

Opponent Modelling in RoShamBo



Timo Bozsolik (timo.boz@gmx.de)

Überblick

- **Einführung**
- **Theoretische Betrachtung – Die optimale Strategie**
- **Basis – Strategien**
- **locaine Powder**
- **Schlussfolgerungen**

RoShamBo – Die Regeln

	Stein	Schere	Papier
Stein	---	Stein	Papier
Schere	Stein	---	Schere
Papier	Papier	Schere	---

- **Stein** gewinnt gegen **Schere**
- **Schere** gewinnt gegen **Papier**
- **Papier** gewinnt gegen **Stein**

Die optimale Strategie

- **Jedes Symbol gewinnt und verliert gegen jeweils ein anderes**
 - ⇒ **alle Symbole gleichwertig**
- **Weiß der Gegner, welche Taktik man verfolgt, wird er jede Runde gewinnen**
- **optimale Strategie ist zufällige Auswahl**
 - ➔ **gleiche Erfolgchancen**
 - ➔ **keine Vorhersehbarkeit**

Die optimale Strategie / 2

- optimale Strategie wird nie mit hoher Differenz gewinnen oder verlieren
 - ➔ 1000 Runden „Immer **Stein**“ gegen „Zufall“:
 - 333 x gewinnt „Immer **Stein**“
 - 333 x gewinnt „Zufall“
 - 333 x Unentschieden
- optimale Strategie erreicht gegen jede andere Strategie in etwa ein Unentschieden
 - ⇒ Ziel ist es, gegen andere suboptimale Strategien zu gewinnen
 - ⇒ *Opponent Modelling*

Basis-Strategien

- **Dummy-Bots**, eingesetzt beim ersten internationalen RoShamBo-Programmierwettbewerb
- **Wissen um einfache Basis-Strategien wichtig, um Spielweise des Gegners zu erkennen**

Random-Bot

- Realisierung der optimalen Strategie durch zufällige Auswahl eines Symbols
- keine erfolgsversprechende Gegenstrategie
- maximale Score: ± 0

Basis-Strategien / 2

PI-Bot

- Vordefinierte Sequenz (Pseudo-Zufall) gegeben durch Zahl $\pi \sim 3,141592$
- vollständige Vorhersagbarkeit bei Erkennung des Systems
- maximale Score: 1000 von 1000

Text-Bot

- Vordefinierte Sequenz gegeben durch englischen Text
- gewisse Vorhersagbarkeit durch Sprachmuster
- maximale Score: ~ 200 von 1000 (empirisch)

Flat-Bot

- Gleichmäßigere Zufallsverteilung durch bevorzugte Wahl bisher weniger benutzter Symbole
- statistische Vorhersagbarkeit
- maximale Score: ~ 420 von 1000 (gemessen $\sim 800 / 1000$)

Basis-Strategien / 3

DeBruijn-Bot

- Vordefinierte DeBruijn-Sequenz
 - Enthält alle möglichen Subsequenzen gegebener Länge
 - 3 Zeichen, alle Substrings der Länge 2: RRPSPPRSS
- keine Wiederholung von Patterns
- maximale Score: 500 von 1000
- konnte vorderen Platz im Wettbewerb erreichen

SwitchALot-Bot

- Auswahl eines anderen als das in der letzten Runde benutzen Symbols mit hoher Wahrscheinlichkeit (88%)
- relativ sichere Vorhersage des in der nächsten Runde nicht auftretenden Symbols, dadurch entweder Gewinn oder Unentschieden
- maximale Score: ~ 320 von 1000

Basis-Strategien / 4

Foxtrott-Bot

- Sequenz $[r \rightarrow p+2 \rightarrow r \rightarrow p+1 \rightarrow r \rightarrow p]$
 - r = zufälliger Zug, p = vorheriger Zug
 - Addition modulo 3
- Vorhersage jedes zweiten Zuges
- maximale Score: 500 von 1000

AntiRotn-Bot

- Behandlung konsekutiver Paare der Gegneraktionen durch Bewertung der Rotation (+1, 0, -1)
- Vorhersage in Wahrscheinlichkeiten
 - $EV[x] = Pr[x+2] - Pr[x+1]$
 - Bsp: $Pr[R] = 0.2$, $Pr[P] = 0.39$, $Pr[S] = 0.41$
 - ⇒ $EV[R]=0.02$, $EV[P] = -0.21$, $EV[S] = 0.19$
- Zusatz: Bei hohem Verlust ($> 4\%$) → Zufall
- maximale Score: ~ 40 von 1000

„Reale“ Bots

- **Generelle Ansätze vs. gemischte Strategien**
 - ➔ **Bsp. für globalen Ansatz: cocaine Powder**
 - ➔ **gemischte Strategie aus mehreren (einfachen) Sub-Strategien, Auswahl der momentan besten über Performance in den letzten Zügen**
- **Direkte Mustererkennung vs. statistische Behandlung der Historie**
- **Direkte Umsetzung der Strategie vs. Abweichung vom Zufall erst beim Erkennen einer Schwäche**
- **Verschleierungstaktiken (zufälliges Rauschen)**

locaine Powder

- **Gewinner des ersten internationalen RoShamBo-Programmierwettbewerbs (1999), dritter beim zweiten Wettbewerb (2000) trotz Offenlegung des Codes**
- **komplexer Bot, bestehend aus drei Ebenen**
 - **untere Ebene: konkrete Prognose-Module**
 - **mittlere Ebene: Meta-Strategien**
 - **obere Ebene: Variation des Zeithorizontes**
- **Wahl der jeweils zu benutzenden Spielweise durch Analyse der Performance in den letzten Runden**
- **vereint Vorteile verschiedener Herangehensweisen**

Meta-Strategien

- Voraussetzung: Vorhersagemodul P auf unterer Ebene bestimmt Symbol p für nächsten Zug des Gegners
- **Stein** $\rightarrow 0$, **Papier** $\rightarrow 1$, **Schere** $\rightarrow 2$
- Schlage Symbol p mit $p + 1 \% 3$

Strategie $P.0$

- direkte Abwehr des von P vorhergesagten Symbols
- P sagt **Stein** voraus, spiele **Papier**
- $p.0 = p + 1 \% 3$

Meta-Strategien / 2

Strategie P.1

- ➔ Abwehr von „second guessing“ (Gegner kennt eigenes Verhalten und reagiert selbst entsprechend)
- ➔ notwendig, falls Gegner P.0 „geknackt“ hat
- ➔ P sagt **Stein** voraus. Gegner denkt, eigener Bot spielt **Papier** und spielt **Schere**. Spiele also **Stein**.
- ➔ $p.1 = p + 3 \% 3 = p$

Strategie P.2

- ➔ Abwehr von „triple guessing“ (Gegner kennt P.1 und reagiert selbst entsprechend)
- ➔ P.1 empfiehlt **Stein**. Gegner weiß das und spielt darauf **Papier**. Spiele also **Schere**.
- ➔ $p.2 = p + 5 \% 3$

■ weitere Strategien: $P.3 = P.0$, $P.4 = P.1$ etc.

Meta-Strategien / 3

Strategie $P'.0$

- ➔ Annahme, Gegner benutzt selbst P
- ➔ Benutze selbst P' als Vorhersage, bei dem die Rollen von eigenem Bot und Gegner vertauscht sind
 - ➔ Vorhersage des nächsten eigenen Zuges nach P
- ➔ P' sagt **Stein** für nächsten eigenen Zug voraus, Gegner spielt also **Papier**. Spiele nach $P'.0$ **Schere**.
- ➔ $p'.0 = p' + 2 \% 3$

Strategien $P'.1$ und $P'.2$

- ➔ „Weiterdenken“ durch weitere Rotationen analog zu $P.1$ und $P.2$
- ➔ $p'.1 = p' + 4 \% 3$
- ➔ $p'.2 = p' + 6 \% 3$

Meta-Strategien / 4

- Insgesamt 6 Meta-Strategien
 - ➔ 2 empfehlen **Stein**, 2 **Schere**, 2 **Papier**
 - ➔ welche Strategie anwenden?

- Annahme: Gegner spielt ein Match nach gleichbleibender Spielweise
 - ➔ Auswahl der Strategie mit den besten Ergebnissen in der Vergangenheit

Methoden zur Vorhersage

- 3 Module, Auswahl des wiederum besten nach bisheriger Performance

Frequenzanalyse

- statistische Methode
- sage die bisher häufigste Aktion des Gegners auch für den nächsten Zug voraus
- Abwehr von Frequenzanalyse des Gegners im Kombination mit Meta-Strategien

Methoden zur Vorhersage / 2

History-Matching

- ➔ direkte Methode, Erkennung von Patterns
 - ➔ Letzten 3 Runden waren bspw.:
Stein vs. Papier, Stein vs. Schere, Schere vs. Papier
 - ➔ Suche nach Vorkommen dieser Sequenz in der Vergangenheit und betrachte folgendes Symbol
- ➔ Längere Sequenzen aussagekräftiger

Zufalls-Modul

- ➔ liefere zufälliges Symbol zurück
- ➔ Absicherung gegen stärkere Spieler
 - ➔ andere Module schneiden dann schlecht ab
 - ➔ Zufall hat daher mit ± 0 beste Performance

Meta-Meta-Strategie

- Grundsätzliche Annahme, dass Gegner sein Verhalten im Zeitverlauf nicht ändert, kann nicht getroffen werden.
- Strategie auf oberer Ebene:
 - Veränderung des zeitl. Horizontes der Bewertung
 - Auswahl der Meta-Strategie
 - Auswahl des Vorhersagers
 - Veränderung der Größe der Historie der Frequenzanalyse und des History-Matchings
- Benutzung des Zeithorizontes, der im Gesamt-Match am erfolgreichsten war

Schlussfolgerungen

- **Komplexes Szenario des Opponent-Modelling hinter trivial wirkendem Spiel**
- **„Reine Anwendung“ des Opponent-Modelling**
- **Erkenntnisse nützlich auch bei anderen Spielen mit unvollkommener Information**

Anwendbarkeit auf Poker

- ◆ **Poker durch viel mehr Aspekte geprägt**
- ◆ **Trotzdem jede verfügbare Information wichtig**
- ◆ **Konkrete Situationen:**
 - **Vermeidung unnötiger Bluffs**
 - **Spiel mit schlechten Karten**
 - **Heads-Up**

Fragen?

Quellen

- [1] Billings, D. Thoughts on RoShamBo, 2000. ICGA Journal, Vol. 23, No. 1, pp. 3-8.
- [2] Egnor, D. locaine Powder, 2000. ICGA Journal, Vol. 23, No. 1, pp. 33-35.
- [3] Billings, D. The First International RoShamBo Programming Competition, 2000. ICGA Journal, Vol. 23, No. 1, pp. 42-50.
- [4] Billings, D. The Second International RoShamBo Programming Competition, 2001. <http://www.cs.ualberta.ca/~darse/rsbpc.html>.