

# Online Lernen

Algorithmische Spieltheorie

Sommer 2018

## Imitation von Experten

No-Regret Algorithmen und grob-korrelierte Gleichgewichte

Nullsummenspiele

Korrelierte Gleichgewichte

## Einführendes Beispiel

In vielen Fällen trifft man Entscheidungen nicht einmal sondern wiederholt, z.B. jeden Tag, ohne zu wissen wie sich andere Teilnehmer/Spieler oder die "Natur" an diesem Tag verhalten werden. Dazu betrachten wir Lernalgorithmen, die mit dieser Art von Unsicherheit umgehen können.

Beispiel:

- ▶ Du mußt jeden Tag von zu Hause zum Campus Bockenheim pendeln.
- ▶ Die Fahrzeit liegt zwischen 30 und 60 Minuten, je nachdem welches Verkehrsmittel und welche Route Du wählst, wie der Verkehr ist oder wieviel Verspätung die Busse und Züge haben.
- ▶ Stell Dir vor, Du kennst drei **Experten**, die auch aus Deiner Straße zum Unicampus pendeln und dafür unterschiedliche Wege benutzen.

Wir werden zeigen, dass Du auf lange Sicht im Durchschnitt **fast so schnell sein kannst wie der beste Experte**, nur indem Du die Wahlen der Experten imitierst.

# Das Experten-Problem

Wir betrachten ein Gegnermodell mit diskreten Zeitschritten  $1, \dots, T$ .  
Sei  $[T] = \{1, \dots, T\}$ .

## Die Experten und ihre Kosten

- ▶ Es gibt  $N$  *Experten*, nummeriert von 1 bis  $N$ .
- ▶ Im Schritt  $t \in [T]$  erfährt Experte  $i \in [N]$  die *Kosten*  $\ell_i^t \in [0, 1]$ , die von einem unbekanntem Gegner gewählt werden (z.B. der "Natur")
- ▶ Sei  $L_i^t = \sum_{k=1}^t \ell_i^k$ .

## Kombination/Imitation von Experten

- ▶ In Schritt  $t$  wählt ein *Online-Algorithmus*  $H$  den Experten  $i \in [N]$  mit Wahrscheinlichkeit  $p_i^t$ .
- ▶ Der Vektor  $p^t$  kann von den vorherigen Kosten  $\ell^1, \dots, \ell^{t-1}$  abhängen.
- ▶ Der (erwartete) Verlust von  $H$  in Schritt  $t$  ist  $\ell_H^t = \sum_{i \in [N]} p_i^t \ell_i^t$ .
- ▶ Sei  $L_H^t = \sum_{k=1}^t \ell_H^k$ .

Imitation von Experten

No-Regret Algorithmen und grob-korrelierte Gleichgewichte

Nullsummenspiele

Korrelierte Gleichgewichte

# Ein Greedy-Algorithmus

Im Folgenden sei  $L_{min}^{t-1} = \min_{i \in [N]} L_i^{t-1}$  für  $1 \leq t \leq T$ .

## Greedy Algorithmus

In jedem Schritt  $t$ ,

- ▶ sei  $S^{t-1} = \{i : L_i^{t-1} = L_{min}^{t-1}\}$ ;
- ▶ sei  $j = \min\{S^{t-1}\}$ ;
- ▶ setze  $p_j^t = 1$ , und  $p_i^t = 0$ , für  $i \neq j$ .

Für die Analyse des Greedy-Algorithmus nehmen wir der Einfachheit halber an, dass alle Verluste entweder 0 oder 1 sind (anstatt reelle Zahlen in  $[0, 1]$ ).

# Greedy Algorithmus

**Beispiel:**

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

# Greedy Algorithmus

## Satz

Der Greedy Algorithmus garantiert für jede Folge von Verlusten aus  $\{0, 1\}$

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

## Beweis:

- ▶ Teile die Zeit in Phasen  $0, \dots, L_{min}^T$  ein, so dass gilt:

Jeder Schritt  $t$  mit  $L_{min}^{t-1} = i$  gehört zu Phase  $i$ .

- ▶ In jeder Phase  $i < L_{min}^T$  erfährt der Greedy-Algorithmus einen Verlust von höchstens  $N$ .
- ▶ In Phase  $L_{min}^T$  ist der Verlust von Greedy höchstens  $N - 1$ .





# Untere Schranke für deterministische Algorithmen

## Satz

*Für jeden deterministischen Online-Algorithmus  $D$  und jedes  $T \geq 1$  gibt es eine Folge von  $T$  Verlusten, so dass*

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

Diese untere Schranke kann man leicht zeigen, indem man das obige Beispiel für Greedy verallgemeinert. (Wie?)

Die untere Schranke zeigt, dass man ohne Randomisierung die Garantie des Greedy Algorithmus nicht schlagen kann.

# Randomized Weighted Majority Algorithmus (RWM)

Sei  $\eta \in (0, \frac{1}{2}]$  ein passend gewählter Parameter.

## Randomized Weighted Majority (RWM) Algorithmus

Am Anfang sei  $w_i^1 = 1$  für jedes  $i \in [N]$ .

In jedem Schritt  $t$ ,

- ▶ sei  $W^t = \sum_{i=1}^N w_i^t$ ;
- ▶ wähle Experten  $i$  mit Wahrscheinlichkeit  $p_i^t = w_i^t / W^t$ ;
- ▶ setze  $w_i^{t+1} = w_i^t \cdot (1 - \eta)^{\ell_i^t}$ .

# Randomized Weighted Majority (RWM) Algorithmus

Satz (Littlestone, Warmuth, 1994)

*Der RWM Algorithmus garantiert für jede Folge von Verlusten aus  $[0, 1]$*

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

Wenn wir  $\eta = \sqrt{\frac{\ln N}{T}}$  wählen, dann ergibt sich

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

# Randomized Weighted Majority (RWM) Algorithmus

Der *Regret* eines Lernalgorithmus  $H$  wird definiert als  $R(T) = L_H^T - L_{min}^T$ .

## Korollar

Der RWM Algorithmus mit  $\eta = \sqrt{\frac{\ln N}{T}}$  hat einen Regret von höchstens  $2\sqrt{T \ln N}$ .

Der *durchschnittliche Regret pro Zeitschritt* ist daher nur  $2\sqrt{\frac{\ln N}{T}}$ .

Beachte: Dieser Term geht gegen 0 für wachsendes  $T$ .

Algorithmen mit dieser Eigenschaft heißen

*No-Regret Lernalgorithmen.*

Im Gegensatz zum einfachen Greedy ist RWM ein No-Regret Algorithmus.

# Randomized Weighted Majority (RWM) Algorithmus

## Beweis des Satzes:

- ▶ Wir analysieren, wie sich die Summe der Gewichte  $W^t$  über die Zeit verringert. Es gilt

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

- ▶ Beachte:  $(1 - \eta)^\ell = (1 - \ell\eta)$  in beiden Fällen  $\ell = 0$  und  $\ell = 1$ .
- ▶ Daneben ist  $(1 - \eta)^\ell$  eine konvexe Funktion in  $\ell$ .
- ▶ Für  $\ell \in [0, 1]$  ergibt dies  $(1 - \eta)^\ell \leq (1 - \ell\eta)$ .
- ▶ Daraus folgt:

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

# Randomized Weighted Majority (RWM) Algorithmus

- ▶ Sei  $F^t$  der erwartete Verlust von RWM in Zeitschritt  $t$ .
- ▶ Es gilt  $F^t = \sum_{i=1}^N \ell_i^t w_i^t / W^t$ .
- ▶ Einsetzen in die Schranke für  $W^{t+1}$  ergibt

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

- ▶ Daraus folgt

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

- ▶ Die Summe der Gewichte nach Schritt  $T$  kann also nach oben beschränkt werden durch den erwarteten Verlust von RWM.

# Randomized Weighted Majority (RWM) Algorithmus

- ▶ Andererseits kann die Summe der Gewichte nach Schritt  $T$  auch nach unten beschränkt werden durch den Verlust des besten Experten:

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

- ▶ Kombination der beiden Schranken und Logarithmieren auf beiden Seiten führt zu

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

- ▶ Wir vereinfachen den Ausdruck mit Hilfe der folgenden Abschätzung

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

die für jedes  $z \in [0, \frac{1}{2}]$  gilt.

# Randomized Weighted Majority (RWM) Algorithmus

- ▶ Diese Vereinfachung ergibt

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

- ▶ Zum Schluß lösen wir die Ungleichung nun nur noch nach  $L_{RWM}^T$  auf:

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





# Erlernen von Gleichgewichten in Spielen

Regret-Minimierung ist ein natürlicher Ansatz für Verhalten bei wiederholten Entscheidungen mit unvollständiger Information.

Wir betrachten Regret-Lernen in einem Spiel  $\Gamma = (\mathcal{N}, (\Sigma_i)_{i \in \mathcal{N}}, (c_i)_{i \in \mathcal{N}})$ , das  $T$  Runden lang wiederholt hintereinander gespielt wird. (sog. *wiederholtes Spiel*).

Anfangs kennt keiner der Spieler  $i \in \mathcal{N}$  das Spiel. In jeder Runde  $t$  wählt jeder Spieler  $i$  eine reine Strategie  $s_i^t \in \Sigma_i$  mit seinem eigenen No-Regret Algorithmus. Der Algorithmus von  $i$  benutzt nur die *beobachteten Kosten von  $i$  in den vorherigen Runden*.

Wird dieses System zu einem (approx.) Nash-Gleichgewicht konvergieren?

Dies wäre eine tolle und sehr plausible Erklärung, wie Nash-Gleichgewichte in der Praxis entstehen können. Leider ist im Allgemeinen die Antwort: Nein.

# Gleichgewichte und Lernen

Allerdings verschwindet für jeden Spieler der durchschnittliche Regret über die Zeit. Aus dieser Eigenschaft kann man ein (allgemeineres) Gleichgewichtskonzept ableiten.

## Definition

Sei  $\mathcal{V}$  eine Wahrscheinlichkeitsverteilung über die Zustände eines endlichen Spiels.  $\mathcal{V}$  heißt **grob-korreliertes Gleichgewicht** wenn für jeden Spieler  $i \in \mathcal{N}$  und jede Strategie  $s'_i \in S_i$  gilt

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

$\mathcal{V}$  heißt **(additives)  $\varepsilon$ -approximatives grob-korreliertes Gleichgewicht** wenn

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

## No-Regret und Grob-Korrelierte Gleichgewichte

Betrachte einen Ablauf des Spiels  $s^1, s^2, \dots, s^T$  über  $T$  Runden. Wir interpretieren ihn als Verteilung über Zustände, in dem wir  $k \in [T]$  uniform zufällig wählen.

Wenn Spieler  $i$  einen Regret von  $R_i(T)$  hat, dann gilt für jede Strategie  $s'_i \in S_i$

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

### Proposition

*Wenn jeder Spieler nach  $T$  Runden Regret höchstens  $R$  hat, dann stellt der Ablauf des Spiels ein  $\frac{R}{T}$ -approximatives grob-korreliertes Gleichgewicht dar.*

Wenn alle Spieler RWM nutzen, dann wird schon nach  $T = \frac{4}{\varepsilon^2} \cdot \log(\max_i |S_i|)$  Runden ein  $\varepsilon$ -approximatives grob-korreliertes Gleichgewicht erreicht.

Imitation von Experten

No-Regret Algorithmen und grob-korrelierte Gleichgewichte

**Nullsummenspiele**

Korrelierte Gleichgewichte

## Erinnerung: Nullsummenspiele mit 2 Spielern

- ▶ In einem **Nullsummenspiel mit 2 Spielern** gilt für die beiden Spieler  $c_I(s) + c_{II}(s) = 0$  in jedem Zustand  $s$  des Spiels.
- ▶ Matrix  $A$  mit  $|\Sigma_I|$  Zeilen und  $|\Sigma_{II}|$  Spalten.  
Spieler I ist Zeilenspieler, Spieler II ist Spaltenspieler.
- ▶  $a_{ij}$  ist **Nutzenwert von Spieler I** in Zustand  $(i, j)$ ,  
 $a_{ij}$  ist **Kostenwert oder Verlust von Spieler II** in Zustand  $(i, j)$ .
- ▶ Wir normalisieren  $A$  so dass  $a_{ij} \in [0, 1]$ :

$A$  wird nicht-negativ wenn wir  $\max |a_{ij}|$  zu jedem Eintrag hinzuaddieren.  
Dann teilen wir durch den entstandenen höchsten Eintrag. Damit erhalten wir alle Einträge  $a_{ij}$  aus dem Intervall  $[0, 1]$ .

Beachte: Diese Anpassung hat keine Auswirkung auf die optimalen Strategien (und damit die Nash-Gleichgewichte) des Spiels.

## Beispiele

Matching Pennies  
(normalisiert)

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

Stein-Papier-Schere  
(normalisiert)

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

Ein Spiel mit  
 $|\Sigma_I| \neq |\Sigma_{II}|$ :

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

# Maximin-Strategien

- ▶ **Gain-Floor** für Spieler I:  $v_I^* = \max_x \min_y x^T Ay$ .  
**Optimale Strategie**  $x^*$  garantiert Gain-Floor für I (**Maximin-Strategie**).
- ▶ **Loss-Ceiling** für Spieler II:  $v_{II}^* = \min_y \max_x x^T Ay$ .  
**Optimale Strategie**  $y^*$  garantiert Loss-Ceiling für II (**Minimax-Strategie**).

## Lemma

*Es gilt:  $v_I^* \leq v_{II}^*$ .*

## Satz (Minimax Theorem)

*In jedem Nullsummenspiel mit 2 Spielern gilt  $v = v_I^* = v_{II}^*$ .*

# Gemischte Nash-Gleichgewichte

## Korollar

*Zustand  $(x, y)$  im Nullsummenspiel mit 2 Spielern ist ein gemischtes Nash-Gleichgewicht*

$\Leftrightarrow$

*$x$  und  $y$  sind optimale Strategien.*

## Korollar

*Jedes Nullsummenspiel mit 2 Spielern hat mindestens ein gemischtes Nash-Gleichgewicht, und alle gemischten Nash-Gleichgewichte ergeben den gleichen erwarteten Nutzenwert für Spieler I.*

## Satz

*In Nullsummenspielen mit 2 Spielern kann ein gemischtes Nash-Gleichgewicht in polynomieller Zeit berechnet werden.*



# Lernen in Nullsummenspielen

- ▶ Spieler kennen das Spiel nicht, in dem sie spielen. Sie verwenden No-Regret Algorithmen für die Wahl der Strategie.

Können die Spieler optimale Strategien und somit ein gemischtes Nash-Gleichgewicht **erlernen?**

# Lernen in Nullsummenspielen

- ▶ Spieler kennen das Spiel nicht, in dem sie spielen. Sie verwenden No-Regret Algorithmen für die Wahl der Strategie.

Können die Spieler optimale Strategien und somit ein gemischtes Nash-Gleichgewicht **erlernen?**

- ▶ Betrachte Spieler II. Experten sind reine Strategien, Gegner ist Spieler I.
- ▶ In jedem Schritt  $t$  wählt der Lernalgorithmus  $H$  von Spieler II eine gemischte Strategie  $y^t$  gegen die unbekannte Strategie  $x^t$  des Gegenspielers I.

# Lernen in Nullsummenspielen

- ▶ Spieler kennen das Spiel nicht, in dem sie spielen. Sie verwenden No-Regret Algorithmen für die Wahl der Strategie.

Können die Spieler optimale Strategien und somit ein gemischtes Nash-Gleichgewicht **erlernen?**

- ▶ Betrachte Spieler II. Experten sind reine Strategien, Gegner ist Spieler I.
- ▶ In jedem Schritt  $t$  wählt der Lernalgorithmus  $H$  von Spieler II eine gemischte Strategie  $y^t$  gegen die unbekannte Strategie  $x^t$  des Gegenspielers I.
- ▶ Verlust in Schritt  $t$  für Strategie (Experte)  $i$  beträgt

$$\ell_i^t = \sum_{j \in \Sigma_I} x_j^t a_{ji} .$$

- ▶ Gesamter Verlust des Lernalgorithmus  $H$  in Schritt  $t$  beträgt

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

# No-Regret und optimale Strategien

- ▶ No-Regret Lernalgorithmus  $H$ :

$$\frac{L_H^T - L_{min}^T}{T} \rightarrow 0 \quad \text{für } T \rightarrow \infty .$$

# No-Regret und optimale Strategien

- ▶ No-Regret Lernalgorithmus  $H$ :

$$\frac{L_H^T - L_{min}^T}{T} \longrightarrow 0 \quad \text{für } T \rightarrow \infty .$$

- ▶ Nach Definition: Der durchschnittliche Verlust pro Schritt eines No-Regret Algorithmus wird so klein wie der durchschnittliche Verlust der (im Nachhinein nach  $t$  Schritten) **besten reinen Strategie**.
- ▶ Wird der durchschnittliche Verlust  $L_H^T/T$  so klein wie der Wert des Spiels?

# No-Regret und optimale Strategien

- ▶ No-Regret Lernalgorithmus  $H$ :

$$\frac{L_H^T - L_{min}^T}{T} \rightarrow 0 \quad \text{für } T \rightarrow \infty .$$

- ▶ Nach Definition: Der durchschnittliche Verlust pro Schritt eines No-Regret Algorithmus wird so klein wie der durchschnittliche Verlust der (im Nachhinein nach  $t$  Schritten) **besten reinen Strategie**.
- ▶ Wird der durchschnittliche Verlust  $L_H^T/T$  so klein wie der Wert des Spiels?

## Satz

*Betrachte ein wiederholtes Nullsummenspiel mit 2 Spielern. Wenn Spieler II über die  $T$  Zeitschritte einen Algorithmus  $H$  mit Regret  $R$  für seine Strategiewahl nutzt, dann beträgt sein durchschnittlicher Verlust*

$$\frac{L_H^T}{T} \leq v_I^* + \frac{R}{T} .$$

*Dies Resultat gilt analog für Spieler I und die Loss-Ceiling.*

# Erlernen des Gain-Floors

## Beweis:

- ▶ Wir zeigen, dass die beste Strategie am Ende der  $T$  Schritte die Gesamtkosten von höchstens  $L_{min}^T \leq T \cdot v_I^*$  hat.
- ▶ Betrachte den Ablauf der Strategiewahlen des Gegenspielers I, also Strategien  $x^1, x^2, \dots, x^T$ . Addiere sie zu einer "durchschnittlichen Strategie"

$$\hat{x}_j = \frac{1}{T} \sum_{t=1}^T x_j^t \quad \text{für jedes } j \in \Sigma_I$$

- ▶ Gesamter Verlust  $L_i^T$  einer reinen Strategie  $i \in \Sigma_{II}$  bleibt gleich, wenn Spieler I in allen Zeitschritten immer nur  $\hat{x}$  gespielt hätte:

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

## Erlernen des Gain-Floors

- ▶ Nehmen wir also an, I spielt immer nur  $\hat{x}$ . Betrachte die im Nachhinein beste reine Strategie für II. Damit ergibt sich nun ein Spiel mit nur noch einem Zeitschritt.
- ▶ In diesem Einschrittspiel muss Spieler I zuerst die durchschnittliche Strategie  $\hat{x}$  wählen. Dann wählt Spieler II eine (reine) beste Antwort  $\hat{i}$  – also muss I zuerst wählen, dann antwortet II.
- ▶ Per Definition des Gain-Floors gibt es immer  $i \in \Sigma_{II}$  so dass der Nutzen von I (und der Verlust von II) höchstens  $v_I^*$  beträgt, d.h.,  $c_{II}(\hat{x}, i) \leq v_I^*$ .
- ▶ Also gibt es eine reine Strategie  $i \in \Sigma_{II}$  mit

$$L_{min}^T \leq L_i^T \leq T \cdot v_I^* .$$

- ▶ Zusammen ergeben die beiden Einsichten:

$$L_H^T \leq L_{min}^T + R \leq T \cdot v_I^* + R .$$





# Ein einfacher Beweis des Minimax-Theorems

## Satz (Minimax Theorem)

In jedem Nullsummenspiel mit 2 Spielern gilt  $v = v_I^* = v_{II}^*$ .

### Beweis:

- ▶ Zum Widerspruch nehmen wir an  $v_I^* + \gamma = v_{II}^*$  für ein  $\gamma > 0$ .
- ▶ Beide Spieler spielen nun das Spiel wiederholt mit einem Lernalgorithmus. Sei  $T$  groß genug, so dass der Regret der Algorithmen  $R/T < \gamma/3$ .
- ▶ Über die Gesamtzeit der  $T$  Runden können wir wie oben zeigen  $L_{min}^T \leq v_I^*$  für Spieler II, und  $L_{min}^T \leq -v_{II}^*$  für Spieler I (“-“ da Verlust).
- ▶ Dies bedeutet, dass die Algorithmen höchstens  $v_I + \gamma/3$  durchschnittliche Kosten für Spieler II und mindestens  $v_{II} - \gamma/3$  durchschnittlichen Nutzen für Spieler I garantieren.
- ▶ Durchschnittliche Kosten für II sind durchschnittlicher Nutzen für I  
→ Widerspruch. □

# Konvergenz

## Korollar

*Wenn beide Spieler einen No-Regret Lernalgorithmus nutzen, konvergiert der durchschnittliche Ablauf des Spiels  $(\hat{x}, \hat{y})$  zu optimalen Strategien und damit zu einem gemischten Nash-Gleichgewicht des Spiels.*

Der Satz garantiert Konvergenz nur für den **durchschnittlichen Ablauf des Spiels**, aber nicht für das **eigentliche Verhalten** in den Strategien  $x^t$  und  $y^t$ !

## Satz

*Es gibt No-Regret Lernalgorithmen für Spieler I und II mit denen die Strategien  $x^t$  und  $y^t$  nicht zu optimalen Strategien konvergieren.*

# Konvergenz für allgemeine No-Regret Algorithmen

Matching Pennies (normalisiert)

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

**Beweis:** Ein "seltsamer" No-Regret Algorithmus  $H$ :

- ▶ Regel: I und II wählen reine Strategien, I wechselt zur anderen reinen Strategie in Runde 1, 3, 5, ..., II wechselt zur anderen reinen Strategie in Runde 2, 4, 6,...
- ▶ Wenn ein Spieler von der Regel abweicht, ruft der andere Spieler den RWM-Algorithmus auf. (Dieser Trick sichert die No-Regret-Eigenschaft für **jeden möglichen** Ablauf).
- ▶  $L_i^T/T \rightarrow 0.5$  für beide Strategien  $i = 1, 2$  von II, durchschnittlicher Verlust des Algorithmus  $L_H^T/T \rightarrow 0.5$ . **No-Regret Algorithmus für II!** (gleiches Argument für I).
- ▶ Keine der Verteilungen  $y^t$  ist nahe an der optimalen Strategie  $(0.5, 0.5)$ , kein Verlust in einer einzelnen Runde  $\ell_H^t$  ist nahe am Wert  $v = 0.5$ . □

# Ein adaptiver RWM Algorithmus

(Freund, Schapire, 1999)

Sei  $\eta_0 \in (0, \frac{1}{2}]$  und  $u$  eine obere Schranke auf den Wert des Spiels.

## Variabler Randomized Weighted Majority (vRWM) Algorithmus

Initialisiere  $w_i^1 = 1$  für alle  $i \in [N]$ .

In jedem Schritt  $t$ ,

- ▶ sei  $W^t = \sum_{i=1}^N w_i^t$ ;
- ▶ wähle Experte  $i$  mit W.keit  $p_i^t = w_i^t / W^t$ ;
- ▶ **if**  $\ell_{vRWM}^t \leq u$  **then** setze  $w_i^{t+1} = w_i^t$ ;
- ▶ **else** setze

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

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

# Vergleich von Verteilungen

## Definition

Die **Kullback-Leibler Divergenz** oder **relative Entropie** von zwei Verteilungen  $y$  und  $y'$  ist gegeben durch

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

Zum Vergleich von  $a, b \in [0, 1]$  nutzen wir die Verteilungen  $(a, 1 - a)$  und  $(b, 1 - b)$ :

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

Die Kullback-Leibler Divergenz für Verteilungen ist **immer nicht-negativ** und  **$RE(y \parallel y') = 0$  genau dann wenn  $y = y'$** .

# Konvergenz von vRWM

## Satz

Sei  $y'$  eine beliebige gemischte Strategie für II, die einen Verlust von höchstens  $u$  garantiert gegen jede beste Antwort von I. In jeder Iteration  $t$  von vRWM, in der  $\ell_{vRWM}^t \geq u$ , sinkt die relative Entropie zwischen  $y'$  und  $y^{t+1}$  um

$$RE(y' \parallel y^{t+1}) \leq RE(y' \parallel y^t) - RE(u \parallel \ell_{vRWM}^t) .$$

In jedem Schritt, in dem der Verlust von vRWM zu hoch ist, bewegt die Anpassung die neue Strategie näher an jede gute Strategie.

## Beweis des Satzes

**Beweis:**

Zur Vollständigkeit beweisen wir den Satz. Betrachte einen Schritt  $t$  mit  $\ell_{vRW M}^t > u$ . Dann gilt

$$\begin{aligned}
 & RE(y' \parallel y^{t+1}) - RE(y' \parallel y^t) \\
 = & \sum_{i \in \Sigma_{II}} y'_i \ln \frac{y'_i}{y_i^{t+1}} - \sum_{i \in \Sigma_{II}} y'_i \ln \frac{y'_i}{y_i^t} \\
 = & \sum_{i \in \Sigma_{II}} y'_i \ln \frac{y_i^t}{y_i^{t+1}} \\
 \leq & \sum_{i \in \Sigma_{II}} y'_i \ln \frac{1 - \eta_t \ell_{vRW M}^t}{(1 - \eta_t)^{\ell_i^t}} .
 \end{aligned}$$

Dabei nutzen wir, dass  $y_i^{t+1} = w_i^t (1 - \eta_t)^{\ell_i^t} / W^{t+1}$  und  $W^{t+1} \leq W^t (1 - \eta_t F^t) = W^t (1 - \eta_t \ell_{vRW M}^t)$  wie oben gesehen.

## Beweis des Satzes

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

weil Strategie  $y'$  nie mehr als  $u$  an Verlust erzeugt.



# Beweis des Satzes

Die Ableitung von

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

nach  $\eta_t$  setzen wir gleich 0. Damit erhalten wir das Minimum

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

wie gewünscht. Einsetzen in der Formel ergibt

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



# Konvergenz von vRWM

## Korollar

Für jede Folge von Strategien  $x^1, x^2, \dots$  beträgt die Anzahl der Runden mit Verlust  $\ell_{vRWM}^t \geq u + \varepsilon$  höchstens

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

Für ein festes  $\varepsilon$  ist diese Zeit unabhängig von  $T$ . Für spätere Zeitschritte  $t$  muss daher der Verlust immer näher an  $u$  herankommen. Dies Verhalten ist viel besser als z.B. beim seltsamen No-Regret Algorithmus, der einen Verlust von 1 in jeder zweiten Runde generiert, sogar für beliebig späte Zeitschritte  $t$ .

Imitation von Experten

No-Regret Algorithmen und grob-korrelierte Gleichgewichte

Nullsummenspiele

Korrelierte Gleichgewichte

# Korrelierte Gleichgewichte

Im Allgemeinen konvergieren No-Regret-Algorithmen zu einem (approximativen) grob-korrelierten Gleichgewicht. Diese Gleichgewichte stellen ein deutlich schwächeres Stabilitätskonzept dar als z.B. gemischte Nash-Gleichgewichte. Sind grob-korrelierte Gleichgewichte das stärkste Konzept, das schnell approximiert werden kann?

## Definition

Sei  $\mathcal{V}$  eine Wahrscheinlichkeitsverteilung über die Zustände eines endlichen Spiels.  $\mathcal{V}$  heißt **korreliertes Gleichgewicht** wenn für jeden Spieler  $i \in \mathcal{N}$  und jede Austauschfunktion  $\sigma : S_i \rightarrow S_i$  gilt

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

$\mathcal{V}$  heißt **(additives)  $\varepsilon$ -approximatives korreliertes Gleichgewicht** wenn

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

## Korrelierte Gleichgewichte

Im gemischten Gleichgewicht wählt jeder Spieler eine **eigene Verteilung  $x_i$  über seine Strategien**. Die Verteilung über Zustände ergibt sich erst aus der **unabhängigen Kombination** der  $x_i$ . Keiner der Spieler möchte zu einer anderen (reinen) Strategie abweichen.

In korrelierten und grob-korrelierten Gleichgewichten wird **direkt eine Verteilung über Zustände** angegeben, die die Strategiewahlen einzelner Spieler miteinander korrelieren kann (z.B. als Resultat von gemeinsamem Lernen über die Zeit).

Keiner der Spieler möchte zu einer anderen (reinen) Strategie abweichen:

- ▶ Korreliert: **Unter der Bedingung, dass für Spieler  $i$  Strategie  $s_i$  ausgewürfelt wurde**, möchte  $i$  nicht zu einer anderen reinen Strategie abweichen.
- ▶ Grob-korreliert: Keiner der Spieler möchte **immer** zu einer reinen Strategie abweichen (hier wird nicht auf eine ausgewürfelte Strategie  $s_i$  bedingt).

# Beispiel

Zwei Autofahrer rasen auf eine Kreuzung zu.

	(G)as	(B)remsen
(G)as	101	2
(B)remsen	0	1

Beispiele für Gleichgewichte:

- ▶ Rein: Zustände (G,B) und (B,G)
- ▶ Gemischt (aber nicht rein): Beide Spieler spielen  $x_G = 0.01$  und  $x_B = 0.99$ .
- ▶ Korreliert (aber nicht gemischt):  
 $\Pr_{s \sim \mathcal{V}}[s = (B, G)] = 0.5$  und  
 $\Pr_{s \sim \mathcal{V}}[s = (G, B)] = 0.5$

# Beispiel

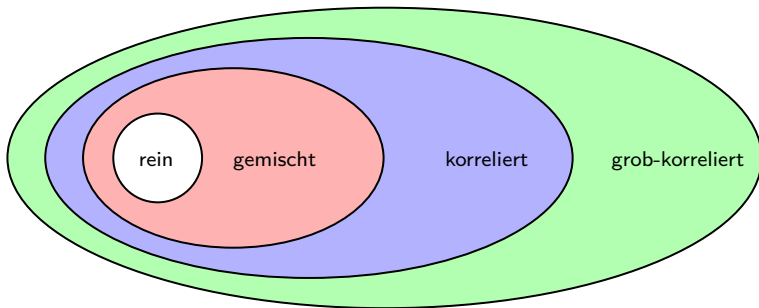
Zwei Autofahrer rasen auf eine Kreuzung zu.

	(G)as	(B)remsen
(G)as	101	2
(B)remsen	0	1

Beispiele für Gleichgewichte:

- ▶ Man kann sich das korrelierte Gleichgewicht vorstellen wie eine **Ampel**: Jedem Spieler wird ein Vorschlag gemacht. Kein Spieler will unilateral die empfohlene Strategie ändern, wenn der andere die Empfehlung befolgt.
- ▶ Grob-Korreliert (aber nicht korreliert): In symmetrischen  $2 \times 2$ -Spiele nicht möglich (jedes grob-korrelierte Gleichgewicht ist korreliert), siehe Übung.

# Eine Hierarchie von Gleichgewichtskonzepten





# Erlernen von Korrelierten Gleichgewichten

## Können wir korrelierte Gleichgewichte erlernen?

Dafür betrachten wir Algorithmen mit einer stärkeren Garantie im Experten Problem. Für Algorithmus  $H$  sei  $H(t) \in [N]$  der in Schritt  $t$  gewählte Experte. Bisher haben wir den Verlust des **besten einzelnen Experten** als Vergleichswert für Regret betrachtet. Jetzt betrachten wir die **beste Funktion von Experten**

$$\sigma^* \in \arg \min_{\sigma: [N] \rightarrow [N]} \sum_{t=1}^T \ell_{\sigma(H(t))}^t ,$$

mit  $L_{s \min}^T = \sum_{t=1}^T \ell_{\sigma^*(H(t))}^t$  dem Verlust der besten Funktion von Experten.

# Swap-Regret

## Definition

Der **Swap-Regret** von  $H$  ist gegeben durch

$$SR(T) = L_H^T - L_{s \min}^T = \sum_{t=1}^t \ell_H^t - \sum_{t=1}^t \ell_{\sigma^*(H(t))}^t .$$

Swap-Regret beschreibt den Verlust gegenüber Austauschen von Experten.  
Wenn  $\sigma^*(i) = j$ , dann bedeutet dies:

Wann immer  $H$  Experte  $i$  gewählt hat, hätte er lieber Experte  $j$  wählen sollen.

Für einen **No-Swap-Regret Algorithmus** gilt  $SR(T)/T \rightarrow 0$  für  $T \rightarrow \infty$ .

Es gilt  $SR(T) \geq R(T)$  und No-Swap-Regret  $\Rightarrow$  No-Regret. (Warum?)

# Swap-Regret und korreliertes Gleichgewicht

Betrachte einen Ablauf des Spiels  $s^1, s^2, \dots, s^T$  über  $T$  Runden und interpretiere ihn als Verteilung über Zustände.

Wenn Spieler  $i$  einen Swap-Regret von  $SR_i(T)$  hat, dann gilt für jede Strategie  $s_i$  und jede Funktion  $\sigma : S_i \rightarrow S_i$ :

$$\begin{aligned} \mathbb{E}_{k \in [T]} [c_i(s^k)] &= \sum_{t=1}^T \frac{1}{T} \cdot c_i(s_i^t, s_{-i}^t) \\ &\leq \sum_{t=1}^T \frac{1}{T} \cdot c_i(\sigma(s_i^t), s_{-i}^t) + \frac{SR_i(T)}{T} \\ &= \mathbb{E}_{k \in [T]} [c_i(\sigma(s_i^k), s_{-i}^k)] + \frac{SR_i(T)}{T}. \end{aligned}$$

## Proposition

Wenn jeder Spieler nach  $T$  Runden Swap-Regret höchstens  $R$  hat, dann stellt der Ablauf des Spiels ein  $\frac{R}{T}$ -approximatives korreliertes Gleichgewicht dar.

# No-Swap-Regret Algorithmen

Wir können No-Regret Algorithmen in No-Swap-Regret Algorithmen verwandeln!

## Satz

Sei  $H$  ein Algorithmus mit Regret  $R(T)$ . Dann existiert ein Algorithmus  $M$  mit Swap-Regret  $SR(T) = N \cdot R(T)$ .

## Beweis:

Wir nutzen  $N$  Kopien  $H_1, \dots, H_N$  von Algorithmus  $H$ . In gewisser Weise sichert uns  $H_j$  zu, dass keine Abbildung  $\sigma$  von Experte  $j$  auf  $\sigma(j)$  allzu großen Swap-Regret erzeugt.

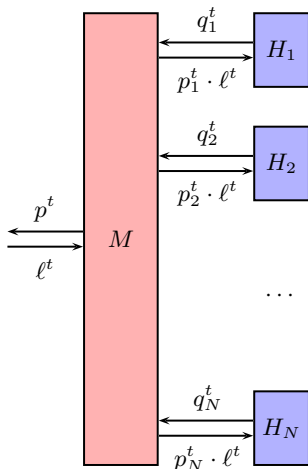
Ein **Master-Algorithmus**  $M$  koordiniert die Kopien von  $H$ .

# Master-Algorithmus

## Master-Algorithmus

In jedem Schritt  $t$ ,

1. Erhalte Verteilungen  $q_1^t, \dots, q_n^t$  von  $H_1, \dots, H_N$ ;
2. Errechne Konsens-Verteilung  $p^t$ ;
3. Wähle Experte  $i$  mit W.keit  $p_i^t$ ;
4. Melde an  $H_j$  einen Verlust von  $p_j^t \cdot \ell_i^t$ , für jeden Experten  $i \in N$ ;



## Master-Algorithmus: No-Swap-Regret

Wir betrachten die Errechnung von  $p^t$  in Zeile 2. am Ende des Beweises.

Für Swap-Regret betrachten wir die folgenden Terme:

- ▶ Verlust des Master-Algorithmus: 
$$L_M^T = \sum_{t=1}^T \sum_{i \in [N]} p_i^t \ell_i^t$$
- ▶ Verlust mit bester Abbildung  $\sigma^*$ : 
$$L_{s \min}^T = \sum_{t=1}^T \sum_{i \in [N]} p_i^t \ell_{\sigma^*(i)}^t$$

Algorithmus  $H_j$  denkt, wir wählen Experte  $i$  mit Wahrscheinlichkeit  $q_{j,i}^t$  und die Kosten dafür wären  $p_j^t \ell_i^t$ .

- ▶ Interner Verlust von  $H_j$ : 
$$L_{H_j}^T = \sum_{t=1}^T \sum_{i \in [N]} q_{j,i}^t \cdot (p_j^t \ell_i^t)$$
- ▶ Bester Experte  $k_j$  für  $H_j$ : 
$$L_{\min,j}^T = \sum_{t=1}^T p_{k_j}^t \ell_{k_j}^t$$

$H_j$  hat Regret höchstens  $R(T)$ , also  $L_{H_j}^T \leq L_{\min,j}^T + R(T)$ .

# Master-Algorithmus: No-Swap-Regret

Sei nun  $\sigma^*$  die beste Abbildung der Experten. Dann gilt mit Aufsummieren:

$$\begin{aligned}
 \sum_{j \in [N]} L_{H_j}^T &\leq \sum_{j \in [N]} L_{\min, j}^T + R(T) \\
 &\leq \sum_{j \in [N]} \sum_{t=1}^T p_j^t \ell_{\sigma^*(j)}^t + \sum_{j \in [N]} R(T) \\
 &= L_{s \min}^T + N \cdot R(T)
 \end{aligned}$$

Für die Summe der internen Verluste haben wir die No-Swap-Regret Garantie.

Wie hängt der tatsächliche Verlust  $L_M^T$  mit der Summe der internen Verluste  $\sum_j L_{H_j}^T$  zusammen?

Dafür betrachten wir nun die Konstruktion der Konsens-Verteilung  $p^t$ .

# Master-Algorithmus: Konsens-Verteilung

Wähle die Konsens-Verteilung  $p^t$  als

$$p_i^t = \sum_{j \in [N]} q_{j,i}^t p_j^t .$$

Dann gilt:

$$\begin{aligned} L_M^T &= \sum_{t=1}^T \sum_{i \in [N]} p_i^t \ell_i^t \\ &= \sum_{t=1}^T \sum_{i \in [N]} \sum_{j \in [N]} q_{j,i}^t p_j^t \ell_i^t \\ &= \sum_{j \in [N]} L_{H_j}^T \\ &\leq L_{s \min}^T + N \cdot R(T) , \end{aligned}$$

und der Satz ist gezeigt. □

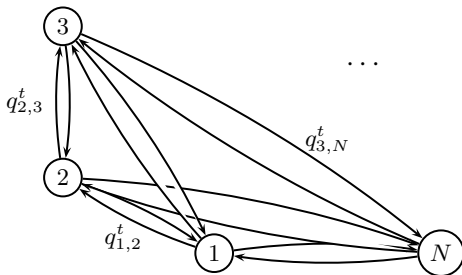


# Master-Algorithmus: Markov-Kette

$p^t$  ist die **stationäre Verteilung** der folgenden Markov-Kette:

Die Zustände sind alle Experten. Wenn wir bei Zustand/Experte  $i$  sind, dann ist die Wahrscheinlichkeit zu Zustand/Experte  $j$  zu wechseln gegeben durch  $q_{i,j}^t$ . Matrix  $Q^t$  mit  $Q^t(i,j) = q_{i,j}^t$  gibt die Übergangswahrscheinlichkeiten von Zustand/Experte  $i$  an.

Wähle  $p^t$  als die stationäre Verteilung der Markov-Kette, der dominante Eigenvektor der Matrix  $Q^t$ .  $p^t$  kann effizient berechnet werden.



# Übersicht: Gleichgewichte, Berechnung und Konvergenz

## Reines Nash-Gleichgewicht

- ▶ Existiert nicht immer, aber bestimmten Spielen (z.B. in Potenzialspielen)
- ▶ Berechnung polynomiell in der Kostenmatrix des Spiels (die ist riesig)
- ▶ PLS-schwer in kompakt repräsentierten Auslastungsspielen
- ▶ Konvergenzzeit von Beste-Antwort-Dynamik exponentiell
- ▶ Polynomielle Konvergenzzeit für spezielle Potenzialspiele

## Gemischtes Nash-Gleichgewicht

- ▶ Existiert in endlichen Spielen (Satz von Nash)
- ▶ Berechnung in PPAD (Sperners Lemma) und PPAD-schwer
- ▶ 2-Spieler-Nullsumme: Berechnung in polynomieller Zeit mit LP
- ▶ 2-Spieler-Nullsumme: Konvergenz in polynomieller Zeit mit No-Regret
- ▶ 2-Spieler-Nullsumme: Konvergenz im Ablauf mit vRWM-Algorithmus

# Übersicht: Gleichgewichte, Berechnung und Konvergenz

## Korreliertes Gleichgewicht

- ▶ Existiert in endlichen Spielen (verallgemeinert gemischtes Ggw.)
- ▶ Verteilung über Zustände ohne profitable "Swap"-Abweichungen
- ▶ Konvergenz in polynomialer Zeit mit No-Swap-Regret

## Grob-korreliertes Gleichgewicht

- ▶ Existiert in endlichen Spielen (verallgemeinert korreliertes Ggw.)
- ▶ Verteilung über Zustände ohne profitable "bester-Experte"-Abweichungen
- ▶ Konvergenz in polynomialer Zeit mit No-Regret

Beachte: Konvergenz für No-Regret oder No-Swap-Regret bezieht sich auf  $\epsilon$ -**approximative** Varianten der Gleichgewichte.  $\epsilon$  sinkt als Funktion von  $T$ .

# Literatur

- ▶ Nisan, Roughgarden, Tardos, Vazirani. Algorithmic Game Theory, 2007. (Kapitel 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. (Kapitel 17+18).
- ▶ Blum, Mansour. From External to Internat Regret. Journal of Machine Learning Research 8:1307–1324, 2007.