

Spieltheorien und Theoreme

Für das Seminar: Randomized Algorithms bei
Prof. Dr. R. Klein

Von Daniel Herrmann

(Anknüpfend an den Beitrag von Alexander Hombach)

Inhalt:

- Blickpunkt auf randomisierte Strategien
- von Neumann's Minimax Theorem
- Theorem von Loomis
- Yao's Techniken
- Yao's Minimax Prinzip
- Yao's Prinzip zu Monte Carlo Algorithmen (kurzer Einblick)

Randomisierte Strategien bei Spieltheorien:

Bisher haben wir uns nur mit deterministischen Spielstrategien auseinandergesetzt (Ausarbeitung von Alex). Jetzt widmen wir uns den randomisierten bzw. gemischten Strategien.

Zuerst eine kleine Erklärung, was eigentlich eine „Gemischte Strategie“ ist:

Eine Gemischte Strategie ist eine Wahrscheinlichkeitsverteilung über alle möglichen Strategien.

Zur Erklärung: der Spaltenspieler wählt einen Vektor $p = (p_1, \dots, p_n)$ der eine Wahrscheinlichkeitsverteilung der Spalten von der im Vortrag von Alexander Hombach vorgestellten Payoff-Matrix M darstellt. Z.B.: p_i ist die Wahrscheinlichkeit, dass Spieler R(oberta) Spielstrategie i wählen wird.

Gleicherweise hat der Zeilenspieler C(harles) einen Vektor $q = (q_1, \dots, q_n)$, welcher eine Wahrscheinlichkeitsverteilung der Zeilen von der Spielmatrix M darstellt.

Nun ergibt sich für das Spiel, in unserem Fall Schere, Stein, Papier folgender Erwartungswert:

$$E[\text{payoff}] = p^T M q = \sum_{i=1}^n \sum_{j=1}^m p_i M_{ij} q_j$$

$$p^T = (p_1, \dots, p_n)$$

$$M = \text{Payoff} - \text{Matrix}$$

$$q = (q_1, \dots, q_n)^T$$

Dazu eine kleine Beispielrechnung:

$$M = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix} \quad \text{und} \quad p = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} \quad \text{und} \quad q = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$$

$$\begin{aligned}
E[\text{Payoff}] &= p^T M q = \sum_{i=1}^3 \sum_{j=1}^3 p_i M_{ij} q_j \\
&= \left(\frac{1}{3} * 0 * \frac{1}{3}\right) + \left(\frac{1}{3} * 1 * \frac{1}{3}\right) + \left(\frac{1}{3} * (-1) * \frac{1}{3}\right) \\
&\quad + \left(\frac{1}{3} * (-1) * \frac{1}{3}\right) + \left(\frac{1}{3} * 0 * \frac{1}{3}\right) + \left(\frac{1}{3} * 1 * \frac{1}{3}\right) \\
&\quad + \left(\frac{1}{3} * 1 * \frac{1}{3}\right) + \left(\frac{1}{3} * (-1) * \frac{1}{3}\right) + \left(\frac{1}{3} * 0 * \frac{1}{3}\right) \\
&= 0 + \frac{1}{9} - \frac{1}{9} - \frac{1}{9} + 0 + \frac{1}{9} + \frac{1}{9} - \frac{1}{9} + 0 \\
&= 0
\end{aligned}$$

Wie man am Ergebnis sehen kann, ist bei dieser Wahl der Spielstrategien das Spiel fair und es gibt keinen Spieler, der einen Nachteil haben wird.

Durch die Wahl eines anderen p bzw. q könnte einer der beiden Spieler seine Gewinnchancen steigern. Ein entsprechendes Beispiel folgt später.

Sei V_R die bestmögliche untere Schranke für den erwarteten Payoff für Spieler R bei Wahl einer Strategie p und sei V_C die bestmögliche obere Schranke für den erwarteten Payoff der Spieler C bei Wahl einer Strategie q . Somit erhalten wir folgendes:

$$V_R = \max_p \min_q p^T M q$$

$$V_C = \min_q \max_p p^T M q$$

Dies führt uns zum Minimax Theorem von von Neumann, welches aussagt, dass ein Spiel immer eine Lösung hat und $V_R = V_C$ gilt.

Zur Erklärung des vorangegangenen:

Bei V_R versucht zuerst der Zeilenspieler durch Wahl einer Spielstrategie den Erwartungswert so niedrig wie möglich zu machen. Danach versucht der Spaltenspieler durch seine Wahl einer Spielstrategie den Erwartungswert zu seinen Gunsten zu verändern, also ihn zu maximieren.

Bei V_C ist es genauso, nur dass hier zuerst der Spaltenspieler seine Wahl trifft und dann der Zeilenspieler.

Theorem 2.2 (von Neumann's Minimax Theorem):

Für jedes Zero-Sum Spiel mit zwei Spielern, welchem eine Payoff-Matrix zugrunde liegt, gilt:

$$V_R = \max_p \min_q p^T M q = \min_q \max_p q^T M q = V_C$$

In Worten:

Der größte Erwartungswert, den der Spaltenspieler beim wählen einer gemischten Strategie erreichen kann, ist **gleich** dem kleinsten Erwartungswert, den der Zeilenspieler beim wählen einer gemischten Strategie erreichen kann.

Dieser Erwartungswert wird „Wert des Spiels“ genannt und mit V bezeichnet.

Sattelpunkte:

Ein paar gemischter Strategien $(\hat{p}, \hat{q})$, welche den linken Term der Gleichung 2.2 maximieren und den rechten Teil minimieren, werden Sattelpunkt genannt und die beiden Strategien werden als optimal bezeichnet.

Stellen wir folgende Überlegung an: Was wäre, wenn einer der Spieler sich auf eine Strategie festgefahren hat? Nehmen wir an, dass der Spaltenspieler mit seinem Vektor p eine feste Entscheidung getroffen hat und nur noch mit dieser Strategie spielt.

Somit wird aus der Gleichung $p^T M q$ eine lineare Funktion von q . Das bringt folgende interessante Neuerung. Wenn Spieler C die Strategie von Spieler R kennt, dann ist seine optimale Strategie eine deterministische. Er kann sich auf sein Gegenüber einstellen und seine Strategie so anpassen, dass sich der Erwartungswert des Spiels zu seinen Gunsten ändert. Er muss nicht einmal eine besonders komplizierte Strategie wählen, damit er häufiger gewinnt. Um das zu erreichen kann er einfach einen Einheitsvektor als seine Strategie q wählen und somit mehr Gewinne für sich einfahren. Dies führt uns zum Theorem von Loomis.

Theorem 2.3 (Loomis' Theorem):

Für jedes Zero-Sum Spiel mit zwei Spielern, welchem eine Payoff-Matrix M zugrunde liegt, gilt:

$$V_R = \max_p \min_j p^T M e_j = \min_q \max_i e_i^T M q = V_C$$

Dabei sei e_k der k -te Einheitsvektor.

Zum bessern Verständnis folgen 2 Beispiele:

1. Generelles Beispiel:

$$M = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix} \quad \text{und} \quad p = \begin{pmatrix} 3/4 \\ 1/4 \\ 0 \end{pmatrix} \quad \text{und} \quad q = \begin{pmatrix} 1/4 \\ 0 \\ 3/4 \end{pmatrix}$$

$$\begin{aligned} E[\text{Payoff}] &= p^T M q = \sum_{i=1}^3 \sum_{j=1}^3 p_i M_{ij} q_j \\ &= \left(\frac{3}{4} * 0 * \frac{1}{4} \right) + \left(\frac{3}{4} * 1 * 0 \right) + \left(\frac{3}{4} * (-1) * \frac{3}{4} \right) \\ &\quad + \left(\frac{1}{4} * (-1) * \frac{1}{4} \right) + \left(\frac{1}{4} * 0 * 0 \right) + \left(\frac{1}{4} * 1 * \frac{3}{4} \right) \\ &\quad + \left(0 * 1 * \frac{1}{4} \right) + \left(0 * (-1) * 0 \right) + \left(0 * 0 * \frac{3}{4} \right) \\ &= 0 + 0 - \frac{9}{16} - \frac{1}{8} + 0 + \frac{3}{16} + 0 - 0 + 0 \\ &= -\frac{9}{16} - \frac{1}{8} + \frac{3}{16} = -\frac{9}{16} - \frac{2}{16} + \frac{3}{16} = -\frac{8}{16} = -\frac{1}{2} \end{aligned}$$

Hier hat sich Spieler R für eine Strategie entschieden und Spieler C hat sich darauf eingestellt. Jedoch kann Spieler C immer noch ab und an verlieren. Doch der Erwartungswert hat sich zugunsten von Spieler C geändert. In diesem Beispiel ist e_k kein Einheitsvektor. Es soll generell die Unterschiede zwischen den Theoremen von von Neuman und Loomis zeigen.

2. Beispiel zu Loomis':

$$M = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix} \quad \text{und} \quad p = \begin{pmatrix} 3/4 \\ 1/4 \\ 0 \end{pmatrix} \quad \text{und} \quad q = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{aligned} E[\text{Payoff}] &= p^T M q = \sum_{i=1}^3 \sum_{j=1}^3 p_i M_{ij} q_j \\ &= \left(\frac{3}{4} * 0 * 1 \right) + \left(\frac{3}{4} * 1 * 0 \right) + \left(\frac{3}{4} * (-1) * 0 \right) \\ &\quad + \left(\frac{1}{4} * (-1) * 1 \right) + \left(\frac{1}{4} * 0 * 0 \right) + \left(\frac{1}{4} * 1 * 0 \right) \\ &\quad + (0 * 1 * 1) + (0 * (-1) * 0) + (0 * 0 * 0) \\ &= 0 + 0 - 0 - \frac{1}{4} + 0 + 0 + 0 - 0 + 0 \\ &= -\frac{1}{4} \end{aligned}$$

Hier hat Spieler C eine Strategie gewählt, die ihn kein Spiel verlieren lässt. Der „schlimmste“ Fall ist ein Unentschieden. Dennoch ist der Erwartungswert niedriger als im Beispiel davor. Also hat Spieler C noch nicht die Optimale Strategie gewählt und sollte eine andere Verteilung wählen.

Yao's Techniken:

Nun widmen wir uns den Ergebnissen und der Anwendung aus den oben genannten Spieltheorien, um untere Schranken der Performance von Randomisierten Algorithmen zu beweisen.

Dazu bedienen wir uns folgender Betrachtungsweise: Wir sehen den Algorithmusdesigner als unseren Spaltenspieler C an und seinen Widersacher als den Zeilenspieler R. Hierbei sind die Spalten die Menge aller möglichen Algorithmen für dieses Spiel, wohingegen die Zeilen als Menge aller möglichen Eingaben einer festen Größe dienen.

Dabei sollte man nicht vergessen, dass jede Spalte für einen deterministischen Algorithmus steht, der immer eine korrekte Lösung liefert.

Nun liefert der Payoff, den C an R zu verrichten hat, ein reellwertiges Maß der Effizienz eines Algorithmus, bezogen auf die Laufzeit, die Güte des Ergebnisses, Bearbeitungs- oder Platzkosten und Komplexität. Wir beschränken uns hier jedoch auf die Laufzeit.

Weiterhin wählt der Algorithmusdesigner einen Algorithmus, welcher den Payoff minimiert, während sein Widersacher diesen gerne maximieren möchte.

Bei genauerer Überlegung fällt natürlich auf, dass es nur endlich viele endliche Inputs gibt, die in Frage kommen, sowie auch nur endlich viele (deterministische, terminierende und korrekte) Algorithmen, um das Problem zu lösen.

Eine optimale „pure“ Strategie für C hängt von einem optimal gewählten deterministischen Algorithmus für das Problem ab.

„Deterministische Komplexität des Problems“:

Damit wird die worst-case Laufzeit V_C jedes deterministischen Algorithmus für das Problem bezeichnet.

V_R bezeichnet die „Nichtdeterministische Komplexität des Problems“.

Wenn das Spiel eine Lösung hat dann sind beide Komplexitäten gleich.

Unser Interesse liegt nun in der Interpretation von gemischten Strategien für den Algorithmusdesigner und seinen Widersacher.

Eine gemischte Strategie ist eine Wahrscheinlichkeitsverteilung über die Gesamtheit aller (immer korrekten) deterministischen Algorithmen, also ein optimaler randomisierter Las Vegas Algorithmus. Eine optimale Strategie für R ist eine Wahrscheinlichkeitsverteilung über dem Raum aller möglichen Eingaben. Das ist aber noch nicht alles. Wir benötigen noch eine weitere Komplexität. Nämlich die

Verteilungskomplexität:

Diese ist definiert als die erwartete Laufzeit eines Problems bei Wahl des besten deterministischen Algorithmus für die schlechteste Verteilung der Eingaben.

Diese Komplexität ist aber kleiner als die deterministische Komplexität, da hier der Algorithmus ja die Eingabe kennt.

Theorem 2.3 impliziert somit, dass die Verteilungskomplexität gleich der am wenigsten möglichen, erreichbaren, erwarteten Laufzeit eines jeden Randomisierten Algorithmus ist. Formulieren wir die uns schon bekannten Theoreme in der Sprache der Algorithmen wie folgt:

Korollar 2.4:

Sei Π ein Problem mit einer endlichen Anzahl I von Eingabeinstanzen, und $\mathcal{A}$ eine endliche Anzahl von deterministischen Algorithmen.

Dann sei $C(I, A)$ die Laufzeit von Algorithmus A bei Eingabe I , wobei $A \in \mathcal{A}$ und $I \in I$ ist. Es seien weiterhin p über I und q über $\mathcal{A}$ Wahrscheinlichkeitsverteilungen. Dabei bezeichne I_p einen zufälligen Input bezüglich p und A_q ein zufälliger Algorithmus bezüglich q .

Dann gelten:

$$\max_p \min_q E[C(I_p, A_q)] = \min_q \max_p E[C(I_p, A_q)]$$

und

$$\max_p \min_{A \in \mathcal{A}} E[C(I_p, A)] = \min_q \max_{I \in \mathcal{I}} E[C(I, A_q)]$$

Aus diesem Korollar erhalten wir folgende Proposition.

Proposition 2.5 (Yao's Minimax Principle):

Für alle Verteilungen p über $\mathcal{I}$ und q über $\mathcal{A}$ gilt:

$$\min_{A \in \mathcal{A}} E[C(I_p, A)] \leq \max_{I \in \mathcal{I}} E[C(I, A_q)]$$

In anderen Worten:

Die erwartete Laufzeit des optimalen deterministischen Algorithmus für eine zufällig gewählte Eingabeverteilung p ist eine untere Schranke für die erwartete Laufzeit des optimalen randomisierten Algorithmus für Π .

Wir sehen, um also die untere Schranke der randomisierten Komplexität zu beweisen reicht es, für beliebiges p die untere Schranke der erwarteten Laufzeit von deterministischen Algorithmen zu beweisen.

Die Stärke dieser Technik liegt in der Flexibilität von p .

Dabei sollte man nicht vergessen, dass der deterministische Algorithmus die gewählte Verteilung p kennt.

Da wir uns jetzt nur mit unteren Schranken der Effizienz von Las Vegas Algorithmen beschäftigt haben, wollen wir noch einen Blick auf die verbleibenden Monte Carlo Algorithmen, mit Fehlerwahrscheinlichkeit $\varepsilon \in [0, \frac{1}{2}]$, werfen.

Sei die minimale erwartete Laufzeit von jedem deterministischen Algorithmus mit Fehlerschranke ε bei Eingabeverteilung p bezeichnet mit:

$$\min_{A \in \mathcal{A}} E[C_\varepsilon(I_p, A)]$$

Sei die erwartete Laufzeit von jedem randomisierten Algorithmus bei schlimmstmöglicher Eingabe mit Fehlerschranke ε bezeichnet mit:

$$\max_{I \in \mathcal{I}} E[C_\varepsilon(I, A_q)]$$

Hierbei wird wiederum der randomisierte Algorithmus als Wahrscheinlichkeitsverteilung q von deterministischen Algorithmen angesehen.

Analog zu Proposition 2.5 erhalten wir

Proposition 2.6:

Für alle Verteilungen p von I und q von A und jedem $\varepsilon \in [0, \frac{1}{2}]$, gilt:

$$\frac{1}{2} \left(\min_{A \in \mathcal{A}} E[C_{2\varepsilon}(I_p, A)] \right) \leq \max_{I \in \mathcal{I}} E[C_\varepsilon(I, A_q)]$$

Dies sollte aber nur einen kleinen Einblick in die Schranken von Monte Carlo Algorithmen geben.

Literatur:

Motwani, Raghavan: *Randomized Algorithms*, Cambridge Press, 1995