Internal Regret and Calibration

Yoav Freund

February 26, 2020

Outline

Outline

Outline

Outline

Outline

External regret

- L_i - The cumulative loss of action i

External regret

- L_i - The cumulative loss of action i
- L_A - The cumulative loss of the algorithm.

External regret

- L_i - The cumulative loss of action i
- L_A - The cumulative loss of the algorithm.
- External Regret $R_i = L_A - L_i$

External regret

- L_i - The cumulative loss of action i
- L_A - The cumulative loss of the algorithm.
- External Regret $R_i = L_A - L_i$
- We seek a uniform bound on the regret: hold simultanously for all $R_i, i = 1 \ldots N$

External regret

- L_i - The cumulative loss of action i
- L_A - The cumulative loss of the algorithm.
- External Regret $R_i = L_A - L_i$
- We seek a uniform bound on the regret: hold simultanously for all $R_i, i = 1 \ldots N$
- For the bounded loss $\ell_i^y \in [0, 1]$ we have $\max_i R_i^n = O(\sqrt{n \ln N})$

External regret

- L_i - The cumulative loss of action i
- L_A - The cumulative loss of the algorithm.
- External Regret $R_i = L_A - L_i$
- We seek a uniform bound on the regret: hold simultanously for all $R_i, i = 1 \ldots N$
- For the bounded loss $\ell_i^y \in [0, 1]$ we have $\max_i R_i^n = O(\sqrt{n \ln N})$
- For log loss we have $\max_i R_i^n = O(\ln N)$

Internal regret

- $R_{(i,j),n}$ regret for not taking action j instead of each action i during iterations $1 \ldots n$

Internal regret

- $R_{(i,j),n}$ regret for not taking action j instead of each action i during iterations $1 \ldots n$
- We want an algorithm such that $\max_{(i,j)} R_{(i,j),n} = o(n)$

Zero-Sum vs. Non Zero-Sum

- In a zero sum game the gain of the row player is the loss of the column player.

Zero-Sum vs. Non Zero-Sum

- In a zero sum game the gain of the row player is the loss of the column player.
- Zero sum games have a unique **MiniMax** equilibrium: neither side has an incentive to move from their optimal mixed strategy. Whether the other side plays optimally or not.

Zero-Sum vs. Non Zero-Sum

- In a zero sum game the gain of the row player is the loss of the column player.
- Zero sum games have a unique **MiniMax** equilibrium: neither side has an incentive to move from their optimal mixed strategy. Whether the other side plays optimally or not.
- In general games the gains of the players are unconstrained.

Zero-Sum vs. Non Zero-Sum

- In a zero sum game the gain of the row player is the loss of the column player.
- Zero sum games have a unique **MiniMax** equilibrium: neither side has an incentive to move from their optimal mixed strategy. Whether the other side plays optimally or not.
- In general games the gains of the players are unconstrained.
- In general games There might not be a **MiniMax** equilibrium

Nash Equilibrium

- In general games There might not be a **MiniMax** equilibrium.

Nash Equilibrium

- In general games There might not be a **MiniMax** equilibrium.
- **Nash** any general game there are one or more equilibria of the folloing form.

Nash Equilibrium

- In general games There might not be a **MiniMax** equilibrium.
- **Nash** any general game there are one or more equilibria of the folloing form.
- Equilibrium *i*: mixed strategies **P**, **Q** such that row player will not deviate from **P** if it knows that column player will play **Q**, and vice versa.

Prisoner's Dilemma

Prisoner's dilemma [edit]

Main article: Prisoner's dilemma

Imagine two prisoners held in separate cells, interrogated simultaneously, and offered deals (lighter jail sentences) for betraying their fellow criminal. They can "cooperate" (with the other prisoner) by not snitching, or "defect" by betraying the other. However, there is a catch; if both players defect, then they both serve a longer sentence than if neither said anything. Lower jail sentences are interpreted as higher payoffs (shown in the table).

Prisoner 1 \ Prisoner 2	Cooperate (with other)	Defect (betray other)
Cooperate (with other)	−1, −1	−3, 0
Defect (betray other)	0, −3	−2, −2

Example PD payoff matrix

The prisoner's dilemma has a similar matrix as depicted for the coordination game, but the maximum reward for each player (in this case, a minimum loss of 0) is obtained only when the players' decisions are different. Each player improves their own situation by switching from "cooperating" to "defecting", given knowledge that the other player's best decision is to "defect". The prisoner's dilemma thus has a single Nash equilibrium: both players choosing to defect.

What has long made this an interesting case to study is the fact that this scenario is globally inferior to "both cooperating". That is, both players would be better off if they both chose to "cooperate" instead of both choosing to defect. However, each player could improve their own situation by breaking the mutual cooperation, no matter how the other player possibly (or certainly) changes their decision.

Driving Game

Driving on a road against an oncoming car, and having to choose either to swerve on the left or to swerve on the right of the road, is also a coordination game. For example, with payoffs 10 meaning no crash and 0 meaning a crash, the coordination game can be defined with the following payoff matrix:

Driver 1 \ Driver 2	Drive on the Left	Drive on the Right
Drive on the Left	10, 10	0, 0
Drive on the Right	0, 0	10, 10

The driving game

In this case there are two pure-strategy Nash equilibria, when both choose to either drive on the left or on the right. If we admit mixed strategies (where a pure strategy is chosen at random, subject to some fixed probability), then there are three Nash equilibria for the same case: two we have seen from the pure-strategy form, where the probabilities are (0%, 100%) for player one, (0%, 100%) for player two; and (100%, 0%) for player one, (100%, 0%) for player two respectively. We add another where the probabilities for each player are (50%, 50%).

Correlated Equilibrium

- **correlated equilibrium** proposed by Aumann [1974] much after **Nash equilibrium** [1951] (earlier by Cournot [1838])

Correlated Equilibrium

- **correlated equilibrium** proposed by Aumann [1974] much after **Nash equilibrium** [1951] (earlier by Cournot [1838])
- A correlation device: a state that can be read by all players.

Correlated Equilibrium

- **correlated equilibrium** proposed by Aumann [1974] much after **Nash equilibrium** [1951] (earlier by Cournot [1838])
- A correlation device: a state that can be read by all players.
- Examples: counter, clock, stoplight.

Correlated Equilibrium

- **correlated equilibrium** proposed by Aumann [1974] much after **Nash equilibrium** [1951] (earlier by Cournot [1838])
- A correlation device: a state that can be read by all players.
- Examples: counter, clock, stoplight.
- Strategy is a mapping from state of device to distribution over actions.

Game Of Chicken

An example [edit]

Correlated Equilibrium

Consider the game of chicken pictured. In this game two individuals are challenging each other to a contest where each can either *dare* or *chicken out*. If one is going to Dare, it is better for the other to chicken out. But if one is going to chicken out it is better for the other to Dare. This leads to an interesting situation where each wants to dare, but only if the other might chicken out.

	Dare	**C**hicken out
Dare	0, 0	7, 2
Chicken out	2, 7	6, 6

A game of Chicken

In this game, there are three Nash equilibria. The two pure strategy Nash equilibria are (*D*, *C*) and (*C*, *D*). There is also a mixed strategy equilibrium where each player Dares with probability 1/3.

Now consider a third party (or some natural event) that draws one of three cards labeled: (*C*, *C*), (*D*, *C*), and (*C*, *D*), with the same probability, i.e. probability 1/3 for each card. After drawing the card the third party informs the players of the strategy assigned to them on the card (but **not** the strategy assigned to their opponent). Suppose a player is assigned *D*, he would not want to deviate supposing the other player played their assigned strategy since he will get 7 (the highest payoff possible). Suppose a player is assigned *C*. Then the other player will play *C* with probability 1/2 and *D* with probability 1/2. The expected utility of Daring is 7(1/2) + 0(1/2) = 3.5 and the expected utility of chickening out is 2(1/2) + 6(1/2) = 4. So, the player would prefer chickening out.

Since neither player has an incentive to deviate, this is a correlated equilibrium. The expected payoff for this equilibrium is 7(1/3) + 2(1/3) + 6(1/3) = 5 which is higher than the expected payoff of the mixed strategy Nash equilibrium.

The following correlated equilibrium has an even higher payoff to both players: Recommend (*C*, *C*) with probability 1/2, and (*D*, *C*) and (*C*, *D*) with probability 1/4 each. Then when a player is recommended to play *C*, she knows that the other player will play *D* with (conditional) probability 1/3 and *C* with probability 2/3, and gets expected payoff 14/3, which is equal to (not less than) the expected payoff when she plays *D*. In this correlated equilibrium, both players get 5.25 in expectation. It can be shown that this is the

Finding the equilibrium mixed strategy

- **MiniMax** known matrix: Linear programming.

Finding the equilibrium mixed strategy

- **MiniMax** known matrix: Linear programming.
- **MiniMax** Minimizing External Regret: Exponential weights or follow the perturbed leader.

Finding the equilibrium mixed strategy

- **MiniMax** known matrix: Linear programming.
- **MiniMax** Minimizing External Regret: Exponential weights or follow the perturbed leader.
- **Nash equilibrium** known matrix - **NP hard**

Finding the equilibrium mixed strategy

- **MiniMax** known matrix: Linear programming.
- **MiniMax** Minimizing External Regret: Exponential weights or follow the perturbed leader.
- **Nash equilibrium** known matrix - **NP hard**
- **Correlated Equilibrium** Minimizing Internal Regret.

Calibration

- Observe a binary sequence $y_1, \ldots, y_{t-1}$ and make prediction q_t for the probability that $y_t = 1$

Calibration

- Observe a binary sequence $y_1, \ldots, y_{t-1}$ and make prediction q_t for the probability that $y_t = 1$
- Average outcomes:

$$\rho_n^\epsilon(x) = \frac{\sum_{t=1}^n y_t \mathbf{1}[q_t \in (x-\epsilon, x+\epsilon)]}{\sum_{t=1}^n \mathbf{1}[q_t \in (x-\epsilon, x+\epsilon)]}$$

Calibration

- Observe a binary sequence $y_1, \ldots, y_{t-1}$ and make prediction q_t for the probability that $y_t = 1$
- Average outcomes:

$$\rho_n^\epsilon(x) = \frac{\sum_{t=1}^n y_t \mathbf{1}[q_t \in (x-\epsilon, x+\epsilon)]}{\sum_{t=1}^n \mathbf{1}[q_t \in (x-\epsilon, x+\epsilon)]}$$

- ϵ-calibrated predictions:

$$\forall x \in [0,1]; \quad \limsup_{n\to\infty} |\rho_n^\epsilon(x) - x| \leq \epsilon$$

Calibration

- Observe a binary sequence $y_1, \ldots, y_{t-1}$ and make prediction q_t for the probability that $y_t = 1$
- Average outcomes:

$$\rho_n^\epsilon(x) = \frac{\sum_{t=1}^n y_t \mathbf{1}[q_t \in (x-\epsilon, x+\epsilon)]}{\sum_{t=1}^n \mathbf{1}[q_t \in (x-\epsilon, x+\epsilon)]}$$

- ϵ-calibrated predictions:

$$\forall x \in [0,1]; \quad \limsup_{n\to\infty} |\rho_n^\epsilon(x) - x| \leq \epsilon$$

- No deterministic algorithm can be calibrated for all sequences.

Calibration through bounded internal regret

- Consider only the predictions $0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N}{N}$ for $N > 1$

Calibration through bounded internal regret

- Consider only the predictions $0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N}{N}$ for $N > 1$
- Use the square loss (Brier Loss) $\sum_t (q_t - y_t)^2$

Calibration through bounded internal regret

- Consider only the predictions $0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N}{N}$ for $N > 1$
- Use the square loss (Brier Loss) $\sum_t (q_t - y_t)^2$
- Use a prediction algorithm that minimizes internal regret.

Calibration through bounded internal regret

- Consider only the predictions $0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N}{N}$ for $N > 1$
- Use the square loss (Brier Loss) $\sum_t (q_t - y_t)^2$
- Use a prediction algorithm that minimizes internal regret.
- If the prediction is not ϵ calibrated, then the internal regret has to be large.

From External to internal Regret

- At time t we need to produce a distribution $\mathbf{p}_t$ over N actions, so that the internal regret is small.

From External to internal Regret

- At time t we need to produce a distribution $\mathbf{p}_t$ over N actions, so that the internal regret is small.
- For each action pair $i \neq j$ we define a *modified strategy* $\mathbf{p}_t^{i \to j}$ by setting the ith coordinate in $\mathbf{p}_t$ to 0 and adding that mass to the jth coordinate.

From External to internal Regret

- At time t we need to produce a distribution $\mathbf{p}_t$ over N actions, so that the internal regret is small.
- For each action pair $i \neq j$ we define a *modified strategy* $\mathbf{p}_t^{i \to j}$ by setting the ith coordinate in $\mathbf{p}_t$ to 0 and adding that mass to the jth coordinate.
- We use **Hedge**(η) to combine the $N(N-1)$ modified strategies. $\sum_{i \neq j} \Delta_{(i,j),t} \mathbf{p}_t^{i \to j}$

From External to internal Regret

- At time t we need to produce a distribution $\mathbf{p}_t$ over N actions, so that the internal regret is small.
- For each action pair $i \neq j$ we define a *modified strategy* $\mathbf{p}_t^{i \to j}$ by setting the ith coordinate in $\mathbf{p}_t$ to 0 and adding that mass to the jth coordinate.
- We use **Hedge**(η) to combine the $N(N-1)$ modified strategies. $\sum_{i \neq j} \Delta_{(i,j),t} \mathbf{p}_t^{i \to j}$
- But we need a distribution over N actions not $N(N-1)$ modified strategies.

From External to internal Regret

- At time t we need to produce a distribution $\mathbf{p}_t$ over N actions, so that the internal regret is small.
- For each action pair $i \neq j$ we define a *modified strategy* $\mathbf{p}_t^{i \to j}$ by setting the ith coordinate in $\mathbf{p}_t$ to 0 and adding that mass to the jth coordinate.
- We use **Hedge**(η) to combine the $N(N-1)$ modified strategies. $\sum_{i \neq j} \Delta_{(i,j),t} \mathbf{p}_t^{i \to j}$
- But we need a distribution over N actions not $N(N-1)$ modified strategies.
- We solve the fixed point equation

$$\mathbf{p}_t = \sum_{i \neq j} \Delta_{(i,j),t} \mathbf{p}_t^{i \to j}$$