LEARNING IN CONCAVE GAMES

P. Mertikopoulos

French National Center for Scientific Research (CNRS)

Laboratoire d'Informatique de Grenoble

French Symposium on Games – Paris, May 29, 2015

Context and motivation

- Concave games:
  - finitely many players
  - continuous action spaces
  - individually concave payoff functions

Context and motivation

- Concave games:
  - finitely many players
  - continuous action spaces
  - individually concave payoff functions
- Context/applications:
  - Standard in economics & finance (multi-portfolio optimization, auctions, oligopolies,...)
  - Networking (routing, tolling, network economics,...)
  - Electrical engineering (wireless communications, electricity grids,...)
  - ...

Context and motivation

- Concave games:
  - finitely many players
  - continuous action spaces
  - individually concave payoff functions
- Context/applications:
  - Standard in economics & finance (multi-portfolio optimization, auctions, oligopolies,…)
  - Networking (routing, tolling, network economics,…)
  - Electrical engineering (wireless communications, electricity grids,…)
  - …

What this talk is about:

How can players converge to a stable, equilibrium state by learning from past instances of play?

Basic Definitions

A *concave game* consists of:

- A finite set of *players* $\mathcal{N} = \{1, \dots, N\}$.
- A compact, convex set of *actions* $x_k \in \mathcal{X}_k$ per player.
- An individually concave *payoff function* $u_k \colon \prod_k \mathcal{X}_k \to \mathbb{R}$ per player, i.e.

$$u_k(x_k; x_{-k}) \text{ is concave in } x_k \text{ for all } x_{-k} \in \textstyle\prod_{\ell \neq k} \mathcal{X}_\ell.$$

Basic Definitions

A *concave game* consists of:

- A finite set of *players* $\mathcal{N} = \{1, \dots, N\}$.
- A compact, convex set of *actions* $x_k \in \mathcal{X}_k$ per player.
- An individually concave *payoff function* $u_k \colon \prod_k \mathcal{X}_k \to \mathbb{R}$ per player, i.e.

$$u_k(x_k; x_{-k}) \text{ is concave in } x_k \text{ for all } x_{-k} \in \textstyle\prod_{\ell \neq k} \mathcal{X}_\ell.$$

Fine print:

- Each $\mathcal{X}_k$ assumed to live in a finite-dimensional *ambient space* $\mathcal{V}_k \equiv \mathbb{R}^{d_k}$.
  (No infinite dimensionalities in this talk!)
- The product $\mathcal{X} \equiv \prod_k \mathcal{X}_k$ will be called the game's *action space*.
  Corresponding ambient space: $\mathcal{V} = \prod_k \mathcal{V}_k$.
- Each ambient space $\mathcal{V}_k$ equipped with a norm $\|\cdot\|$.

Example 1: Finite (Linear) Games

A *finite game* consists of:

- A finite set of players $\mathcal{N} = \{1, \dots, N\}$.
- A finite set of actions $\alpha_k \in \mathcal{A}_k$ per player.
- Each player's payoff function $u_k \colon \prod_k \mathcal{A}_k \to \mathbb{R}$.

In the *mixed extension* of a finite game, players can play *mixed strategies* $x_k \in \Delta(\mathcal{A}_k)$.

Corresponding (expected) payoff:

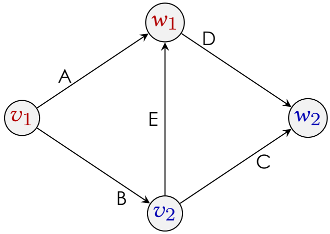
$$u_k(x) = \sum_{\alpha_1 \in \mathcal{A}_1} \cdots \sum_{\alpha_N \in \mathcal{A}_N} x_{1,\alpha_1} \cdots x_{N,\alpha_N} \, u_k(\alpha_1, \dots, \alpha_N)$$

The mixed strategy space $\mathcal{X}_k = \Delta(\mathcal{A}_k)$ is convex and $u_k(x_k; x_{-k})$ is linear in x_k
$\implies$ mixed extensions of finite games are concave

Example 2: Routing

Consider the following model of Internet congestion:

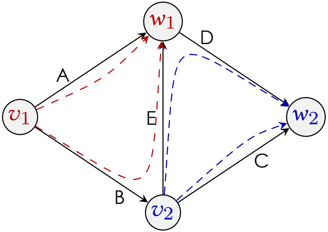
- *Origin nodes* (v_k) generate traffic that must be routed to intended *destination nodes* (w_k)



Example 2: Routing

Consider the following model of Internet congestion:

- *Origin nodes* (v_k) generate traffic that must be routed to intended *destination nodes* (w_k)
- Set of *paths* $\mathcal{A}_k$ joining $v_k \rightsquigarrow w_k$.

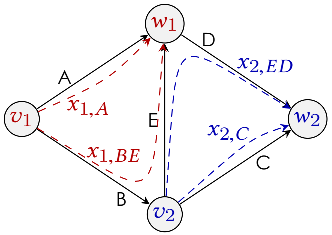


Example 2: Routing

Consider the following model of Internet congestion:

- *Origin nodes* (v_k) generate traffic that must be routed to intended *destination nodes* (w_k)
- Set of *paths* $\mathcal{A}_k$ joining $v_k \rightsquigarrow w_k$.
- Actions: split traffic flows over different paths

$$\mathcal{X}_k = \{x_k : x_{k\alpha} \geq 0 \text{ and } \sum_{\alpha \in \mathcal{A}_k} x_{k\alpha} = \rho_k\}$$



Example 2: Routing

Consider the following model of Internet congestion:

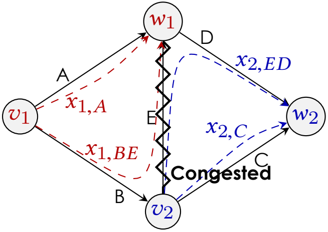
- *Origin nodes* (v_k) generate traffic that must be routed to intended *destination nodes* (w_k)
- Set of *paths* $\mathcal{A}_k$ joining $v_k \rightsquigarrow w_k$.
- Actions: split traffic flows over different paths

$$\mathcal{X}_k = \{x_k : x_{k\alpha} \geq 0 \text{ and } \sum_{\alpha\in\mathcal{A}_k} x_{k\alpha} = \rho_k\}$$

- *Path latency:*

$$\ell_{k\alpha}(x) \equiv \sum_{r\in\alpha} \ell_r(y_r)$$

where $y_e = \sum_k \sum_{\alpha\ni e} x_{k\alpha}$ is the total load on link e and $\ell_e(y_e)$ is the induced delay.



Example 2: Routing

Consider the following model of Internet congestion:

- *Origin nodes* (v_k) generate traffic that must be routed to intended *destination nodes* (w_k)
- Set of *paths* $\mathcal{A}_k$ joining $v_k \rightsquigarrow w_k$.
- Actions: split traffic flows over different paths

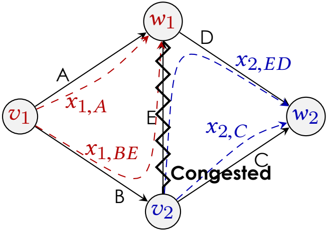
$$\mathcal{X}_k = \{x_k : x_{k\alpha} \geq 0 \text{ and } \sum_{\alpha \in \mathcal{A}_k} x_{k\alpha} = \rho_k\}$$

- *Path latency:*

$$\ell_{k\alpha}(x) \equiv \sum_{r \in \alpha} \ell_r(y_r)$$

where $y_e = \sum_k \sum_{\alpha \ni e} x_{k\alpha}$ is the total load on link e and $\ell_e(y_e)$ is the induced delay.

- Payoff: $u_k(x) = -\sum_{\alpha \in \mathcal{A}_k} x_{k\alpha} \ell_{k\alpha}(x)$.



Example 2: Routing

Consider the following model of Internet congestion:

- *Origin nodes* (v_k) generate traffic that must be routed to intended *destination nodes* (w_k)
- Set of *paths* $\mathcal{A}_k$ joining $v_k \rightsquigarrow w_k$.
- Actions: split traffic flows over different paths

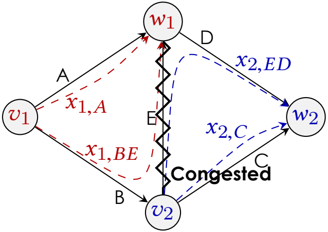
$$\mathcal{X}_k = \{x_k : x_{k\alpha} \geq 0 \text{ and } \sum_{\alpha\in\mathcal{A}_k} x_{k\alpha} = \rho_k\}$$

- *Path latency:*

$$\ell_{k\alpha}(x) \equiv \sum_{r\in\alpha} \ell_r(y_r)$$

where $y_e = \sum_k \sum_{\alpha\ni e} x_{k\alpha}$ is the total load on link e and $\ell_e(y_e)$ is the induced delay.

- Payoff: $u_k(x) = -\sum_{\alpha\in\mathcal{A}_k} x_{k\alpha}\ell_{k\alpha}(x)$.



Under standard assumptions for ℓ_e (convex, increasing), u_k is concave in x_k
$\implies \mathcal{G}(\mathcal{N},\mathcal{X},u)$ is a concave game

Nash Equilibrium and Marginal Utilities

A *Nash equilibrium* is an action profile $x^* \in \mathcal{X}$ such that

$$u_k(x_k^*; x_{-k}^*) \geq u_k(x_k; x_{-k}^*) \quad \text{for every unilateral deviation } x_k \in \mathcal{X}_k, k \in \mathcal{N}$$

$\Longrightarrow$ There is no direction that could unilaterally increase a player's payoff.

Nash Equilibrium and Marginal Utilities

A *Nash equilibrium* is an action profile $x^* \in \mathcal{X}$ such that

$$u_k(x_k^*; x_{-k}^*) \geq u_k(x_k; x_{-k}^*) \quad \text{for every unilateral deviation } x_k \in \mathcal{X}_k, k \in \mathcal{N}$$

$\implies$ There is no direction that could unilaterally increase a player's payoff.

Alternative characterization: define the *marginal utility vector* of player k as:

$$v_k(x) = \nabla_k u_k(x) \equiv \nabla u_k(x_k; x_{-k}),$$

(differentiation taken only w.r.t. x_k; the opponents' profile x_{-k} is kept fixed)

Since u_k is concave, x^* is an equilibrium if and only if

$$\langle v_k(x^*) | x_k - x_k^* \rangle \leq 0 \quad \text{for all } x_k \in \mathcal{X}_k, k \in \mathcal{N}.$$

Assumption

$v_k(x)$ *assumed Lipschitz throughout this talk.*

Fine print: $x_k \in \mathcal{X}_k \subseteq \mathcal{V}_k$ treated as primal variables; payoff gradients $v_k \in \mathcal{V}_k^*$ treated as duals.

Equilibrium Existence and Uniqueness

Every concave game admits a Nash equilibrium (Debreu, 1952; Rosen, 1965).

What about uniqueness?

Equilibrium Existence and Uniqueness

Every concave game admits a Nash equilibrium (Debreu, 1952; Rosen, 1965).

What about uniqueness?

Theorem (Rosen, 1965)

Suppose that the players' marginal utilities satisfy the monotonicity property:

$$\sum_k \lambda_k \left\langle v_k(x') - v_k(x) \middle| x'_k - x_k \right\rangle < 0 \tag{R}$$

for some $\lambda \succ 0$ and for all $x, x' \in \mathcal{X}$. Then, the game admits a unique Nash equilibrium.

Equilibrium Existence and Uniqueness

Every concave game admits a Nash equilibrium (Debreu, 1952; Rosen, 1965).

What about uniqueness?

Theorem (Rosen, 1965)

Suppose that the players' marginal utilities satisfy the monotonicity property:

$$\sum_k \lambda_k \left\langle v_k(x') - v_k(x) \middle| x'_k - x_k \right\rangle < 0 \qquad \text{(R)}$$

for some $\lambda \succ 0$ and for all $x, x' \in \mathcal{X}$. Then, the game admits a unique Nash equilibrium.

- Rosen (1965) calls this condition *diagonal strict concavity*.
- Define the *λ-weighted Hessian* $H(x;\lambda) = \bigoplus_{j,k} H_{jk}(x;\lambda)$ of the game as:

$$H_{jk}(x;\lambda) = \lambda_j \nabla_k v_j(x) + \lambda_k \left(\nabla_j v_k(x)\right)^\top$$

If $H(x;\lambda) \prec 0$ for all $x \in \mathcal{X}$, the game admits a unique equilibrium.

Learning via Payoff Gradient Ascent

A natural idea to improve one's payoff: ascend the payoff gradient

$$x \leftarrow x + \gamma\, v(x)$$

where γ is a (possibly variable) step-size parameter.

Problem: must respect players' action constraints ($x \in \mathcal{X}$)

Learning via Payoff Gradient Ascent

A natural idea to improve one's payoff: ascend the payoff gradient

$$x \leftarrow x + \gamma\, v(x)$$

where γ is a (possibly variable) step-size parameter.

Problem: must respect players' action constraints ($x \in \mathcal{X}$)

To do that, rewrite the gradient ascent process

$$y \leftarrow y + \gamma\, v(x),$$

$$x \leftarrow y,$$

Learning via Payoff Gradient Ascent

A natural idea to improve one's payoff: ascend the payoff gradient

$$x \leftarrow x + \gamma\, v(x)$$

where γ is a (possibly variable) step-size parameter.

Problem: must respect players' action constraints ($x \in \mathcal{X}$)

To do that, rewrite the gradient ascent process and project:

$$y \leftarrow y + \gamma\, v(x),$$

$$x \leftarrow \arg\min_{x' \in \mathcal{X}} \|y - x'\|_2,$$

Learning via Payoff Gradient Ascent

A natural idea to improve one's payoff: ascend the payoff gradient

$$x \leftarrow x + \gamma\, v(x)$$

where γ is a (possibly variable) step-size parameter.

Problem: must respect players' action constraints ($x \in \mathcal{X}$)

To do that, rewrite the gradient ascent process and project:

$$y \leftarrow y + \gamma\, v(x),$$

$$x \leftarrow \arg\max_{x' \in \mathcal{X}} \left\{ \langle y | x' \rangle - \tfrac{1}{2} \|x'\|_2^2 \right\},$$

Learning via Payoff Gradient Ascent

A natural idea to improve one's payoff: ascend the payoff gradient

$$x \leftarrow x + \gamma\, v(x)$$

where γ is a (possibly variable) step-size parameter.

Problem: must respect players' action constraints ($x \in \mathcal{X}$)

To do that, rewrite the gradient ascent process **and regularize:**

$$y \leftarrow y + \gamma\, v(x),$$

$$x \leftarrow \arg\max_{x' \in \mathcal{X}} \left\{ \langle y | x' \rangle - h(x') \right\},$$

where the *penalty function* $h\colon \mathcal{X} \to \mathbb{R}$ is smooth and strongly convex:

$$h(tx + (1-t)x') \leq th(x) + (1-t)h(x') - \tfrac{1}{2} K t(1-t) \|x' - x\|^2$$

for some $K > 0$ and for all $t \geq 0$, $x, x' \in \mathcal{X}$.

Examples

1. The quadratic penalty $h(x) = \frac{1}{2}\sum_\alpha x_\alpha^2 = \frac{1}{2}\|x\|_2^2$ gives the *Euclidean projection*

$$\Pi(y) = \arg\max_{x\in\mathcal{X}}\{\langle y|x\rangle - \tfrac{1}{2}\|x\|_2^2\} = \arg\min_{x\in\mathcal{X}}\|y-x\|_2^2$$

Examples

1. The quadratic penalty $h(x) = \frac{1}{2}\sum_\alpha x_\alpha^2 = \frac{1}{2}\|x\|_2^2$ gives the *Euclidean projection*

$$\Pi(y) = \arg\max_{x \in \mathcal{X}} \{\langle y|x\rangle - \tfrac{1}{2}\|x\|_2^2\} = \arg\min_{x \in \mathcal{X}} \|y - x\|_2^2$$

2. If $\mathcal{X} = \Delta_d$, the (negative) *Gibbs entropy* $h(x) = \sum_\alpha x_\alpha \log x_\alpha$ gives the *logit map*

$$G(y) = \frac{(\exp(y_1), \dots, \exp(y_d))}{\sum_{\alpha=1}^d \exp(y_\alpha)}$$

Examples

1. The quadratic penalty $h(x) = \frac{1}{2}\sum_\alpha x_\alpha^2 = \frac{1}{2}\|x\|_2^2$ gives the *Euclidean projection*

$$\Pi(y) = \arg\max_{x\in\mathcal{X}}\{\langle y|x\rangle - \tfrac{1}{2}\|x\|_2^2\} = \arg\min_{x\in\mathcal{X}}\|y-x\|_2^2$$

2. If $\mathcal{X} = \Delta_d$, the (negative) *Gibbs entropy* $h(x) = \sum_\alpha x_\alpha \log x_\alpha$ gives the *logit map*

$$G(y) = \frac{(\exp(y_1),\dots,\exp(y_d))}{\sum_{\alpha=1}^d \exp(y_\alpha)}$$

3. If $\mathcal{X} = \{\mathbf{X} : \mathbf{X} \succcurlyeq 0, \operatorname{tr}(\mathbf{X}) \leq 1\}$, the *von Neumann entropy* $h(\mathbf{X}) = \operatorname{tr}[\mathbf{X}\log\mathbf{X}]$ gives

$$Q(\mathbf{Y}) = \frac{\exp(\mathbf{Y})}{1+\operatorname{tr}[\exp(\mathbf{Y})]}$$

Examples

1. The quadratic penalty $h(x) = \frac{1}{2}\sum_\alpha x_\alpha^2 = \frac{1}{2}\|x\|_2^2$ gives the *Euclidean projection*

$$\Pi(y) = \arg\max_{x\in\mathcal{X}}\{\langle y|x\rangle - \tfrac{1}{2}\|x\|_2^2\} = \arg\min_{x\in\mathcal{X}}\|y-x\|_2^2$$

2. If $\mathcal{X} = \Delta_d$, the (negative) *Gibbs entropy* $h(x) = \sum_\alpha x_\alpha \log x_\alpha$ gives the *logit map*

$$G(y) = \frac{(\exp(y_1), \dots, \exp(y_d))}{\sum_{\alpha=1}^d \exp(y_\alpha)}$$

3. If $\mathcal{X} = \{\mathbf{X} : \mathbf{X} \succcurlyeq 0, \operatorname{tr}(\mathbf{X}) \leq 1\}$, the *von Neumann entropy* $h(\mathbf{X}) = \operatorname{tr}[\mathbf{X}\log\mathbf{X}]$ gives

$$Q(\mathbf{Y}) = \frac{\exp(\mathbf{Y})}{1 + \operatorname{tr}[\exp(\mathbf{Y})]}$$

etc.

Important: if $\|dh(x)\| \to \infty$ as $x \to \operatorname{bd}(\mathcal{X})$, we say that h is *steep*.
Steep penalty functions induce interior point “projections”: $\operatorname{im} Q = \operatorname{rel\,int}\mathcal{X}$.

Learning via Mirror Descent

Formally, we get the multi-agent version of *mirror descent*:

$$\begin{aligned} y_k(n+1) &= y_k(n) + \gamma_n \, v_k(x(n)) \\ x_k(n+1) &= Q_k(y_k(n+1)) \end{aligned} \tag{MD}$$

where the *choice map* Q_k is defined as:

$$Q_k(y_k) = \arg\max_{x_k \in \mathcal{X}_k} \left\{ \langle y_k | x_k \rangle - h_k(x_k) \right\}$$

- Long history in optimization (Nemirovski, Yudin, Nesterov, Juditski, Beck, Teboulle, ...) and, more recently, in machine learning (Warmuth, Littlestone, Shalev-Shwartz,...)
- Well-understood for single-agent problems

Learning via Mirror Descent

Formally, we get the multi-agent version of *mirror descent*:

$$
\begin{aligned}
y_k(n+1) &= y_k(n) + \gamma_n \, v_k(x(n)) \\
x_k(n+1) &= Q_k(y_k(n+1))
\end{aligned}
\tag{MD}
$$

where the *choice map* Q_k is defined as:

$$
Q_k(y_k) = \underset{x_k \in \mathcal{X}_k}{\arg\max} \left\{ \langle y_k | x_k \rangle - h_k(x_k) \right\}
$$

- Long history in optimization (Nemirovski, Yudin, Nesterov, Juditski, Beck, Teboulle, ...) and, more recently, in machine learning (Warmuth, Littlestone, Shalev-Shwartz,...)
- Well-understood for single-agent problems
- **Multi-agent** problems (games): ???

Variational Stability

No uncoupled dynamics can always lead to equilibrium $\implies$ must refine

Definition

We say that $x^* \in \mathcal{X}$ is *variationally stable* if

$$\sum_k \lambda_k \left\langle v_k(x) \middle| x_k - x_k^* \right\rangle < 0$$

for some $\lambda \succ 0$ and for all x in a neighborhood U of x^* in $\mathcal{X}$. If $U = \mathcal{X}$, we will say that x^* is *globally variationally stable*.

Variational Stability

No uncoupled dynamics can always lead to equilibrium $\implies$ must refine

Definition

We say that $x^* \in \mathcal{X}$ is *variationally stable* if

$$\sum_k \lambda_k \left\langle v_k(x) \middle| x_k - x_k^* \right\rangle < 0$$

for some $\lambda \succ 0$ and for all x in a neighborhood U of x^* in $\mathcal{X}$. If $U = \mathcal{X}$, we will say that x^* is *globally variationally stable*.

- Contrast with Nash equilibrium (which it refines):

$$\sum_k \lambda_k \left\langle v_k(x_k; x_{-k}^*) \middle| x_k - x_k^* \right\rangle < 0$$

- Strict equilibria of finite games are variationally stable.
- Compare with notion of (Taylor) evolutionary stability in multi-population games:

$$\sum_k \left\langle v_k(x) \middle| x_k - x_k^* \right\rangle < 0 \quad \text{for all } x \neq x^* \text{ near } x^*$$

- Global/local ESSs are globally/locally variationally stable.
- Rosen's condition implies global variational stability.

Local convergence

Proposition

Suppose that (MD) is run with steep penalty functions and a small enough step-size sequence γ_n such that $\sum_{j=1}^{\infty} \gamma_j = \infty$ and $\sum_{j=1}^{n} \gamma_j^2 \Big/ \sum_{j=1}^{n} \gamma_j \to 0$. If $x(n) \to x^*$, then x^* is a Nash equilibrium of the game.

Local convergence

Proposition

Suppose that (MD) is run with steep penalty functions and a small enough step-size sequence γ_n such that $\sum_{j=1}^{\infty} \gamma_j = \infty$ and $\sum_{j=1}^{n} \gamma_j^2 \big/ \sum_{j=1}^{n} \gamma_j \to 0$. If $x(n) \to x^*$, then x^* is a Nash equilibrium of the game.

Theorem

Suppose that $x^* \in \mathcal{X}$ is variationally stable and (MD) is run with the same conditions as above. Then, x^* is locally attracting.

Corollary

In finite games, strict equilibria are locally attracting.

Global convergence

Theorem

Suppose that $x^* \in \mathcal{X}$ is globally variationally stable and the algorithm's step-size sequence γ_j satisfies $\sum_{j=1}^{\infty} \gamma_j = \infty$ and $\sum_{j=1}^{n} \gamma_j^2 \Big/ \sum_{j=1}^{n} \gamma_j \to 0$.
Then, $x(n) \to x^*$ for every initialization of the players' learning algorithm.

Corollary

If Rosen's condition holds, players converge to equilibrium from any initial condition.

Global convergence

Theorem

Suppose that $x^* \in \mathcal{X}$ is globally variationally stable and the algorithm's step-size sequence γ_j satisfies $\sum_{j=1}^{\infty} \gamma_j = \infty$ and $\sum_{j=1}^{n} \gamma_j^2 \Big/ \sum_{j=1}^{n} \gamma_j \to 0$.
Then, $x(n) \to x^*$ for every initialization of the players' learning algorithm.

Corollary

If Rosen's condition holds, players converge to equilibrium from any initial condition.

Proof idea.

- $x(n)$ is an asymptotic pseudo-trajectory of the continuous-time dynamics

$$\dot{y} = v(Q(y))$$

- A (global) Lyapunov function is given by the (λ-weighted) *Fenchel coupling*

$$F(y) = \sum_k \lambda_k \left[h_k(x_k^*) + h_k^*(y_k) - \langle y_k | x_k^* \rangle \right]$$

- Standard stochastic approximation results do not suffice for convergence of $x(n)$.
- Show that $x(n)$ visits a neighborhood of x^* infinitely often directly.
- Use steepness and Benaïm's theory of attractors on the flow induced by Q on $\mathcal{X}$. □

Learning with Imperfect Information

The above analysis relies on perfect observations of the payoff gradients $v_k(x)$.

- In finite games, it is not too hard to deduce $u_k(\alpha_k; \alpha_{-k})$ for every action $\alpha_k \in \mathcal{A}_k$ given a fixed action $\alpha_{-k} \in \mathcal{A}_{-k}$ of his opponents.
- However, knowing $u_k(x_k; x_{-k})$ is much more demanding.

Imperfect information: players only have access to noisy estimates of their payoff gradients, i.e.

$$\hat{v}_k(n) = v_k(x(n)) + z_k(n)$$

Learning with Imperfect Information

The above analysis relies on perfect observations of the payoff gradients $v_k(x)$.

- In finite games, it is not too hard to deduce $u_k(\alpha_k; \alpha_{-k})$ for every action $\alpha_k \in \mathcal{A}_k$ given a fixed action $\alpha_{-k} \in \mathcal{A}_{-k}$ of his opponents.
- However, knowing $u_k(x_k; x_{-k})$ is much more demanding.

Imperfect information: players only have access to noisy estimates of their payoff gradients, i.e.

$$\hat{v}_k(n) = v_k(x(n)) + z_k(n)$$

Statistical hypotheses for the noise process $z_k(n)$:

(H1) Unbiasedness: $\mathbb{E}[z(n+1) \mid \mathcal{F}_n] = 0.$

(H2) Finite mean squared error: $\mathbb{E}\left[\|z(n+1)\|^2 \mid \mathcal{F}_n\right] < \infty.$

(H2+) Finite errors: $\sup_n \|z(n)\| < \infty$ (a.s.).

Convergence Analysis

Run (MD) with steep penalty functions and a small enough step-size sequence γ_n such that $\sum_{n=1}^{\infty} \gamma_n^2 < \sum_{n=1}^{\infty} \gamma_n = \infty$.

Convergence Analysis

Run (MD) with steep penalty functions and a small enough step-size sequence γ_n such that $\sum_{n=1}^{\infty} \gamma_n^2 < \sum_{n=1}^{\infty} \gamma_n = \infty$.

Theorem

Suppose that x^* is globally variationally stable and (H1), (H2) hold. Then, with probability 1, $x(n)$ converges to x^* for every initialization of (MD).

Convergence Analysis

Run (MD) with steep penalty functions and a small enough step-size sequence γ_n such that $\sum_{n=1}^{\infty} \gamma_n^2 < \sum_{n=1}^{\infty} \gamma_n = \infty$.

Theorem

Suppose that x^* is globally variationally stable and (H1), (H2) *hold. Then, with probability 1, $x(n)$ converges to x^* for every initialization of* (MD).

Theorem

Suppose that x^* is variationally stable and (H1), (H2) *hold. Then, for every $\varepsilon > 0$, there exists a neighborhood U of x^* such that*

$$\mathbb{P}(\lim_{n\to\infty} x(n) = x^* \mid x(0) \in U) \geq 1 - \varepsilon,$$

i.e. x^* attracts all nearby initializations of (MD) *with high probability. If* (H2+) *also holds, the above holds for $\varepsilon = 0$ as well.*

Applications to finite games

Suppose that players play repeatedly a finite game $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{A}, u)$:

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. Players estimate (noisily) the payoff of each of their actions:

$$\hat{v}_{k\alpha}(n+1) = u_k(\alpha; \alpha_{-k}(n+1)) + z_{k\alpha}(n+1) \quad \alpha \in \mathcal{A}_k$$

3. Players update their mixed strategies using (MD) and the process repeats.

Applications to finite games

Suppose that players play repeatedly a finite game $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{A}, u)$:

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. Players estimate (noisily) the payoff of each of their actions:

$$\hat{v}_{k\alpha}(n+1) = u_k(\alpha; \alpha_{-k}(n+1)) + z_{k\alpha}(n+1) \quad \alpha \in \mathcal{A}_k$$

3. Players update their mixed strategies using (MD) and the process repeats.

Corollary

With assumptions as before, strict equilibria are locally attracting with high probability.

Perspectives

- Nonsteep penalty functions (e.g. Euclidean projections)?
  - Continuous-time analysis works fine.
  - Stochastic approximation harder.
  - Direct analysis easier.

Perspectives

- Nonsteep penalty functions (e.g. Euclidean projections)?
  - Continuous-time analysis works fine.
  - Stochastic approximation harder.
  - Direct analysis easier.

- Rates of convergence?
  - Doable for empirical frequencies of play.
  - Less so for last iterate.

Perspectives

- Nonsteep penalty functions (e.g. Euclidean projections)?
  - Continuous-time analysis works fine.
  - Stochastic approximation harder.
  - Direct analysis easier.
- Rates of convergence?
  - Doable for empirical frequencies of play.
  - Less so for last iterate.
- Constraints on the players' action sets?
  - Seems possible, but lose some distributedness (because of constraint coupling).

Perspectives

- Nonsteep penalty functions (e.g. Euclidean projections)?
  - Continuous-time analysis works fine.
  - Stochastic approximation harder.
  - Direct analysis easier.
- Rates of convergence?
  - Doable for empirical frequencies of play.
  - Less so for last iterate.
- Constraints on the players' action sets?
  - Seems possible, but lose some distributedness (because of constraint coupling).
- Observations of realized payoffs only?
  - Standard estimators often fail because of infinite variance.
  - Not a problem in online learning – maybe in multi-agent case as well?
  - Two-time-scale stochastic approximation can help (at the cost of convergence speed).

Perspectives

- Nonsteep penalty functions (e.g. Euclidean projections)?
  - Continuous-time analysis works fine.
  - Stochastic approximation harder.
  - Direct analysis easier.
- Rates of convergence?
  - Doable for empirical frequencies of play.
  - Less so for last iterate.
- Constraints on the players' action sets?
  - Seems possible, but lose some distributedness (because of constraint coupling).
- Observations of realized payoffs only?
  - Standard estimators often fail because of infinite variance.
  - Not a problem in online learning – maybe in multi-agent case as well?
  - Two-time-scale stochastic approximation can help (at the cost of convergence speed).
- ...