NO-REGRET LEARNING IN GAMES

Panayotis Mertikopoulos[1]

[1]French National Center for Scientific Research (CNRS)

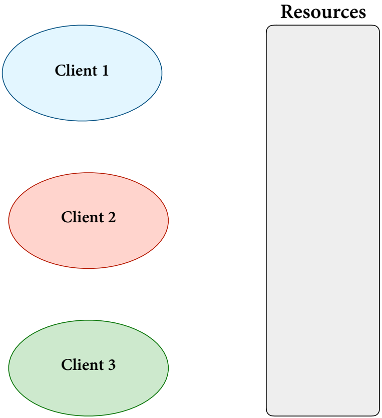
Laboratoire d'Informatique de Grenoble (LIG)

NPCG '19 – Paris, April 16, 2019

Outline

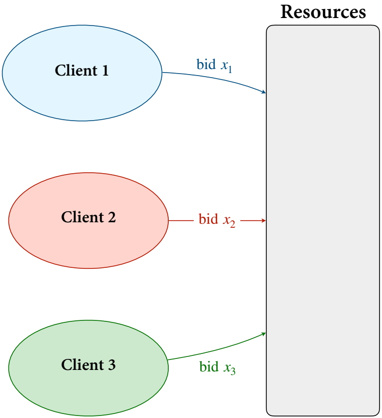
The Kelly auction

Proportionally fair allocation of resources to different clients [Kelly, 1998]:



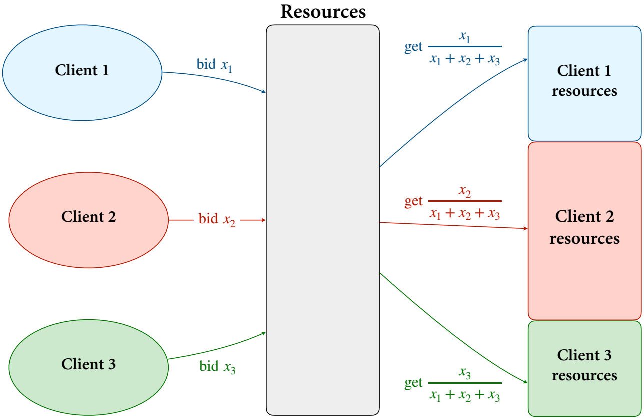
The Kelly auction

Proportionally fair allocation of resources to different clients [Kelly, 1998]:



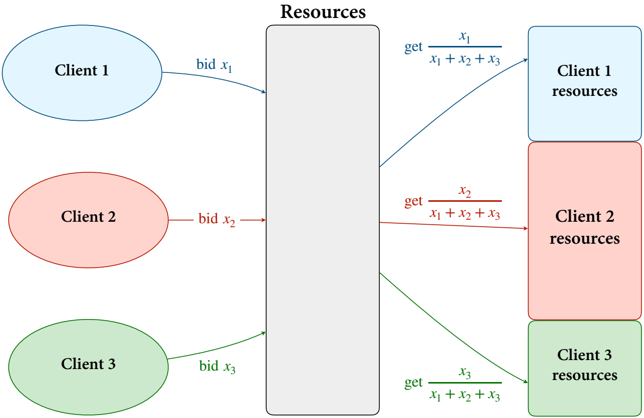
The Kelly auction

Proportionally fair allocation of resources to different clients [Kelly, 1998]:



The Kelly auction

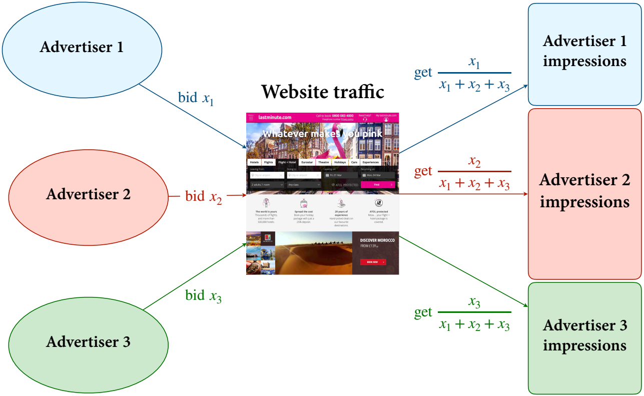
Proportionally fair allocation of resources to different clients [Kelly, 1998]:



Resources could be processor cores, bandwidth

The Kelly auction

Proportionally fair allocation of resources to different clients [Kelly, 1998]:



Resources could be processor cores, bandwidth , or even **anonymous web traffic**

Online decision processes

Agents called to take repeated decisions with minimal information:

---

repeat

At each epoch $t = 1, 2, \ldots$

Choose action X_t

Get payoff $u_t(X_t)$

until end

Online decision processes

Agents called to take repeated decisions with minimal information:

---

repeat

At each epoch $t = 1, 2, \ldots$

  Choose **action** X_t

  Get **payoff** $u_t(X_t)$

until end

Main question: *How to choose a "good" action at each epoch?*

- Uncertain world: no beliefs, feedback, knowledge of future, etc.
- Obliviousness: are payoffs affected by the agent's previous actions?
- Optimality: what is "optimal" in this setting?

Regret minimization

Performance often quantified by the agent's **regret**

$$u_t(x) - u_t(X_t)$$

Regret minimization

Performance often quantified by the agent's **regret**

$$\sum_{t=1}^{T} [u_t(x) - u_t(X_t)]$$

Regret minimization

Performance often quantified by the agent's **regret**

$$\max_{x \in \mathcal{X}} \sum_{t=1}^{T} [u_t(x) - u_t(X_t)]$$

Regret minimization

Performance often quantified by the agent's **regret**

$$\mathrm{Reg}(T) = \max_{x \in \mathcal{X}} \sum_{t=1}^{T} [u_t(x) - u_t(X_t)]$$

Regret minimization

Performance often quantified by the agent's **regret**

$$\mathrm{Reg}(T) = \max_{x \in \mathcal{X}} \sum_{t=1}^{T} [u_t(x) - u_t(X_t)]$$

No regret: $\mathrm{Reg}(T) = o(T)$

"The sequence of chosen actions is as good as the best fixed action in hindsight."

Regret minimization

Performance often quantified by the agent's **regret**

$$\mathrm{Reg}(T) = \max_{x \in \mathcal{X}} \sum_{t=1}^{T} [u_t(x) - u_t(X_t)]$$

No regret: $\mathrm{Reg}(T) = o(T)$

"The sequence of chosen actions is as good as the best fixed action in hindsight."

Prolific literature:

- Economics [Hannan, Blackwell, Hart & Mas-Colell,...]
- Machine learning & computer science [Littlestone & Warmuth, Vovk,...]
- Online learning & optimization [Cesa-Bianchi & Lugosi, Zinkevich,...]

Multi-agent learning

- **Multiple** agents, individual objectives
- Payoffs determined by actions of **all** agents
- Agents receive payoffs, **adjust actions**, and the process repeats

Multi-agent learning

- **Multiple** agents, individual objectives

  Example: *place a bid in a repeated auction*

- Payoffs determined by actions of **all** agents

  Example: *outcome of auction revealed*

- Agents receive payoffs, **adjust actions**, and the process repeats

  Example: *change bid if unsatisfied*

No-regret and equilibrium

The golden rule:

No-regret learning leads to equilibrium

No-regret and equilibrium

The golden rule:

No-regret learning leads to equilibrium*

*If it's ok to:

No-regret and equilibrium

The golden rule:

No-regret learning leads to equilibrium*

*If it's ok to:

✗ Assign positive weight only to strictly dominated strategies

[Viossat & Zapechelnyuk, 2013]

No-regret and equilibrium

The golden rule:

No-regret learning leads to equilibrium*

*If it's ok to:

✗ Assign positive weight only to strictly dominated strategies

[Viossat & Zapechelnyuk, 2013]

✗ Be arbitrarily far from equilibrium infinitely often

[too many to list]

No-regret and equilibrium

The golden rule:

No-regret learning leads to equilibrium*

*If it's ok to:

- ✗ Assign positive weight only to strictly dominated strategies

  [Viossat & Zapechelnyuk, 2013]

- ✗ Be arbitrarily far from equilibrium infinitely often

  [too many to list]

- ✗ …

No-regret and equilibrium

When does no-regret learning converge to Nash equilibrium?

Outline

N-player games

The game

- Finite set of *players* $i \in \mathcal{N} = \{1, \dots, N\}$
- Each player selects an *action* x_i from a compact convex set $\mathcal{X}_i$
- Reward of player i determined by *payoff function* $u_i \colon \mathcal{X} \equiv \prod_i \mathcal{X}_i \to \mathbb{R}$

Examples

- Finite games (mixed extensions)
- Power control/allocation problems
- Traffic routing
- Generative adversarial networks (two-player zero-sum games)
- Divisible good auctions (Kelly,…)
- Cournot oligopolies
- …

Kelly auctions

The Kelly auction as an N-player game:

- **Players:** $i = 1, \dots, N$ [bidders]
- **Resources** $\mathcal{S} = \{1, \dots, S\}$ [websites]
- **Action spaces:** $\mathcal{X}_i = \{x_i \in \mathbb{R}_+^S : \sum_s x_{is} \leq b_i\}$ [b_i: budget of i-th bidder]
- **Resource allocation ratio:**

$$\rho_{is}(x) = \frac{q_s x_{is}}{c_{is} + \sum_{j \in \mathcal{N}} x_{js}}$$

[c_{is}: entry barrier]

- **Payoff functions:**

$$u_i(x) = \sum_{s \in \mathcal{S}} [g_i \rho_{is}(x) - x_i]$$

[utility from resources minus cost]

Nash equilibrium

Nash equilibrium

Action profile $x^* = (x_1^*, \dots, x_t^*) \in \mathcal{X}$ that is **unilaterally stable**

$$u_i(x_i^*; x_{-i}^*) \geq u_i(x_i; x_{-i}^*) \quad \text{for every player } i \in \mathcal{N} \text{ and every deviation } x_i \in \mathcal{X}_i$$

Nash equilibrium

Nash equilibrium

Action profile $x^* = (x_1^*, \dots, x_t^*) \in \mathcal{X}$ that is **unilaterally stable**

$$u_i(x_i^*; x_{-i}^*) \geq u_i(x_i; x_{-i}^*) \quad \text{for every player } i \in \mathcal{N} \text{ and every deviation } x_i \in \mathcal{X}_i$$

Individual payoff gradients

$$V_i(x) = \nabla_{x_i} u_i(x_i; x_{-i})$$

Interpretation: direction of individually steepest payoff ascent

Nash equilibrium

Nash equilibrium
Action profile $x^* = (x_1^*, \ldots, x_t^*) \in \mathcal{X}$ that is unilaterally stable

$$u_i(x_i^*; x_{-i}^*) \geq u_i(x_i; x_{-i}^*) \quad \text{for every player } i \in \mathcal{N} \text{ and every deviation } x_i \in \mathcal{X}_i$$

Individual payoff gradients

$$V_i(x) = \nabla_{x_i} u_i(x_i; x_{-i})$$

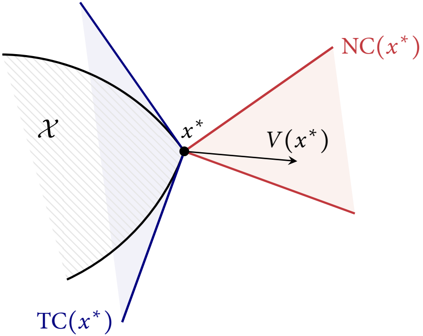
Interpretation: direction of individually steepest payoff ascent

Variational characterization
If x^* is a Nash equilibrium, then

$$\langle V_i(x^*), x_i - x_i^* \rangle \leq 0 \quad \text{for all } i \in \mathcal{N}, x_i \in \mathcal{X}_i$$

Intuition: $u_i(x_i; x_{-i}^*)$ decreasing along all rays emanating from x_i^*

Geometric interpretation



At Nash equilibrium, individual payoff gradients are outward-pointing

Monotonicity

A key assumption for games is **monotonicity:**

$$\langle V(x') - V(x), x' - x \rangle \leq 0 \quad \text{for all } x \in \mathcal{X} \tag{MC}$$

Monotonicity

A key assumption for games is **monotonicity:**

$$\langle V(x') - V(x), x' - x \rangle \leq 0 \quad \text{for all } x \in \mathcal{X} \tag{MC}$$

Equivalently: $H(x) \preccurlyeq 0$ where H is the game's **Hessian matrix:**

$$H_{ij}(x) = \frac{1}{2}\nabla_{x_j}\nabla_{x_j}u_i(x) + \frac{1}{2}(\nabla_{x_i}\nabla_{x_j}u_j(x))^\top$$

Interpretation: concavity for games

Monotonicity

A key assumption for games is **monotonicity:**

$$\langle V(x') - V(x), x' - x \rangle \leq 0 \quad \text{for all } x \in \mathcal{X} \tag{MC}$$

Equivalently: $H(x) \preccurlyeq 0$ where H is the game's **Hessian matrix:**

$$H_{ij}(x) = \frac{1}{2} \nabla_{x_j} \nabla_{x_j} u_i(x) + \frac{1}{2} (\nabla_{x_i} \nabla_{x_j} u_j(x))^\top$$

Interpretation: concavity for games

Examples: Kelly auctions, Cournot oligopolies, routing, power control, …

Close relatives:

- Stable games [Hofbauer & Sandholm, 2009]
- Contractive games [Sandholm, 2015];
- Dissipative [Sorin & Wan, 2016]

Monotonicity

Theorem (Rosen, 1965)

If a game is strictly monotone, it admits a **unique Nash equilibrium.**

[+ extensions to {…}-monotone games, generalized equilibrium problems,…]

Outline

Bacgkround

N-player games

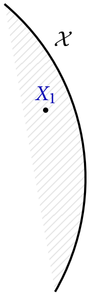
No-regret learning in games

Learning with limited feedback

How to achieve no regret?

Take a gradient step and project: [Zinkevich, ICML 2003]

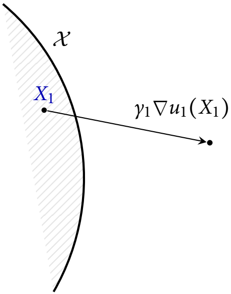
$$X_{t+1} = \Pi(X_t + \gamma_t \nabla u_t(X_t)) \tag{OGD}$$



How to achieve no regret?

Take a gradient step and project: [Zinkevich, ICML 2003]

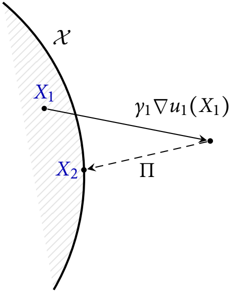
$$X_{t+1} = \Pi(X_t + \gamma_t \nabla u_t(X_t)) \qquad \text{(OGD)}$$



How to achieve no regret?

Take a gradient step and project: [Zinkevich, ICML 2003]

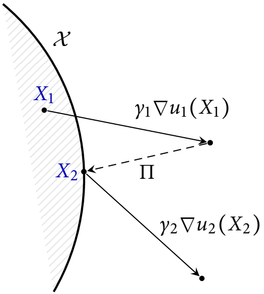
$$X_{t+1} = \Pi(X_t + \gamma_t \nabla u_t(X_t)) \qquad \text{(OGD)}$$



How to achieve no regret?

Take a gradient step and project: [Zinkevich, ICML 2003]

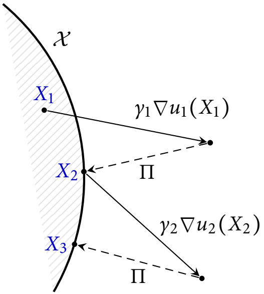
$$X_{t+1} = \Pi(X_t + \gamma_t \nabla u_t(X_t)) \tag{OGD}$$



How to achieve no regret?

Take a gradient step and project: [Zinkevich, ICML 2003]

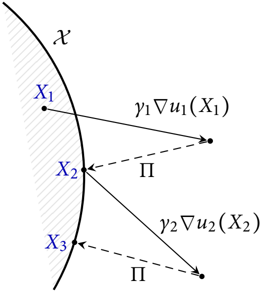
$$X_{t+1} = \Pi(X_t + \gamma_t \nabla u_t(X_t)) \tag{OGD}$$



How to achieve no regret?

Take a gradient step and project: [Zinkevich, ICML 2003]

$$X_{t+1} = \Pi(X_t + \gamma_t \nabla u_t(X_t)) \tag{OGD}$$

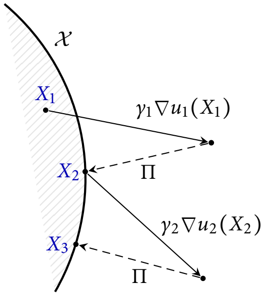


$\mathrm{Reg}(T) = \mathcal{O}(T^{1/2})$ for suitable γ_t; optimal in T [Abernethy et al, 2008]

How to achieve no regret?

Take a gradient step and project: [Zinkevich, ICML 2003]

$$X_{t+1} = \Pi(X_t + \gamma_t \nabla u_t(X_t)) \quad \text{(OGD)}$$



$\mathrm{Reg}(T) = \mathcal{O}(T^{1/2})$ for suitable γ_t; optimal in T [Abernethy et al, 2008]

...but what about convergence?

A dynamical systems viewpoint

Vector flow of V (simplest case: no constraints, smooth, etc.):

$$\frac{dX_i}{dt} = -V_i(X(t)) \tag{GD}$$

Energy function:

$$E(x) = \frac{1}{2}\|x - x^*\|^2$$

Lyapunov property:

$$\frac{dE}{dt} = -\langle V(X(t)), X(t) - x^* \rangle \leq 0$$

Distance to solutions is (weakly) decreasing along trajectories of (GD)

Cycles

Roadblock: the energy might be a constant of motion [Hofbauer et al, 2009]

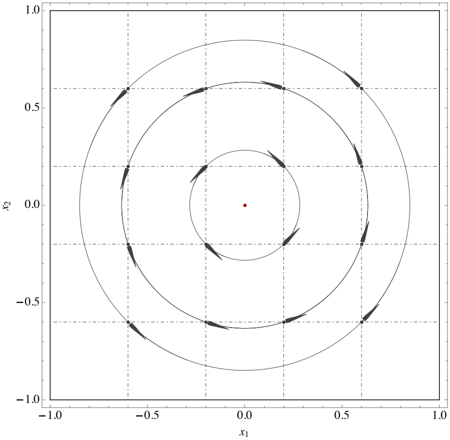


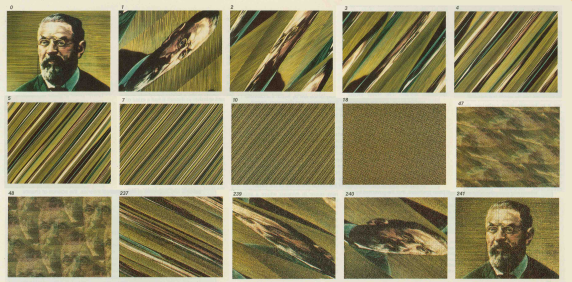
Figure: Hamiltonian flow of $f(x_1, x_2) = x_1 x_2$.

Poincaré recurrence

Cycles are an example of **recurrence:**

Definition (Poincaré, 1890's)

A dynamical system is *Poincaré recurrent* if almost all solution trajectories return *arbitrarily close* to their starting point *infinitely many times.*

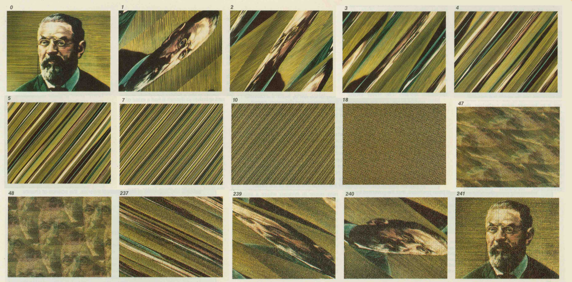


Poincaré recurrence

Cycles are an example of **recurrence:**

Definition (Poincaré, 1890's)

A dynamical system is *Poincaré recurrent* if almost all solution trajectories return *arbitrarily close* to their starting point *infinitely many times.*



Theorem (M, Papadimitriou, Piliouras, SODA 2018; bare-bones version)

(GD) *is recurrent in all bilinear saddle-point problems with an interior solution.*

OGD in games

OGD as a forward (Euler) scheme:

$$X^{+} = X - \gamma V(X)$$

OGD in games

OGD as a forward (Euler) scheme:

$$X^{+} = X - \gamma V(X)$$

Energy no longer a constant:

$$\frac{1}{2}\|X^{+} - x^{*}\|^{2} = \frac{1}{2}\|X - x^{*}\|^{2} - \gamma \underbrace{\langle V(X), X - x^{*}\rangle}_{\text{from (GD)}} + \frac{1}{2} \underbrace{\gamma^{2}\|V(X)\|^{2}}_{\text{discretization error}}$$

...even worse

OGD in games

OGD as a forward (Euler) scheme:

$$X^{+} = X - \gamma V(X)$$

Energy no longer a constant:

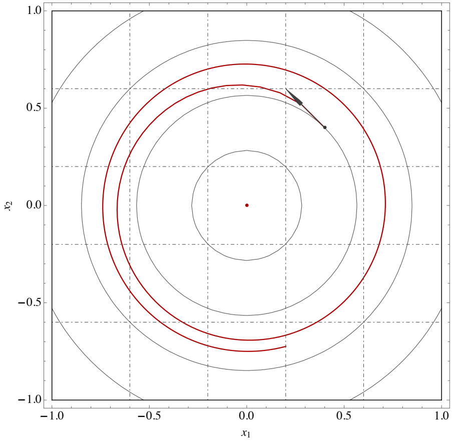
$$\frac{1}{2}\|X^{+} - x^{*}\|^{2} = \frac{1}{2}\|X - x^{*}\|^{2} - \gamma \underbrace{\cancel{\langle V(X), X - x^{*}\rangle}}_{\text{from (GD)}} + \frac{1}{2}\underbrace{\gamma^{2}\|V(X)\|^{2}}_{\text{discretization error}}$$

...even worse

OGD in games

OGD as a forward (Euler) scheme:

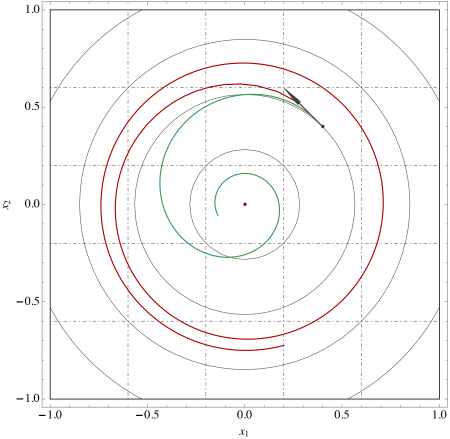
$$X_{t+1} = X_t - \gamma V(X_t)$$



Time averages: a very different story

No-regret captures behavior of time-averaged process:

$$\bar{X}_t = \frac{1}{t}\sum_{s=1}^{t} X_s$$



Convergence to equilibrium

Behavior different under **strict** monotonicity:

$$\frac{1}{2}\|X_{t+1}-x^*\|^2 = \frac{1}{2}\|X_t - x^*\|^2 - \gamma_t \underbrace{\langle V(X_t), X_t - x^*\rangle}_{<\,0 \text{ if } X_t \text{ not Nash}} + \frac{1}{2}\underbrace{\gamma_t^2 \|V(X_t)\|^2}_{\text{discretization error}}$$

Can the drift overcome the discretization error?

Convergence to equilibrium

Behavior different under **strict** monotonicity:

$$\frac{1}{2}\|X_{t+1} - x^*\|^2 = \frac{1}{2}\|X_t - x^*\|^2 - \gamma_t \underbrace{\langle V(X_t), X_t - x^* \rangle}_{<\, 0 \text{ if } X_t \text{ not Nash}} + \frac{1}{2} \underbrace{\gamma_t^2 \|V(X_t)\|^2}_{\text{discretization error}}$$

Can the drift overcome the discretization error?

Theorem (M & Zhou, MathProg 2019)

- **Assume:** *game strictly monotone,* $\sum_t \gamma_t = \infty$, $\sum_t \gamma_t^2 < \infty$
- **Then:** X_t *converges to a Nash equilibrium from any initial condition*

Convergence to equilibrium

Behavior different under **strict** monotonicity:

$$\frac{1}{2}\|X_{t+1}-x^*\|^2 = \frac{1}{2}\|X_t - x^*\|^2 - \gamma_t \underbrace{\langle V(X_t), X_t - x^*\rangle}_{<\,0 \text{ if } X_t \text{ not Nash}} + \frac{1}{2}\underbrace{\gamma_t^2 \|V(X_t)\|^2}_{\text{discretization error}}$$

Can the drift overcome the discretization error?

Theorem (M & Zhou, MathProg 2019)

- **Assume:** *game strictly monotone,* $\sum_t \gamma_t = \infty$, $\sum_t \gamma_t^2 < \infty$
- **Then:** X_t *converges to a Nash equilibrium from any initial condition*

In strictly monotone games, no-regret $\rightsquigarrow$ Nash equilibrium

Outline

Feedback

(OGD) requires gradient information, which may be difficult to come by:

- Other players' actions unknown
- Measurement errors
- Stochastic utilities (realized vs. expected gradients)
- …

Imperfect gradient feedback:

$$\hat{V}_t = V(x_t) + U_t$$

with the following hypotheses:

[H1] Zero-mean error: $\mathbb{E}[U_t \mid \mathcal{F}_{t-1}] = 0$ $\quad [\implies \mathbb{E}[\hat{V}_t \mid \mathcal{F}_{t-1}] = V(x_t)]$

[H2] Finite mean squared error: $\mathbb{E}[\|U_t\|_*^2 \mid \mathcal{F}_{t-1}] \leq \sigma^2$ $\quad [\implies \mathbb{E}[\|\hat{V}_t\|_*^2 \mid \mathcal{F}_{t-1}] \leq V^2]$

Learning with imperfect gradients

Algorithm 1 Stochastic gradient descent

Require: step-size sequence $\gamma_t > 0$

1: choose $X \in \mathcal{X}$ # initialization
2: **for** $t = 1, 2, \dots$ **do**
3: oracle query at state X returns V # gradient feedback
4: set $X \leftarrow \Pi(X + \gamma_t V)$ # new state
5: **end for**
6: **return** X

Learning with imperfect gradients

Algorithm 1 Stochastic gradient descent

Require: step-size sequence $\gamma_t > 0$

1: choose $X \in \mathcal{X}$ # initialization
2: **for** $t = 1, 2, \dots$ **do**
3: oracle query at state X returns V # gradient feedback
4: set $X \leftarrow \Pi(X + \gamma_t V)$ # new state
5: **end for**
6: **return** X

Guarantees:

- $\mathbb{E}[\mathrm{Reg}(T)] = \mathcal{O}(\sqrt{T})$ [folk]
- Strict monotonicity $\implies$ **X_t converges to Nash (a.s.)** [M & Zhou, 2019]

No gradient feedback whatsoever

In many cases, even stochastic gradients are out of reach:

- Multi-armed bandits (clinical trials, …)
- Other players' actions unknown (auctions, …)
- …

No gradient feedback whatsoever

In many cases, even stochastic gradients are out of reach:

- Multi-armed bandits (clinical trials, ...)
- Other players' actions unknown (auctions, ...)
- ...

Possible fixes:

- Two-time-scale approach: fast samples, slow updates [can be slow ☹]

No gradient feedback whatsoever

In many cases, even stochastic gradients are out of reach:

- Multi-armed bandits (clinical trials, …)
- Other players' actions unknown (auctions, …)
- …

Possible fixes:

- Two-time-scale approach: fast samples, slow updates [can be slow ☹]
- Multiple-point estimates [needs synchronization ☹]

No gradient feedback whatsoever

In many cases, even stochastic gradients are out of reach:

- Multi-armed bandits (clinical trials, …)
- Other players' actions unknown (auctions, …)
- …

Possible fixes:

- Two-time-scale approach: fast samples, slow updates [can be slow ☹]
- Multiple-point estimates [needs synchronization ☹]
- **Simultaneous perturbation stochastic approximation** [Spall, 1997]

Simultaneous perturbation stochastic approximation

Estimate $u'(x)$ at target point $x \in \mathbb{R}$

$$u'(x) \approx \frac{u(x+\delta) - u(x-\delta)}{2\delta}$$

Simultaneous perturbation stochastic approximation

Estimate $u'(x)$ at target point $x \in \mathbb{R}$

$$u'(x) \approx \frac{u(x+\delta) - u(x-\delta)}{2\delta}$$

Pick $z = \pm 1$ with probability $1/2$. Then:

$$\mathbb{E}[u(x+\delta z)z] = \frac{1}{2}u(x+\delta) - \frac{1}{2}u(x-\delta)$$

$\Longrightarrow$ Estimate $u'(x)$ up to $\mathcal{O}(\delta)$ by sampling u at $\hat{x} = x + \delta z$ and looking at $\frac{1}{\delta}u(\hat{x})z$

Simultaneous perturbation stochastic approximation

Estimate $u'(x)$ at target point $x \in \mathbb{R}$

$$u'(x) \approx \frac{u(x+\delta) - u(x-\delta)}{2\delta}$$

Pick $z = \pm 1$ with probability $1/2$. Then:

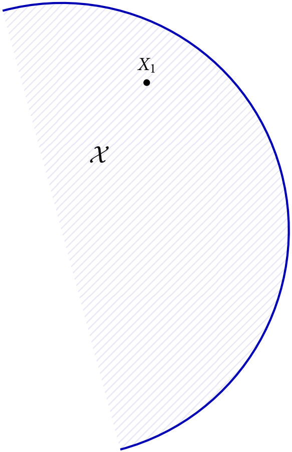
$$\mathbb{E}[u(x+\delta z)z] = \frac{1}{2}u(x+\delta) - \frac{1}{2}u(x-\delta)$$

$\implies$ Estimate $u'(x)$ up to $\mathcal{O}(\delta)$ by sampling u at $\hat{x} = x + \delta z$ and looking at $\frac{1}{\delta}u(\hat{x})z$

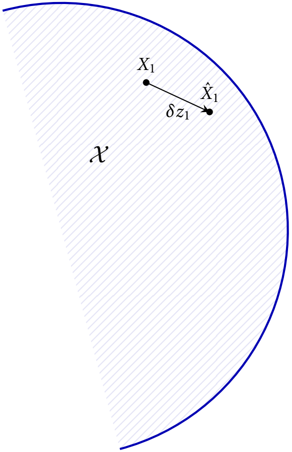
Algorithm 2 Single-point estimator of ∇u at X

1: Draw z uniformly from $\mathbb{S}^d$
2: Play $\hat{X} = X + \delta z$
3: Get $\hat{u} = u(\hat{X})$
4: Set $\hat{V} = \frac{d}{\delta}\hat{u}z$

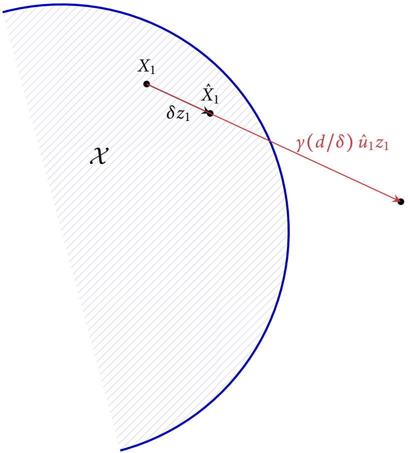
Learning with bandit feedback



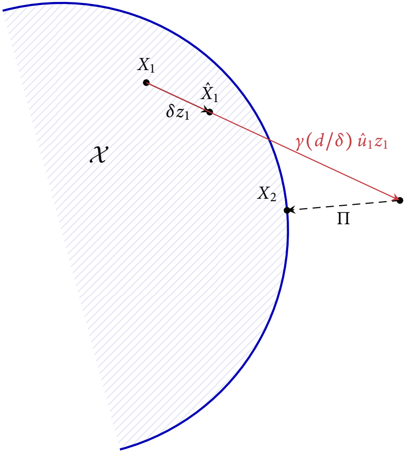
Learning with bandit feedback



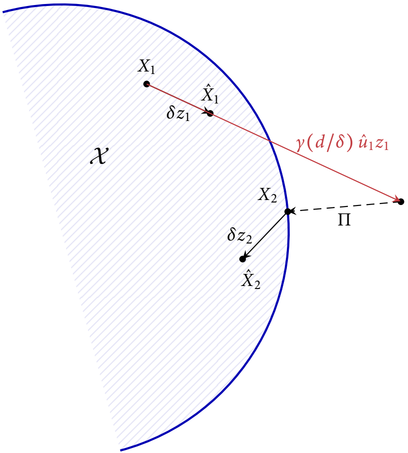
Learning with bandit feedback



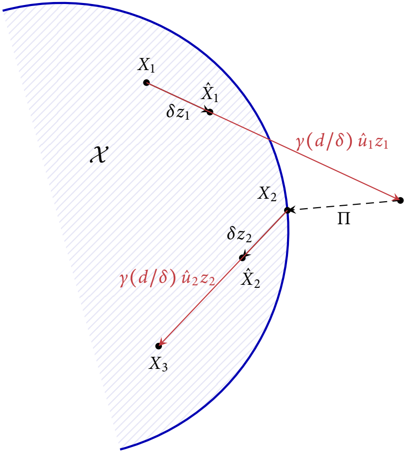
Learning with bandit feedback



Learning with bandit feedback



Learning with bandit feedback



Bandit gradient descent

Algorithm 3 Multi-agent gradient ascent with bandit feedback

Require: step-size $\gamma_t > 0$, query radius $\delta_t > 0$, safety ball $\mathbb{B}_r(p) \subseteq \mathcal{X}$

```
choose X ∈ 𝒳                                  # initialization
repeat at each stage t = 1, 2, ...
    draw Z uniformly from 𝕊^d                  # perturbation direction
    set W ← Z − r^{-1}(X − p)                  # feasibility adjustment
    play X̂ ← X + δ_t W                         # choose action
    receive û ← u(X̂)                           # get payoff
    set V̂ ← (d/δ_t) û · Z                      # estimate gradient
    update X ← Π(X + γ_t V̂)                    # update pivot
until end
```

Challenges

Key difficulty:

- One-point estimates may be biased (no more than $\mathcal{O}(\delta)$ accuracy)

Challenges

Key difficulty:

- One-point estimates may be biased (no more than $\mathcal{O}(\delta)$ accuracy)
- Can eliminate bias by taking decreasing $\delta_t \to 0$

Challenges

Key difficulty:

- One-point estimates may be biased (no more than $\mathcal{O}(\delta)$ accuracy)
- Can eliminate bias by taking decreasing $\delta_t \to 0$ but variance explodes

$$\mathbb{E}[\|\hat{V}_t\|^2] = \mathcal{O}(1/\delta_t^2)$$

Challenges

Key difficulty:

- One-point estimates may be biased (no more than $\mathcal{O}(\delta)$ accuracy)
- Can eliminate bias by taking decreasing $\delta_t \to 0$ but variance explodes

$$\mathbb{E}[\|\hat{V}_t\|^2] = \mathcal{O}(1/\delta_t^2)$$

- Stochastic approximation analysis requires bounded variance

Challenges

Key difficulty:

- One-point estimates may be biased (no more than $\mathcal{O}(\delta)$ accuracy)
- Can eliminate bias by taking decreasing $\delta_t \to 0$ but variance explodes

$$\mathbb{E}[\|\hat{V}_t\|^2] = \mathcal{O}(1/\delta_t^2)$$

- Stochastic approximation analysis requires bounded variance
- Bias-variance dilemma: accuracy vs. stability?

Convergence analysis

Must balance step-size γ_t against query radius δ_t:

- $\lim_{t\to\infty} \gamma_t = \lim_{t\to\infty} \delta_t = 0$ # vanishing noise and bias
- $\sum_{t=1}^{\infty} \gamma_t = \infty$ # the process doesn't stop
- $\sum_{t=1}^{\infty} \gamma_t^2/\delta_t^2 < \infty$ # variance control
- $\lim_{t\to\infty} \gamma_t \delta_t = 0$ # bias control

Convergence analysis

Must balance step-size γ_t against query radius δ_t:

- $\lim_{t\to\infty} \gamma_t = \lim_{t\to\infty} \delta_t = 0$ — # vanishing noise and bias
- $\sum_{t=1}^{\infty} \gamma_t = \infty$ — # the process doesn't stop
- $\sum_{t=1}^{\infty} \gamma_t^2/\delta_t^2 < \infty$ — # variance control
- $\lim_{t\to\infty} \gamma_t \delta_t = 0$ — # bias control

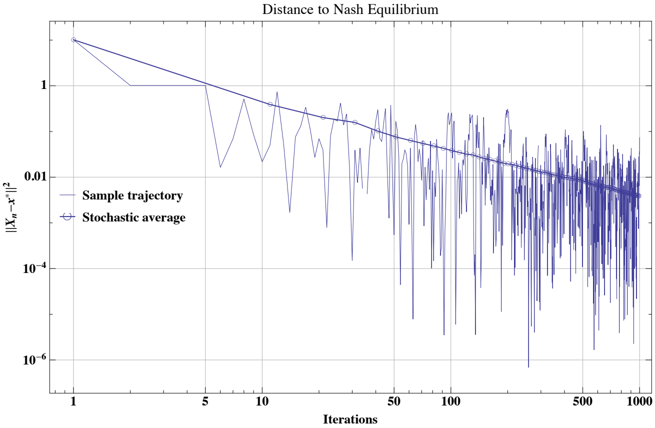
Theorem (Bravo, Leslie & M, NIPS 2018)

1. *Under* **strict monotonicity,** X_t *converges to Nash equilibrium with probability* 1.
2. *Under* **strong monotonicity** ($H(x) \prec -\beta I$), $\gamma_t \propto 1/t$, $\delta_t \propto 1/t^{1/3}$, *we have:*

$$\mathbb{E}[\|X_t - x^*\|^2] = \mathcal{O}(1/t^{1/3}).$$

Convergence rate

Speed of convergence in a repeated Kelly auction



Conclusions and perspectives

Conclusions

- No-regret learning does not guarantee stability by itself ✗
- No-regret learning plus suitable monotonicity does ✓
- Convergence to equilibrium does not require gradient feedback ✓

Conclusions and perspectives

Conclusions

- No-regret learning does not guarantee stability by itself ✗
- No-regret learning plus suitable monotonicity does ✓
- Convergence to equilibrium does not require gradient feedback ✓

Open questions

- Faster rates?
- Delayed payoff observations?
- Beyond monotonicity?
- ???

NetEcon 2019

The 14th Workshop on the Economics of Networks, Systems and Computation

Phoenix, Arizona, 28th June 2019

In conjunction with ACM EC 2019 & SIGMETRICS 2019

Keynote speakers: Itai Ashlagi *** David Parkes *** Nicolas Stier

Topics: Networks & … learning, resource pricing, market design, auctions

https://netecon19.inria.fr