Seminar über Algorithmen 19. November 2013

Graphische Spiele

Michael Brückner *Wolfgang Mulzer, Yannik Stein*

1 Einführung

Da in Mehrspielerspielen mit einer hohen Anzahl $n \in \mathbb{N}$ an Spielern die Auszahlungsdarstellungen enorm groß werden, und es häufig vor kommt, dass nicht alle Spieler mit allen Gegenspielern interagieren, eignet es sich, die Auszahlungsdarstellungen auf miteinander interagierende Spieler zu reduzieren. Grundidee hierfür ist ein Graph mit der Knotenmenge $\{1, \ldots, n\}$ der Spieler. Die Kantenmenge repräsentiere dabei diejenigen Spielerpaare, die tatsächlich im Spiel interagieren. Die Auszahlungsmatrizen beziehen sich dabei nur auf einen Spieler und seine direkte Nachbarschaft.

1.1 Begriffe

Die folgende Liste zeigt kurz zusammengefasst die wichtigsten Erkenntnisse und Konventionen der letzten Kapitel.

- $N = \{1, \ldots, n\}$ sei die Menge der Spieler.
- $a_i \in \{0, 1\}$ sei die Strategie, für die sich Spieler $i \in N$ entschieden hat.
- M_i sei die Gewinnmatrix für Spieler i, beschriftet durch die Strategievektoren $a \in \{0, 1\}^n$.
- $M_i(a) \in [0, 1]$ gebe den Gewinn für Spieler i an, wenn a gespielt wird. (Spiel in Normalform)
- $\{0, 1\}$ seien die einzigen reinen Strategien, $p_i \in [0, 1]$ sei die Wahrscheinlichkeit, dass i die Strategie 0 spielt. (gemischte Strategien)
- $p \in [0, 1]^n$ heiße Produktverteilung und enthalte die entsprechenden gemischten Strategien aller Spieler.
- $M_i(p) := \mathbb{E}_{a \sim p}[M_i(a)]$ sei der erwartete Gewinn für Spieler i.
- $p[i : p'_i]$ sei der Vektor p, wobei p_i durch p'_i ersetzt werde.
- Sei $p \in [0, 1]^n$ gemischter Strategievektor. Dann ist p ein Nash-Gleichgewicht, wenn für alle Spieler i und alle alternativen Wahrscheinlichkeiten $p'_i \in [0, 1]$ gilt:

$$M_i(p) \geq M_i(p[i : p'_i])$$

- p heißt ε-Nash-Gleichgewicht, wenn stattdessen gilt:

$$M_i(p) + \varepsilon \geq M_i(p[i : p'_i])$$

  Alle Spieler können ihren Gewinn also durch Wechsel der Strategie um höchstens ε erhöhen.
- $G = (N, E)$ sei ein ungerichteter Graph. Die Kantenmenge E gebe an, welche Spieler im betrachteten Spiel interagieren.
- $N(i) := \{i\} \cup \{j \in N \mid ij \in E\}$ sei die Nachbarschaft des Spielers i.
- $d := \max_{i=1}^{n} |N(i)|$ sei die Größe der größten Nachbarschaft in G.
- Sei a ein Strategievektor, dann sei a^i die Projektion von a auf $N(i)$.

Diese Mittel reichen, um graphische Spiele zu definieren:

Definition 1. *Ein graphisches Spiel ist ein Paar $(G, \mathcal{M})$, wobei G ein ungerichteter Graph über der Knotenmenge $\{1, \ldots, n\}$ ist und $\mathcal{M}$ die Menge der lokalen Gewinnmatrizen. Für einen Strategievektor $a = (a_1, \ldots, a_n) \in A$ gebe die lokale Gewinnmatrix $M_i \in \mathcal{M}$ den Gewinn $M_i(a^i)$ für Spieler i an.*

Von Spielen in Normalform ist Darstellungsgröße der Gewinnmatrizen exponentiell in der Anzahl der Spieler. Ist die Größe d der größten Nachbarschaft in G jedoch viel kleiner als n, so wirkt sich das drastisch auf die Darstellungsgröße des Spiels aus. Während Gewinne jetzt nur noch von lokalen Umgebungen abhängen, sind die betrachteten Gleichgewichte immer noch eine globale Größe, da sich Gewinne von Spielern auch indirekt beeinflussen können durch die Strategien gemeinsamer Nachbarn.

Zu Nash-Gleichgewichten, die sich nur dadurch auszeichnen, dass kein Spieler sich verbessern kann, wenn alle anderen bei ihrer Entscheidung bleiben, gibt es die Erweiterung der korrelierten Gleichgewichte. Korrelierte Gleichgewichte liefern im allgemeinen etwas fairere Spielergebnisse, berühmtes Beispiel ist das Ampelspiel. Bei korrelierten Gleichgewichten geht es darum, dass jedem Spieler eine gemischte Strategie zugewiesen wird und er für sich das beste Resultat erzielt, wenn er die ihm zugewiesene Strategie beibehält.

Definition 2. *Ein korreliertes Gleichgewicht ist eine Wahrscheinlichkeitsverteilung $P(a)$ über alle Strategievektoren a, die für alle Spieler i und beide Strategien $b \in \{0, 1\}$ folgende Gleichung erfüllt:*

$$\mathbb{E}_{a \sim P_{a_i = b}}[M_i(a)] \geq \mathbb{E}_{a \sim P_{a_i = b}}[M_i(a[i : 1 - b])]$$

2 Nash-Gleichgewichte in graphischen Spielen

Wir betrachten in diesem Kapitel nur Spiele, deren Graph ein Baum ist. Für die gibt es einen Algorithmus, der im folgenden vorgestellt werden soll:

2.1 Algorithmus TreeNash

Für die Notation des Algorithmus' ist es sinnvoll, die Spieler statt durch Nummern, durch Buchstaben U, V, W zu bezeichnen. Dabei bezeichne u, v, w die entsprechend gewählten gemischten Strategien.
Vorbereitung:

1. Wähle einen beliebigen Knoten V zur Wurzel des Baums.
2. Berechne eine Knotenreihenfolge von der Wurzel abwärts in die Blätter.

Aufwärtsübergabe

Führe für alle Knoten V (beginnend in den Blättern) die folgenden Schritte durch:

1. Sei W der Ursprung von V (oder **null**, wenn W die Wurzel ist)
2. Für alle $w, v \in [0, 1]$, initialisiere $T(w, v) = 0$ und setze die setze die Zeugenliste für $T(w, v)$ auf die leere Liste.
3. Ist V ein Blatt?
   (a) Für alle $w, v \in [0, 1]$, setze $T(w, v)$ auf 1 genau dann, wenn $V = v$ eine beste Antwort auf $W = w$ ist.
4. Ist V kein Blatt?
   (a) Seien $U = (U_1, \ldots, U_k)$ die Kinder von V und $T(v, u_i)$ die übergebenen Tabellen.
   (b) Für alle $w, v \in [0, 1]$, für alle gemischten Strategievektoren $u = (u_1, \ldots, u_k)$ mit $T(v, u_i) = 1\ \forall i$: Ist $V = v$ eine beste Antwort auf $U = u$ und $W = w$, setze $T(w, v) = 1$ und füge u der Zeugenliste für $T(w, v)$ hinzu.

Abwärtsübergabe

Führe für alle Knoten V (beginnend in der Wurzel) die folgenden Schritte durch:

1. Seien $U = (U_1, \ldots, U_k)$ die Kinder von V (oder leer, wenn V ein Blatt ist), W der Ursprung von V (null, wenn V die Wurzel ist). Seien (w, v) die Werte, die von W nach V übergeben wurden.
2. Beschrifte V mit v
3. Wähle einen Zeugen u zu $T(w, v) = 1$.
4. Für $i = 1, \ldots, k$: Gebe (v, u_i) an U_i weiter.

Der Algorithmus bringt ein wesentliches Problem mit sich: Es gibt keine endliche Darstellung für alle $w, v \in [0, 1]$, die der Algorithmus abarbeiten könnte. Dies lässt sich umgehen, indem wir statt $[0, 1]$ das diskretisierte Intervall $\{0, \tau, 2\tau, \ldots, 1\}$ für ein von einem vorgegebenen ε abhängiges τ wählen. Das folgende Lemma zeigt den Vorteil dieses Gitters:

Lemma 3. *Sei G ein Graph mit maximalem Grad d, $(G, \mathcal{M})$ ein graphisches Spiel. p sei ein (exaktes) Nash-Gleichgewicht für $(G, \mathcal{M})$ und q der nächstgelegene Punkt zu p in dem τ-Gitter. Dann ist q ein $d\tau$-Nash-Gleichgewicht.*

Der vollständige Beweis für das Lemma befindet sich in [1]. Der entscheidende Vorteil ist, dass die Wahl des τ nur von d und ε abhängt, aber nicht von der Gesamtzahl der Spieler n. Daraus lässt sich ein approximativer Algorithmus ApproxTreeNash entwickeln, der sich vom obigen Algorithmus lediglich in der Wahl der Punkte unterscheidet: Es werden nur vielfache von τ berücksichtigt. Außerdem muss die Berechnung der Nash-Gleichgewichte abgeändert werden zur Berechnung von ε-Nash-Gleichgewichten. Es lässt sich berechnen, dass dieser Algorithmus approximative Nash-Gleichgewichte in einer Laufzeit findet, die polynomiell in der Darstellungsgröße des graphischen Spiels ist.

Für einen exakten Algorithmus wird eine geometrische Betrachtung der Menge $\{(u, v) \in [0, 1]^2 \mid T(u, v) = 1\}$ relevant. [1] zeigt, dass die Menge eine Vereinigung von achsenparallelen Rechtecken ist. Mit dieser Eigenschaft kann ein exakter Algorithmus ExactTreeNash entwickelt werden, der alle Nash-Gleichgewichte exakt ausrechnet, allerdings in der Spielerzahl exponentielle Laufzeit innehat.

Einen Algorithmus, der sich mit Graphen beschäftigt, die kein Baum sind, liefert NashProp. Er führt die Berechnungen von TreeNash für jeden Knoten mehrfach durch, wobei er *jeden* Nachbarn einmal als den Ursprung im Baum ansieht. Außerdem werden die Tabellen anfangs komplett mit 1 gefüllt und es werden Gleichgewichte gestrichen, bei denen die Antwort eines Spielers keine beste Antwort auf die Strategien seiner Nachbarn ist. Dieser Algorithmus läuft in polynomieller Zeit, berechnet allerdings nur eine Obermenge der Nash-Gleichgewichte. Ein zweiter Schritt, der diese Menge dann auf Richtigkeit überprüft, wird dann rechnerisch aufwändig.

3 Korrelierte Gleichgewichte in graphischen Spielen

Definition 4. *Zwei Wahrscheinlichkeitsverteilungen P und Q über Strategievektoren a heißen äquivalent bezüglich erwartetem Gewinn $P \equiv_{\mathrm{EP}} Q$, wenn sie jedem Spieler den selben erwarteten Gewinn zuweisen:*

$$\mathbb{E}_{a \sim P}[M_i(a)] = \mathbb{E}_{a \sim Q}[M_i(a)]$$

Definition 5. *Zwei Wahrscheinlichkeitsverteilungen P und Q über Strategievektoren a heißen lokalnachbarschaftlich äquivalent $P \equiv_{\mathrm{LN}} Q$, wenn für alle Spieler i und alle Strategievektoren a^i*

über $N(i)$ gilt:

$$P(a^i) = Q(a^i)$$

Lemma 6. *Für alle Graphen G, alle Verteilungen P und Q über Strategievektoren gilt: Ist $P \equiv_{\mathrm{LN}} Q$, so auch $P \equiv_{\mathrm{EP}} Q$. Außerdem gilt für jeden Graphen G, wenn $P \not\equiv_{\mathrm{LN}} Q$, dann gibt es eine Gewinnmatrixmenge $\mathcal{M}$, sodass $P \not\equiv_{\mathrm{EP}} Q$.*

Lemma 7. *Sei $(G, \mathcal{M})$ ein graphisches Spiel, P ein korreliertes Gleichgewicht für $(G, \mathcal{M})$ und $P \equiv_{\mathrm{LN}} Q$, dann ist Q ebenfalls ein korreliertes Gleichgewicht für $(G, \mathcal{M})$.*

Definition 8. *Ein lokales Markov-Netzwerk ist ein Paar $M := (G, \Psi)$, wobei G ein ungerichteter Graph auf der Knotenmenge N ist und $\Psi = \{\psi_1, \ldots, \psi_n\}$ eine Menge von Potentialfunktionen ist. Jede Potentialfunktion $\psi_i : \{a^i\} \to [0, \infty)$ weise dabei jedem Strategievektor der Nachbarschaft von i eine positive reelle Zahl zu.*

Ein lokales Markov-Netzwerk M definiert eine Wahrscheinlichkeitsverteilung

$$P_M(a) := \frac{1}{Z} \left(\prod_{i=1}^{n} \psi_i(a^i) \right),$$

wobei $Z = \sum_a \prod_{i=1}^n \psi_i(a^i)$ der Normalisierungsfaktor ist.

Lemma 9. *Sei G Graph und P Wahrscheinlichkeitsverteilung über Strategievektoren. Dann gibt es eine Verteilung Q, die sich als lokales Markov-Netzwerk darstellen lässt, sodass $Q \equiv_{\mathrm{LN}} P$ gilt.*

Satz 10. *Sei $(G, \mathcal{M})$ ein graphisches Spiel und P ein korreliertes Gleichgewicht. Dann gibt es eine Verteilung Q mit den folgenden Eigenschaften:*

1. *Q ist ebenfalls korreliertes Gleichgewicht.*
2. *Q gibt allen Spielern die selben Gewinne wie P: $Q \equiv_{\mathrm{EP}} P$*
3. *Q kann als lokales Markov-Netzwerk dargestellt werden.*

Zum einen erlaubt uns Satz 10 eine Vielzahl an Möglichkeiten aus der Stochastik oder der Theorie maschinellen Lernens [3, 2]. Außerdem sind Markov-Netzwerke extrem sparsam in der Beschreibung, nämlich linear in der Größe des Spiels, weshalb sich korrelierte Gleichgewichte so sparsam formulieren lassen.

Korrelierte Gleichgewichte lassen sich in Spielen in Normalform per LP ausrechnen. Dies führt zu einer exponentiellen Laufzeit in der Spielgröße. Für graphische Spiele, denen ein Baum zugrunde liegt, lässt sich dies jedoch auf lineare Laufzeit beschränken, denn für Bäume lassen sich die Nebenbedingungen des LP wie folgt aufstellen:

Variablen: Für jeden Spieler i und jeden Strategievektor a^i gibt es eine Variable $P(a^i)$.

Nebenbedingungen:

1. P ist Korreliertes Gleichgewicht. Für alle Spieler i und alle Strategien a, a':

$$\sum_{a^i : a^i_i = a} P(a^i) M_i(a^i) \geq \sum_{a^i : a^i_i = a} P(a^i) M_i(a^i[i : a'_i])$$

2. P ist Wahrscheinlichkeitsverteilung. Für alle Spieler i:

$$\forall a^i : P(a^i) \geq 0; \quad \sum_{a^i} P(a^i) = 1$$

3. P weist gleiche Werte in Nachbarschaften zu. Für alle Spieler i, j, für alle Strategievektoren y^{ij} der gemeinsamen Nachbarschaft $N(i) \cap N(j)$:

$$\sum_{a^i : a^{ij} = y^{ij}} P_i(a^i) = \sum_{a^j : a^{ij} = y^{ij}} P_j(a^j)$$

Lemma 11. *Sei* $(G, \mathcal{M})$ *ein graphisches Spiel. Jede Lösung zu den obigen Bedingungen liefert ein korreliertes Gleichgewicht für* $(G, \mathcal{M})$*. Dabei entscheidet die Zielfunktion darüber, welches Gleichgewicht erreicht wird.*

4 Literatur

[1] Michael Kearns, Michael L. Littman, and Satinder Singh. Graphical models for game theory, 2001.

[2] S.L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

[3] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Representation and Reasoning Series. Morgan Kaufmann, 1988.