Learning and Correlated Equilibria

Algorithmic Game Theory

Winter 2019/20

Imitating Experts

No-Regret Algorithms and Coarse-Correlated Equilibria

Zero-Sum Games

Concave Games

Expert Problem: Example

In many applications we make decisions not once but repeatedly, say, every day, without knowing the behavior of other actors or "nature" on that day. We consider learning algorithms that enable us to cope with such problems.

Example:

- Suppose you are commuting day by day from home to Campus Bockenheim and back.
- The traveling time per day is between 30 and 60 minutes depending on the chosen route and the traffic situation.
- Suppose you know, say, three **experts** that are also commuting from your area to campus and use different strategies for choosing the route.

We will show that you can become almost as fast as the best expert just by imitating the expert choices.

Expert Problem: Definition

Assume an adversarial online model with discrete time steps $1, \dots, T$. Let $[T]$ denote $\{1, \dots, T\}$.

Experts and Losses

- There are N *experts* numbered from 1 to N.
- In step $t \in [T]$, expert $i \in [N]$ experiences a *loss* of $\ell_i^t \in [0, 1]$ (as chosen by an adversary or "nature").
- Let $L_i^t = \sum_{k=1}^{t} \ell_i^k$.

Combining Experts

- In step t, an *online algorithm* H chooses expert $i \in [N]$ with probability p_i^t.
- The vector p^t might depend on the loss vectors $\ell^1, \dots, \ell^{t-1}$.
- The (expected) loss of H in step t is $\ell_H^t = \sum_{i \in [N]} p_i^t \ell_i^t$.
- Let $L_H^t = \sum_{k=1}^{t} \ell_H^k$.

Imitating Experts

No-Regret Algorithms and Coarse-Correlated Equilibria

Zero-Sum Games

Concave Games

Greedy Algorithm

In the following, let $L_{min}^{t-1} = \min_{i \in [N]} L_i^{t-1}$, for $1 \leq t \leq T$.

Greedy Algorithm

At every time t,

- let $S^{t-1} = \{i : L_i^{t-1} = L_{min}^{t-1}\}$;
- let $j = \min\{S^{t-1}\}$;
- set $p_j^t = 1$, and $p_i^t = 0$, for $i \neq j$.

In the analysis of the Greedy algorithm, we assume for simplicity that all losses are either 0 or 1 instead of real numbers from $[0, 1]$.

Greedy Algorithm

Example:

ℓ_1	1	0	0	1	0	0	1	0	0	1	0	0	1	0
L_1	1	1	1	2	2	2	3	3	3	4	4	4	5	5
ℓ_2	0	1	0	0	1	0	0	1	0	0	1	0	0	1
L_2	0	1	1	1	2	2	2	3	3	3	4	4	4	5
ℓ_3	0	0	1	0	0	1	0	0	1	0	0	1	0	0
L_3	0	0	1	1	1	2	2	2	3	3	3	4	4	4
j	1	2	3	1	2	3	1	2	3	1	2	3	1	2
ℓ_{Greedy}	1	1	1	1	1	1	1	1	1	1	1	1	1	1
L_G	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Greedy Algorithm

Theorem

The Greedy algorithm, for any sequence of losses from $\{0,1\}$*, has*

$$L_G^T \leq N \cdot L_{min}^T + (N-1).$$

Proof:

- Partition the sequence into phases $0, \ldots, L_{min}^T$ such that

  Every step t with $L_{min}^{t-1} = i$ belongs to phase i.

- In each phase $i < L_{min}^T$, the Greedy algorithm incurs a loss of at most N.
- In phase L_{min}^T, the loss of Greedy is at most $N-1$.

□

Lower Bound for Deterministic Algorithms

Theorem

For any deterministic online algorithm D and every $T \geq 1$, there exists a sequence of T losses such that

$$L_D^T = T \quad \textit{and} \quad L_{min}^T \leq \lfloor T/N \rfloor.$$

This lower bound can be shown quite easily by generalizing the example that we have given for the Greedy algorithm. (How?)

The lower bound shows that one cannot get better than the Greedy algorithm without using randomization.

Randomized Weighted Majority (RWM) Algorithm

Let $\eta \in (0, \frac{1}{2}]$ be a suitably chosen parameter.

Randomized Weighted Majority (RWM) Algorithm

Initially, set $w_i^1 = 1$, for every $i \in [N]$.
At every time t,

- let $W^t = \sum_{i=1}^{N} w_i^t$;
- choose expert i with probability $p_i^t = w_i^t / W^t$;
- set $w_i^{t+1} = w_i^t \cdot (1-\eta)^{\ell_i^t}$.

Randomized Weighted Majority (RWM) Algorithm

Theorem (Littlestone, Warmuth, 1994)

The RWM algorithm, for any sequence of losses from $[0,1]$*, has*

$$L_{RWM}^T \leq (1+\eta) L_{min}^T + \frac{\ln N}{\eta} \ .$$

Setting $\eta = \sqrt{\frac{\ln N}{T}}$ yields

$$L_{RWM}^T \leq L_{min}^T + 2\sqrt{T \ln N} \ .$$

Randomized Weighted Majority (RWM) Algorithm

The *regret* of a learning algorithm H is defined as $L_H^T - L_{min}^T$.

Corollary

The RWM algorithm with $\eta = \sqrt{\frac{\ln N}{T}}$ *has regret at most* $2\sqrt{T \ln N}$.

The *average regret per step* is thus only $2\sqrt{\frac{\ln N}{T}}$.

Observe that this quantity is going to zero when increasing T.

Algorithms with this property are called

no-regret learning algorithms.

Thus, in contrast to the simple greedy algorithms is RWM a no-regret algorithm.

Randomized Weighted Majority (RWM) Algorithm

Proof of the theorem:

- Let us analyze how the sum of weights W^t decreases over time. It holds

$$W^{t+1} = \sum_{i=1}^{N} w_i^{t+1} = \sum_{i=1}^{N} w_i^t (1-\eta)^{\ell_i^t} .$$

- Observe that $(1-\eta)^\ell = (1-\ell\eta)$, for both $\ell = 0$ and $\ell = 1$.
- Furthermore, $(1-\eta)^\ell$ is a convex function in ℓ.
- For $\ell \in [0,1]$ this implies $(1-\eta)^\ell \leq (1-\ell\eta)$.
- This gives

$$W^{t+1} \leq \sum_{i=1}^{N} w_i^t (1-\ell_i^t \eta) .$$

Randomized Weighted Majority (RWM) Algorithm

- Let F^t denote the expected loss of RWM in step t.
- It holds $F^t = \sum_{i=1}^{N} \ell_i^t w_i^t / W^t$.
- Substituting this into the bound for W^{t+1} gives

$$W^{t+1} \leq W^t - \eta F^t W^t = W^t(1 - \eta F^t) .$$

- As a consequence,

$$W^{T+1} \leq W^1 \prod_{t=1}^{T}(1 - \eta F^t) = N \prod_{t=1}^{T}(1 - \eta F^t) .$$

- The sum of weights after step T can be upper bounded in terms of the expected loss of RWM.

Randomized Weighted Majority (RWM) Algorithm

- On the other hand, the sum of weights after step T can be lower bounded in terms of the loss of the best expert as follows:

$$W^{T+1} \geq \max_{1 \leq i \leq N} (w_i^{T+1}) = \max_{1 \leq i \leq N} \left((1-\eta)^{\sum_{t=1}^{T} \ell_i^t} \right) = (1-\eta)^{L_{min}^T} .$$

- Combining the bounds and taking the logarithm on both sides gives

$$L_{min}^T \ln(1-\eta) \leq (\ln N) + \sum_{t=1}^{T} \ln(1-\eta F^T) .$$

- In order to simplify, we will now use the following estimation

$$-z - z^2 \leq \ln(1-z) \leq -z$$

holding for every $z \in [0, \frac{1}{2}]$.

Randomized Weighted Majority (RWM) Algorithm

- This gives

$$\begin{aligned} L_{min}^T(-\eta - \eta^2) &\leq (\ln N) + \sum_{t=1}^{T}(-\eta F^t) \\ &= (\ln N) - \eta L_{RWM}^T \ . \end{aligned}$$

- Finally, solving for L_{RWM}^T gives

$$L_{RWM}^T \leq (1+\eta)L_{min}^T + \frac{\ln N}{\eta} \ .$$

Learning Equilibria in Games

Regret minimization is a natural model for behavior in cases where we have to make repeated decisions with incomplete information.

We consider regret learning when a game $\Gamma = (\mathcal{N}, (\Sigma_i)_{i\in\mathcal{N}}, (c_i)_{i\in\mathcal{N}})$ is played over and over again for T rounds (called *repeated game*).

Initially, no player $i \in \mathcal{N}$ knows the game. In each round t he picks a pure strategy $s_i^t \in \Sigma_i$ using a no-regret algorithm. The algorithm of player i is based *only on the costs observed by i in previous rounds*.

Does the system converge to (approx.) Nash equilibrium in this case?

This would be a nice and plausible explanation how Nash equilibria can evolve in practice. Unfortunately, in general, the answer is "No".

Learning and Equilibria

For every player the average regret over time is going to 0. Based on this property, we can derive a (more general) equilibrium concept.

Definition

Let $\mathcal{V}$ be a probability distribution over the states of a finite game. $\mathcal{V}$ is called coarse-correlated equilibrium if for every player $i \in \mathcal{N}$ and every strategy $s_i' \in S_i$ it holds

$$\mathbb{E}_{s\sim\mathcal{V}}[c_i(s)] \quad \leq \quad \mathbb{E}_{s\sim\mathcal{V}}[c_i(s_i', s_{-i})] \ .$$

$\mathcal{V}$ is called (additive) ε-approximate coarse-correlated equilibrium if

$$\mathbb{E}_{s\sim\mathcal{V}}[c_i(s)] \quad \leq \quad \mathbb{E}_{s\sim\mathcal{V}}[c_i(s_i', s_{-i})] + \varepsilon \ .$$

No-Regret and Coarse-Correlated Equilibria

Consider the history of play $s^1, s^2, \ldots, s^T$ in a repeated game over T rounds. We interpret the history as a distribution over states by choosing $k \in [T]$ uniformly at random.

If player i has regret $R_i(T)$, then for every strategy $s_i' \in S_i$

$$\begin{aligned} \mathbb{E}_{k\in[T]}[c_i(s^k)] &= \sum_{t=1}^{T} \frac{1}{T} \cdot c_i(s^t) \leq \sum_{t=1}^{T} \frac{1}{T} \cdot c_i(s_i', s_{-i}^t) + \frac{R_i(T)}{T} \\ &= \mathbb{E}_{k\in[T]}[c_i(s_i', s_{-i}^k)] + \frac{R_i(T)}{T} \quad . \end{aligned}$$

Proposition

After T rounds if every player has regret at most R, then the history of play represents a $\frac{R}{T}$-approximate coarse-correlated equilibrium.

Suppose all players are using RWM, then after at most $T = \frac{4}{\varepsilon^2} \cdot \log(\max_i |S_i|)$ rounds the history of play represents a ε-approximate coarse-correlated equilibrium.

Imitating Experts

No-Regret Algorithms and Coarse-Correlated Equilibria

Zero-Sum Games

Concave Games

Recall 2-Player Zero-Sum Games

- A 2-player zero-sum game is a strategic game with 2 players, where $c_{\mathcal{I}}(s) + c_{\mathtt{II}}(s) = 0$ for every state s.
- Matrix A with $|\Sigma_{\mathtt{I}}|$ rows and $|\Sigma_{\mathtt{II}}|$ columns.
  Player I is row player, player II is column player.
- a_{ij} is utility for player I in state (i, j),
  a_{ij} is cost or loss for player II in state (i, j).
- We here normalize A to have $a_{ij} \in [0, 1]$:

  Make A non-negative by adding $\max |a_{ij}|$ to every entry. Then divide by the resulting largest entry scaling all a_{ij} to $[0, 1]$.

  Observe that this does not alter the optimal strategies (and thereby the Nash equilibria) of the game.

Examples

Matching Pennies (normalized)

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Rock-Paper-Scissors (normalized)

$$\begin{pmatrix} 1/2 & 0 & 1 \\ 1 & 1/2 & 0 \\ 0 & 1 & 1/2 \end{pmatrix}$$

A game with $|\Sigma_{\text{I}}| \neq |\Sigma_{\text{II}}|$:

$$\begin{pmatrix} 0 & 1/2 & 1 \\ 1/4 & 1/2 & 3/4 \end{pmatrix}$$

Maximin Strategies

- Gain-Floor for player I: $v_{\mathtt{I}}^* = \max_x \min_y x^T A y$.

  Optimal strategy x^* guarantees the gain-floor for I (maximin strategy).

- Loss-Ceiling for player II: $v_{\mathtt{II}}^* = \min_y \max_x x^T A y$.

  Optimal strategy y^* guarantees the loss-ceiling for II (minimax strategy).

Lemma

It holds that $v_{\mathtt{I}}^* \leq v_{\mathtt{II}}^*$.

Theorem (Minimax Theorem)

In every 2-player zero-sum game it holds that $v = v_{\mathtt{I}}^* = v_{\mathtt{II}}^*$.

Mixed Nash equilibrium

Corollary

State (x, y) in a 2-player zero-sum game is a mixed Nash equilibrium

$$\Leftrightarrow$$

x and y are optimal strategies.

Corollary

Every 2-player zero-sum game has at least one mixed Nash equilibrium. All mixed Nash equilibria in such a game yield the same expected utility for player `I`.

Theorem

In 2-player zero-sum games a mixed Nash equilibrium can be computed in polynomial time.

Learning in Zero-Sum Games

- Players do not know the game they are playing. They use no-regret learning algorithms to make their strategy choice.

  Can they learn to play optimally (i.e., learn a Nash equilibrium) ?

Learning in Zero-Sum Games

- Players do not know the game they are playing. They use no-regret learning algorithms to make their strategy choice.

  Can they learn to play optimally (i.e., learn a Nash equilibrium) ?

- Consider player II, experts are pure strategies, adversary is player I.
- In each step t learning algorithm H of player II picks mixed strategy y^t against an unknown adversary strategy x^t of player I.

Learning in Zero-Sum Games

- Players do not know the game they are playing. They use no-regret learning algorithms to make their strategy choice.

  Can they learn to play optimally (i.e., learn a Nash equilibrium) ?

- Consider player II, experts are pure strategies, adversary is player I.
- In each step t learning algorithm H of player II picks mixed strategy y^t against an unknown adversary strategy x^t of player I.
- Loss in round t for strategy (expert) i is

$$\ell_i^t = \sum_{j \in \Sigma_{\mathrm{I}}} x_j^t a_{ji} \ .$$

- Total loss in round t of learning algorithm H is

$$\ell_H^t = c_{\mathrm{II}}(x^t, y^t) = \sum_{i \in \Sigma_{\mathrm{II}}} \sum_{j \in \Sigma_{\mathrm{I}}} x_j^t a_{ji} y_i^t \ .$$

No-Regret and Optimal Strategies

- No-regret learning algorithm H:

$$\frac{L_H^T - L_{min}^T}{T} \longrightarrow 0 \qquad \text{when } T \to \infty \ .$$

No-Regret and Optimal Strategies

- No-regret learning algorithm H:

$$\frac{L_H^T - L_{min}^T}{T} \longrightarrow 0 \qquad \text{when } T \to \infty \ .$$

- By definition, the average loss per round of any no-regret learning algorithm becomes as small as the best average loss of any pure strategy in hindsight.
- Is the average loss L_H^T/T as small as the value of the game?

No-Regret and Optimal Strategies

- No-regret learning algorithm H:

$$\frac{L_H^T - L_{min}^T}{T} \longrightarrow 0 \qquad \text{when } T \to \infty \ .$$

- By definition, the average loss per round of any no-regret learning algorithm becomes as small as the best average loss of any pure strategy in hindsight.
- Is the average loss L_H^T/T as small as the value of the game?

Theorem

For a 2-player zero-sum game with gain floor $v_{\mathtt{I}}^*$, if player `II` *plays for T steps using algorithm H with regret R, then the average loss*

$$\frac{L_H^T}{T} \ \leq \ v_{\mathtt{I}}^* + \frac{R}{T} \ .$$

The result applies similarly for player `I` *and the loss-ceiling.*

Learning the Gain Floor

Proof:

- We will show that the best pure strategy in hindsight has total loss at most $L_{min}^T \leq T \cdot v_{\mathtt{I}}^*$.
- Consider the history of play of the adversary player $\mathtt{I}$, i.e., strategies $x^1, x^2, \ldots, x^T$, and combine them to an "average strategy"

$$\hat{x}_j = \frac{1}{T} \sum_{t=1}^{T} x_j^t \qquad \text{for all } j \in \Sigma_{\mathtt{I}}.$$

- Total loss L_i^T of a single strategy $i \in \Sigma_{\mathtt{II}}$ in hindsight is the same if player $\mathtt{I}$ had always played $\hat{x}$ in all time steps:

$$L_i^t = \sum_{t=1}^{T} \sum_{j \in \Sigma_{\mathtt{I}}} x_j^t \cdot a_{ji} = \sum_{j \in \Sigma_{\mathtt{I}}} \left(\sum_{t=1}^{T} x_j^t \right) \cdot a_{ji} = T \cdot \sum_{j \in \Sigma_{\mathtt{I}}} \hat{x}_j \cdot a_{ji} \ .$$

Learning the Gain Floor

- If we assume I plays always $\hat{x}$ and consider the best pure strategy for II in hindsight, then the scenario reduces to a one-step game.

- In this one-step game, player I first determines the average history of play $\hat{x}$ and then player II picks best pure strategy against $\hat{x}$ – i.e., I moves first, then II answers.
- By definition of gain floor, there is always $i \in \Sigma_{\text{II}}$ such that gain of I/loss of II is reduced to at most v_{I}^*, i.e., $c_{\text{II}}(\hat{x}, i) \leq v_{\text{I}}^*$.

- Hence, there is a pure strategy $i \in \Sigma_{\text{II}}$ such that

$$L_{min}^T \leq L_i^T \leq T \cdot v_{\text{I}}^* \ .$$

- Combining these insights:

$$L_H^T \ \leq \ L_{min}^T + R \ \leq \ T \cdot v_{\text{I}}^* + R \ .$$

A Simple Proof of the Minimax Theorem

Theorem (Minimax Theorem)

In every 2-player zero-sum game it holds that $v = v_{\mathtt{I}}^* = v_{\mathtt{II}}^*$.

Proof:

- For contradiction, assume $v_{\mathtt{I}}^* + \gamma = v_{\mathtt{II}}^*$ for some $\gamma > 0$.
- Let both players play the game iteratively for T steps with a learning algorithm that has regret $R/T < \gamma/3$.
- Using the average history of play as before we note that $L_{min}^T \leq v_{\mathtt{I}}^*$ for player II, and $L_{min}^T \leq -v_{\mathtt{II}}^*$ for player I ("$-$" because of loss).
- But this means the algorithms yield at most $v_{\mathtt{I}} + \gamma/3$ average cost for player II and at least $v_{\mathtt{II}} - \gamma/3$ average gain for player I.
- Average cost of II is average gain of I $\rightarrow$ Contradiction. □

Convergence

Corollary

If both players use a no-regret learning algorithm, the average histories of play $(\hat{x}, \hat{y})$ converge to optimal strategies and, thus, to a mixed Nash equilibrium of the game.

This shows convergence only for the history of play, but not for the actual behavior in the distributions x^t and y^t!

Theorem

There are no-regret algorithms for players `I` *and* `II` *such that the actual behavior x^t and y^t does not converge to optimal strategies.*

Convergence for General No-Regret Algorithms

Matching Pennies (normalized)

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Proof: A "weird" no-regret algorithm H:

- Rule: `I` and `II` pick pure strategies, `I` moves to the other pure strategy in rounds 1, 3, 5, ..., `II` moves to the other pure strategy in rounds 2, 4, 6,...
- If one player deviates from the rule, the other invokes the RWM algorithm. (trick to ensure no-regret property for every possible sequence of play).

- $L_i^T/T \to 0.5$ for both strategies $i = 1, 2$ of `II`, average loss of algorithm $L_H^T/T \to 0.5$. No-regret algorithm for `II`! (similar argument for `I`).
- None of the distributions y^t is close to optimal strategy $(0.5, 0.5)$, no single round loss ℓ_H^t is close to $v = 0.5$. □

An Adaptive RWM Algorithm (Freund, Schapire, 1999)

Let $\eta_0 \in (0, \frac{1}{2}]$ and u be an upper bound on the value of the game.

Variable Randomized Weighted Majority (vRWM) Algorithm

Initially, set $w_i^1 = 1$, for every $i \in [N]$.
At every time t,

- let $W^t = \sum_{i=1}^N w_i^t$;
- choose expert i with probability $p_i^t = w_i^t / W^t$;
- **if** $\ell_{vRWM}^t \leq u$ **then** set $w_i^{t+1} = w_i^t$;
- **else** set

$$\eta_t = 1 - \frac{u(1 - \ell_{vRWM}^t)}{(1-u)\ell_{vRWM}^t}$$

and $w_i^{t+1} = w_i^t \cdot (1 - \eta_t)^{\ell_i^t}$.

Comparison of Distributions

Definition

The relative entropy or Kullback-Leibler divergence of two distributions y and y' is defined as

$$RE(y \mid\mid y') = \sum_{i=1}^{n} y_i \cdot \ln\left(\frac{y_i}{y'_i}\right) \quad .$$

To compare $a, b \in [0,1]$ we use distributions $(a, 1-a)$ and $(b, 1-b)$:

$$\begin{aligned} RE(a \mid\mid b) &= RE((a,1-a) \mid\mid (b,1-b)) \\ &= a \ln\left(\frac{a}{b}\right) + (1-a)\ln\left(\frac{1-a}{1-b}\right) \quad . \end{aligned}$$

The relative entropy for distributions is always non-negative and $RE(y \mid\mid y') = 0$ if and only if $y = y'$.

Convergence of vRWM

Theorem

Let y' be any mixed strategy for `II` *which generates a loss of at most u against every best response of* `I`*. Then in any iteration t of vRWM in which $\ell^t_{vRWM} \geq u$ the relative entropy between y' and y^{t+1} satisfies*

$$RE(y' \mid\mid y^{t+1}) \leq RE(y' \mid\mid y^t) - RE(u \mid\mid \ell^t_{vRWM}) \ .$$

In every step, in which the loss of vRWM is too high, the adjustment moves the next distribution closer to a good strategy.

Proof of the Theorem

Proof:
For completeness, we provide a proof of the last theorem. Consider a step, in which $\ell^t_{vRWM} > u$ and bound

$$\begin{aligned} & RE(y' \mid\mid y^{t+1}) - RE(y' \mid\mid y^t) \\ = & \sum_{i \in \Sigma_{II}} y'_i \ln \frac{y'_i}{y_i^{t+1}} - \sum_{i \in \Sigma_{II}} y'_i \ln \frac{y'_i}{y_i^t} \\ = & \sum_{i \in \Sigma_{II}} y'_i \ln \frac{y_i^t}{y_i^{t+1}} \\ \leq & \sum_{i \in \Sigma_{II}} y'_i \ln \frac{1 - \eta_t \ell^t_{vRWM}}{(1-\eta_t)^{\ell_i^t}} \quad , \end{aligned}$$

where we use that $y_i^{t+1} = w_i^t (1-\eta_t)^{\ell_i^t} / W^{t+1}$ and $W^{t+1} \leq W^t(1 - \eta_t F^t) = W^t(1 - \eta_t \ell^t_{vRWM})$ as observed above.

Proof of the Theorem

$$
\begin{aligned}
& RE(y' \mid\mid y^{t+1}) - RE(y' \mid\mid y^t) \\
\leq & \sum_{i \in \Sigma_{\text{II}}} y'_i \ln \frac{1 - \eta_t \ell^t_{vRWM}}{(1-\eta_t)^{\ell^t_i}} \\
= & \sum_{i \in \Sigma_{\text{II}}} y'_i \ln \left(\frac{1}{1-\eta_t} \right)^{\ell^t_i} + \ln(1 - \eta_t \ell^t_{vRWM}) \\
= & \left(\ln \frac{1}{1-\eta_t} \right) \cdot \sum_{i \in \Sigma_{\text{II}}} y'_i \ell^t_i + \ln(1 - \eta_t \ell^t_{vRWM}) \\
\leq & \left(\ln \frac{1}{1-\eta_t} \right) u + \ln(1 - \eta_t \ell^t_{vRWM}) \ ,
\end{aligned}
$$

because strategy y' never generates more loss than u.

Proof of the Theorem

We take the derivative of

$$\left(\ln \frac{1}{1-\eta_t}\right) u + \ln(1-\eta_t \ell^t_{vRWM})$$

for η_t and equate it with 0. This implies a minimum is attained at

$$\eta_t = 1 - \frac{u(1-\ell^t_{vRWM})}{(1-u)\ell^t_{vRWM}}$$

as desired. Plugging in this expression for η_t yields

$$\begin{aligned} & - u \ln \left(\frac{u}{\ell^t_{vRWM}} \cdot \frac{1-\ell^t_{vRWM}}{1-u} \right) + \ln \frac{1-\ell^t_{vRWM}}{1-u} \\ = & - RE(u \mid\mid \ell^t_{vRWM}) \ . \end{aligned}$$

Convergence of vRWM

Corollary

For any sequence of strategies $y^1, y^2, \ldots$ the number of rounds in which the loss $\ell^t_{vRWM} \geq u + \varepsilon$ is at most

$$\frac{\ln |\Sigma_{\text{II}}|}{RE(u \,||\, u + \varepsilon)} \ .$$

For fixed ε this time is independent of T. Thus, the loss suffered in time steps t must get closer to u when t gets larger and larger. This is a much more desirable behavior than, e.g., the weird no-regret algorithm, which yields a loss of 1 every second round, even for arbitrarily large t.

Imitating Experts

No-Regret Algorithms and Coarse-Correlated Equilibria

Zero-Sum Games

Concave Games

Broadcast Game

A set $\mathcal{N}$ of n users want to download a video broadcast over a common link.

- Link has *capacity* C, we let wlog $C = 1$.
- As strategy player i places a bid $s_i \in \Sigma_i$, where $\Sigma_i = [b_{\min}, 1]$ with $b_{\min} > 0$ some minimum bid.
- Service provider M collects vector s of all bids, determines a proportional throughput rate for each player:

$$M_i(s) = \frac{s_i}{\sum_{j \in \mathcal{N}} s_j}.$$

  Note: $M_i(s) > 0$ and $\sum_i M_i(s) = 1 = C$.

- Player i gets his rate and pays his bid to the provider. Utility

$$u_i(s) = \alpha_i \cdot M_i(s) - s_i \quad = \quad \frac{\alpha_i \cdot s_i}{\sum_{j \in \mathcal{N}} s_j} - s_i$$

  (Player i values money vs. rate with factor $\alpha_i > 0$).

Example: (let, e.g., $b_{\min} = 0.01$)

Player	α_i	Bid s_i	Rate $M_i(s)$	Utility $u_i(s)$
1	2	0.9	0.45	0.90 - 0.9 = 0
2	3	0.7	0.35	1.05 - 0.7 = 0.35
3	4	0.4	0.25	1.00 - 0.4 = 0.60

Functions $u_i(s_i, s_{-i})$ for $s = (0.9, 0.7, 0.4)$:

- $u_1(x, 0.7, 0.4) = \frac{2x}{x+1.1} - x$
- $u_2(0.9, x, 0.4) = \frac{3x}{x+1.3} - x$
- $u_1(0.9, 0.7, x) = \frac{4x}{x+1.6} - x$

Observe that utility functions are concave.

Concavity/Convexity

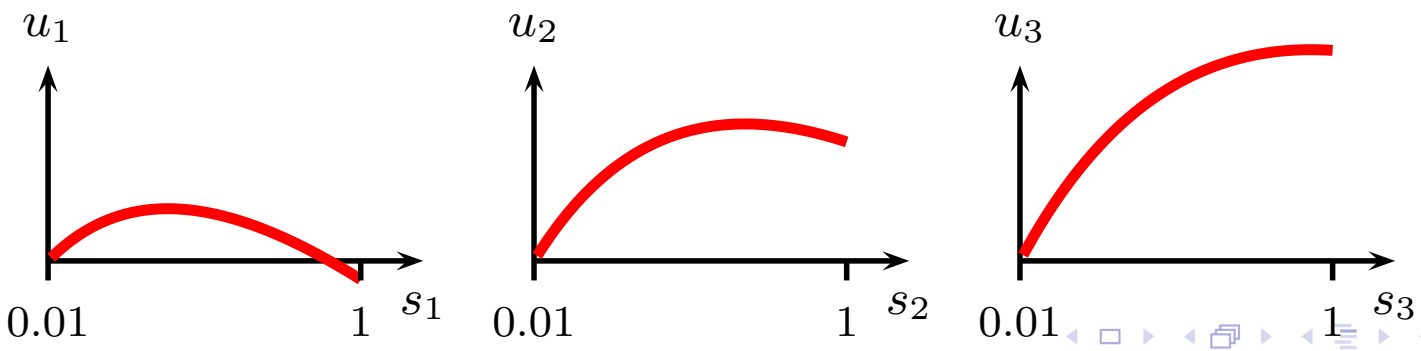
Definition (Concave/Convex)

A function $f : X \to \mathbb{R}$ is convex (concave) on $X \subset \mathbb{R}^k$ if the direct connection between $f(x)$ and $f(y)$ lies above (below) f for every $x, y \in X$.
Formally, for all $x, y \in \mathbb{R}$

$$\begin{array}{lrcll} \text{Convex:} & \lambda f(x) + (1-\lambda) f(y) & \geq & f(\lambda x + (1-\lambda) y), & \text{for all } \lambda \in (0,1) \\ \text{Concave:} & \lambda f(x) + (1-\lambda) f(y) & \leq & f(\lambda x + (1-\lambda) y), & \text{for all } \lambda \in (0,1) \end{array}$$

For strictly concave/convex, $\geq$ and $\leq$ are replaced with $>$ and $<$. Obviously if f is (strictly) concave, then $-f$ is (strictly) convex, and vice versa.

Utilities in the bandwidth game are strictly concave, e.g., in the previous example:



Existence of Equilibrium

A broadcast game is not a finite game, it has infinite strategy spaces.

Does a Nash equilibrium always exist?

One can use concavity of utility functions and Brouwer's Theorem to prove:

Lemma
Every broadcast game has at least one (pure) Nash equilibrium.

This raises the question how a Nash equilibrium can evolve in these games.

Do players using no-regret learning converge to a Nash equilibrium?

To answer this question, we first have to find no-regret learning algorithms for an infinite numbers of experts...

Online Convex Minimization

An experts problem with infinitely many experts:

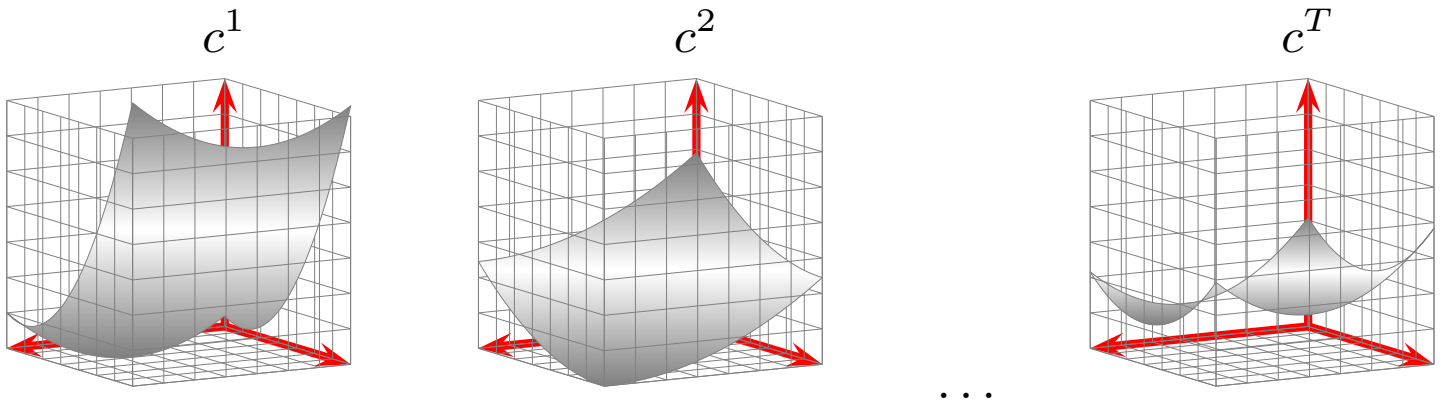
- "Experts" are all points in a convex and compact set $D \subset \mathbb{R}^k$
- Every round $t = 1, \ldots, T$ we pick a point $x^t \in D$
- Then we learn a differentiable and convex cost function $c^t : D \to \mathbb{R}$.

Goal: Pick x^t's to minimize total cost $\sum_{t=1}^{T} c^t(x^t)$.

Note that instead of minimizing convex costs, we can equivalently think of maximizing concave utility $-c^t(x^t)$.

Examples

For example, we could have $D = [0,1]^2$ and learn convex cost functions:



In the repeated broadcast game

- Player i has a compact and convex $D = [b_{\min}, 1]$
- His utility $u_i(s_i, s^t_{-i})$ is concave and differentiable. Also, it depends on unknown bids s^t_{-i}, which are learned only after strategy s_i is chosen.

No-Regret Property

- Our cost: $\sum_{t=1}^{T} c^t(x^t)$ Best expert x^*: $\sum_{t=1}^{T} c^t(x^*) = \min_{x \in D} \sum_{t=1}^{T} c^t(x)$
- Average regret per time step:

$$\frac{R(T)}{T} = \frac{1}{T}\left(\sum_{t=1}^{T} c^t(x^t) - \sum_{t=1}^{T} c^t(x^*)\right)$$

- No-regret algorithm if $\frac{R(T)}{T} \to 0$ for $T \to \infty$.

We will derive a no-regret algorithm using projected gradient descent.

Gradient Descent

Definition (Gradient)

For a differentiable function $f : \mathbb{R}^k \to \mathbb{R}$, the gradient at a point $x \in \mathbb{R}^k$ is a vector $\nabla f(x) \in \mathbb{R}^k$ that points in the opposite direction of steepest descent.

In the broadcast game, the cost $c^t(x) = -u_i(x, s^t_{-i})$ is a function only of $x \in \mathbb{R}$, as s^t_{-i} cannot be influenced by player i. Then ∇c^t is the derivative of c^t:

$$\nabla c^t(x) = 1 - \alpha_i \cdot \frac{\sum_{j \neq i} s^t_j}{\left(x + \sum_{j \neq i} s^t_j\right)^2} \quad .$$

Gradient Descent

To minimize a *single* convex function $f : \mathbb{R}^k \to \mathbb{R}$, it is sufficient to start at any point $x^1 \in \mathbb{R}^k$ and iteratively update it using the gradient by making steps in the direction of steepest descent

$$x^{t+1} = x^t - \eta \cdot \nabla f(x^t) \ ,$$

with sufficiently small step size $\eta \in \mathbb{R}$. We then "sink into the valley" of the convex function and arrive close to (depending on η) the global minimum.

When we use gradients to minimize a single convex function f *over a convex subspace* D, following the gradient descent could lead us outside D. A trick to get around this is to map the point back into D as follows.

Projected Gradient Descent

Definition (Projection)

We define a projection $P : \mathbb{R}^k \to D$ such that $P(x) = x'$, where $x \in \mathbb{R}^k$ is arbitrary and $x' \in D$ is closest to x.

In the broadcast game $P(x) = x$ for $x \in [b_{\min}, 1]$, $P(x) = 1$ for $x > 1$, and $P(x) = b_{\min}$ for $x < b_{\min}$.

Intuitively, the projection returns x to the closest point in D.

If we minimize a single convex function f over a convex and compact space D using projected gradient updates

$$x^{t+1} = P(x^t - \eta \nabla f(x^t)) \ ,$$

the convexity lets us "circle around the border" of D to sink to the minimum of f within D.

GIGA

For the online scenario with changing functions, we use the generalized infinitesimal gradient ascent (GIGA) algorithm:

$$\text{Pick } x^1 \in D \text{ arbitrary} \quad \text{and} \quad x^{t+1} = P(x^t - \eta \cdot \nabla c^t(x^t)) \ .$$

It uses the gradient for c^t for optimizing c^{t+1}. This seems like a stupid idea, as c^{t+1} can be completely different from c^t. Nevertheless, ...

Theorem (Zinkevich, 2003)

Let $G \geq \|\nabla c^t(x)\|$ and $\Delta \geq \|x - y\|$ for all $x, y \in D$ and $t = 1, \ldots, T$. If $\eta = \frac{\Delta}{G\sqrt{T}}$, then GIGA experiences a regret of

$$R(T) \leq \Delta \cdot G \cdot \sqrt{T}.$$

If G and Δ are independent of T, GIGA is a no-regret algorithm.

Proof of the Theorem

Intuition why GIGA "works":

- Projected gradient descent works if all cost functions c^t are similar.
- If functions are highly different, we get high cost, but then optimum $x^* \in D$ must have high cost, too.

Potential argument to capture this intuition:

- W.l.o.g. label optimum as origin $x^* = 0$ of the coordinate system.
- Consider "potential" as $\Phi_t = \frac{1}{2\eta}\|x^t\|^2$.
- Φ_t measures distance of x^t to $x^* = 0$

Lemma (Cost-vs.-Distance)

$$c^t(x^t) - c^t(0) + \Phi_{t+1} - \Phi_t \leq \eta \cdot G^2/2$$

Either cost is close to optimal, or x^{t+1} is closer to x^* than x^t.

Proof of the Cost-vs.-Distance Lemma

Note that $\|P(x)\| \leq \|x\|$, because D is convex and the projection always moves x towards D and, thus, the origin $x^* = 0 \in D$.

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= \frac{1}{2\eta}(\|x^{t+1}\|^2 - \|x^t\|^2) \\
&\leq \frac{1}{2\eta}(\|x^t - \eta \nabla c^t(x^t)\|^2 - \|x^t\|^2) \qquad \text{(as } \|P(x)\| \leq \|x\|\text{)} \\
&= \frac{1}{2\eta}(\|x^t\|^2 + \eta^2 \|\nabla c^t(x^t)\|^2 - 2\eta \nabla c^t(x^t) x^t - \|x^t\|^2)
\end{aligned}$$

where the last step uses the vector law of cosines, $\|u+v\|^2 = \|u\|^2 + \|v\|^2 + 2u^T v$.

Proof of the Cost-vs.-Distance Lemma

Simplifying and using the definition of G yields

$$\begin{aligned}\Phi_{t+1} - \Phi_t &\leq \frac{1}{2\eta}(\|x^t\|^2 + \eta^2\|\nabla c^t(x^t)\|^2 - 2\eta\nabla c^t(x^t)x^t - \|x^t\|^2) \\ &\leq \frac{1}{2}\eta G^2 - \nabla c^t(x^t)\cdot x^t \ .\end{aligned}$$

Convexity means a function grows "super-linear", formally

$$c^t(0) - c^t(x^t) \geq \nabla c^t(x^t)\cdot(0 - x^t) \ .$$

Using this insight, we have

$$\begin{aligned}\Phi_{t+1} - \Phi_t &\leq \eta G^2/2 - \nabla c^t(x^t)\cdot x^t \\ &\leq \eta G^2/2 + c^t(0) - c^t(x^t) \ ,\end{aligned}$$

which proves the lemma. □ (Lemma)

Proof of the Theorem

Summing up from $t = 1, \dots, T$, we get a telescopic sum and the lemma yields

$$\begin{aligned}\sum_{t=1}^{T}(c^t(x^t) - c^t(0) + \Phi_{t+1} - \Phi_t) &= \Phi_{T+1} - \Phi_1 + \sum_{t=1}^{T}(c^t(x^t) - c^t(0)) \\ &\leq T \cdot \eta G^2/2 \ .\end{aligned}$$

We recall that $x^* = 0$ and use $\frac{\Delta^2}{2\eta} \geq \Phi_t \geq 0$ and $\eta = \frac{\Delta}{G\sqrt{T}}$ to get

$$\begin{aligned}R(T) &= \sum_{t=1}^{T}(c^t(x^t) - c^t(0)) \\ &\leq \Phi_1 - \Phi_{T+1} + \frac{T\eta G^2}{2} \leq \frac{\Delta^2}{2\eta} + \frac{T\eta G^2}{2} \\ &= \frac{\Delta G\sqrt{T}}{2} + \frac{\Delta G\sqrt{T}}{2} \ ,\end{aligned}$$

which proves the theorem. □ (Theorem)

Convergence to Equilibrium

Now that we have derived a no-regret algorithm for infinite strategy spaces, we can tackle the question how a Nash equilibrium might evolve in Broadcast Games (and more general variants).

Do players using no-regret learning converge to a Nash equilibrium?

A quick experiment in the example broadcast game with 3 players given above turns out as follows.

t	Player 1		Player 2		Player 3	
	$\nabla u_1(s^{t-1})$	s_1^t	$\nabla u_2(s^{t-1})$	s_2^t	$\nabla u_3(s^{t-1})$	s_3^t
1		0,01000		0,01000		0,01000

t	Player 1		Player 2		Player 3	
	$\nabla u_1(s^{t-1})$	s_1^t	$\nabla u_2(s^{t-1})$	s_2^t	$\nabla u_3(s^{t-1})$	s_3^t
1		0,01000		0,01000		0,01000
2	(21,22222)	1,00000	(43,44444)	1,00000	(65,66667)	1,00000

t	Player 1		Player 2		Player 3	
	$\nabla u_1(s^{t-1})$	s_1^t	$\nabla u_2(s^{t-1})$	s_2^t	$\nabla u_3(s^{t-1})$	s_3^t
1		0,01000		0,01000		0,01000
2	(21,22222)	1,00000	(43,44444)	1,00000	(65,66667)	1,00000
3	(-0,77778)	0,45003	(-0,55556)	0,60716	(-0,33333)	0,76430

t	Player 1		Player 2		Player 3	
	$\nabla u_1(s^{t-1})$	s_1^t	$\nabla u_2(s^{t-1})$	s_2^t	$\nabla u_3(s^{t-1})$	s_3^t
1		0,01000		0,01000		0,01000
2	(21,22222)	1,00000	(43,44444)	1,00000	(65,66667)	1,00000
3	(-0,77778)	0,45003	(-0,55556)	0,60716	(-0,33333)	0,76430
4	(-0,58664)	0,11133	(-0,26800)	0,45243	(-0,04408)	0,73885

t	Player 1		Player 2		Player 3	
	$\nabla u_1(s^{t-1})$	s_1^t	$\nabla u_2(s^{t-1})$	s_2^t	$\nabla u_3(s^{t-1})$	s_3^t
1		0,01000		0,01000		0,01000
2	(21,22222)	1,00000	(43,44444)	1,00000	(65,66667)	1,00000
3	(-0,77778)	0,45003	(-0,55556)	0,60716	(-0,33333)	0,76430
4	(-0,58664)	0,11133	(-0,26800)	0,45243	(-0,04408)	0,73885
5	(-0,29793)	0,01000	(0,00210)	0,45348	(-0,00324)	0,73723

t	Player 1		Player 2		Player 3	
	$\nabla u_1(s^{t-1})$	s_1^t	$\nabla u_2(s^{t-1})$	s_2^t	$\nabla u_3(s^{t-1})$	s_3^t
1		0,01000		0,01000		0,01000
2	(21,22222)	1,00000	(43,44444)	1,00000	(65,66667)	1,00000
3	(-0,77778)	0,45003	(-0,55556)	0,60716	(-0,33333)	0,76430
4	(-0,58664)	0,11133	(-0,26800)	0,45243	(-0,04408)	0,73885
5	(-0,29793)	0,01000	(0,00210)	0,45348	(-0,00324)	0,73723
6	(-0,17409)	0,01000	(0,03659)	0,46985	(-0,03555)	0,72133
7	(-0,17441)	0,01000	(0,01375)	0,47546	(-0,00227)	0,72040
8	(-0,17759)	0,01000	(0,00461)	0,47720	(0,00157)	0,72099
9	(-0,17917)	0,01000	(0,00154)	0,47775	(0,00128)	0,72145
10	(-0,17984)	0,01000	(0,00051)	0,47792	(0,00075)	0,72170
11	(-0,18012)	0,01000	(0,00016)	0,47797	(0,00040)	0,72182
12	(-0,18024)	0,01000	(0,00004)	0,47798	(0,00021)	0,72189
13	(-0,18029)	0,01000	(0,00000)	0,47798	(0,00011)	0,72192
14	(-0,18031)	0,01000	(-0,00001)	0,47798	(0,00006)	0,72193
15	(-0,18032)	0,01000	(-0,00001)	0,47797	(0,00003)	0,72194
16	(-0,18033)	0,01000	(-0,00001)	0,47797	(0,00002)	0,72195
17	(-0,18033)	0,01000	(-0,00001)	0,47797	(0,00001)	0,72195
18	(-0,18033)	0,01000	(-0,00000)	0,47797	(0,00000)	0,72195

Concave Games

Broadcast games are a special case of a larger class of games called *socially concave games*. Every socially concave game has a pure Nash equilibrium.

Definition

A socially concave game is a strategic game $\Gamma = (\mathcal{N}, (\Sigma_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}})$, where

- $\mathcal{N}$ is a finite set of n players,
- every strategy set Σ_i is compact and convex,
- utility $u_i(s_i, s_{-i})$ is concave in s_i, for every fixed s_{-i},
- utility $u_i(s_i, s_{-i})$ is convex in s_{-i}, for every fixed $s_i \in \Sigma_i$,
- there exist $(\lambda_i)_{i\in\mathcal{N}}$ with $\lambda_i > 0$, $\sum_i \lambda_i = 1$ such that $g(s) = \sum_{i\in\mathcal{N}} \lambda_i u_i(s)$ is a concave function in s.

No-Regret Learning in Socially Concave Games

Theorem

Consider a socially concave game played repeatedly for T rounds. If every player plays according to a no-regret algorithm, then as $T \to \infty$:

1. *The average history of play $\hat{s}$ converges to a (mixed) Nash equilibrium.*
2. *The average utility of each player converges to her utility at the mixed Nash equilibrium.*

If all players use GIGA to pick their strategies in a repeated socially concave game, the average history of play converges to a Nash equilibrium of the game.

Furthermore, it is known that if the utility functions u_i are strictly concave in s_i, there is a unique mixed Nash equilibrium which is also a pure one.

Thus, the theorem proves the intuition from our experiment, i.e., in the broadcast game the set of players can learn a pure Nash equilibrium using GIGA.

Literature

- Nisan, Roughgarden, Tardos, Vazirani. Algorithmic Game Theory, 2007. (Chapter 4).
- Littlestone, Warmuth. The Weighted Majority Algorithm. Information & Computation 108(2):212–261, 1994.
- Freund, Schapire. Adaptive Game Playing using Multiplicative Weights. Games and Economic Behavior, 29:79–103, 1999.
- Roughgarden. Twenty Lectures on Algorithmic Game Theory, 2016. (Chapter 17+18).
- Blum, Mansour. From External to Internat Regret. Journal of Machine Learning Research 8:1307–1324, 2007.

Literatur

- Rosen. Existence and Uniqueness of Equilibrium Points for Concave n-Person Games. Econometrica, 33(3):520–534, 1965.
- Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. ICML 2003.
- Even-Dar, Mansour, Nadav. On the Convergence of Regret Minimization Dynamics in Concave Games. STOC 2009.