Algorithmic game theory

Martin Balko

6th lecture

November 16th 2023

Regret minimization

Regret minimization

Regret minimization

- We introduce a completely new model of interactions based on so-called regret minimization.

Regret minimization

- We introduce a completely new model of interactions based on so-called regret minimization. We apply online learning.

Sources: https://towardsdatascience.com/

Regret minimization

- We introduce a completely new model of interactions based on so-called regret minimization. We apply online learning.

Sources: https://towardsdatascience.com/

- Later, we apply these new methods to design new fast algorithms to approximate correlated equilibria.

Regret minimization

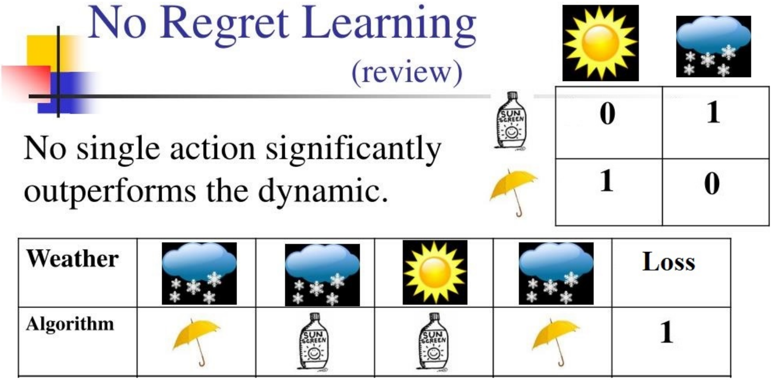
- We introduce a completely new model of interactions based on so-called regret minimization. We apply online learning.

Sources: https://towardsdatascience.com/

- Later, we apply these new methods to design new fast algorithms to approximate correlated equilibria.
- Today, we introduce the model and some basic algorithms how to minimize regret.

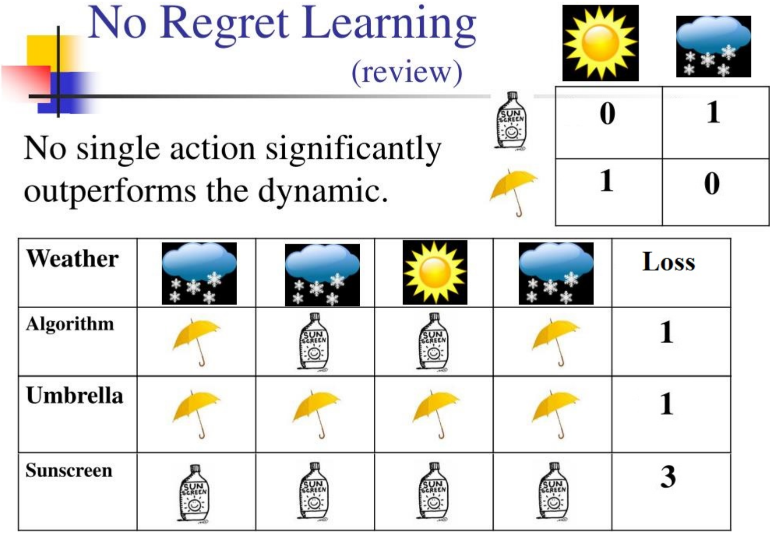
Example

Example



Source: No regret algorithms in games (Georgios Piliouras)

Example



Source: No regret algorithms in games (Georgios Piliouras)

The greedy algorithm

The greedy algorithm

Algorithm 0.2: GREEDY ALGORITHM(X, T)

Input : A set of actions $X = \{1, \dots, N\}$ and number of steps $T \in \mathbb{N}$.
Output : A probability distribution p^t for every $t \in \{1, \dots, T\}$.
$p^1 \leftarrow (1, 0, \dots, 0)$,
for $t = 2, \dots, T$

do $\begin{cases} L_{min}^{t-1} \leftarrow \min_{j \in X}\{L_j^{t-1}\}, \\ S^{t-1} \leftarrow \{i \in X \colon L_i^{t-1} = L_{min}^{t-1}\}, \\ k \leftarrow \min S^{t-1}, \\ p_k^t \leftarrow 1, p_i^t \leftarrow 0 \text{ for } i \neq k, \end{cases}$

Output $\{p^t \colon t \in \{1, \dots, T\}\}$.

The randomized greedy algorithm

The randomized greedy algorithm

Algorithm 0.4: RANDOMIZED GREEDY ALGORITHM(X, T)

Input : A set of actions $X = \{1, \ldots, N\}$ and number of steps $T \in \mathbb{N}$.
Output : A probability distribution p^t for every $t \in \{1, \ldots, T\}$.
$p^1 \leftarrow (1/N, \ldots, 1/N)$,
for $t = 2, \ldots, T$

$$\textbf{do} \begin{cases} L_{min}^{t-1} \leftarrow \min_{j \in X}\{L_j^{t-1}\}, \\ S^{t-1} \leftarrow \{i \in X \colon L_i^{t-1} = L_{min}^{t-1}\}, \\ p_i^t \leftarrow 1/|S^{t-1}| \text{ for every } i \in S^{t-1} \text{ and } p_i^t \leftarrow 0 \text{ otherwise.} \end{cases}$$

Output $\{p^t \colon t \in \{1, \ldots, T\}\}$.

The randomized greedy algorithm

Algorithm 0.5: RANDOMIZED GREEDY ALGORITHM(X, T)

Input : A set of actions $X = \{1, \dots, N\}$ and number of steps $T \in \mathbb{N}$.
Output : A probability distribution p^t for every $t \in \{1, \dots, T\}$.
$p^1 \leftarrow (1/N, \dots, 1/N)$,
for $t = 2, \dots, T$

$$\textbf{do} \begin{cases} L_{min}^{t-1} \leftarrow \min_{j \in X}\{L_j^{t-1}\}, \\ S^{t-1} \leftarrow \{i \in X \colon L_i^{t-1} = L_{min}^{t-1}\}, \\ p_i^t \leftarrow 1/|S^{t-1}| \text{ for every } i \in S^{t-1} \text{ and } p_i^t \leftarrow 0 \text{ otherwise.} \end{cases}$$

Output $\{p^t \colon t \in \{1, \dots, T\}\}$.

The polynomial weights algorithm

The polynomial weights algorithm

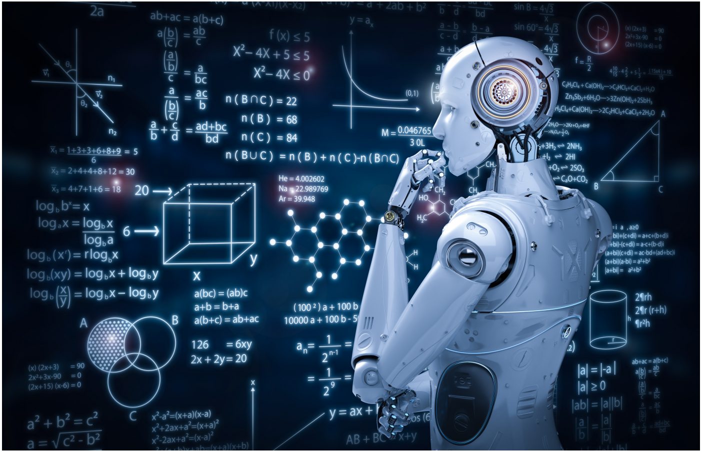
Algorithm 0.7: POLYNOMIAL WEIGHTS ALGORITHM(X, T, η)

Input : A set of actions $X = \{1, \ldots, N\}$, $T \in \mathbb{N}$, and $\eta \in (0, 1/2]$.
Output : A probability distribution p^t for every $t \in \{1, \ldots, T\}$.
$w_i^1 \leftarrow 1$ for every $i \in X$,
$p^1 \leftarrow (1/N, \ldots, 1/N)$,
for $t = 2, \ldots, T$

$$\textbf{do} \begin{cases} w_i^t \leftarrow w_i^{t-1}(1 - \eta \ell_i^{t-1}), \\ W^t \leftarrow \sum_{i \in X} w_i^t, \\ p_i^t \leftarrow w_i^t / W^t \text{ for every } i \in X. \end{cases}$$

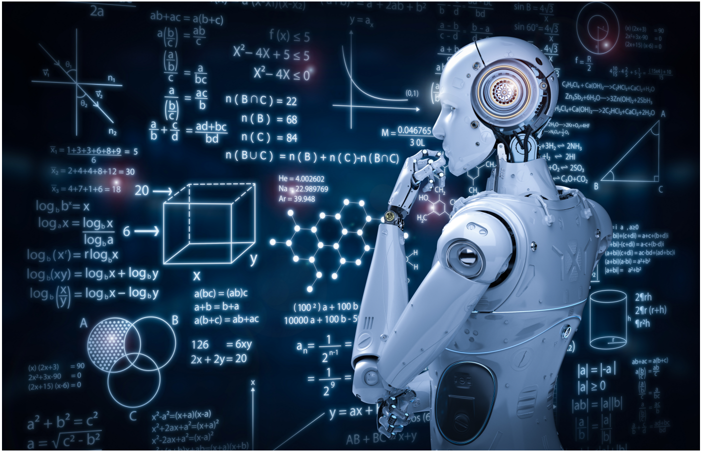
Output $\{p^t : t \in \{1, \ldots, T\}\}$.

- Besides game theory, the “multiplicative weight update method” has many applications in various fields of science, for example in optimization, theoretical computer science, and machine learning.



Sources: https://clubitc.ro

- Besides game theory, the “multiplicative weight update method” has many applications in various fields of science, for example in optimization, theoretical computer science, and machine learning.



Sources: https://clubitc.ro

- See https://en.wikipedia.org/wiki/Multiplicative_weight_update_method#Applications

The No-regret dynamics

The No-regret dynamics

- "Players in a normal-form game play against each other by selecting actions according to the Polynomial-weights algorithm."

The No-regret dynamics

- "Players in a normal-form game play against each other by selecting actions according to the Polynomial-weights algorithm."

Algorithm 0.10: NO-REGRET DYNAMICS(G, T, ε)

Input : A normal-form game $G = (P, A, C)$ of n players, $T \in \mathbb{N}$, and $\varepsilon > 0$.
Output : A prob. distribution p_i^t on A_i for each $i \in P$ and $t \in \{1, \ldots, T\}$.
for every step $t = 1, \ldots, T$

do Each player $i \in P$ independently chooses a mixed strategy p_i^t using an algorithm with average regret at most ε, with actions corresponding to pure strategies.
Each player $i \in P$ receives a loss vector $\ell_i^t = (\ell_i^t(a_i))_{a_i \in A_i}$, where $\ell_i^t(a_i) \leftarrow \mathbb{E}_{a_{-i}^t \sim p_{-i}^t}[C_i(a_i; a_{-i}^t)]$ for the product distribution $p_{-i}^t = \prod_{j \neq i} p_j^t$.

Output $\{p^t : t \in \{1, \ldots, T\}\}$.

"ENROLL IN AGT" THEY SAID

Algorithm 2.6.4: No-regret dynamics(G, T, ε)

Input : A normal-form game $G = (P, A, C)$ of n players, $T \in \mathbb{N}$ and $\varepsilon > 0$.
Output : A probability distribution p_i^t on A_i for each $i \in P$ and $t \in \{1, \ldots, T\}$.
for every step $t = 1, \ldots, T$

do
- Each player $i \in P$ independently chooses a mixed strategy p_i^t using an algorithm with average regret at most ε, with actions corresponding to pure strategies.
- Each player $i \in P$ receives a loss vector $\ell_i^t = (\ell_i^t(a_i))_{a_i \in A_i}$, where $\ell_i^t(a_i) \leftarrow \mathbb{E}_{a_{-i}^t \sim \sigma_{-i}^t}[C_i(a_i; a_{-i}^t)]$ for the product distribution $\sigma_{-i}^t = \prod_{j \neq i} p_j^t$.

Output $\{p^t : t \in \{1, \ldots, T\}\}$.

"THERE'LL BE NO REGRET" THEY SAID

Sources: Students of MFF UK

"ENROLL IN AGT" THEY SAID

Algorithm 2.6.4: NO-REGRET DYNAMICS(G, T, ε)

Input : A normal-form game $G = (P, A, C)$ of n players, $T \in \mathbb{N}$ and $\varepsilon > 0$.
Output : A probability distribution p_i^t on A_i for each $i \in P$ and $t \in \{1, \dots, T\}$.
for every step $t = 1, \dots, T$

do {
Each player $i \in P$ independently chooses a mixed strategy p_i^t using an algorithm with average regret at most ε, with actions corresponding to pure strategies.
Each player $i \in P$ receives a loss vector $\ell_i^t = (\ell_i^t(a_i))_{a_i \in A_i}$, where $\ell_i^t(a_i) \leftarrow \mathbb{E}_{a_{-i}^t \sim \sigma_{-i}^t}[C_i(a_i; a_{-i}^t)]$ for the product distribution $\sigma_{-i}^t = \prod_{j \neq i} p_j^t$.
}

Output $\{p^t : t \in \{1, \dots, T\}\}$.

"THERE'LL BE NO REGRET" THEY SAID

Sources: Students of MFF UK

Thank you for your attention.