

The Challenge of Poker

Björn Heidenreich

B.HEIDENREICH@OCEAN2.DE

1. The Challenge of Poker

Diese Arbeit stellt eine Zusammenfassung des Artikels "The challenge of poker"¹ dar. Sie beschränkt sich auf die wesentlichen Aspekte, welche für die Computer Poker Challenge im Sommersemester 2008 an der Technischen Universität Darmstadt von Interesse sind und soll eine Einordnung der anderen Seminarthemen ermöglichen.

Poker stellt ein interessantes Forschungsfeld der Künstlichen Intelligenz dar. Im Gegensatz zu anderen Spielen, wie z.B. Schach - wo Künstliche Intelligenzen bereits auf Weltklasse-Niveau spielen - existieren beim Poker nur unvollständige Informationen: Karten anderer Spieler sind nicht sichtbar. Deshalb reichen Methoden wie reine Tiefensuche nicht aus, um Poker auf einem ähnlichen Niveau spielen zu können. Aber noch andere Eigenschaften machen Poker für die Forschung interessant. So gibt es mehrere konkurrierende Agenten (Spieler). Desweiteren zeichnen Risikomanagement, Modellierung der Gegner und der Umgang mit nur unvollständigen oder nicht vorhandenen Informationen Poker aus.

Es gibt viele Ansätze, eine Künstliche Poker-Intelligenz zu entwickeln. In dieser Arbeit soll auf Poki, ein Programm der University of Alberta, näher eingegangen werden. Daneben gibt es viele weitere Ansätze um der Komplexität und Schwierigkeit des Poker Spielens Herr zu werden: Es kann z.B. mit vereinfachten Poker-Varianten gearbeitet werden, aber auch Subsets des Spiels können unabhängig voneinander betrachtet werden. Die Gefahr dieser Vereinfachungen liegt jedoch darin, dass sie das eigentliche Poker-Problem zerstören können. Ein anderer Ansatz wäre deshalb der, ein Programm zu entwickeln, welches gegen die besten realen Pokerspieler bestehen kann. Dieser Ansatz wird von Poki verfolgt.

2. Anforderungen an einen guten Poker-Spieler

Es können fünf verschiedene Anforderungen an einen guten Poker-Spieler gestellt werden. Diese müssen nicht explizit in der KI designed, sollten jedoch aber zumindest zufriedenstellend gelöst werden, um eine starke KI entwickeln zu können.

2.1 Hand Strength

Hand Strength misst die Stärke der Hand in Relation zu anderen. Eine einfache Lösung könnte aus einer Funktion mit den eigenen Karten und den community cards als Input bestehen. Eine bessere Funktion berücksichtigt unter anderem zusätzlich die verbleibende Anzahl an Spieler, die Position am Tisch sowie die Wahrscheinlichkeitsverteilung der gegnerischen Karten.

1. Billings, Davidson, Schaeffer, Szafron: The Challenge of Poker, Department of Computer Science, University of Alberta

2.2 Hand Potential

Mit Hand Potential wird die Wahrscheinlichkeit bezeichnet, dass eine Hand verbessert oder verschlechtert wird, nachdem weitere community cards gegeben werden. Beispielsweise kann eine Hand mit vier gleichen Farben eine geringe Hand Strength besitzen. Die Wahrscheinlichkeit mit einem Flush zu gewinnen, ist jedoch entsprechend hoch.

2.3 Bluffing

Bluffing ermöglicht es, auch mit einer schwachen Hand zu gewinnen. Um die optimale Bluffing-Frequenz in einer bestimmten Situation herauszufinden, kann die Spieltheorie verwendet werden. Ein minimaler Ansatz besteht darin, mit einer der Bluffing Frequenz entsprechenden Anzahl von Händen zu bluffen. Ausgefeiltere Systeme versuchen hingegen profitable Situationen herauszufinden, indem sie die gegnerische Hand Strength und ihre Folding-Wahrscheinlichkeit vorherzusagen versuchen.

2.4 Unberechenbarkeit

Die Unberechenbarkeit macht es den Poker-Gegnern schwer, ein akkurates Modell des eigenen Spielers zu formen. Indem der Stil des eigenen Spiels über die Zeit variiert wird, wird die Wahrscheinlichkeit erhöht, dass der Gegner aufgrund von falschen Annahmen Fehler macht.

2.5 Gegner-Modellierung

Opponent Modeling versucht eine Wahrscheinlichkeitsverteilung der Kartenpaare der gegnerischen Hand zu erhalten. Der einfachste Ansatz besteht darin, für jeden Gegner das gleiche Modell zu verwenden. Dieses kann im Laufe des Spiels durch die Modifikation der Wahrscheinlichkeitsverteilung aufgrund des Verlaufs der investierten Einsätze und gesammelten Statistiken der Gegner verbessert werden.

3. Pokis Architektur

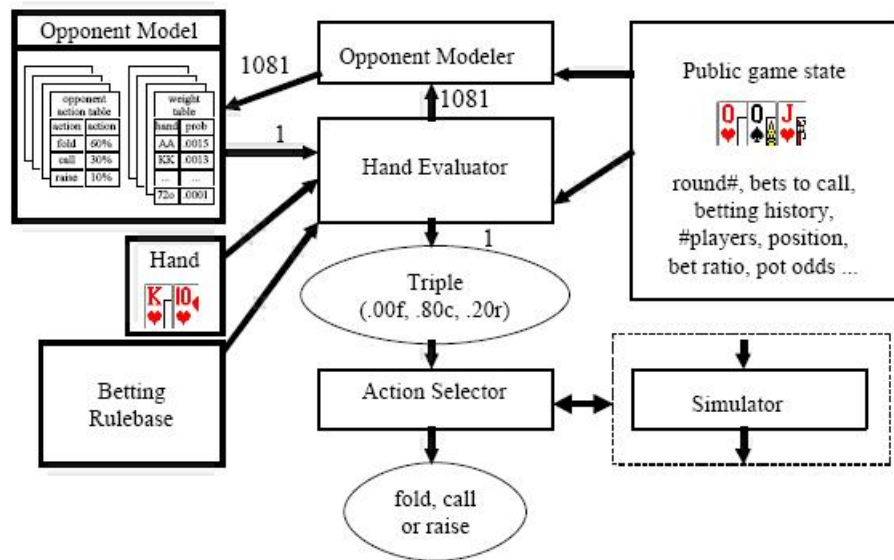


Abbildung 1: Pokis Architektur

Hand Evaluator dient dazu, die Güte der eigenen Hand zu bestimmen. Hierzu wird neben dem Kontext des Spiels (public game state) auch die Gegner-Modellierung mit einbezogen. Letztere versucht möglichst genau die Karten der Gegner und deren Strategie vorherzusagen. Der Action-Selector bzw. der Simulator schließlich wählt die eigene Aktion (fold, call oder raise) aus.

4. Spiel-Strategien

Spiel-Strategien vor und nach dem Flop sind sehr unterschiedlich: Vor dem Flop sind nur sehr wenige Informationen, welche zur Bestimmung des Einsatzes herangezogen werden können, vorhanden. Entsprechend wird nur ein einfaches System zur Bestimmung des Einsatzes benötigt. Post-Flop-Einsätze werden durch wesentlich mehr Faktoren determiniert. Eine entsprechende Strategie verwendet den vollständigen Kontext des Spiels, die eigene Hand und Gegner-Modelle zur Entscheidungsbildung.

4.1 Pre-Flop-Strategien

Es existieren 1326 ($52 \text{ über } 2$) verschiedene mögliche Kartenpaare für die private hand. Der Wert solch einer Hand wird income rate genannt und mittels einer roll-out Simulation, einer Offline-Berechnung, welche aus mehreren Millionen Händen besteht, bestimmt. Hierbei callen alle Spieler den ersten Einsatz. Im Anschluss werden alle weiteren Karten ohne einen Einsatz der Spieler verteilt. Diese sehr unrealistische always call assumption liefert nicht zwangsweise ein akkurates Ergebnis für den Wert eines Kartenpaares. Jedoch reicht das Ergebnis für eine relative Gewichtung der Karten in der Pre-Flop-Phase.

4.1.1 VERGLEICH VON PRE-FLOP-STRATEGIEN

David Slansky beschreibt in seinem Buch "Texas Hold'em for the Advanced Player" ein Klassifikationsschema für die private hand mit einer hohen Korrelation zwischen seinem Ranking und den roll-out Simulationen. Es gibt jedoch keine Bewertung von Karten, welche in allen Situationen zutrifft. Ein fortgeschrittener Spieler wird seine Strategie den vorangegangenen Spielen anpassen und damit die Bewertung entsprechend beeinflussen. Desweiteren hängt der Kartenwert auch von dem Einsatz der vorherigen Spieler ab. Eine Hand wird beispielsweise anders gewertet, nachdem alle vorherigen Spieler gefolded haben, als wenn alle gecallt haben.

Group 1		Group 2		Group 3		Group 4	
+2112	AA*	+714	TT*	+553	99*	+481	T9s
+1615	KK*	+915	AQs	+657	JTs	+515	KQo
+1224	QQ*	+813	AJs	+720	QJs	+450	88*
+935	JJ*	+858	KQs	+767	KJs	+655	QTs
+1071	AKs	+718	AKo	+736	ATs	+338	98s
				+555	AQo	+449	J9s
						+430	AJo
						+694	KTs
Group 5		Group 6		Group 7		Group 8	
+364	77*	+304	66*	+214	44*	-75	87o
+270	87s	+335	ATo	+92	J9o	+87	53s (> 43s)
+452	Q9s	+238	55*	+41	43s	+119	A9o
+353	T8s	+185	86s	+141	75s	+65	Q9o
+391	KJo	+306	KTo	+127	T9o	-129	76o
+359	QJo	+287	QTo	+199	33*	-42	42s (< 52s)
+305	JTo	+167	54s	-15	98o	-83	32s (< 52s)
+222	76s	+485	K9s	+106	64s	+144	96s
+245	97s	+327	J8s	+196	22*	+85	85s
+538	A9s			+356	K8s	-51	J8o
+469	A8s			+309	K7s	+206	J7s
+427	A7s			+278	K6s	-158	65o
+386	A6s			+245	K5s	-181	54o
+448	A5s			+227	K4s	+41	74s
+422	A4s			+211	K3s	+85	K9o
+392	A3s			+192	K2s	-10	T8o
+356	A2s			+317	Q8s		
+191	65s						

Abbildung 2: Vergleich von Slankys Rating und roll-out Simulation

Die Tabelle vergleicht Slankys Bewertung mit einer roll-out Simulation. Der Buchstabe s steht hierbei für "suited hand", also ein Kartenpaar gleicher Farbe, o steht für "offsuit hand", zwei Karten unterschiedlicher Farbe, und * markiert zwei Karten mit dem gleichen Wert. Die Tabelle ist nach Slansky in acht Gruppen unterteilt, wobei Gruppe 1 die beste und Gruppe 8 die schlechteste Hand, welche nur unter bestimmten Umständen gespielt werden sollte, darstellt. Im Allgemeinen lässt sich eine starke Korrelation zwischen Slankys Ranking und den income rates der roll-out Simulationen feststellen.

Allerdings zeigt die Simulation eine Verzerrung zu Gunsten von Karten, die gut gegen mehrere Spieler gespielt werden können. Dies sind in der Regel Karten welche leicht zu

entsprechend starken ausgebaut werden können, wie z.B. flush, straight, three of a kind, usw. Da bei einer roll-out Simulation alle Spieler im Spiel bleiben, muss die durchschnittliche Gewinner-Hand deutlich stärker sein, als in einem realen Spiel. Aus dem gleichen Grund sind hohe Paare entsprechend unterbewertet. Umgekehrt liefert Slankys Ranking eine Verzerrung zu Gunsten von sogenannten Connectors unterschiedlicher Farben, also Karten, die sich im Wert nur um 1 unterscheiden (bspw. Herz 7 und Kreuz 8). Desweiteren existieren in Slankys Wertung auch logische Fehler. So befindet sich zum Beispiel 43s in Gruppe 7 vor 53s in Gruppe 8. Es kann jedoch gezeigt werden, dass letzteres Paar 43s dominiert.

Da der Unterschied zwischen roll-out Simulation und Slankys Rating nicht allzu groß ist, lässt sich eine roll-out Simulation gut nutzen um den Pre-Flop-Kartenwert zu bestimmen. Von der Benutzung Slankys Ratings sei abgeraten, da einerseits menschliches Wissen ausgeschlossen werden sollte und andererseits eine roll-out Simulation spezifisch für verschiedene Situationen durchgeführt werden kann. Slankys Klassifikation wiederum gilt nur für bestimmte Spielsituationen.

4.1.2 ITERIERTE ROLL-OUT SIMULATIONEN

Eine Verbesserung der roll-out Simulation stellt die iterierte roll-out Simulation dar. Hierbei wird die Simulation mehrmals ausgeführt, wobei die vorherigen Ergebnisse die nachfolgenden Simulationen beeinflussen. Erhält man beispielsweise für einen Spieler einen negativen Wert, so wird davon ausgegangen, dass er die Hand folded. Dies führt zu einer realistischeren Verteilung, da im Gegensatz zur einfachen roll-out Simulation Spieler aus dem Spiel ausscheiden können. Es werden verschiedene Simulationsrunden durchgeführt, solange, bis das Ergebnis gegen ein Gleichgewicht konvergiert. Man erhält so eine Menge von gewinnträchtigen Karten, also eine Entscheidungshilfe ob gespielt werden sollte oder nicht. Dieses Verfahren ist noch lange nicht perfekt, aber doch besser wie die roll-out Simulation.

Eine weitere Verbesserung besteht darin, einen Noise-Faktor einzuführen. Damit erhalten auch Karten mit negativer Bewertung eine Chance gespielt zu werden.

Hand	IR-10	Iterated	Hand	IR-10	Iterated	Hand	IR-10	Iterated
AA*	+2112	+2920	ATs	+736	+640	KQo	+515	+310
KK*	+1615	+2180	99*	+553	+630	QTs	+655	+280
QQ*	+1224	+1700	KQs	+858	+620	QJs	+720	+270
JJ*	+935	+1270	AQo	+555	+560	A9s	+538	+220
TT*	+714	+920	KJs	+767	+480	ATo	+335	+200
AKs	+1071	+860	88*	+450	+450	KTs	+694	+190
AKo	+718	+850	77*	+364	+390	KJo	+391	+160
AQs	+915	+780	AJo	+430	+380	A8s	+469	+110
AJs	+813	+680	JTs	+657	+360	66*	+304	+40

Abbildung 3: Vergleich roll-out Simulation und iterierte roll-out Simulation

Die Ergebnisse solch eines Versuches sind in der obigen Tabelle zusammengefasst. IR-10 bezeichnet die income-rates für eine roll-out Simulation mit zehn Spielern, iterated die Ergebnisse einer iterierten roll-out Simulation. Hierbei sind die Werte in Milli-Bets angegeben: Karten mit einem Wert von +1000 gewinnen im Durchschnitt jedes Mal 1 Small Bet. Im Gegensatz dazu stellen die Werte der simplen roll-out Simulation nur ein relatives Maß dar.

4.2 Post-Flop-Strategien

Die Entscheidung für den Einsatz nach dem Flop kann beispielsweise wie folgt getroffen werden:

1. Effective hand strength (EHS) von Pokis Hand berechnen (Hand Evaluator).
2. Mittels des Spiel-Kontexts, Regeln und Funktionen lässt sich die EHS in ein Wahrscheinlichkeits-Tripel übersetzen: $\{P(\text{fold}), P(\text{call}), P(\text{raise})\}$
3. Generiere eine Zufallszahl zwischen 0 und 1 und wähle mit ihr eine Entscheidung entsprechend der Wahrscheinlichkeitsverteilung (Action Selector).

EHS ist ein Maß, wie gut Pokis Karten in Relation zu den verbleibenden Spielern und den community cards sind.

4.2.1 HAND STRENGTH (HS)

HS ist die Wahrscheinlichkeit mit der eine gegebene Hand besser ist, als die eines Gegners. Angenommen die Wahrscheinlichkeit möglicher Kartenpaare des Gegners ist gleichverteilt, so lassen sich diese aufzählen und wie folgt gewichten: Wenn Pokis Hand besser ist +1, bei unentschieden +0.5 und wenn sie schlechter ist 0. Werden diese Werte nun aufsummiert und schließlich durch die Zahl aller möglichen Hände des Gegners geteilt, so erhält man die ungewichtete Hand Strength.

Angenommen Pokis Hand ist Karo Ass und Kreuz Dame, der Flop ist Herz Bube, Kreuz 4 und Herz 3 und es gibt nur einen aktiven Gegner. Es verbleiben also 47 unbekannte Karten und damit 1081 (47 über 2) mögliche Kartenpaare des Gegners. In unserem Beispiel ist jedes three of a kind, two pair, one pair oder Ass König besser (444 Fälle), andere Ass Dame Kombinationen sind gleichwertig (9 Fälle) und die restlichen Kombinationen (628 Fälle) sind schlechter. Die Hand Strength beträgt folglich $(0 \cdot 444 + 0,5 \cdot 9 + 1 \cdot 628)/1081 = 0,585$. Mit anderen Worten: Es besteht eine 58,5 prozentige Wahrscheinlichkeit, dass das Kartenpaar Karo Ass und Kreuz Dame besser ist als eine zufällige Hand. Sind mehrere Gegner vorhanden, so muss die Hand Strength einfach mit der entsprechenden Gegner-Anzahl potenziert werden. Bei fünf Gegnern beträgt die HS in unserem Beispiel $0,585^5 = 0,069$.

4.2.2 HAND POTENTIAL (HP)

Nach dem Flop verbleiben immer noch zwei aufzudeckende Karten, nach dem Turn noch eine. Diese sollen nun vorhergesagt werden. Positive Potential (PPot) ist hierbei die Wahrscheinlichkeit, dass eine Hand, welche momentan nicht die beste ist, im Showdown gewinnt. Negative Potential (NPot) bezeichnet analog die Wahrscheinlichkeit, dass eine momentan führende Hand im Showdown verliert.

PPot und NPot werden berechnet, indem alle möglichen Karten der Gegner und des Boards aufgezählt werden. Für alle Kombinationen zählt man die Fälle, in denen die eigene Hand nicht die beste ist, aber am Ende gewinnt (PPot) und die Fälle in denen sie die beste ist, aber am Ende verliert (NPot).

A♠-Q♣ hole cards		J♥-4♣-3♥ board cards		
5 Cards	7 Cards			
	Ahead	Tied	Behind	Sum
Ahead	449,005	3,211	169,504	628x990 = 621,720
Tied	0	8,370	540	9x990 = 8,910
Behind	91,981	1,036	346,543	444x990 = 439,560
Sum	540,986	12,617	516,587	1,081x990 = 1,070,190

Abbildung 4: Beispiel für Hand Potential

Anhand der Tabelle lässt sich die Berechnung leicht erläutern: Benutzen wir die Form T (Zeile, Spalte) und B für Behind, T für Tied, A für Ahead und S für Sum, so ergeben sich die folgenden Formeln:²

$$PPot = \frac{T(B,A)+T(B,T)/2+T(T,A)/2}{T(B,S)+T(T,S)/2}$$

$$NPot = \frac{T(A,B)+T(A,T)/2+T(T,B)/2}{T(A,S)+T(T,S)/2}$$

Wenn die in der Tabelle gegebenen Karten besser als die des Gegners nach 5 gegebenen Karten sind, dann beträgt die Wahrscheinlichkeit $449005/621720 = 72\%$, dass sie es auch noch nach 7 Karten sind.

Die entsprechenden Potentiale nach dem Pot zu berechnen kann sehr aufwendig sein, weshalb meistens eine schnellere Berechnung benutzt wird, z.B. wird nur die nächste aufzudeckende Karte betrachtet.

4.2.3 EFFECTIVE HAND STRENGTH (EHS)

EHS kombiniert Hand Strength und Hand Potential in einem Wert um eine einzelne Metrik für die relative Stärke einer Hand zu bekommen. Ein möglicher Ansatz für die Gewinn-Wahrscheinlichkeit ist der folgende:

$$\begin{aligned} P(\text{Gewinn}) &= P(\text{gute Hand}) \cdot P(\text{Gegner verbessert sich nicht}) \\ &\quad + P(\text{keine gute Hand}) \cdot P(\text{Hand verbessert sich}) \\ &= HS \cdot (1 - NPot) + (1 - HS) \cdot PPot \end{aligned}$$

Nun interessiert uns aber lediglich die Wahrscheinlichkeit, dass unsere Hand die beste ist, oder sich zur besten verbessert. Wir können also $NPot = 0$ setzen:

$$EHS = HS + (1 - HS) \cdot PPot$$

Für n verbleibende Gegner lässt sich diese Formel verallgemeinern zu:

$$EHS = HS^n + (1 - HS^n) \cdot PPot$$

Diese Formel nimmt an, dass die entsprechenden Variablen für alle Gegner gleich sind. Dieser Ansatz ist jedoch nicht gut geeignet, da verschiedene Spieler verschiedene Stile besitzen. Wir führen also für jeden Spieler eine eigene HS und PPot ein:

$$EHS_i = HS_i + (1 - HS_i) \cdot PPot_i$$

2. vgl. <http://www.cs.ualberta.ca/~jonathan/Grad/Papers/ai98.poker.html>, 27. März 2008

4.2.4 WEIGHT TABLES

Die bisherige Berechnung von HS und HP geht davon aus, dass alle Kombinationen von Kartenpaaren gleich wahrscheinlich sind. Im Poker-Spiel ist dies jedoch in der Regel nicht der Fall. Zum Beispiel ist die Wahrscheinlichkeit, dass ein Gegner nach dem Flop Ass und König auf der Hand hat größer, als dass er eine sieben und eine zwei besitzt; da die meisten Spieler mit sieben und zwei vor dem Flop folden würden.

Um dies zu berücksichtigen, besitzt Poki eine weight table für jeden Gegner, welche die Wahrscheinlichkeit für jedes Kartenpaar zu einem Zeitpunkt des Spiels beinhaltet. Entsprechend wird bei der Berechnung von HS und HP jedes mögliche Kartenpaar mit seinem Gewicht multipliziert. Uns interessieren nur relative Werte, so dass es sich bei den Gewichten nicht um eine Wahrscheinlichkeitsverteilung handeln muss. Ihre Summe muss also nicht zwingend 1 ergeben. Deshalb verwendet Poki eine Zahl zwischen 0 und 1 als Gewicht. Am Anfang einer neuen Runde wird jedes Kartenpaar-Gewicht mit 1 initialisiert. Wenn mehr Karten sichtbar werden, werden entsprechende Paare unmöglich und deren Gewichte werden auf 0 gesetzt.

Nach jedem Einsatz werden die Gewichte des entsprechenden Gegners neu berechnet, dies wird re-weighting genannt. Das entsprechende Gewicht wird anhand der Aktion des Spielers gesenkt oder erhöht. Die mögliche Hand jedes Gegners wird mittels dieser Gewichte neu berechnet und eine Wahrscheinlichkeitsverteilung der gegnerischen Aktionen erstellt. Mittels dieser werden die Gewichte nach jeder Aktion entsprechend berichtigt.

4.2.5 WAHRSCHEINLICHKEITS-TRIPPEL

Gesucht ist ein Wahrscheinlichkeits-Tripel $P = \{P(fold), P(call), P(raise)\}$, so dass $P(fold) + P(call) + P(raise) = 1$. Es repräsentiert die Wahrscheinlichkeiten der entsprechenden Aktionen. Hand Strength, Hand Potential und Effective Hand Strength stellen hierbei nur einige der Informationen, welche zur Berechnung der Wahrscheinlichkeiten nötig sind, zur Verfügung. Weitere beeinflussende Faktoren sind z.B. pot odds, relative Position am Tisch, Verlauf der bisherigen Einsätze.

Ein professioneller Spieler bewertet seine Optionen beispielsweise nach dem erwarteten Return on Investment. Hierzu kann eine Regel- oder Formel-basierte Strategie verwendet werden. Ein großer Vorteil der Verwendung dieses Wahrscheinlichkeits-Trippels besteht darin, dass verschiedenartiges Wissen in einer Information zusammengefasst wird.

4.3 simulationsbasierte Strategien

Es ist zeitintensiv und (fast) unmöglich, einen Experten alle nötigen Regeln zur Platzierung des Einsatzes identifizieren zu lassen. Das Spiel ist sehr komplex und Entscheidungen müssen im jeweiligen Kontext getroffen werden. Aus diesem Grunde müssen dynamische, adaptive Techniken eingesetzt werden um eine gute Poker-Software generieren zu können.

Eine wissensbasierte Strategie ist sinngemäß das gleiche wie eine evaluation function in deterministischen Spielen mit vollständigen Informationen: Vom aktuellen Kontext des Spiels würde sie die Aktionen wählen, welche zu dem besten Ergebnis führen. Es findet also eine Suche statt. Dies ist bei Poker jedoch nicht möglich, da grundlegende Unterschiede in Game-Trees mit unvollständigen Informationen und der Anzahl der zu betrachtenden Möglichkeiten existieren.

Poki benutzt deshalb eine simulations-basierte Strategie: Es werden viele Szenarien simuliert und berechnet, wie viel Geld verloren oder gewonnen wird. Jedesmal wenn Poki eine Entscheidung treffen muss, wird dieser Simulator verwendet, um einen Erwartungswert (expected value, EV) für den Einsatz zu gewinnen. Hierbei werden von dem aktuellen Spiel-Kontext ausgehend verschiedene Aktionen bis zum Ende durchgespielt. In Abbildung 1 ersetzt der Simulator hierbei den Action Selector. Jede Runde wird zwei Mal simuliert um die Konsequenzen von check oder call und bett oder raise zu simulieren. Die gegnerischen Karten werden hierbei entsprechend der weight table angenommen. Der durchschnittliche Gewinn bei call oder check wird call EV, der durchschnittliche Gewinn bei bet oder raise wird raise EV genannt. Fold EV lässt sich einfach berechnen, da für die Hand keine zukünftigen Gewinne oder Verluste existieren. Die aktuelle Implementierung von Poki wählt die Aktion mit der größten Gewinn-Erwartung aus. Wenn die Erwartung zweier Aktionen gleich ist, wird die aggressivere gewählt. Um die Unvorhersehbarkeit des Programms zu steigern, könnte man bei ähnlich hohen Erwartungen die Aktion zufällig wählen. Jedoch birgt das Rauschen in den Simulationsdaten schon eine entsprechende Zufälligkeit.

Ein Problem besteht darin, dass sich aufgrund der Komplexität nicht alle Kartenverteilungen und Aktionen der Gegner simulieren lassen. Poki verwendet Opponent Modeling um dieses Problem zu umgehen indem die Aktionen der Gegner vorhergesagt werden. Die Simulation erlaubt komplexe Strategien ohne entsprechendes Expertenwissen zu benötigen.

5. Gegner-Modellierung

Jede Poker-Strategie benötigt eine gute Modellierung des Gegners, um dessen Schwächen identifizieren und ausnutzen zu können. Da jeder Gegner andere Strategien benutzt und diese gegebenenfalls im Laufe des Spiels wechselt, muss diese Modellierung adaptiv erfolgen.

5.1 Statistisch-basierte Gegner-Modellierung

Opponent Modeling wird beim Poker in mindestens zwei Weisen genutzt: Einerseits soll auf die Stärke der gegnerischen Karten geschlossen, andererseits sollen ihre Aktionen vorhergesagt werden. Im Zentrum steht hierbei wieder das Wahrscheinlichkeits-Tripel $\{P(\text{fold}), P(\text{call}), P(\text{raise})\}$, welches den Spiel-Kontext in eine Wahrscheinlichkeit der gegnerischen Aktionen abbildet.

Ein Weg besteht darin, die eigene Strategie zu unterstellen. Dieses Verfahren heißt generic opponent modeling (GOM). Ein anderes Verfahren basiert auf dem vergangenen Verhalten des einzelnen Gegners: specific opponent modeling (SOM). Beispielsweise wird beobachtet, dass ein Gegner in 40% der Fälle direkt nach dem Flop seinen Einsatz macht. Es lässt sich annehmen, dass er in dieser Spielsituation mit den Top 40% der Kartenpaare einen Bet ausführt. Soll das Verhalten der Gegner in bestimmten Situationen gelernt werden, so tritt die folgende Problematik auf: Wird der Kontext zu weit definiert, so lernt das System vielleicht nicht alle wichtigen Aspekte des Gegners. Wird er zu eng definiert, so lernt das System nur sehr langsam und manches vielleicht überhaupt nicht, da die zu lernenden Situationen zu selten auftreten.

Opponent Modeling beim Poker besitzt viele Eigenschaften, welches es für maschinelles Lernen so schwierig macht, gute Algorithmen zu entwickeln: Rauschen, Unberechenbarkeit

der Gegner, Forderung nach schnellem Lernen und den Anspruch von relativ wenigen Trainings-Beispielen gute Vorhersagen zu bilden.

5.2 Gegner-Modellierung basierend auf neuronalen Netzen

Um ein allgemeineres System zur Modellierung des Gegners zu erhalten, implementiert Poki ein neuronales Netz für die Vorhersage der gegnerischen Aktionen in jedem Kontext. Als Input dienen Eigenschaften des Spiel-Kontexts, welche sich auf die Spieler-Entscheidungen auswirken können oder mit ihnen korreliert sind. Der Output-Layer besteht aus drei Knoten, welche für fold, call und raise stehen. Legt man die entsprechenden Werte auf die Input-Knoten, so erhält man durch Normalisierung der Werte der Output-Knoten das bereits bekannte Wahrscheinlichkeits-Tripel.

Nachdem das Netz mittels verschiedener Gegner trainiert wurde, sieht man, dass manche Input-Knoten für die Vorhersage dominant sind, während andere fast keine Rolle spielen. Die Vorhersagegenauigkeit kann mittels Cross-Validation der realen Spielzüge gemessen werden.³ Man erhält so eine relativ kleine Klasse an wichtigen Faktoren, welche die statistische Gegner-Modellierung signifikant verbessert.

6. Performance Evaluation

Auf die Performance Evaluation soll an dieser Stelle nur kurz eingegangen werden. Die Performanz einer Poker-KI ist nur schwer messbar. Poki ist ein komplexes System, wo kleine Änderungen der Software zu großen, unvorhergesehenen Auswirkungen führen können.

Der Faktor Glück hat bei Poker einen großen Einfluss, was es schwer macht, entsprechende Software miteinander zu vergleichen. Ebenso handelt es sich um adaptive Systeme, welche sich auf die Gegner einstellen.

Eine Methode neue Features zu testen besteht darin, die Software gegen eine alte Version ihrer selbst spielen zu lassen. Für die Evaluation der Spielstärke ist dieser Ansatz aber nur bedingt geeignet, da verschiedene Spielstile nicht abgedeckt werden. Die Software kann beispielsweise gute Ergebnisse gegen sich selbst erzielen, aber gegen einen bestimmten Spieltyp versagen. Die Streuung dieser Strategien ist bei menschlichen Spielern größer als bei künstlichen, weshalb es sinnvoll ist die Software gegen menschliche Spieler antreten zu lassen. Dies kann zum Beispiel mittels eines Java Web Applets erfolgen. Hier herrscht allerdings keine kontrollierte Umgebung: Ein gutes Resultat kann darin begründet liegen, dass lediglich gegen schwache Gegner gespielt wurde. Selbst leichte Verbesserungen lassen sich erst nach dem Spielen von mehreren tausend Händen gegen verschiedene Gegner erkennen. Ein Maß für die Performanz besteht zum Beispiel in den durchschnittlich gewonnen small bets pro Hand (sb/hand): Bei \$10/\$20 Hold'em mit 40 Händen pro Stunde und +0,05 sb/hand gewinnt der Spieler durchschnittlich \$20 die Stunde. Je nach Gegnergruppen (z.B. sehr gute, mittlere, schwache Spieler) kann sich sb/hand stark unterscheiden. Beispielsweise kann es sein, dass ein Programm gegen starke Gegner gut spielen kann, aber gegen mittlere Gegner schlechter abschneidet, da es deren Aktionen schlechter voraussagen kann. Bei mindestens 20.000 gespielten Händen gegen frühere Versionen der Software kann eine Verbesserung um +0,05 sb/hand als signifikant angenommen werden.

3. vgl. Davidson, Billings, Schaeffer, Szafron: Improved opponent modeling in poker, in International Conference on Artificial Intelligence, Seiten 1467 - 1473, 2000

7. Fazit

Poker ist ein komplexes Spiel mit vielen Anforderungen. Um erfolgreich zu sein, müssen die Aktionen des Spielers einerseits unvorhersagbar sein, andererseits müssen die gegnerischen Aktionen jedoch bestmöglich vorhergesagt werden.

Bei der Entwicklung von Poki wurde eine zyklische Vorgehensweise benutzt. Eine Komponente wurde so lange verbessert, bis eine andere den Performanzengpass darstellte. Aktuell sind manche Funktionen in Poki noch extrem einfach und bestehen beispielsweise aus einer einfachen Konstante, was die Performanz jedoch nicht negativ beeinflusst. Trotz den Fortschritten, gerade auch im Bereich der Gegner-Modellierung, existieren noch zahlreiche möglichen Verbesserungen in der Funktionsweise von Poki.

Es ist möglich umfangreiche Daten über seine Gegner zu sammeln. Das Problem besteht darin, diese auszuwerten und die nützlichen Features zu finden. Gegebenfalls sagt eine einfachere Metrik das Verhalten der Gegner besser voraus, als eine komplexere, welche mehr Features mit einbezieht.

Momentan besteht in diesem Bereich noch viel Forschungsbedarf.