Regret Minimization

Alexander Kauer
Freie Universität Berlin

05.11.2013

Überblick

Allgemeines
- Motivation
- Definitionen

External Regret Minimization
- Allgemeines
- Algorithmen zur External Regret Minimization
- Das Partial Information Model

Swap Regret Minimization
- Definitionen
- Reduktion von External Regret zu Swap Regret

Motivation

- Spiele können in unsicheren oder unbekannten Umgebungen stattfinden:
  - Keine Informationen über die Strategien der anderen Spieler gegeben.
  - Keine Informationen über mögliche Gewinne oder Verluste gegeben.
- Es werden Online-Algorithmen benötigt, die trotz dieser Einschränkungen möglichst gute Ergebnisse erzielen.

Definitionen

- Menge der verfügbaren Aktionen des Spielers: $X = \{1, 2, \ldots, N\}$.
- Online-Algorithmus H wählt zum Zeitpunkt t eine Wahrscheinlichkeitsverteilung p^t über X (sprich: die Strategie für den Spieler).
- Verlustvektor $l^t \in [0,1]^N$ wird anschließend ermittelt und H erfährt Verlust $l_H^t = \sum_{i=1}^{N} p_i^t l_i^t$.
- Gesamtverlust bis Zeitpunkt T der i-ten Aktion ist $L_i^T = \sum_{t=1}^{T} l_i^t$.
- Gesamtverlust von H bis Zeitpunkt T ist $L_H^T = \sum_{i=1}^{T} l_H^t$

Definitionen

- Je nach *Information Model* erhält H eine unterschiedliche Menge an Informationen:
  - Im *Full Information Model* erhält H den kompletten Verlustvektor l^t und erfährt einen Verlust von $l_H^t = \sum_{i=1}^{N} p_i^t l_i^t$.
  - Im *Partial Information Model* erhält H hingegen ausschließlich den gesamten Verlust l_H^t.
- Wenn nicht anderweitig vermerkt, dann wird das *Full Information Model* genutzt.

Allgemeines zur External Regret Minimization

- Gegeben: Menge an Algorithmen $\mathcal{G}$ für das Spiel.
  - Im einfachsten Fall Algorithmen, welche immer die gleiche Aktion aus X wählen.
- Ziel: Möglichst nah an den besten dieser Vergleichsalgorithmen bezüglich der Verlustminimierung zu kommen.
- Sei $L^T_{\mathcal{G},\min} = \min_{g \in \mathcal{G}} L^T_g$ der geringste Verlust, der von einem der Vergleichsalgorithmen erzeugt wird.
- Dann bezeichnet das External Regret $R_{\mathcal{G}} = L^T_H - L^T_{\mathcal{G},\min}$ den Abstand zum besten dieser Vergleichsalgorithmen, welchen es zu minimieren gilt.

Probleme mit $\mathcal{G}_{\text{all}}$

- Das erste Problem hierbei stellt schon die Wahl der Vergleichsalgorithmen dar:

- **Satz**

  Sei $\mathcal{G}_{all}$ die Menge aller möglichen Algorithmen bis Zeitpunkt T. Für jeden beliebigen Online-Algorithmus H gibt es bei der Ausführung T Verlustvektoren mit einem Regret von mindestens $R_{\mathcal{G}_{all}} = T(1 - \frac{1}{N})$.

- **Beweis.**

  Wähle l^t folgendermaßen: $l_i^t = \begin{cases} 0 & \text{wenn } i = \min\{1 \leq k \leq N \mid p_k^t = \min_j p_j^t\} \\ 1 & \text{sonst} \end{cases}$

  Da weiterhin gilt $\min_i\{p_i^t\} \leq \frac{1}{N}$, beträgt der Verlust von H pro Zeitpunkt t mindestens $1 - \frac{1}{N}$ und bis Zeitpunkt T folglicherweise $T(1 - \frac{1}{N})$. In $\mathcal{G}_{\text{all}}$ gibt es hingegen einen Algorithmus, welcher einen Verlust von insgesamt 0 erzielt. $\square$

- Aus diesem Grund wird im Folgenden von einer Menge $\mathcal{G}$ ausgegangen, dessen Algorithmen ihren Index in $\mathcal{G}$ als Aktion zu jedem Zeitpunkt wählen (mit $|\mathcal{G}| = N$).

External Regret Minimization: Greedy-Algorithmus

- Sei x^t der Index der zum Zeitpunkt t selektierten Aktion:

$$
\begin{array}{lll}
\text{Start:} & x^1 & := 1 \\
\text{Zeitpunkt } t\text{:} & S^{t-1} & := \{i : L_i^{t-1} = L_{\min}^{t-1}\} \\
 & x^t & := \min S^{t-1}
\end{array}
$$

- **Satz**

Für den Greedy-Algorithmus gilt für jede beliebige Reihenfolge an Verlusten

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

Beweis.

Zu jedem Zeitpunkt, an dem dieser Algorithmus einen Verlust (von bis zu 1) erhält und $L_{\min}$ sich nicht verändert, muss mindestens eine Aktion aus S^t entfernt werden. Dies kann maximal $|N|$ mal geschehen, bevor $L_{\min}^t$ erhöht werden muss. □

Probleme bei deterministischen Algorithmen

- Problem hierbei:

- **Satz**

Für jeden beliebigen deterministischen Algorithmus H existiert eine Verlustvektorsequenz, für welche sich $L_H^T = T$ und $L_{\min}^T \leq \lfloor \frac{T}{N} \rfloor$ ergeben.

Beweis.

Sei x^t die Aktion, welche der Algorithmus zu Zeitpunkt t auswählt. Setze l^t folgendermaßen:

$$l_i^t = \begin{cases} 1 & \text{wenn } i = x^t \\ 0 & \text{sonst} \end{cases}$$

Somit ist $L_H^T = T$. Da es N verschiedene Aktionen gibt, muss es mindestens eine Aktion geben, welche von H maximal $\lfloor \frac{T}{N} \rfloor$ ausgewählt wurde. Da ausschließlich die Aktionen von H einen Verlust bedeuten, ergibt sich $L_{\min}^T \leq \lfloor \frac{T}{N} \rfloor$. □

- Erweiterung des Greedy-Algorithmus mit Auswahl einer Wahrscheinlichkeitsverteilung p^t statt einer festen Aktion x^t:

  Start: $p_i^1 := \frac{1}{N}$

  Zeitpunkt t: $S^{t-1} := \{i : L_i^{t-1} = L_{\min}^{t-1}\}$

$$p_i^t := \begin{cases} \frac{1}{|S^{t-1}|} & \text{falls } i \in S^{t-1} \\ 0 & \text{sonst} \end{cases}$$

- **Satz**

Für den Randomized-Greedy-Algorithmus gilt für jede beliebige Reihenfolge an Verlusten

$$L_{RG}^T \leq (\ln N) + (1 + \ln N) L_{\min}^T$$

(ohne Beweis)

Polynomial-Weights-Algorithmus

- Einbeziehung aller Aktionen in die Auswahl über Gewichte w^t über eine Wahrscheinlichkeitsverteilung:

  Start: $w_i^1 := 1,\ p_i^1 := \frac{1}{N}, i \in X$

  Zeitpunkt t: $w_i^t := w_i^{t-1}(1 - \eta l_i^{t-1})$

  $p_i^t := \frac{w_i^t}{W^t}$ mit $W^t := \sum_{i \in X} w_i^t$

- **Satz**

  Für den Polynomial-Weights-Algorithmus gilt mit $\eta \leq 0.5$, jeder Verlustvektorsequenz und jedem $k \in X$

  $$L_{PW}^T \leq L_k^T + \eta Q_k^T + \frac{\ln N}{\eta}$$

  mit $Q_k^T = \sum_{t=1}^T (l_k^t)^2$. Setzt man $\eta = \min\{\sqrt{\frac{\ln N}{T}}, 0.5\}$, dann ergibt sich mit dem Wissen von $Q_k^T \leq T$ die Ungleichung $L_{PW}^T \leq L_{\min}^T + 2\sqrt{T \ln N}$.

Freie Universität Berlin

Beweis.

Sei $F^t = \frac{\sum_i w_i^t l_i^t}{W^t}$ der Verlust zum Zeitpunkt t. Mit dieser Definition und den Aktualisierungsregeln ergeben sich $W^{t+1} = W^t - \sum_i \eta w_i^t l_i^t = W^t(1-\eta F^t)$ und $W^{T+1} = N \prod_{t=1}^T (1-\eta F^t)$.

Obere Schranke:

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

Untere Schranke (mit $k \in X$):

$$\begin{aligned} \ln W^{T+1} \quad & \geq \quad \ln w_k^{T+1} = \sum_{t=1}^{T} \ln(1-\eta l_k^t) \\ & \overset{\ln(1-z)\geq -z-z^2, 0\leq z\leq 0.5)}{\geq} -\sum_{t=1}^{T} \eta l_k^t - \sum_{t=1}^{T} (\eta l_k^t)^2 = -\eta L_k^T - \eta^2 Q_k^T \end{aligned}$$

Kombiniert man die beiden Ungleichungen, ergibt sich die Formel aus dem Satz. □

Untere Grenzen für External Regret Minimization

- Der Polynomial-Weights-Algorithmus verhält sich bzgl. unterer Grenze nahezu optimal. Es gilt für jeden Algorithmus:
- Für kleine $T < \log_2 N$: $R \in \Omega(T)$
- Für alle T: $R \in \Omega(\sqrt{T})$

Untere Grenzen für External Regret Minimization

- **Satz**

  Sei $T < \log_2 N$. Dann gibt es für jeden Algorithmus A_1 eine Verlustvektorsequenz, wodurch sich zwingend $E[L^T_{A_1}] = \frac{T}{2}$ und $L_{\min} = 0$ ergeben.

- **Beweis.**

  Setze zu jedem Zeitpunkt t den Verlustvektor auf 1, falls es im Zeitpunkt $t-1$ einen Verlustvektor gab und i in diesem 1 war. Setze von den restlichen Elementen jeweils die Hälfte auf 0 und 1.
  Somit ergeben sich $E[L^T_{A_1}] = \frac{T}{2}$ als auch $L_{\min} = 0$. □

Untere Grenzen für External Regret Minimization

- **Satz**

  Angenommen $N = 2$. Dann gibt es für jeden Algorithmus A_2 eine Verlustvektorsequenz, wodurch sich zwingend $E[L^T_{A_2} - L_{\min}] \in \Omega(\sqrt{T})$ ergibt.

- **Beweis.**

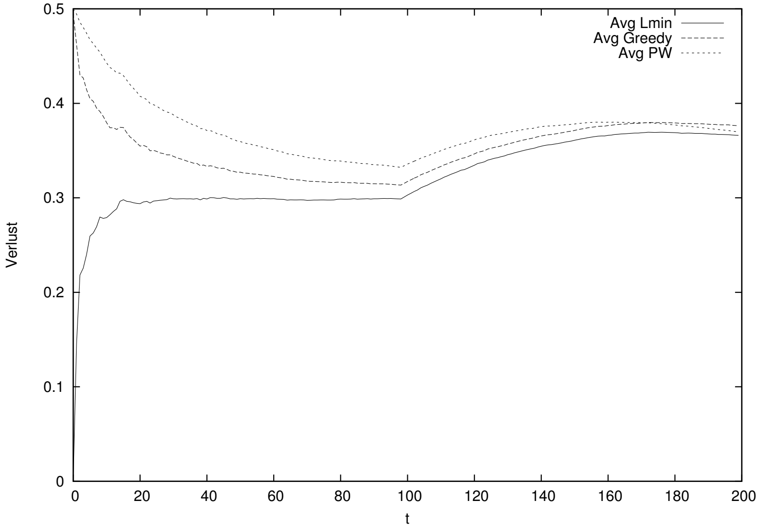
  - Zu Zeitpunkt t werfe Münze und setze $l^t := z_1 = (0, 1)$ oder $l^t := z_2 = (1, 0)$.
  - Nach T Würfen ergeben sich $\frac{T}{2} + y$ Verluste z_1 und $\frac{T}{2} - y$ Verluste z_2 mit $\frac{T}{2} - L_{\min} = |y|$.
  - Nun wird die untere Grenze für $E[|y|]$ benötigt; Diese ergibt sich aus der Wahrscheinlichkeit von y mit $\binom{T}{0.5T+y} \cdot 2^{-T} \in \mathcal{O}(\frac{1}{\sqrt{T}})$ nach Sterling zu

  $$E[|y|] \in \Omega(\sqrt{T})$$

  □

- Weiterhin kann jeder andere Fall mit $N > 2$ auf $N = 2$ zurückgeführt werden, indem an allen Postionen außer den ersten beiden ausschließlich immer ein Verlust von 1 über den Verlustvektor generiert wird und ansonsten wie oben verfahren wird.

Ein Beispiel



Das Partial Information Model

- Problem bei diesen Betrachtungen: Oft bekommt Algorithmus H nur den Gesamtverlust $\sum_i p_i^t l_i^t$ und nicht $l^t \in [0,1]^N$ mitgeteilt.
- Idee: Zerteilen Zeitpunkte $1, \dots T$ in K Blöcke B^τ und führen pro Block "Erkundungen" aus und nutzen sonst während jedes Blocks die gleiche Auswahl p^τ.
- Anschließend geben wir den durschnittlichen Verlust $c_i^\tau = \sum_{t \in B^\tau} \frac{p_i^\tau \cdot l_i^t}{|B^\tau|}$ an einen R Regret Algorithmus O, welcher anhand dessen die Verteilung für den nächsten Block $B^{\tau+1}$ bestimmt (und ignorieren zunächst, dass wir kein l^t erhalten..).

Das Partial Information Model

- Weiterhin sei $C_i^K = \sum_{\tau=1}^{K} c_i^\tau$ und $C_{\min}^K = \min_i C_i^K$.
- Dann ergibt sich für O für die Sequenz der c^τ maximal ein Verlust von $C_{\min} + R$.
- Und unter Einbeziehung aller Zeitpunkte ergibt sich somit ein Verlust von $L_O^T = \sum_{\tau=1}^{K} \sum_{t \in B^\tau} p^\tau \cdot l^t = \frac{T}{K} \sum_{\tau=1}^{K} p^\tau \cdot c^\tau \leq \frac{T}{K} \left(C_{\min}^K + R \right)$

Das Partial Information Model

- Jetzt muss H so entworfen werden, sodass trotz der Erkundungsphase L_O^T erhalten bleibt.
- Statt c^τ zu generieren, generiere stattdessen eine Zufallsvariable c'^τ mit $E[c'^\tau] = c^\tau$:
  1. Wähle für jedes ein $i \in X$ zufällig $t_i \in B^\tau$ mit $i \neq j \to t_i \neq t_j$.
  2. Spiele zum Zeitpunkt t_i Aktion i mit Wahrscheinlichkeit 1 und setze $c_i'^\tau$ auf den erzielten Verlust. Somit gilt $E[c'^\tau] = c^\tau$.
  3. Ansonsten spiele nach Verteilung p^τ.
- Somit ergibt sich im Erwartungswert L_O^T zzgl. der Verluste beim Erkunden: $E[L_H^T] \leq NK + \frac{T}{K}\left(C_{\min}^K + R\right)$.

Swap regret: Definitionen

- Eine Modification Rule ist eine Funktion F, welche sämtliche in Vergangenheit als auch die aktuell vom Algorithmus selektierte Strategie erhält und womöglich eine neue Strategie zurückliefert, welche stattdessen gespielt werden soll. F^t beschreibt die Funktion zum Zeitpunkt t.
- Für $x_1, b_1, b_2 \in X$ sei $\text{switch}_i(x_1, b_1, b_2)$ für Spieler i die folgende Funktion:

$$\text{switch}_i(x_1, b_1, b_2) = \begin{cases} b_2, & \text{falls } x_1 = b_1 \\ x_1, & \text{sonst} \end{cases}$$

- Gegeben eine Modification Rule F (und $s_i(x) = s_i(x_i, x_{-i}) = l_H^t$ mit x als Verteilungssvektor aller Spieler, siehe vorherige Vorträge), lässt sich Regret allgemeiner definieren:

$$\text{regret}_i(x, F) = s_i(x) - s_i(F(x_i), x_{-i})$$

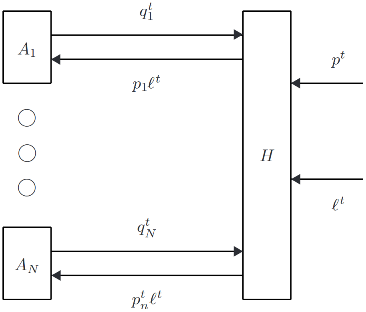
- Falls $F(x_1) = \text{switch}_i(x_1, b_1, b_2)$, dann handelt es sich um Swap Regret.

Swap regret: Definitionen

- Mit diesen Definitionen kann man nun das korrellierte Equilibrium anders auffassen:
- Ein Spiel mit den Spielern $M = \{1, \ldots, m\}$ und den gewählten Strategien x der Spieler befindet sich in einem korrelliertem Equilibrium genau dann, wenn gilt $\forall i \in M \; \forall F : X_i \rightarrow X_i \; E[\text{regret}_i(x, F)] \leq 0$

External Regret zu Swap Regret: Idee

- Um einen Algorithmus H mit gutem Swap Regret zu erhalten, kann man einen Algorithmus A mit (gutem) External Regret R wiederverwenden.
- Idee: Wir nutzen N Instanzen $A_1, \ldots, A_N$, kombinieren die gewählten Wahrscheinlichkeiten q_i zu einem p und geben den Verlust l abhängig von p wieder an die A_i zurück.



[1]

[1] aus Nisan, Roughgarden, Tardos, Vazirani: Algorithmic Game Theory, Seite 92

External Regret zu Swap Regret

- Führe zu jedem Zeitpunkt t die A_i aus und erhalte jeweils eine Verteilung q_i
- Berechne einzelne Verteilung p^t mit $p_j^t = \sum_i p_i^t q_{i,j}^t$ bzw. $p^t = p^t Q^t$, wobei Q^t eine Matrix aus den $q_{i,j}$ darstellt.
  - $p^t = p^t Q^t$ ist immer und effizient lösbar.
  - Erste Möglichkeit zur Betrachtung: p_j^t selektiert Aktion j mit gewisser Wahrscheinlichkeit.
  - Zweite Möglichkeit zur Betrachtung: p_i^t selektiert A_i mit gewisser Wahrscheinlichkeit, welche dann die Aktion wählt.
- Nachdem l erhalten, gebe jedem A_i sein $p_i^t l^t$ zurück.
  - Somit erhält jedes A_i einen Verlust von $(p_i^t l^t) \cdot q_i^t = p_i^t (q_i^t \cdot l^t)$
- Zusammen ergibt sich $\forall i, j \in \{1, \dots, N\}$ $\sum_{t=1}^T p_i^t (q_i^t \cdot l^t) \leq \sum_{t=1}^T p_i^t l_j^t + R$.

External Regret zu Swap Regret

- Zusammen ergibt sich $\forall i, j \in \{1, \ldots, N\}$ $\sum_{t=1}^{T} p_i^t (q_i^t \cdot l^t) \leq \sum_{t=1}^{T} p_i^t l_j^t + R$.
- Wenn man die Verluste über alle A_i summiert, ergibt sich $\sum_i p_i^t (q_i^t \cdot l^t) = p^t Q^t l^t = p^t l^t$, da $p^t = p^t Q^t$.
- Summiert man nun die obige Formel über alle i und beachtet, dass die Aussage für beliebige j und somit auch für alle $F : \{1, \ldots, N\} \to \{1, \ldots, N\}$ gilt, so ergibt sich

$$L_H^T \leq \sum_{i=1}^{N} \sum_{t=1}^{T} p_i^t l_{F(i)}^t + NR = L_{H,F}^T + NR$$