropy-Driven Learning Dynamics

If the agent's payoffs are coming from a finite game, the continuous-time learning dynamics is

Score Dynamics

$$\dot{y}_{k\alpha} = u_{k\alpha}(x) - T y_{k\alpha},$$
$$x_k = Q_k(y_k),$$

where

- Q_k is the choice map of the driving entropy of player k, $h_k : \mathcal{X}_k \to \mathbb{R}$,
- T is the discounting parameter.

This is the score-based formulation of the dynamics.

Strategy-based Formulation of the Dynamics

Let us assume that h is decomposable: $h(x) = \sum_{\alpha} \theta(x_\alpha)$. By setting

$$\Theta''(x) = \left[\sum\nolimits_{\beta} 1/\theta''(x_\beta)\right]^{-1}.$$

Then, a little algebra yields:

Strategy Dynamics

$$\begin{aligned}\dot{x}_\alpha \quad = \quad & \frac{1}{\theta''(x_\alpha)}\left[u_\alpha(x) - \Theta''(x)\sum\nolimits_{\beta}\frac{u_\beta(x)}{\theta''(x_\beta)}\right] \\ & - \frac{T}{\theta''(x_\alpha)}\left[\theta'(x_\alpha) - \Theta''_k(x)\sum\nolimits_{\beta}\frac{\theta'(x_\beta)}{\theta''(x_\beta)}\right],\end{aligned}$$

Boltzmann-Gibbs Entropy

Let $h(x) = \sum_{\alpha} x_\alpha \log(x_\alpha)$. Then the dynamics become:

$$\dot{y}_\alpha = u_\alpha(x) - Ty_\alpha,$$
$$x = \mathit{Gibbs}(y),$$

and

$$\dot{x}_\alpha = x_\alpha \left[u_\alpha(x) - \sum\nolimits_\beta x_\beta u_\beta(x) \right] - T \; x_\alpha \left[\log x_\alpha - \sum\nolimits_\beta x_\beta \log x_\beta \right]$$

Replicator Dynamics

Entropy-adjustment term

Rest Points of the Dynamics

Proposition

1. For $T = 0$, the rest points of the entropy-driven dynamics are the restricted Nash equilibria of the game.

Rest Points of the Dynamics

Proposition

1. For $T = 0$, the rest points of the entropy-driven dynamics are the restricted Nash equilibria of the game.
2. For $T > 0$, the rest points coincide with the restricted QRE (solutions of $x = Q(u/T)$, that converge to restricted NE when $T \to 0$)

Rest Points of the Dynamics

Proposition

1. For $T = 0$, the rest points of the entropy-driven dynamics are the restricted Nash equilibria of the game.
2. For $T > 0$, the rest points coincide with the restricted QRE (solutions of $x = Q(u/T)$, that converge to restricted NE when $T \to 0$)
3. For $T < 0$, the rest points are the restricted QRE of the opposite game (opposite payoffs).

Rest Points of the Dynamics

Proposition

1. For $T = 0$, the rest points of the entropy-driven dynamics are the restricted Nash equilibria of the game.
2. For $T > 0$, the rest points coincide with the restricted QRE (solutions of $x = Q(u/T)$, that converge to restricted NE when $T \to 0$)
3. For $T < 0$, the rest points are the restricted QRE of the opposite game (opposite payoffs).

The parameter T plays a double role:

- it reflects the importance given by players to past events (the discount factor),
- it determines the rationality level of rest points.

Convergence for Potential Games

Lemma

Let $U : \mathcal{X} \to \mathbb{R}$ be the potential of a finite potential game. Then, $F(x) \equiv U(x) - Th(x)$ is strictly Lyapunov for the entropy-driven dynamics.

Convergence for Potential Games

Lemma

Let $U : \mathcal{X} \to \mathbb{R}$ be the potential of a finite potential game. Then, $F(x) \equiv U(x) - Th(x)$ is strictly Lyapunov for the entropy-driven dynamics.

- the dynamics tend to move towards the maximizers of F for every parameter T,
- the parameter modifies the set of maximizers.

Convergence for Potential Games

Lemma

Let $U : \mathcal{X} \to \mathbb{R}$ be the potential of a finite potential game. Then, $F(x) \equiv U(x) - Th(x)$ is strictly Lyapunov for the entropy-driven dynamics.

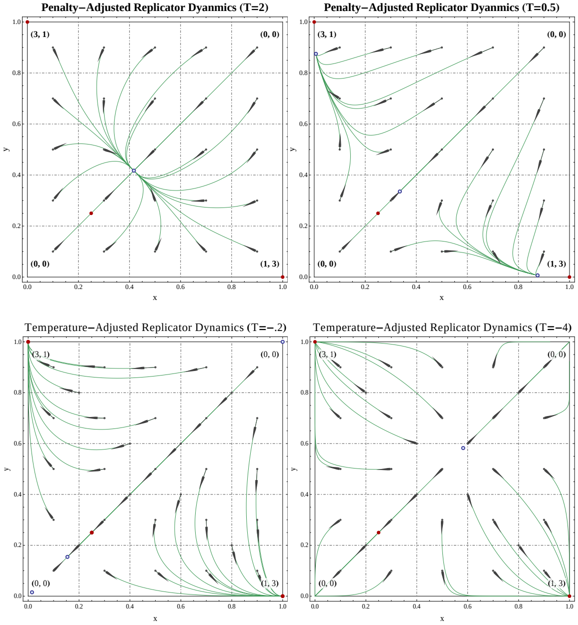
- the dynamics tend to move towards the maximizers of F for every parameter T,
- the parameter modifies the set of maximizers.

Proposition

Let $x(t)$ be a trajectory of the entropy-driven dynamics for a potential game. Then:

1. For $T > 0$, $x(t)$ converges to a QRE.
2. For $T = 0$, $x(t)$ converges to a restricted Nash equilibrium.
3. For $T < 0$, $x(t)$ converges to a vertex or is stationary.

Phase portrait of the parameter-adjusted replicator dynamics in a 2×2 potential game.



From Continuous-Time Dynamics to Distributed Algorithm

- Aim: discretization of the entropy-driven dynamics with the same convergence properties than the continuous dynamics.

From Continuous-Time Dynamics to Distributed Algorithm

- Aim: discretization of the entropy-driven dynamics with the same convergence properties than the continuous dynamics.
- Main difficulty: a distributed algorithm should only use random samples of the payoff $u(x)$, i.e. $u(A)$ where A is a random action profile with distribution x.

We derive two stochastic approximations of the dynamics respectively with scores and strategies updates.

The Stochastic Approximation Tool

We will consider the random process

$$Z(n+1) = Z(n) + \gamma_{n+1}\left(f(Z(n)) + V(n+1)\right)$$

as a stochastic approximation of the ODE

$$\dot{z} = f(z)$$

where

- (γ_n) is the sequence of step-sizes assumed to be $(\ell^2 - \ell^1)$ summable series (typically, $\gamma_n = 1/n$),
- $\mathbb{E}[V(n+1) \mid \mathcal{F}_n] = 0$.

The Stochastic Approximation Tool

Theorem (Benaim, 99)

Assume that

- the ODE admits a strict Lyapunov function,
- weak condition on the Lyapunov function,
- $\sup_n \mathbb{E}[\|V(n)\|^2] < \infty$,
- $\sup_n \|Z(n)\| < \infty$ a.s.,

Then the set of accumulation points of the sequence $(Z(n))$ generated by the stochastic approximation of the ODE

$$\dot{z} = f(z)$$

almost surely belongs to a connected invariant set of the ODE.

Score-Based learning with imperfect payoff monitoring

Recall the score based version of the dynamics:

$$\dot{y}_{k\alpha} = u_{k\alpha}(x) - T y_{k\alpha},$$

$$x_k = Q_k(y_k),$$

A stochastic approximation yields the following algorithm:

Score-based Algorithm

1. At stage $n+1$, each player selects an action $\alpha_k(n+1) \in \mathcal{A}_k$ based on a mixed strategy $X_k(n) \in \mathcal{X}_k$.
2. Every player gets bounded and unbiased estimates $\hat{u}_{k\alpha}(n+1)$ s.t.
   2.1 $\mathbb{E}\left[\hat{u}_{k\alpha}(n+1) \mid \mathcal{F}_n\right] = u_{k\alpha}(X(n))$,
   2.2 $|\hat{u}_{k\alpha}(n+1)| \leq C$ (a.s.),
3. For each action α, the score is updated: $Y_{k\alpha}(n+1) \leftarrow Y_{k\alpha}(n) + \gamma_n(\hat{u}_{k\alpha}(n+1) - T Y_{k\alpha}(n))$;
4. Each player updates its mixed strategy $X_k(n+1) \leftarrow Q_k(Y_k(n+1))$; and the process repeats ad infinitum.

Score-Based Algorithm Assessment

Theorem

The Score-Based Algorithm converges (a.s.) to a connected set of QRE of $\mathfrak{G}$ with rationality parameter $1/T$.

- Each players needs to know its payoff vector. Stateless: no
- Convergence to QRE in the presence of noise. Robust: yes
- Players play simultaneously. Asynchronous: no
- Convergence may also fail with delayed observations. Delay-oblivious: no
- Convergence is fast when the number of players is large. Scalable: yes (based on numerical evidence).

Score-Based learning using in-game payoff

Assume now that the only information at the players' disposal is the payoff of their chosen actions (possibly perturbed by some random noise process).

$$\hat{u}_k(n+1) = u_k(\alpha_1(n+1), \ldots, \alpha_N(n+1)) + \xi_k(n+1),$$

where the noise ξ_k is a bounded, $\mathcal{F}$-adapted martingale difference (i.e. $\mathbb{E}\left[\xi_k(n+1) \mid \mathcal{F}_n\right] = 0$ and $|\xi_k| \leq C$ for some $C > 0$) with $\xi_k(n+1)$ independent of $\alpha_k(n+1)$.
We can use the the unbiased estimator

$$\hat{u}_{k\alpha}(n{+}1) = \hat{u}_k(n{+}1) \cdot \frac{\mathbb{1}(\alpha_k(n+1) = \alpha)}{\mathbb{E}\left(\alpha_k(n+1) = \alpha \mid \mathcal{F}_n\right)} = \mathbb{1}(\alpha_k(n{+}1) = \alpha) \cdot \frac{\hat{u}_k(n+1)}{X_{k\alpha}(n)}$$

Score-Based learning using in-game payoff (II)

This allows us to replace the inner action-sweeping loop of the Scored-Based Algorithm with the update score:

$$Y_{k\alpha_k} \leftarrow Y_{k\alpha_k} + \gamma_n(\hat{u}_k - TY_{k\alpha_k})/X_{k\alpha_k}$$

This algorithm works well in practice (it converges to a QRE in potential games whenever $T > 0$).

But both conditions

1. $\sup_n \mathbb{E}[\|V(n)\|^2] < \infty$,
2. $\sup_n \|Z(n)\| < \infty$ a.s.,

are hard (impossible?) to prove (in fact $2 \Rightarrow 1$).

Strategy-Based Algorithm

The strategy-based version of the dynamics is

$$\dot{x}_\alpha = \frac{1}{\theta''(x_\alpha)} \left[u_\alpha(x) - \Theta''(x_k) \sum\nolimits_\beta \frac{u_\beta(x)}{\theta''(x_\beta)} - Tg_{\alpha,\theta}(X_k) \right],$$

A stochastic approximation of the strategy based dynamics is

$$X_{k\alpha} \stackrel{+}{=} \gamma_n \times \frac{1}{\theta''(X_{k\alpha})} \left[\frac{u_k(A)}{X_{kA_k}} \left(\mathbb{1}(A_k = \alpha) - \frac{\Theta''(X_k)}{\theta''(X_{kA_k})} \right) - Tg_{\alpha,\theta}(X_k) \right]$$

where A is the randomly chosen action profile with distribution X.

Algorithm assessment

Theorem

When $T > 0$, the sequence $(X(n))$ generated by the strategy-based algorithm remains positive and converges almost surely to a connected set of QRE of $\mathfrak{G}$ with rationality level $1/T$.

- Each players only observes in-game payoff. Stateless: yes
- Convergence to QRE in the presence of noise. Robust: yes
- Players play simultaneously Asynchronous: no
- no delay in observations Delay-oblivious: no
- Convergence is fast when the number of players is large. Scalable: yes (based on numerical evidence).

Asynchronous version of the Algorithm

Using extensions of the stochatic approximation theorem (Borkar, 2008), converge to QRE is still true for the asynchronous and delayed version of the algorithm:
let $d_{j,k}(n)$ de the (integer-valued) lag between player j and k when k plays at stage n. The observed payoff $\hat{u}_k(n+1)$ of k at stage $n+1$ depends on his opponents' past actions:

$$\mathbb{E}\left[\hat{u}_k(n+1) \mid \mathcal{F}_n\right] = u_k\left(X_1(n-d_{1,k}(n)), \ldots, X_k(n), \ldots, X_N(n-d_{N,k}(n))\right). \tag{1}$$

Asynchronous version of the Algorithm

Algorithm 1: Strategy-based learning for one player (with asynchronous and imperfect in-game observations)

$n \leftarrow 0$;

Initialize $X_k \in \operatorname{rel\,int}(\mathcal{X}_k)$ as a mixed strategy with full support;

repeat

until Next **Play** occurs at time $\tau_k(n+1) \in \mathcal{T}$; # Local time clock

$n \leftarrow n+1$;

select new action α_k according to mixed strategy X_k; # current action

observe $\hat{u}_k$; # receive measured payoff

foreach action $\alpha \in \mathcal{A}_k$ **do**

$$X_{k\alpha} {+}{=}\, \gamma_n \left[\left(\mathbb{1}_{\alpha_k=\alpha} - X_{k\alpha} \right) \cdot \hat{u}_k - T X_{k\alpha} \left(\log X_{k\alpha} - \textstyle\sum_\beta^k X_{k\beta} \log X_{k\beta} \right) \right]$$

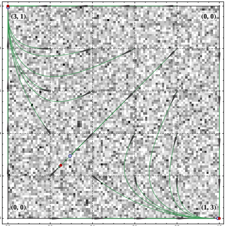
update strategy

until termination criterion is reached;

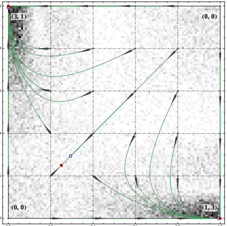
Algorithm assessment

- Each players only observes in-game payoff. Stateless: yes
- Convergence to QRE in the presence of noise. Robust: yes
- Players play according to local timers. Asynchronous: yes
- Convergence with delayed observations. Delay-oblivious: yes
- Convergence is fast when the number of players is large. Scalable: yes (based on numerical evidence).

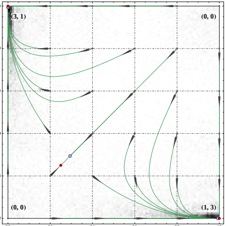
Numerical experiments



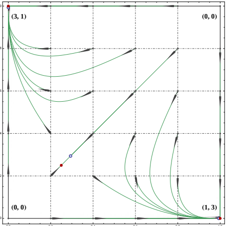
(a) Initial db ($n = 0$).



(b) Db. with $n = 2$.



(c) Db with $n = 5$.



(d) Db. with $n = 10$.

Merci!