

Masterarbeit zum Thema

Ein selbstkalibrierender Spieler für Skat

Fabian Knorr

24. April 2018



Lehrstuhl für Mathematik
mit Schwerpunkt Digitale Bildverarbeitung

Autor

Fabian Knorr
Matrikelnummer 64370
knorrfab@fim.uni-passau.de

Betreuung und Erstprüfung

Prof. Dr. Tomas Sauer
Lehrstuhl für Mathematik mit Schwerpunkt Digitale Bildverarbeitung
Universität Passau

Zweitprüfung

Prof. Dr. Tobias Kaiser
Lehrprofessur für Reine Mathematik
Universität Passau

Danksagung Besonderer Dank gilt Dr. Peter Heinlein für die hervorragende fachliche Beratung und Unterstützung während der Forschungsarbeit, der Isar Interactive GmbH & Co. KG für die Bereitstellung von Material, Software und Infrastruktur sowie Daniel Schäfer für den Beitrag von Expertenwissen und Validierungsdaten.

Ich versichere hiermit, dass ich diese Arbeit selbstständig verfasst, noch nicht anderweitig zu Prüfungszwecken vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt, sowie wörtliche und sinngemäße Zitate auch als solche gekennzeichnet habe.

24. April 2018

Fabian Knorr

Inhaltsverzeichnis

1	Einleitung	1
1.1	Inhaltsübersicht	2
1.2	Skat	3
1.3	Spieltheoretische Optimierung	5
1.4	Auswertung offener Situationen	6
1.5	Spielfluss	7
1.6	Existierende Literatur	8
1.7	Testdaten	9
2	Perfect Information Monte Carlo Search	11
2.1	Grundlagen	12
2.2	Tischgleichverteilung	14
2.3	Kombinatorische Simulation	16
2.4	Stichprobenauswertung	19
2.5	Zugwahl	20
2.6	Schwächen des Verfahrens	23
2.7	Zusammenfassung	24
3	Zustandsinferenz	25
3.1	Inferenz anhand eines Ausspielmoduls	26
3.2	Merkmalbasierte Inferenz	27
3.3	Simulation der Wahrscheinlichkeitsverteilung	31
3.4	Inferenz in der Spielfindungsphase	32
3.5	Inferenz in der Ausspielphase	41
3.6	Evaluation	45
3.7	Zusammenfassung	47
4	Spielstärkemaße	49
4.1	Mittelung der Listenpunkte	50
4.2	Perfect Information Post Mortem Analysis	52
4.3	Monte Carlo Post Mortem Analysis	57
4.4	Vergleich der Verfahren	61

4.5 Vergleich mit einem professionellen Spieler	64
4.6 Zusammenfassung	65
5 Adaptive Spielweise	67
5.1 Maximale Spielstärke des Computers	67
5.2 Verrauschen der Stichprobenergebnisse	71
5.3 Schrittweises Anpassen der Spielstärke	76
5.4 Zusammenfassung	76
6 Ausblick	77
6.1 Zusammenfassung	77
6.2 Weitere Forschung	78
Anhang	i
Abkürzungsverzeichnis	i
Abbildungsverzeichnis	ii
Literaturverzeichnis	iv

1 Einleitung

Kartenspiele wie Skat oder Bridge sind ein interessantes Thema in der Forschung zu künstlicher Intelligenz, da sie Unsicherheit über die Spielsituation mit einer starken Strategiekomponente verbinden. Im Gegensatz zu klassischen Domänen der KI-Forschung wie Schach bleiben dem Spieler in dieser Disziplin Teile des Zustands in Form der gegnerischen Karten verborgen.

Dieser Umstand führt zu einer Zufallskomponente, die auf stochastischen Verfahren basierende Strategien erfordert. So ist bereits einige Forschungsarbeit in die Implementierung starker Computerspieler auf diesem Gebiet geflossen; Kupferschmid und Helmert (2006), Ginsberg (1999) und Long (2011) kombinieren Spielbaumsuche mit Monte-Carlo-Simulation, um Bridge- und Skatspieler auf Expertenniveau zu implementieren. Diese Technik bildet auch die Grundlage der vorliegenden Arbeit.

Fähige Computerspieler sind nicht nur von akademischem Interesse, sondern kommen auch zu Unterhaltungszwecken als Gegner für Amateur- bis Profispieler zum Einsatz. Dieser Anwendungszweck bietet ein neues Entwicklungsziel: Der technisch bestmöglichen Spielweise ist nur ein Experte gewachsen, während dem Einsteiger keine realistische Gewinnchance bleibt. Um Herausforderung und Freude am Spiel zu balancieren, muss sich die Leistung eines geeigneten Computergegners deshalb einem menschlichen Spieler anpassen können.

Einen Skatspieler mit diesen Fähigkeiten zu entwickeln ist das Ziel dieser Arbeit. Dazu muss die Stärke eines menschlichen Spielers in annehmbarer Zeit korrekt eingeschätzt werden, um anschließend die Spielweise des Computers entsprechend zu kalibrieren. Das Einschätzen der Spielerfähigkeiten wird durch den Zufallsaspekt des Skatspiels erschwert; eine Reduktion der Spielstärke erfordert Finesse, um keine offensichtlichen Anfängerfehler zu verursachen.

1.1 Inhaltsübersicht

In dieser Arbeit wird ein starker Skatspieler entwickelt, der auf der in der Literatur etablierten Perfect-Information-Monte-Carlo-Suche basiert. Um den naiven Algorithmus aufzuwerten, wird er durch merkmalsbasierte Zustandsinferenz erweitert, die ihm eine verbesserte Schätzung der Kartenverteilung und damit eine präzisere Zugwahl erlaubt. Mit einem in dieser Arbeit neu entwickelten, probabilistischen Zugbewertungsverfahren wird es dem Spieler außerdem ermöglicht, die Spielstärke seiner Kontrahenten nach wenigen beobachteten Partien einzuschätzen und die eigene Stärke dem gewünschten Niveau anzupassen.

Einleitung Das erste Kapitel stellt die wichtigsten Grundlagen des Skatspiels und die spieltheoretischen Überlegungen hinter der KI-Entwicklung vor. Es wird auf eine existierende Möglichkeit zur effizienten Auswertung offener Spielsituationen eingegangen und das Entwicklungsziel des *Spielflusses* diskutiert. Schließlich wird eine Übersicht über die existierende Literatur auf dem Entwicklungsgebiet gegeben und die Testdaten vorgestellt, die zur Evaluation verwendet werden.

Perfect Information Monte Carlo Search In Kapitel 2 wird die für die Arbeit grundlegende Perfect-Information-Monte-Carlo-Suche (kurz PIMC-Suche) zur Zugbewertung vorgestellt. Für die praktische Implementierung wird ein neues Verfahren zur Simulation der betrachteten Wahrscheinlichkeitsverteilungen entwickelt und verschiedene Möglichkeiten zur Konsolidierung der Stichprobenergebnisse verglichen. Zuletzt werden theoretische und praktische Schwächen der PIMC-Suche diskutiert.

Zustandsinferenz In Kapitel 3 wird die PIMC-Suche um merkmalsbasierte Zustandsinferenz auf beobachteten Zügen der Mitspieler erweitert, um die Kartenverteilung besser schätzen zu können. Dazu wird aus einer umfangreichen Datenbank aufgezeichneter Spiele ein Modell erstellt, mit dem die Wahrscheinlichkeitsverteilung hinter der Monte-Carlo-Simulation an gelernte Charakteristiken angepasst wird. Zunächst wird ein existierendes Inferenzverfahren auf Basis der Spielansage vorgestellt, verbessert und die Annahmen des Verfahrens verifiziert. Anschließend wird der Ansatz verallgemeinert und zum ersten Mal auf Züge in der Ausspielphase erweitert. Schließlich wird der Gewinn durch die Zustandsinferenz anhand eines geeigneten Maßes quantifiziert.

Spielstärkemaße Um die Spielstärke eines menschlichen Spielers einschätzen zu können, wird in Kapitel 4 die herkömmliche Größe der *Listenpunkte* sowie ein alternativer Ansatz auf Basis von Spielfehlern aus der Literatur vorgestellt. Mit diesem Wissen

wird auf Grundlage der PIMC-Suche ein neues Spielstärkemaß entwickelt, das Spielerfähigkeiten deutlich schneller einschätzen kann als die existierende Methoden. Die vorgestellten Verfahren werden dann anhand von echten Spielerdaten verglichen.

Adaptive Spielweise In Kapitel 5 wird die maximale Spielstärke des implementierten PIMC-Spielers evaluiert. Anschließend wird zum ersten Mal ein Verfahren entwickelt, um die Leistung des Computerspielers auf ein vorgegebenes Niveau einzustellen und dabei eine möglichst plausible Spielweise beizubehalten. Mit diesem Schritt werden die Einzelkomponenten der Arbeit zusammengeführt.

Ausblick Im letzten Kapitel werden die Ergebnisse dieser Arbeit zusammengefasst und Fragestellungen für zukünftige Forschung skizziert, die während der Umsetzung des Spielers aufgekommen sind.

1.2 Skat

Da im Rahmen dieser Arbeit großzügig Gebrauch von Begriffen aus der Welt des Skat gemacht wird, wird hier kurz die Spielmechanik zusammen mit den relevanten Regeln vorgestellt.

Skat ist ein stichbasiertes Spiel für drei Spieler, die sich in jedem Spiel neu in zwei Parteien zusammenfinden: Dem *Alleinspieler*, der die Spielart bestimmt, und der *Gegenpartei* aus den beiden verbleibenden Spielern, die eine Allianz für die Zeit der Partie bilden um den Sieg des Alleinspielers zu verhindern.

Skat wird mit einem französischen oder deutschen Blatt mit 32 Karten gespielt. Jeder Spieler erhält zu Beginn zehn Karten, die verbleibenden zwei werden verdeckt in den namensgebenden *Skat* gelegt, den der Alleinspieler optional aufnehmen darf.

Zunächst wird in der **Spielfindungsphase** die Partezuordnung und die Spielart festgelegt. Abhängig vom Blatt und der eigenen Risikobereitschaft geben alle Spieler im Schritt des *Reizens* Gebote ab, um die Position des Alleinspielers zu erlangen.

Ist die Spielerverteilung entschieden, so darf der Alleinspieler optional den Skat aufnehmen und anschließend wieder zwei Karten ablegen (auch *Drücken* genannt). Entschieden er sich dafür, den Skat nicht aufzunehmen, spricht man von einem *Handspiel*. Schließlich sagt der Alleinspieler die Spielart für diese Partie an.

Dabei existieren sechs Spielarten: Der *Grand*, in dem nur Buben Trumpf sind, *Farbspiele* für jede der Trumpffarben ♣, ♠, ♥, ♦ und das *Nullspiel*, in dem der Alleinspieler gewinnt, indem er keinen Stich macht. Optional kann der Alleinspieler sein Spiel *Ouvert* (offen) deklarieren, was bedeutet, dass er seine Karten für das Spiel aufdeckt.

Karten, die nicht Teil der Trumpffarbe sind, werden als *Beiblatt* bezeichnet. Das Ausweisen einer Kartenmenge als Trumpf motiviert den Begriff der *Farbklasse*:

Definition 1.1. Sei die Spielart bekannt. Die **Farbklasse** $a \in \{T, \clubsuit, \spadesuit, \heartsuit, \diamondsuit\}$ einer Karte ist gleich T, falls es sich um eine Trumpfkarte handelt und gleich der aufgedruckten Kartenfarbe aus $\{\clubsuit, \spadesuit, \heartsuit, \diamondsuit\}$, falls die Karte Teil des Beiblatts ist.

In der **Ausspielphase**, bestehend aus zehn *Stichen*, geben die Spieler reihum Karten zu. Dabei herrscht *Farbzwang*: Die Farbklasse der zweiten (*Mittelhand*) und dritten Karte (*Hinterhand*) im Stich muss mit der Farbklasse der ersten (*Vorhand*) übereinstimmen, falls der Ausspieler eine solche Karte zugeben kann. Kann er nicht bedienen, wird er *farbfrei* genannt und kann eine beliebige andere Karte legen. Der Stich geht an die Partei, die die höchste Karte der Farbklasse oder den höchsten Trumpf gelegt hat.

Interessant im Kontext dieser Arbeit ist der große strategische Unterschied zwischen dem Spiel aus Allein- und Gegenspielersicht. Der Alleinspieler legt sich vor Ausspielbeginn eine Strategie zurecht und sitzt durch sein überlegenes Blatt in den meisten Situationen in der günstigen Vorhandposition. Die Gegenpartei ist dagegen stark auf den Informationsgewinn aus dem Spiel und die Kommunikation mit dem Partner angewiesen, um Lücken im Blatt des Alleinspielers auszumachen.

Sind alle Karten ausgespielt, entscheidet sich die gewinnende Partei in Grand- und Farbspielen durch die Anzahl der *Augen*, also Kartenpunkte, die sie während der Ausspielphase einsammeln konnte. Der Alleinspieler gewinnt genau dann, wenn er 61 oder mehr der 120 maximal möglichen Augen vorweisen kann. In Nullspielen gewinnt der Alleinspieler genau dann, wenn er keinen Stich macht.

In Grand- und Farbspielen kennt Skat verschiedene *Gewinnstufen*. Kann eine Partei bei Spielende 30 oder weniger Augen vorweisen, so befindet sie sich im *Schneider*. Ist die Gegenpartei im Schneider, so erhöht sich die Zahl der Spielpunkte für den Alleinspieler gegenüber einem regulären Gewinn. Erreicht er selbst keine 30 Augen, so verliert er zu seinem normalen Verlust zusätzliche Spielpunkte.

Ein detailliertes, offizielles Regelwerk zu Skat findet sich bei DSKV (2010).

1.3 Spieltheoretische Optimierung

Die Entwicklung eines Computerspielers arbeitet im Allgemeinen auf die Näherung einer Strategie hin, die eine spielabhängige Auszahlungsfunktion maximiert. Im Skat sind das die erreichten Punkte durch Gewinn oder Verlust einer Partie.

Eine optimale Strategie im spieltheoretischen Sinne ist dadurch charakterisiert, dass sie bei seinerseits optimalem Verhalten des Gegners das bestmögliche Ergebnis liefert¹. Dies führt zu einem Gleichgewicht, in dem das einseitige Abweichen einer Partei von der optimalen Strategiemenge immer zu ihrem eigenen Nachteil verläuft (vgl. Sauer (2017)). Dieses Verhältnis wird auch als *Nash-Lösung* eines Spiels bezeichnet; ein Akteur, der seine Strategie nach dieser Vorgabe wählt, heißt *Nash-Spieler*.

Zu einfachen Spielen können Nash-Lösungen bestimmt werden; bei Skat ist das jedoch aufgrund des großen Zustandsraums im Allgemeinen nicht realisierbar. Um dies in Zahlen zu fassen, wird nun eine untere Schranke für die Zahl der auszuwertenden Zustände bei Beginn der Ausspielphase bestimmt. Dabei sind 30 Züge zu spielen, wobei in jeder Spielsituation für jeden Zug die Antworten der Mitspieler in jeder plausiblen, realen Kartenverteilung bestimmt werden müssen.

In jeder Situation kennen alle Spieler ihre eigenen Karten, die verbleibenden sind unbekannt. Wird der Skat ignoriert, gibt es zu Beginn aus Sicht jedes Spielers $\binom{20}{10}$ mögliche Kartenverteilungen. Diese Zahl verringert sich mit jedem Zug, sodass der nachfolgende Spieler noch $\binom{19}{9}$ Zustände observieren kann. Pro Stich ist nur die Anzahl der Züge in Vorhandposition bekannt, beginnend mit 10. Unter Umständen müssen die nachfolgenden Spieler Farbe zugeben und haben damit keine Wahlfreiheit. Mit diesen Einschränkungen entwickelt sich die Anzahl der Zustände K in etwa wie folgt:

$$\begin{aligned} K &\gtrsim 10 \binom{20}{10} \binom{19}{9} \binom{18}{9} \cdot 9 \binom{18}{9} \binom{17}{8} \binom{16}{8} \dots \\ &= \prod_{k=1}^{10} k \binom{2k}{k} \binom{2k-1}{k-1} \binom{2k-2}{k-1} \approx 7 \cdot 10^{79} \end{aligned} \quad (1.1)$$

Diese Zahlen sind weit jenseits der Größenordnung, die mit technischen Mitteln ausgewertet werden kann. Es sind also signifikante Vereinfachungen nötig, um eine Nash-Lösung anzunähern.

¹Dies kann auch für eine probabilistische Interpretation gelten, beispielsweise durch Maximierung des Erwartungswerts einer Auszahlungsfunktion.

Der Vollständigkeit halber bleibt zu erwähnen, dass ein Nash-Spieler auch nur gegen einen gleichwertigen Gegner ideal spielt: Der durchschnittliche menschliche Skatspieler ist weit von diesem Optimum entfernt, sodass das theoretisch optimale Verhalten zu defensiv wäre. Aggressiveres Reizen und riskantere Spielzüge können die Auszahlung vergrößern, falls die Gegner auf einem niedrigeren Niveau operieren.

1.4 Auswertung offener Situationen

In dieser Arbeit werden alle Spielentscheidungen auf die einfach zu modellierende Klasse der offenen Spielsituationen zurückgeführt. Deren Nash-Lösung kann mit bekannten Verfahren effizient bestimmt werden.

Definition 1.2. *Ist allen Spielern die vollständige Kartenverteilung am Tisch bekannt, so liegt eine **offene Spielsituation** vor.*

Da sich die Spieler der Gegenpartei in einer offenen Situation in der gleichen Informationslage befinden, können sie für alle Belange zu einem einzelnen Spieler zusammengefasst werden. Damit entspricht ein offenes Skatspiel einem endlichen Zweipersonen-Nullsummenspiel mit vollständiger Information. Für solche Spiele existiert immer eine deterministische Optimallösung mit bekannten Auszahlungen für beide Spieler (vgl. Sauer (2017)). Diese Lösung ist durch Kartenverteilung, Sitzposition und Spieltyp vollständig definiert und hängt nicht weiter von der bisherigen Spielhistorie ab.

Ist die Kartenverteilung vollständig bekannt, so ist die Komplexität der Auswertung nur von der Anzahl regelkonformer Zugfolgen abhängig und damit wesentlich geringer als beim verdeckten Spiel. Dazu lässt sich bei Spielbeginn eine obere Schranke für die Zustandsanzahl K_{offen} finden, indem angenommen wird, dass jeder Spieler in jeder Situation freie Wahl zwischen seinen Karten hat:

$$K_{\text{offen}} \leq 10 \cdot 10 \cdot 10 \cdot 9 \cdot 9 \cdot 9 \dots = (10!)^3 \approx 5 \cdot 10^{19} \quad (1.2)$$

Die maximale Komplexität liegt damit um viele Größenordnungen unter der eines verdeckten Spiels in (1.1) und kann mit gängigen Verfahren wie der *Alpha-Beta-Suche* weiter reduziert werden (vgl. Knuth und Moore (1975)). Richtig angewandt erlauben derartige Optimierungen die exakte Lösung eines offenen Skatspiels innerhalb von Sekundenbruchteilen, ohne ab einer gewissen Suchtiefe auf heuristische Bewertungsfunktionen ausweichen zu müssen.

Details und Realisierung dieses Verfahrens sind nicht Teil dieser Arbeit, da sie für Skat bereits ausführlich studiert wurden (vgl. Frank und Basin (1998), Long (2011)) und das Ergebnis nicht von der konkreten Implementierung abhängt. Im Folgenden wird stattdessen auf einen existierenden, effizienten Löser zurückgegriffen.

1.5 Spielfluss

Eine gute Spielsituation entsteht aus einer angemessenen Balance zwischen Herausforderung und Chancen. Unbezwingbare Gegner sorgen beim Spieler für Frustration, während das Schlagen weit unterlegener Gegner nicht als Erfolg wahrgenommen wird. Der Eindruck eines ebenbürtigen Gegners dagegen stärkt den Ehrgeiz zu gewinnen und erhöht den Wiederaufnahmewert des Spiels. Dies ist als *Spielfluss* oder *Game Flow* bekannt (siehe Kapp (2012)).

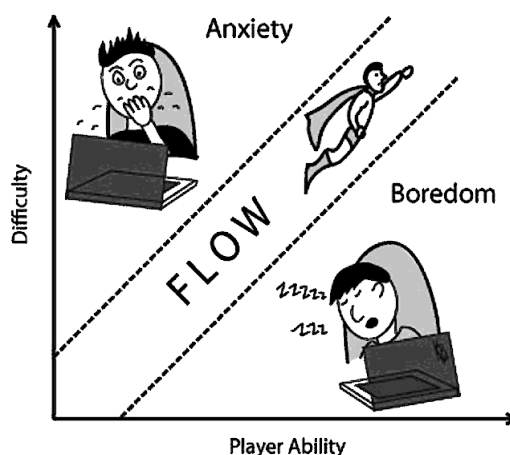


Abbildung 1.1: Der *Spielfluss* oder *Game Flow* ist die erwünschte Balance zwischen Spielerfähigkeiten und Schwierigkeitsgrad (Grafik aus Kapp (2012)). Im Skat wird er durch das Spiel gegen ebenbürtige Mitspieler ermöglicht.

Ziel des hier entwickelten Algorithmus ist damit nicht die bloße Maximierung erreichbarer Punkte, sondern die Aufrechterhaltung des Spielflusses.

Um dieser Anforderung gerecht zu werden, muss der Computer in der Lage sein, sich den menschlichen Fähigkeiten anzupassen. Das setzt voraus, dass die Spielmechanik

- ihre eigene Spielstärke kennt,
- die Spielstärke der Mitspieler korrekt einschätzen kann
- und die eigene Spielstärke auf einen vorgegebenen Wert einstellen kann.

Jede dieser Anforderungen wird mit den Entwicklungen der folgenden Kapitel erfüllt.

1.6 Existierende Literatur

Ausgangspunkt für diese Arbeit ist die Dissertation von Long (2011). Dort wird die Entwicklung des Referenzspielers *Kermit* detailliert beschrieben, nach dessen Grundaufbau auch der Spieler in dieser Arbeit entworfen wurde. Unter anderem wird dort auf die spieltheoretischen Überlegungen eingegangen und die schnelle Lösung offener Situationen (siehe Abschnitt 1.4) detailliert beleuchtet.

Der dort vorgestellte Spieler greift auf Perfect-Information-Monte-Carlo-Suche (kurz PIMC-Suche, siehe Kapitel 2) zurück, um in Situationen mit unvollständiger Information Entscheidungen zu treffen. Dieser Ansatz wird in der Dissertation gegen eine Reihe anderer, existierender Konzepte aus der Literatur verglichen, gegenüber denen er sich eindeutig behaupten kann. Diese Gegenüberstellung begründet auch die Wahl der PIMC-Suche für die vorliegende Arbeit.

Aus der Dissertation entstanden eine Reihe an Folgeveröffentlichungen. Buro u. a. (2009) gehen auf das Konzept der merkmalsbasierten Zustandsinferenz ein. Dabei konstruieren die Autoren aus Beobachtungen vergangener Spiele einen Schätzer für die Wahrscheinlichkeitsverteilung am Tisch und können damit die Spielstärke des Computers deutlich erhöhen. Dieser Ansatz wird in Kapitel 3 beschrieben, erweitert und auf neue Situationen angewendet, um Schwächen des PIMC-Verfahrens auszugleichen.

Long und Buro (2011) führen ein Spielstärkemaß ein, um das Reizverhalten des Computers an die Fähigkeiten seiner Gegner anzupassen. Dieses Maß wird in Kapitel 4 im Kontext einer adaptiven Spielweise besprochen. Aufbauend auf diese Technik wird ein stark verbessertes Fehlermaß entwickelt, das erstmals in der Lage ist, einen menschlichen Gegenspieler nach einer überschaubaren Anzahl von Spielen einzuschätzen. Diese Technik ist zentrale Voraussetzung eines adaptiven Spielers, der sich nach kurzer Zeit seinen Kontrahenten anpasst.

1.7 Testdaten

Um Aussagen über die Qualität der entwickelten Verfahren treffen zu können, müssen sie auf realistische Spielsituationen angewendet werden. Alle für diese Arbeit erforderlichen Auswertungen können anhand von alleinstehenden Situationen durchgeführt werden und erfordern keine Entscheidungen über Zugfolgen hinweg, weshalb auf eine Offline-Datenbank an aufgezeichneten Spielen zurückgegriffen werden kann.

Die Online-Skatplattform *International Skat Server*² (ISS, beschrieben von Long (2011)) stellt eine umfangreiche Datenbank von Spielverläufen zum Download bereit³. Die Spielhistorien bestehen aus den zugeteilten Karten, einer exakten Beschreibung des Reizvorgangs, der Spielansage und dem Verlauf der Ausspielphase. Alle teilnehmenden Spieler tragen Pseudonyme, anhand derer sie über Spiele hinweg identifiziert werden können.

Neben Menschen befinden sich auch einige Computerspieler unter den Teilnehmern, die einen Großteil der aufgezeichneten Spiele ausmachen. Besonders hervorzuheben sind darunter *Kermit* und *TheCount*, zwei Spieler aus der *Kermit*-Familie, die von Long (2011) entwickelt wurden (siehe Abschnitt 1.6) und konzeptionelle Ähnlichkeit mit dem Computerspieler aufweisen, der in dieser Arbeit vorgestellt wird. Nennenswert ist außerdem *XSkat*, ein quelloffener, heuristischer Spieler von Gunter Gerhardt⁴, der ebenfalls in den Historien auftaucht.

Zusätzlich hat der professionelle Skatspieler und viertplatzierte in der Skat-Europameisterschaft 2017⁵ Daniel Schäfer eigens für diese Arbeit Spielverläufe aus 16 Serien aufgezeichnet. In diesen 576 individuellen Spielen tritt er gegen einen Perfect-Information-Monte-Carlo-Spieler an, dessen Fähigkeiten mit *Kermit* und der hier entwickelten KI vergleichbar ist.

Um die Datenquellen im Rahmen dieser Arbeit benutzbar zu machen, wurden die Rohdaten in Textform eingelesen und interpretiert. Fehlerhafte und unvollständige Spielhistorien wurden entfernt und die verbleibenden Daten in eine relationale Datenbank exportiert, um sie einfacher durchsuchen zu können.

²<https://skatgame.net/iss/>, abgerufen am 15.12.2017

³<https://skatgame.net/iss/iss-games-11-2017.sgf.bz2>, fünf Millionen Spielverläufe von Menschen und Computerspielern, Stand November 2017

⁴<http://http://www.xskat.de/>, abgerufen am 14.03.2018

⁵<https://www.ispaworld.info/images/Downloads/ISPA-EM/2017/EM-Finale-2017---Serie-5.pdf>, abgerufen am 13.03.2018

2 Perfect Information Monte Carlo Search

In diesem Kapitel wird die zentrale Komponente des Zugwahl- und Bewertungsalgorithmus vorgestellt, die Perfect-Information-Monte-Carlo-Suche (kurz PIMC-Suche). Obwohl das grundlegende Verfahren gut studiert ist (siehe etwa Ginsberg (1999), Frank und Basin (1998) oder Long (2011)), erfordert die praktische Umsetzung im Rahmen des Skatspiels Entwicklungen, die in dieser Arbeit formalisiert werden.

Spielsituationen, in denen die vollständige Kartenverteilung bekannt ist, werden als offene Situationen bezeichnet. Für diese kann ein optimaler Spielzug effizient bestimmt werden (siehe Abschnitt 1.4). Die PIMC-Suche verallgemeinert diese Methode auf normale, verdeckte Spielsituationen, indem sie eine Menge zufälliger Stichproben als offene Situationen auswertet und die Einzellösungen zu einer Aktion konsolidiert.

Die einzelnen Schritte des Algorithmus sind in Abbildung 2.1 skizziert.

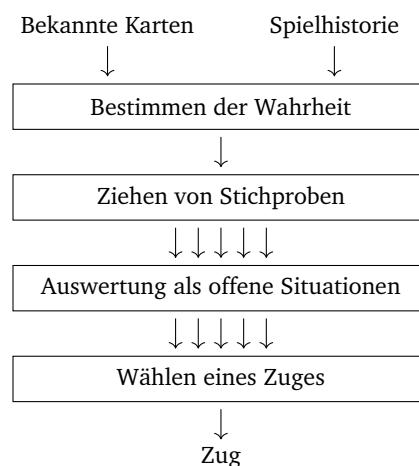


Abbildung 2.1: Strategiefindung mit Perfect Information Monte Carlo Search

Abschnitt 2.1 behandelt die Bestimmung des Suchraums anhand der *Wahrheit*. Abschnitt 2.2 führt einen einfachen Schätzer für die Wahrscheinlichkeitsverteilung ein, auf

dem die PIMC-Simulation aufsetzt. In Abschnitt 2.3 wird ein neues, effizientes Verfahren zur Erzeugung der Monte-Carlo-Stichproben entwickelt. Diese werden anschließend als offene Spiele ausgewertet, wie in Abschnitt 2.4 beschrieben wird. In Abschnitt 2.5 wird eine verbesserte Möglichkeit erarbeitet, um die gewonnenen Informationen zu einem Zug zu konsolidieren. Die theoretischen und praktischen Schwächen der PIMC-Suche werden in Abschnitt 2.6 beleuchtet.

2.1 Grundlagen

Die PIMC-Methode zieht Stichproben plausibler Kartenverteilungen, die mit dem bisherigen Wissen über Farbfreiheit der Gegner und bereits ausgespielte Karten übereinstimmen. In den resultierenden Verteilungen wird das Spielergebnis der möglichen Züge im optimalen offenen Spiel bestimmt. Die Annahme ist, dass ein Zug, der in den meisten dieser Stichproben vorteilhaft ist, auch ein guter Kandidat für den optimalen Zug in der ursprünglichen Situation ist.

Um formal über die Menge der möglichen Kartenverteilungen sprechen zu können, werden zunächst einige Begriffe eingeführt.

Definition 2.1. *Das Deck D bezeichnet die Menge aller 32 Karten des Skatblatts.*

Definition 2.2. *Eine Hand $H \subset D$ ist eine Kartenmenge, die einem Spieler oder dem Skat zugeordnet ist.*

Definition 2.3. *Eine Kartenverteilung oder auch Tisch¹ $t = (H_j)_{j \in J}$ ist ein Tupel paarweise disjunkter Hände H an Positionen J , die die Verteilung der Karten in einer konkreten Spielsituation repräsentieren. \mathbb{T} bezeichnet die Menge aller Tische.*

Bei einem Tisch muss es sich nicht um die vollständige Verteilung zu Spielbeginn handeln, vielmehr liegt jeder Spielsituation eine eigene Kartenverteilung zugrunde — deshalb wird die Mächtigkeit der Hände auch nicht weiter eingeschränkt.

Definition 2.4. *Die Längensignatur $L_H : \{\clubsuit, \spadesuit, \heartsuit, \diamondsuit, T\} \rightarrow \mathbb{N}$ einer Hand H ist eine Abbildung, die die Karten gleicher Farbklasse (siehe Definition 1.1) in H zählt.*

¹Die Wahl dieser Bezeichnung soll einer Verwechslung von *Kartenverteilung* und *Wahrscheinlichkeitsverteilung* vorbeugen, zweier Begriffe, die im Folgenden häufig gemeinsam auftreten.

Beispiel 2.1. Im Kontext eines Grandspiels sei die folgende Hand H gegeben:

$$H = \{\diamond 7, \diamond 10, \diamond B, \diamond A, \heartsuit 9, \spadesuit B, \clubsuit 8, \clubsuit D, \clubsuit K\}$$

Die beiden Buben gehören als Trumpfkarten zur Farbklasse T , während die verbleibenden Beiblattkarten ihrer Kartenfarbe aus $\{\clubsuit, \spadesuit, \heartsuit, \diamond\}$ zugeordnet werden. H erhält damit die Längensignatur

$$L_H = \{(\diamond, 3), (\heartsuit, 1), (\spadesuit, 0), (\clubsuit, 3), (T, 2)\}.$$

Der Farbzwang des Skatspiels erfordert, dass ein Zug in Mittel- und Hinterhand — falls möglich — immer mit der Farbklasse der ersten Karte im Stich erfolgt. Kann ein Spieler nicht bedienen, wird dadurch seine *Farbfreiheit* in dieser Klasse bekannt:

Definition 2.5. Sei a eine Farbklasse. Ein Spieler mit Hand H heißt **farbfrei** auf a wenn gilt $L_H(a) = 0$.

Mit diesen Konzepten kann nun der Begriff der *Wahrheit* eingeführt werden; jener Tischmenge, in der auch die tatsächliche Kartenverteilung enthalten ist.

Definition 2.6. In einer Spielsituation ist die **Wahrheit ohne Farbfreiheitsinformation** $\overline{W} \subseteq \mathbb{T}$ die Menge aller Tische $t \in \mathbb{T}$, die die folgenden Voraussetzungen erfüllen:

1. Bereits ausgespielte Karten treten nicht erneut in t auf.
2. Die Hand des Spielers selbst (und falls aufgenommen, der Skat) reproduziert in t die bekannten Karten.
3. Jede Hand in t enthält die bekannte Anzahl verbleibender Karten. Für die Spielerhände entspricht das der Anzahl der Stiche, an denen der Spieler noch teilnehmen muss; für den Skat sind es bis zu zwei Karten.

Die **Wahrheit** $W \subseteq \overline{W}$ enthält diejenigen Tische $t \in \overline{W}$, für die weiter gilt

4. Ist aus der Historie bekannt, dass der Spieler mit Hand H in Farbklasse a farbfrei ist, so gilt in t ebenfalls $L_H(a) = 0$.

Die Wahrheit ist damit die Menge der Kartenverteilungen, die mit Kenntnis einer partiellen Spielhistorie regelkonform auftreten können. Die Wahrheit ohne Farbfreiheitsinformation ist eine Obermenge, die sich leichter beschreiben lässt — eine Eigenschaft, die für die Algorithmen in Abschnitt 2.2 und 2.3 benötigt wird.

2.2 Tischgleichverteilung

Die größte Herausforderung beim Übergang zu Perfect-Information-Monte-Carlo-Suche ist das Ziehen geeigneter Stichproben. Dazu muss ein Schätzer für die reale, unbekannte Wahrscheinlichkeitsverteilung konstruiert werden. Der Einfachheit halber wird zunächst angenommen, dass der Tisch gleichverteilt über der Wahrheit ist. In Kapitel 3 wird dann ein Inferenzverfahren vorgestellt, um die Verteilung präziser zu bestimmen.

Im Wahrscheinlichkeitsraum Ω bezeichnet die Zufallsvariable $T : \Omega \rightarrow \mathbb{T}$ den tatsächlichen Tisch einer Situation und $t \in \mathbb{T}$ den Tisch einer Monte-Carlo-Stichprobe. Weiter notiert allgemein $\#W$ die Mächtigkeit einer endlichen Menge W .

Definition 2.7. Sei (Ω, \mathcal{A}, P) ein Wahrscheinlichkeitsraum und W die Wahrheit einer Spielsituation. Die **Tischgleichverteilung** \mathcal{U}_W einer Zufallsvariablen $T : \Omega \rightarrow \mathbb{T}$ nimmt an, dass jeder regelkonform mögliche Tisch mit der gleichen Wahrscheinlichkeit auftritt:

$$T \sim \mathcal{U}_W \Leftrightarrow P(T = t) = \begin{cases} \frac{1}{\#W} & t \in W \\ 0 & \text{sonst} \end{cases} \quad (2.1)$$

Ein simples Aufzählen von W und anschließendes Ziehen von Stichproben ist nicht praktikabel: Im Worst Case (Gegenspielerposition bei Spielbeginn) sind nur 10 von 32 Karten bekannt und W hat die Mächtigkeit

$$\max_W \#W = \binom{22}{10} \binom{12}{10} \binom{2}{2} = 42.678.636 \quad (2.2)$$

Liegt keine Farbfreiheitsinformation vor und ist damit $W = \overline{W}$, so kann die Verteilung simuliert werden, in dem die unbekanntes Hände des Tisches durch einfaches Ziehen ohne Zurücklegen aus den verbleibenden Karten befüllt werden.

Gilt dagegen $W \neq \overline{W}$, so muss die Zuteilung der Karten mit Bedacht erfolgen, um die Gleichverteilungseigenschaft nicht zu verletzen. Im Folgenden wird dazu eine existierende, simple Methode vorgestellt und anschließend ein neuer, effizienterer Algorithmus entwickelt.

Verwerfungsmethode

Zunächst wird das etablierte Verfahren der *Verwerfungsmethode* eingeführt. Dabei handelt es sich um einen bekannten Ansatz, um ausgehend von einer Gleichverteilung auf einer einfach zu beschreibenden Menge die Gleichverteilung auf beliebig komplexen Teilmengen zu simulieren (vgl. Georgii (2009)).

Verfahren 2.1. Sei $W \subseteq \bar{W}$ die Wahrheit einer Spielsituation. Die *Verwerfungsmethode* erzeugt so lange Stichprobenkandidaten $t' \in \bar{W}$ durch zufälliges Ziehen der Spielerkarten, bis eine Stichprobe den Einschränkungen der Farbfreiheit genügt und $t \in W$ gefunden ist.

Dieses Verfahren reproduziert die Tischgleichverteilung perfekt, hat als Brute-Force-Ansatz jedoch situationsabhängig Performanceprobleme: Ist \bar{W} sehr viel größer als W und damit der Anteil zulässiger Tische klein, wird nur selten eine Stichprobe gefunden.

Beispiel 2.2. Ein pathologischer Fall ist das folgende Kreuzspiel von Spieler p_1 :

- Stich 1: (p_1 : ♣7, p_2 : ♦A, p_3 : ♣D)
- Stich 2: (p_3 : ♥7, p_1 : ♣A, p_2 : ♦9)
- Bekannte Hand $H_1 = \{\clubsuit B, \heartsuit B, \spadesuit A, \spadesuit K, \clubsuit A, \clubsuit 10, \clubsuit K, \clubsuit 9, \clubsuit 8, \clubsuit 7\}$
- Bekannter Skat $H_S = \{\diamond 7, \diamond 8\}$

Da p_2 zweimal nicht bedienen konnte, ist mit der Farbfreiheitsinformation

$$L_{H_2}(T) = L_{H_2}(\heartsuit) = 0$$

der Tisch vollständig bekannt und es gilt

$$\begin{aligned} H_2 &= \{\diamond 10, \diamond K, \diamond D, \spadesuit 10, \spadesuit D, \spadesuit 9, \spadesuit 8, \spadesuit 7\} \\ H_3 &= \{\heartsuit A, \heartsuit 10, \heartsuit K, \heartsuit D, \heartsuit 9, \heartsuit 8, \spadesuit B, \diamond B\}. \end{aligned}$$

Ohne Berücksichtigung der Farbfreiheit gibt es $\binom{16}{8} = 12.870$ Tischkandidaten, von denen genau einer zulässig ist – im Mittel benötigt die Verwerfungsmethode hier also 12.870 Iterationen, um eine Stichprobe zu produzieren.

Ein solcher trivialer Fall kann gesondert behandelt werden; es ist jedoch nicht schwer, Situationen mit kleinen Freiheitsgraden und ähnlicher Komplexität zu konstruieren.

2.3 Kombinatorische Simulation

Um die Schwächen der Verwerfungsmethode zu überwinden, wird nun ein neues Verfahren entwickelt, das die Struktur der Farbfreiheitsinformation ausnutzt. Dazu wird die Menge der potentiell validen Kartenverteilungen \bar{W} (siehe Definition 2.6) anhand von Längensignaturen partitioniert, sodass die Farbfreiheit nicht mehr für jede Kartenverteilung einzelnen entschieden werden muss. Es entsteht eine zweistufige Simulation, die die bekannten Pathologien umgeht.

Definition 2.8. Für zwei Kartenverteilungen $t, t' \in \mathbb{T}$ mit $t = (H_j)_{j \in J}$ und $t' = (H'_j)_{j \in J}$ ist die Äquivalenzrelation $t \sim_L t'$ definiert als

$$t \sim_L t' \Leftrightarrow \forall j \in J : L_{H_j} = L_{H'_j}$$

Es gilt also $t \sim_L t'$ genau dann, wenn die entsprechenden Hände in t und t' paarweise identische Längensignaturen aufweisen. Damit zerfällt \bar{W} in Äquivalenzklassen über \sim_L ; diese werden für einen Tisch $t \in W$ im Folgenden mit $[t]_L$ bezeichnet.

Definition 2.9. Für die Kartenverteilung $t \in \mathbb{T}$ ist die **Kombination** $K(t)$ definiert als

$$K(t) := [t]_L \cap \bar{W}.$$

Kombinationen haben die Eigenschaft, dass ihre Zulässigkeit gegenüber der Farbfreiheit ohne Kenntnis des konkreten Tisches entschieden werden kann:

$$t \in W \Leftrightarrow K(t) \subset W \tag{2.3}$$

Für eine Kombination k kann die Gleichverteilung \mathcal{U}_k problemlos simuliert werden: Die Längensignatur L_H jeder Hand H ist bekannt; die konkreten Karten von H innerhalb einer Farbklasse können durch Ziehen ohne Zurücklegen aus den verbleibenden Karten dieser Klasse bestimmt werden.

Verfahren 2.2 (Kombinatorische Simulation). Sei (Ω, \mathcal{A}, P) ein Wahrscheinlichkeitsraum und W die Wahrheit einer Spielstituation.

1. Zähle alle Kombinationen $k \subset W$ auf, wobei $k \in \{K(t) \mid t \in W\}$.
2. Für jede zu ziehende Stichprobe:
 - 2.1. Ziehe eine Kombination $k \subset W$, wobei $P(k) = \#k / \#W$.
 - 2.2. Realisiere $t \in k$ durch getrenntes Ziehen der Farbklassen innerhalb jeder Hand.

Das Aufzählen der Kombinationen ist in Algorithmus 2.1 implementiert. Dieser Schritt stellt eine deutliche Vereinfachung in der Komplexität gegenüber dem Aufzählen aller Kartenverteilungen aus W dar: Für die Worst-Case-Annahme aus Gleichung 2.2 wurde die maximale Anzahl möglicher Kombinationen durch exhaustive Suche in allen Startverteilungen ermittelt und in Tabelle 2.1 dargestellt. Die über 40 Millionen möglichen Tische verteilen sich auf höchstens 5.269 Kombinationen.

Situation mit maximaler Kombinationszahl		
Spieltyp	Längensignatur bekannte Hand	Kombinationen
Grand	$(\diamond, 0), (\heartsuit, 0), (\spadesuit, 0), (\clubsuit, 6), (T, 4)$	5.269
Farbspiele	$(\diamond, 0), (\heartsuit, 0), (\spadesuit, 0), (T, 10)$	1.377
Null	$(\diamond, 0), (\heartsuit, 0), (\spadesuit, 2), (\clubsuit, 8)$	1.377

Tabelle 2.1: Maximale Anzahl Kombinationen für den gesamten Tisch im Worst-Case-Fall von nur 10 bekannten Karten. Die Längensignaturen sind Beispiele; durch Symmetrie zwischen den Beiblattfarben $\clubsuit\spadesuit\heartsuit\diamond$ wird das Maximum für mehrere Kombinationen angenommen.

```

# Iteriere über alle Möglichkeiten, n Karten auf m Hände mit vorgegebener Kapazität zu
# verteilen. Gib die Anzahl Belegungen pro Möglichkeit zurück.
def distribute(n, slots):
    if len(slots) == 1:
        # Die verbleibenden Karten der Farbklasse müssen in die letzte Hand passen
        if n <= slots[0]:
            yield 1, [n]
    else:
        # Packe 0 . . . n Karten in die erste Hand, fülle die verbleibenden rekursiv auf
        for k in range(min(slots[0], n) + 1):
            for ch, tail in distribute(n-k, slots[1:]):
                yield ch * binom(n, k), [k] + tail

# Iteriere über alle Kombinationen einer Wahrheit, gegeben durch
# items: Liste verbleibender Karten als [(Farbklasse, Kartenzahl)]
# slots[i]: Anzahl noch fehlender Karten in Hand i
# forbidden: Menge von Tupeln {(i, Farbklasse)} für die gilt  $L_{H_i}(\text{Farbklasse}) = 0$ .
# Gib die Anzahl Belegungen zusammen mit den Längensignaturen aller Hände zurück.
def combinations(items, slots, forbidden):
    # Farbklasse, die in diesem Schritt verteilt wird
    suit, count = items[0]
    remaining_items = items[1:]

    # Abwandlung der slots-Liste: Hände der farbfreien Spieler erhalten keine Karten
    suit_slots = [n if (i, suit) not in forbidden else 0 for i, n in enumerate(slots)]

    # Prüfe alle Möglichkeiten, die gewählte Farbklasse auf die Hände zu verteilen
    for ch, dist in distribute(count, suit_slots):
        distributed_cards = [{suit: n} for n in dist]

        # Verteile die verbleibenden Farbklassen rekursiv
        if remaining_items:
            rem_slots = [s - d for s, d in zip(slots, dist)]
            for ch_next, c in combinations(remaining_items, rem_slots, forbidden):
                yield ch * ch_next, [n | m for n, m in zip(distributed_cards, c)]
        else:
            yield ch, distributed_cards

```

Algorithmus 2.1: Ermitteln aller Kombinationen einer Situation

2.4 Stichprobenauswertung

Mit der effizienten kombinatorischen Simulation der Tischgleichverteilung wird nun die Auswertung von Monte-Carlo-Stichproben möglich (zum Kontext siehe Abbildung 2.1).

Long (2011) erzielt mit 40–50 Stichproben pro Auswertung gute Ergebnisse; für diese Arbeit wird die Stichprobenzahl in allen Anwendungen auf 100 festgelegt. Diese Anzahl ist absichtlich großzügig gewählt, um den Einfluss des Zufalls auf Auswertungsergebnisse gering zu halten.

Die gezogenen Stichproben werden im Auswertungsschritt als offene Spiele betrachtet. Unter dieser Annahme kann für jeden Zug der Situation das Spielergebnis bei optimalem Gegenspiel eindeutig bestimmt werden (siehe 1.4). Je nach Spieltyp manifestiert es sich in Form der Augenzahl der ausspielenden Partei (Grand- und Farbspiele) oder lediglich der Information, ob das Spiel gewonnen oder verloren wurde (Nullspiele).

Es ist wichtig zu verstehen, dass hier eine nicht unerhebliche Verzerrung der Spielmechanik stattfindet. Da allen Spielern die Kartenverteilung bekannt gemacht wird und sie damit die Nash-Lösung des Spiels deterministisch berechnen können, liegt die optimale Strategie und das Ergebnis des Skatspiels bereits vor dem ersten Zug fest. Das steht im Gegensatz zum realen Spiel, wo die Unvollständigkeit der Information essentieller Teil jeder Spielstrategie ist. Beispielsweise sind die meisten Nullspiele im Offenen ungewinnbar, während der Alleinspieler im Verdeckten eine gute Chance hat, eine Lücke in seinem Blatt vor den Gegenspielern zu verbergen.

Die theoretischen und praktischen Auswirkungen dieses Vereinfachungsschritts werden in Abschnitt 2.6 im Detail diskutiert.

2.5 Zugwahl

Wurde die Menge der Stichproben ausgewertet und das Ergebnis jedes offenen Spiels bestimmt, müssen die gewonnenen Informationen zu einem Zug konsolidiert werden (zum Kontext siehe Abbildung 2.1). Dieser Schritt ist nicht trivial, da ein Zielkonflikt zwischen Gewinnwahrscheinlichkeit und Maximierung der Punktzahl existiert.

Bei allen Verfahren, die in diesem Abschnitt vorgestellt werden, können abhängig von der Spielsituation mehrere Züge optimal sein. Deshalb lösen sie nach Zugmengen anstatt nach individuellen Zügen auf; welche dieser Karten ein Ausspielmodul konkret in den Stich gibt ist unerheblich.

Für eine Monte-Carlo-Auswertung bezeichnet $S \subset \mathbb{T}$ die Menge der Stichproben und M die Menge der möglichen Züge in der aktuellen Situation. Für $t \in S$ und $m \in M$ notiert $g(t, m)$, ob die Partei des Computerspielers mit Zug m das offene Spiel gewinnt (1) oder verliert (0). Im selben Kontext bezeichnet $a(t, m)$ die erreichte Augenzahl der Computerspielerpartei durch Zug m .

Maximieren der Gewinnwahrscheinlichkeit

Begonnen wird mit dem allgemeinsten Verfahren. Für Nullspiele liefert jede Stichprobe lediglich, welche Partei das offene Spiel gewinnt, sodass nur auf $g(t, m)$ zurückgegriffen werden kann. Damit lässt sich jedoch schon eine Methode beschreiben, die den Gewinnanteil an den ausgewerteten Stichproben maximiert.

Verfahren 2.3. Die Zugwahl durch *Maximieren der Gewinnwahrscheinlichkeit* bestimmt die Menge der optimalen Züge als

$$M_{\text{Gewinn}} = \arg \max_{m \in M} \sum_{t \in S} g(t, m) \quad (2.4)$$

Diese Methode ist auch auf Grand- und Farbspiele anwendbar, ignoriert dabei aber die Möglichkeit, höhere Gewinnstufen zu erreichen (siehe Abschnitt 1.2): Ein Zug, der 91 Augen verspricht, kann dem Alleinspieler die Gewinnstufe *Schneider* verschaffen und damit mehr Punkte einbringen als ein Zug mit 61 Augen — auch wenn beide die gleichen Gewinnchancen versprechen.

Maximieren der Augenzahl

Um Züge zu bevorzugen, die eine höhere Gewinnstufe ermöglichen, muss die erreichbare Augenzahl der Stichproben betrachtet werden. Eine einfache Methode ist, die mittlere Augenzahl des Computerspielers zu maximieren:

Verfahren 2.4. Die Zugwahl durch **Maximieren der mittleren Augenzahl** bestimmt die Menge der optimalen Züge als

$$M_{\text{Augen}} = \arg \max_{m \in M} \sum_{t \in S} a(t, m) \quad (2.5)$$

Diese Methode ist nur auf Grand- und Farbspiele anwendbar, da für Nullspiele kein Äquivalent zur Augenzahl existiert. Gleichzeitig sind dort auch keine unterschiedlichen Gewinnstufen möglich, weshalb eine weitere Optimierung nicht notwendig ist.

Weisen alle Züge die gleiche Gewinnwahrscheinlichkeit auf, so bevorzugt das Maximieren der mittleren Augenzahl den Zug, der eine höhere Gewinnstufe verspricht. Sind die Wahrscheinlichkeiten unterschiedlich, so kann das Verfahren aber die Gewinnwahrscheinlichkeit schmälern:

Beispiel 2.3. Es liege eine Situation vor, in denen die Züge ♡9, ♡K und ♡A möglich sind und die Stichproben folgende Augenzahlen versprechen:

$m \in M$	$a(t_1, \dots, t_n, m)$					$\frac{1}{n} \sum_t g(t, m)$	$\frac{1}{n} \sum_t a(t, m)$
♡9	56	58	111	116	114	60%	91,0
♡K	61	67	64	71	61	100%	64,8
♡A	68	74	71	78	68	100%	71,8

Fett gedruckt ist jeweils das beste Spaltenergebnis. Die geschätzte Gewinnwahrscheinlichkeit beträgt für ♡K und ♡A 100%, während ♡9 nur zu 60% gewinnt. Die mittlere Augenzahl verspricht mit ♡K und ♡A jedoch je 64,8 bzw. 71,2 Augen, während ♡9 mit 92,0 Augen im Durchschnitt um eine Gewinnstufe höher liegt.

Hier liegt ein Zielkonflikt vor: Obwohl der Mittelwert der erreichten Augen steigt, führt eine gleichzeitig erhöhte Varianz zwischen den Stichprobenergebnissen dazu, dass durch Maximieren der Augenzahl ein riskanterer Zug gewählt wird.

Konservative Zugwahl

Um das Potential der beiden Vorgängerverfahren zu kombinieren, ohne die Nachteile zu übernehmen, wird im Folgenden das Verfahren der *konservativen Zugwahl* entwickelt. Es basiert auf der informellen Beschreibung der Zugwahl von Long (2011) und wählt unter allen Karten mit maximaler Gewinnwahrscheinlichkeit M_{Gewinn} diejenigen, die die höchste mittlere Augenzahl versprechen:

Verfahren 2.5. Die *konservative Zugwahl* bestimmt die Menge der optimalen Züge als

$$M_{\text{konservativ}} = \arg \max_{m \in M_{\text{Gewinn}}} \sum_{t \in S} a(t, m) \quad (2.6)$$

Diese Methode erhöht das Risiko gegenüber dem Maximieren der Gewinnwahrscheinlichkeit nicht, optimiert dabei aber die erwarteten Spielpunkte.

In Beispiel 2.3 würde die konservative Zugwahl $\heartsuit A$ wählen, um einen (aus PIMC-Sicht) sicher gewinnenden Zug auszuspielen und dabei die höchstmögliche Augenzahl zu sichern. In den häufig anzutreffenden Situationen, in denen jeder Zug sicher gewinnt oder jeder Zug sicher verliert, entspricht die konservative Zugwahl dem Maximieren der erwarteten Augenzahl.

Da $a(t, m)$ im verdeckten Spiel nur einen Schätzer für die tatsächliche Augenzahl darstellt und vom tatsächlichen Wert abweichen kann, steigert die Wahl eines Zuges mit höherer erwarteter Augenzahl unter Umständen sogar die realen Gewinnchancen. Das entspricht dem intuitiven Verhalten, in einem gewonnenen Stich die Augen-höhere Karte zu spielen, sofern die Wahl den weiteren Spielverlauf nicht beeinflusst.

Die Effektivität der drei besprochenen Methoden wird in Abschnitt 5.1 erstmals gegeneinander verglichen. Dort kann die konservative Methode aus den erwähnten Gründen ihre Überlegenheit demonstrieren.

2.6 Schwächen des Verfahrens

Perfect-Information-Monte-Carlo-Suche nimmt vereinfachend an, dass nach dem Zug des aktuellen Spielers alle Karten aufgedeckt werden und ein offenes Spiel folgt. Je größer die vorherige Unsicherheit der Spieler über die Situation war, desto stärker verzerrt diese Annahme den weiteren Spielverlauf. Frank und Basin (1998) arbeiten die theoretischen Schwächen heraus, die auf der deterministischen Natur der Stichprobenauswertung basieren.

Praktisch äußern sich die Nachteile der PIMC-Methode dadurch, dass der Algorithmus den Informationsfluss im Spiel nicht abbilden kann — schließlich sind aus seiner Sicht alle Karten bekannt. Das wird vor allem im Gegenspiel sichtbar, wo das Ausloten der Alleinspielerkarten und die Kommunikation mit dem Teampartner für den Erfolg unabdingbar sind. Ein PIMC-Spieler wird dagegen immer die Karte wählen, die bei aktuellem Kenntnisstand am ehesten zum Sieg im offenen Spiel führt.

Skatexperte Daniel Schäfer (siehe Abschnitt 1.7) beschreibt das Spiel an einem Tisch mit zwei PIMC-Spielern, deren Implementierung der hier entwickelten KI konzeptionell ähnelt, wie folgt:

„Das Alleinspiel beherrscht der Computerspieler relativ gut, auch wenn er dabei recht passiv auftritt. Als Partner im Gegenspiel ist er jedoch völlig unbrauchbar; macht grobe, offensichtliche Fehler und spielt nicht teamorientiert. Diese Schwächen sind untypisch und nicht allein durch die Anforderungen der Gegenspielersituation zu rechtfertigen.

Dass das Fehlerpotential von Mensch und Computer im Gegenspiel generell höher ist, ist aber plausibel: Der Alleinspieler weiß von Anfang an exakt, welche Karten bei welcher Partei liegen, während sich der einzelne Gegenspieler darüber im Unklaren ist.“

Hier wird der Unterschied zwischen Amateurspieler und Profi deutlich: Anstatt einfach Karten zu legen und auf Punkte zu hoffen, sind fähige Gegenspieler in der Lage mit ihrem Teampartner zu kooperieren und eine Gewinnstrategie zu entwickeln. Im Alleinspiel ist die optimale Spielweise dagegen wenig von der Reaktion der Gegenpartei abhängig und kann damit vom PIMC-Spieler gut modelliert werden.

In Kapitel 3 wird mit der Zustandsinferenz eine Möglichkeit vorgestellt, die Spielweise im Gegenspiel durch Herleitung der Alleinspielerkarten zu verbessern.

2.7 Zusammenfassung

In diesem Kapitel wurde mit der Perfect-Information-Monte-Carlo-Suche der Grundstein für das Spiel in verdeckten Situationen gelegt. Das Verfahren löst Skatsituationen mit unvollständiger Information näherungsweise durch stichprobenhafte Auswertung der Kartenverteilung als offene Spiele. Dies erlaubt eine einfache Beschreibung von Zügen durch Gewinnwahrscheinlichkeiten und erwartete Augenzahlen.

Dazu wurde zunächst mit der Tischgleichverteilung (Definition 2.7) ein Schätzer für die Wahrscheinlichkeitsverteilung am Tisch konstruiert. Dann wurde mit der kombinatorischen Simulation (Verfahren 2.2) ein neuer Algorithmus entwickelt, um diese Wahrscheinlichkeitsverteilung effizient zu simulieren. Nach Auswertung durch einen existierenden Löser für offene Situationen wurde schließlich mit der konservativen Zugwahl (Verfahren 2.5) ein optimiertes Verfahren zur Stichprobenkonsolidierung vorgestellt.

Die PIMC-Methode liefert ein Zugverhalten, das vor allem im Gegenspiel an einer fehlenden Modellierung des Informationsflusses im Spiel leidet. Die praktischen Auswirkungen wurden in Abschnitt 2.6 mit einem professionellen Skatspieler diskutiert.

In Kapitel 3 wird nun ein verbesserter Schätzer für die Wahrscheinlichkeitsverteilung konstruiert, der in der Lage ist, Schlüsse aus dem Verhalten der Mitspieler zu ziehen. Damit kann das Spiel des Computers weiter gestärkt werden.

3 Zustandsinferenz

Zustandsinferenz ist die Fähigkeit eines Spielers, aus dem Verhalten seiner Mitspieler Schlüsse auf die Kartenverteilung (siehe Definition 2.3) zu ziehen. Das Auflösen der Wahrheit (siehe Definition 2.6) ist eine einfache Form der Zustandsinferenz. Sie beschränkt sich auf Schlüsse, die aus der Regelkonformität des aktuellen Spiels ableitbar sind, wie etwa der Einhaltung des Farbwangs (zu den Regeln siehe Abschnitt 1.2).

Ein geübter Skatspieler kann jedoch deutlich mehr Informationen aus den Aktionen seiner Gegner und Partner ableiten. So würde ein rationaler Spieler etwa selten ein Trumpfspiel ohne Trumpfkarten ansagen oder ein teures Ass in einem verlorenen Stich abwerfen, sofern er noch billigere Karten zugeben kann. Solche Annahmen sind immer heuristisch oder aus beobachtetem Verhalten abgeleitet, da die Regeln durchaus eine abweichende Spielweise zulassen.

Im Kontext der Perfect-Information-Monte-Carlo-Suche (PIMC-Suche, siehe Kapitel 2) bedeutet Zustandsinferenz die Konstruktion eines Schätzers für die reale Wahrscheinlichkeitsverteilung auf Basis beobachteter Züge. Damit können bevorzugt Stichproben ausgewertet werden, die mit dem bisherigen Verhalten der Mitspieler konsistent sind.

Definition 3.1. *Ein Zug m ist die Aktion eines Spielers in der Vergangenheit des Spiels.*

Diese Definition ist mit Absicht allgemein gehalten, da verschiedenste Aktionen Züge darstellen. Beispiele sind das Reizen bis zu einem bestimmten Wert, die Ansage eines Spieltyps oder das Ausspielen einer Karte.

Definition 3.2. *Sei (Ω, \mathcal{A}, P) ein Wahrscheinlichkeitsraum, $T : \Omega \rightarrow \mathbb{T}$ die reale Kartenverteilung in einer Situation, m ein beobachteter Zug und $t \in \mathbb{T}$ eine Stichprobe. Dann bezeichnet **Zustandsinferenz** das Problem, die Größe $P(T = t \mid m)$ zu schätzen.*

In diesem Kapitel wird zunächst in Abschnitt 3.1 eine simple, aber unzureichende Inferenzmethode skizziert und mit ihr die Anforderungen an die Zustandsinferenz erörtert. In Abschnitt 3.2 wird dann ein merkmalsbasiertes Inferenzverfahren aus der Literatur

eingeführt, das auf Offline-Daten realer Spieler trainiert wird. In Abschnitt 3.4 wird die Anwendung der merkmalsbasierten Inferenz auf die Spielfindungsphase realisiert, der Ansatz aus der Literatur verbessert und die Korrektheit der zugrundeliegenden Annahmen überprüft. In Abschnitt 3.5 wird die Methode dann in einer Neuentwicklung verallgemeinert, um sie erstmals zur Inferenz in der Ausspielphase nutzbar zu machen. Schließlich wird in Abschnitt 3.6 ein neues, passendes Maß für die Qualität der Zustandsinferenz definiert und die Effektivität der vorgestellten Verfahren ausgewertet.

3.1 Inferenz anhand eines Ausspielmoduls

Ist ein existierender Computerspieler in der Lage, alle Züge in einer Situation zu bewerten, so kann er auch zur Zustandsinferenz herangezogen werden. Dazu werden Stichproben zunächst gleichverteilt gezogen und die beobachteten Züge der Mitspieler an diesen Tischen nachvollzogen. Folgt der Schluss, dass ein beobachteter Zug unter dieser Kartenverteilung unplausibel wäre, wird die Stichprobe verworfen.

Diesen Ansatz wählen Ginsberg (1999) und (in Kombination mit merkmalsbasierten Verfahren) auch Long (2011), um Stichproben zu verwerfen, die nicht mit dem angesagten Spieltyp kompatibel erscheinen. Da in der vorliegenden Arbeit kein Algorithmus zur Spielfindung vorliegt, wird dieser spezielle Anwendungsfall nicht weiter verfolgt.

Der in Kapitel 2 entwickelte Spieler kann jedoch eine Zugsbewertung in der Ausspielphase (siehe Abschnitt 1.2) vornehmen, etwa in Form der erwarteten Gewinnwahrscheinlichkeit pro Zug. Damit liefert er einen Wert für $P(m | T = t)$, aus dem mit dem Satz von Bayes die gesuchte Größe $P(T = t | m)$ abgeleitet werden kann (vgl. Long (2011)). Es ist also denkbar, in jeder Stichprobe jede beobachtete Karte der Mitspieler mit PIMC-Suche zu bewerten, um diese Wahrscheinlichkeit zu bestimmen.

Das führt jedoch zu einer Explosion der Laufzeitkosten. Wurden beispielsweise 10 Züge beobachtet und die Zahl von 100 Stichproben aus Abschnitt 2.4 beibehalten, so müssten mindestens $100 \cdot 10 \cdot 100 = 100.000$ offene Auswertungen stattfinden, um einen Zug zu wählen — das tausendfache der üblichen 100 Auswertungen.

Außerdem nimmt diese Methode an, dass das beobachtete Verhalten einem PIMC-Spieler ähnlich ist, was ein Trugschluss sein kann (siehe Abschnitt 4.5).

Aufgrund dieser Schwächen wird die Zustandsinferenz anhand des Ausspielmoduls hier nicht weiter betrachtet.

3.2 Merkmalbasierte Inferenz

Um den Problemen aus Abschnitt 3.1 aus dem Weg zu gehen, greifen Long (2011) und Buro u. a. (2009) auf ein merkmalsbasiertes Verfahren zurück, das Stichprobenwahrscheinlichkeiten anhand des Reizverhaltens und der Spielansage bewertet.

Dazu definieren die Autoren einen Satz von Merkmalen für jede Hand (siehe Definition 2.2), wie etwa die Anzahl der Asse oder Trümpfe im Blatt. Diese Merkmale werden dann an einer großen Zahl von Kartenverteilungen aus aufgezeichneten Spielverläufen ausgewertet und die Häufigkeit jedes Merkmals unter dem beobachteten Zug notiert. Mit diesem Modell können den Stichproben einer PIMC-Simulation Wahrscheinlichkeiten anhand ihrer Merkmale zugeordnet werden.

Damit wird das Inferenzproblem von konkreten Kartenverteilungen auf Merkmalskombinationen reduziert. Wie im Folgenden gezeigt wird, verringert dieser Übergang die Komplexität der benötigten Daten und erlaubt auch die Bewertung von Händen, die so in den aufgezeichneten Spielverläufen nicht vorkamen.

Definition 3.3. Eine *Merkmalsklasse* $F : \mathbb{T} \rightarrow \mathbb{N}_0$ ist eine Abbildung, die jeder Kartenverteilung $t \in \mathbb{T}$ ein *Merkmal* $f \in \mathbb{N}_0$ zuordnet.

Ein Beispiel für eine Merkmalsklasse ist etwa die Anzahl der Trumpfkarten in der Hand des Alleinspielers. Ein zugehöriges Merkmal könnte das Vorhandensein von genau fünf Trumpfkarten in dieser Hand sein.

Der folgende Satz formalisiert die Inferenzgrundlage von Long (2011) und stellt das zentrale Fundament für die Algorithmen in diesem Kapitel dar, weswegen er mit Beweis aufgeführt wird.

Satz 3.1. Sei (Ω, \mathcal{A}, P) ein Wahrscheinlichkeitsraum, $F_i : \mathbb{T} \rightarrow \mathbb{N}_0, i \in I$ eine Familie von Merkmalsklassen, $t \in \mathbb{T}$ eine Stichprobe, $T : \Omega \rightarrow \mathbb{T}$ eine Zufallsvariable und m ein Zug.

Mit dem Hilfsereignis

$$\widehat{F}(t) := \bigcap_{i \in I} \{F_i \circ T = F_i(t)\} \quad (3.1)$$

und den Voraussetzungen

1. Zug und Tisch sind über den Merkmalen unabhängig:

$$P(\{T = t\} \cap m \mid \widehat{F}(t)) = P(T = t \mid \widehat{F}(t)) \cdot P(m \mid \widehat{F}(t)) \quad (3.2)$$

2. Merkmalsklassen sind untereinander unabhängig:

$$P(\widehat{F}(t)) = \prod_{i \in I} P(F_i \circ T = F_i(t)) \quad (3.3)$$

3. Merkmalsklassen sind unter dem Zug unabhängig:

$$P(\widehat{F}(t) \mid m) = \prod_{i \in I} P(F_i \circ T = F_i(t) \mid m) \quad (3.4)$$

gilt

$$P(T = t \mid m) = P(T = t) \cdot \prod_{i \in I} \frac{P(F_i \circ T = F_i(t) \mid m)}{P(F_i \circ T = F_i(t))} \quad (3.5)$$

Der Term $\widehat{F}(t)$ entspricht dabei dem Ereignis, dass alle Merkmale von T und t übereinstimmen. Voraussetzung (3.2) formuliert die Annahme, dass die gewählten Merkmalsklassen den Zusammenhang zwischen Zug und Tisch vollständig beschreiben. Voraussetzungen (3.3) und (3.4) drücken aus, dass die Merkmalsklassen vollständig orthogonale Aspekte der Kartenverteilung repräsentieren — die zentrale Voraussetzung dafür, dass sie getrennt voneinander aufgezeichnet und ausgewertet werden können.

Beweis. Der Ausgangsterm ist (3.2):

$$P(\{T = t\} \cap m \mid \widehat{F}(t)) = P(\{T = t\} \mid \widehat{F}(t)) \cdot P(m \mid \widehat{F}(t)) \quad (3.6)$$

Der Satz von Bayes, angewendet auf die Faktoren des Produkts, liefert

$$P(\{T = t\} \cap m \mid \widehat{F}(t)) = \frac{P(\widehat{F}(t) \mid T = t) \cdot P(T = t)}{P(\widehat{F}(t))} \cdot \frac{P(\widehat{F}(t) \mid m) \cdot P(m)}{P(\widehat{F}(t))} \quad (3.7)$$

Da $F_i, i \in I$ Abbildungen sind, folgt mit (3.1) die Präsenz von $\widehat{F}(t)$ direkt aus $T = t$ und es gilt $P(\widehat{F}(t) \mid T = t) = 1$. Damit lässt sich (3.7) vereinfachen zu

$$P(\{T = t\} \cap m \mid \widehat{F}(t)) = \frac{P(\widehat{F}(t) \mid m) \cdot P(m) \cdot P(T = t)}{P(\widehat{F}(t))^2} \quad (3.8)$$

Einsetzen der Definition der bedingten Wahrscheinlichkeit auf der linken Seite erlaubt das Kürzen von $P(\widehat{F}(t))$:

$$P(\{T = t\} \cap m \mid \widehat{F}(t)) = \frac{P(\widehat{F}(t) \mid m) \cdot P(m) \cdot P(T = t)}{P(\widehat{F}(t))} \quad (3.9)$$

Aus $P(\widehat{F}(t) | T = t) = 1$ folgt $\{T = t\} \subseteq \widehat{F}(t)$; die linke Seite kann vereinfacht werden zu

$$P(\{T = t\} \cap m) = \frac{P(\widehat{F}(t) | m) \cdot P(m) \cdot P(T = t)}{P(\widehat{F}(t))} \quad (3.10)$$

Erneutes Anwenden der Definition der bedingten Wahrscheinlichkeit auf der linken Seite und Kürzen von $P(m)$ liefert

$$P(T = t | m) = P(T = t) \cdot \frac{P(\widehat{F}(t) | m)}{P(\widehat{F}(t))} \quad (3.11)$$

Mit Einsetzen der Voraussetzungen (3.3) und (3.4) folgt schließlich die Behauptung

$$P(T = t | m) = P(T = t) \cdot \prod_{i \in I} \frac{P(F_i \circ T = F_i(t) | m)}{P(F_i \circ T = F_i(t))} \quad (3.12)$$

□

Mit der Wahrheit W einer gegebenen Situation (siehe Definition 2.6) wird im Folgenden für die Zufallsvariable T die Tischgleichverteilung $T \sim \mathcal{U}_W$ angenommen (siehe Definition 2.7). Der Faktor $P(T = t)$ in Gleichung (3.5) ist damit wie in Abschnitt 2.3 aus der Wahrheit zu bestimmen.

Der Term $P(F_i \circ T = F_i(t))$ ist vom Zug m unabhängig und beschreibt, mit welcher Wahrscheinlichkeit das beobachtete Merkmal $F_i(t)$ unter \mathcal{U}_W auftritt. Diese Wahrscheinlichkeit bestimmt sich ebenfalls aus der Wahrheit; dabei kann wieder die Zerlegung von W in Kombinationen (siehe Definition 2.9) ausgenutzt werden, um ein vollständiges Aufzählen der möglichen Tische zu vermeiden. Eine Beispielimplementierung für das Merkmal *TrumpCardCount*, das die Trumpfkarten des Alleinspielers bei Spielbeginn zählt (siehe Abschnitt 3.4), ist in Algorithmus 3.1 gegeben.

Die Wahrscheinlichkeiten $P(F_i \circ T = F_i(t) | m)$ sind damit als einzige vom Zug m abhängig und beschreiben nach Satz 3.1 den Einfluss des Zuges auf $P(T = t | m)$ vollständig. Diese Größen werden bei Simulation aus einem Offline-Modell bezogen, das vorab mit beobachteten Wahrscheinlichkeiten auf Trainingsdaten konstruiert wurde.

Dieses Modell kann jedoch schwer aus den Wahrscheinlichkeiten $P(F_i \circ T = F_i(t) | m)$ direkt aufgebaut werden, da T über $T \sim U_W$ von der Wahrheit W abhängt. Damit müsste W Teil des Modells sein und die beobachteten Daten könnten nicht auf Spiele verallgemeinert werden, die eine abweichende Historie aufweisen. Es müssen also weitere Vereinfachungen vorgenommen werden, um die Wahrscheinlichkeiten im Modell präsentierbar zu machen.

```

def trump_card_count_probability(truth, n):
    # Die Anzahl Trumpfkarten bezieht sich immer auf den Zeitpunkt der Spielansage,
    # sodass bereits ausgespielte Trumpfkarten berücksichtigt werden müssen
    known = truth.known_world_at_start[truth.soloist]
    n_known_trump = sum(1 for c in known if c.suit_class() == SuitClass.TRUMP)

    frequency = 0
    total = 0

    # Die Zerlegung nach Kombinationen liefert eine Menge an Längensignaturen zusam-
    # men mit der Anzahl konkreter Kartenverteilungen mit der jeweiligen Signatur.
    for n_variants, cmb in truth.combinations():
        n_trump_cards = n_known_trump + cmb[truth.soloist][SuitClass.TRUMP]
        if n_trump_cards == n:
            frequency += n_variants
            total = n_variants

    return frequency / total

```

Algorithmus 3.1: Bestimmung des Faktors $P(\text{TrumpCardCount} \circ T = n)$ für $n \in \mathbb{N}_0$.
 Durch Ausnutzung der Kombinationsstruktur der Wahrheit W bleibt die Laufzeitkomplexität der Auswertung gering.

Definition 3.4. Seien $F_i, i \in I$ Merkmale und m ein Zug. Dann bezeichnet $\mathbb{T}|_m \subset \mathbb{T}$ die Menge von Kartenverteilungen, in denen der Zug m beobachtet werden kann.

Für das oben erwähnte Merkmal *TrumpCardCount* handelt es sich dabei exemplarisch um Kartenverteilungen zum Zeitpunkt der Spielansage, also jene, in denen jeder Spieler 10 Karten hält.

Mit dieser Definition kann der letzte Vereinfachungsschritt beschrieben werden. Nach wie vor gilt $T \sim \mathcal{U}_W$, nun wird aber zusätzlich $V \sim \mathcal{U}_{\mathbb{T}|_m}$ eingeführt mit der vereinfachenden Annahme

$$P(F_i \circ T = F_i(t) \mid m) \approx P(F_i \circ V = F_i(t) \mid m) \quad (3.13)$$

Dadurch wird die Abhängigkeit zu W effektiv ignoriert. Damit kann das Modell aus Daten konstruiert werden, die bis auf die Wahrheit unter vergleichbaren Umständen aufgezeichnet wurden. Dieses Modell lässt sich formalisieren:

Definition 3.5. Sei (Ω, \mathcal{A}, P) ein Wahrscheinlichkeitsraum, m ein Zug, $V : \Omega \rightarrow \mathbb{T}$ eine Zufallsvariable mit $V \sim \mathcal{U}_{\mathbb{T}|m}$ und $F_i, i \in I$ eine Familie von Merkmalsklassen. Dann ist das **Gegnermodell** von m und $F_i, i \in I$, definiert durch die Werte

$$P(F_i \circ V = f \mid m), \quad f \in F_i(\mathbb{T}|m), i \in I \quad (3.14)$$

Solche Gegnermodelle werden aus aufgezeichneten Spielverläufen erstellt, indem die relative Häufigkeit der Merkmale unter den Zügen bestimmt wird. Nachdem in diesen vollständigen Historien die konkrete Kartenverteilung immer bekannt ist, ist das Konzept der Wahrheit dort nicht relevant — die Werte aus Gleichung (3.14) können direkt ermittelt werden und die Vereinfachung (3.13) muss nicht angewendet werden.

3.3 Simulation der Wahrscheinlichkeitsverteilung

Mit der Einführung von Zustandsinferenz wird die Simulationsgrundlage der Tischgleichverteilung \mathcal{U}_W aus Definition 2.7 durch eine komplexere Wahrscheinlichkeitsverteilung ersetzt, die beobachtete Züge berücksichtigt.

Definition 3.6. Sei (Ω, \mathcal{A}, P) ein Wahrscheinlichkeitsraum, W eine Wahrheit, m ein Zug und $t \in \mathbb{T}$ eine Stichprobe. Für $T, R : \Omega \rightarrow \mathbb{T}$ mit $T \sim \mathcal{U}_W$ ist die **Inferenzverteilung** $\mathcal{I}_{W,m}$ definiert als

$$R \sim \mathcal{I}_{W,m} \Leftrightarrow P(R = t) = P(T = t \mid m) \quad (3.15)$$

Die naheliegende Idee zur Simulation von $\mathcal{I}_{W,m}$ wäre nun, alle Kartenverteilungen aus W aufzuzählen und Stichproben gemäß den inferierten Wahrscheinlichkeiten zu ziehen. Wie in Abschnitt 2.2 bereits angesprochen, ist dieser Ansatz aufgrund der hohen Anzahl möglicher Tische nicht durchführbar.

In dieser Arbeit wird die Wahrscheinlichkeitsverteilung stattdessen durch ein zweistufiges Verfahren angenähert. Dazu werden zunächst *Stichprobenkandidaten* aus der Tischgleichverteilung gezogen, um diese dann mit ihrer inferierten Wahrscheinlichkeit zu gewichten und daraus die endgültigen Stichproben zu ziehen.

Unter der Inferenzverteilung $\mathcal{I}_{W,m}$ können keine Tische auftreten, die nicht auch durch die Tischgleichverteilung \mathcal{U}_W produziert werden können:

$$P(R = t) > 0 \Rightarrow P(T = t) > 0, \quad T \sim \mathcal{U}_W, R \sim \mathcal{I}_{W,m} \quad (3.16)$$

Das folgt direkt aus Definition 3.6. Damit eignet sich die kombinatorische Simulation von \mathcal{U}_W aus Abschnitt 2.3 als Grundlage für die Simulation von $\mathcal{I}_{W,m}$:

Verfahren 3.1 (Simulation der Inferenzverteilung). Sei (Ω, \mathcal{A}, P) ein Wahrscheinlichkeitsraum, W die Wahrheit einer Spielsituation, m ein beobachteter Zug und n die Anzahl zu ziehender Stichproben. Dann wird die Inferenzverteilung $\mathcal{I}_{W,m}$ wie folgt angenähert:

1. Simuliere die Tischgleichverteilung \mathcal{U}_W nach Verfahren 2.2 und ziehe daraus $N \gg n$ Stichprobenkandidaten, bezeichnet mit $(t_k)_{k \in \mathbb{N}}$.
2. Bestimme für $t_k, k \in \mathbb{N}$, die Wahrscheinlichkeit $P(T = t_k \mid m)$ mit $T \sim \mathcal{U}_W$.
3. Partitioniere das Intervall $[0,1]$ in N disjunkte Teilintervalle $(I_k)_{k \in \mathbb{N}}$, wobei die Länge von $I_k, k \in \mathbb{N}$, proportional zu $P(T = t_k \mid m)$ ist.
4. Für jede zu ziehende Stichprobe $i = 1, \dots, n$:
 - 4.1. Ziehe gleichverteilt $x \in \mathbb{R}$ aus $\mathcal{U}_{[0,1]}$.
 - 4.2. Finde $k \in \mathbb{N}$, sodass $x \in I_k$ und gib die Stichprobe t_k zurück.

Für $N \rightarrow \infty$ entspricht das Verfahren 3.1 der exakten Simulation der Inferenzverteilung, da es im Grenzwert äquivalent zum Aufzählen von W und anschließendem Ziehen von Stichproben t mit Wahrscheinlichkeit $P(R = t), R \sim \mathcal{I}_{W,m}$ ist. Um eine brauchbare Näherung zu gewährleisten, wurde für alle Auswertungen in dieser Arbeit die Anzahl der Stichprobenkandidaten auf $N = 100 \cdot n = 10.000$ festgelegt (siehe auch Abschnitt 2.4).

3.4 Inferenz in der Spielfindungsphase

Um das theoretische Konzept der merkmalsbasierten Zustandsinferenz praktisch anwenden zu können, bedarf es geeigneter Definitionen für Züge und Merkmalsklassen. Dazu werden zunächst Züge aus der Spielfindungsphase evaluiert, um die Inferenz anhand von Merkmalen der Kartenverteilung bei Ausspielbeginn zu ermöglichen.

In der Literatur

In der Implementierung von Long (2011) wird aus Gegenspielerposition die Spielansage als Zug betrachtet und aus Alleinpielerposition die Reizwerte der Gegenspieler. Die

Merkmalsklassen werden auf den Händen der betrachteten Spieler definiert, Beispiele sind etwa die Länge jeder Kartenfarbe, die exakte Konfiguration vorhandener Buben oder die Anzahl der Asse in einer Hand.

Hervorzuheben ist dabei, dass die Merkmale spieltypagnostisch gewählt sind: Sie werden logisch vor der Spielansage bestimmt, sodass der Spieltyp nicht in ihre Definition einfließt und nicht nach Farbklassen (siehe Definition 1.1) unterschieden werden kann. Lediglich die aufgedruckten Kartenfarben können Teil der Merkmalsdefinition sein. Diese Einschränkung ist notwendig, um den Reizvorgang der Gegenspieler zu bewerten, für dessen Entscheidungsgrundlage der (angestrebte) Spieltyp nicht bekannt ist. Dadurch können jedoch Merkmalsklassen wie die Länge der Trumpffarbe, die vom Spieltyp abhängig sind, nicht ausgedrückt werden. Gerade solche Größen werden aber von menschlichen Spielern für Zugentscheidungen herangezogen.

Auch erfüllen die gewählten Merkmalsklassen mindestens die Forderung nach Unabhängigkeit aus Gleichung (3.3) nicht — die verschiedenen Farblängen einer Hand können nicht unabhängig sein, da ihre Summe bei Spielbeginn immer 10 betragen muss. Die Annahmen aus Satz 3.1 sind daher in der Praxis als Vereinfachungen zu verstehen, die notwendig sind, um die Trennung von Merkmalsklassen zu ermöglichen. Long (2011) bleibt jedoch eine Untersuchung der Auswirkungen zur Rechtfertigung dieser Vereinfachungen schuldig.

Um Korrektheit zu garantieren, müsste stattdessen der Produktraum der abhängigen Merkmale betrachtet werden. Das würde zu einer mehrdimensionalen Merkmalsklasse führen, die etwa die genaue Längensignatur (siehe Definition 2.4) einer Hand beschreibt. Solche Produktklassen vervielfachen die Anzahl der möglichen Zustände, erfordern damit entschieden mehr Trainingsdaten und extrapolieren schlechter auf ungewöhnliche Merkmalskombinationen. Dies ist ein häufiges Problem in der Klassifikationstheorie und dort als *Curse of Dimensionality* bekannt.

Neben der merkmalsbasierten Inferenz greift Long (2011) zusätzlich auf ein Spielwahlmodul zurück, um Stichproben, in denen die beobachtete Spielansage unrealistisch erscheint, kategorisch auszuschließen (siehe Abschnitt 3.1).

Diese Umsetzung verbessert die Fähigkeiten des implementierten Spielers bereits merklich, sodass er sein Gegenstück ohne Inferenz mit einem deutlichen Punktevorsprung besiegen kann. Der Gewinn ergibt sich dabei fast ausschließlich aus der Beobachtung des Spielansagezugs; der Reizvorgang der Gegenpartei leistet nur einen geringfügigen Beitrag zur Spielstärke.

In dieser Arbeit

Im Folgenden wird das vorgestellte Verfahren auf die Spielansage eingeschränkt, um aussagekräftigere Merkmale in Abhängigkeit vom Spieltyp definieren zu können. Ein Verwerfen von Stichproben wie in der Literatur findet dabei nicht statt. Für die gewählten Merkmalskombinationen wird anschließend die Reproduktion des Gegnermodells durch Simulation der Inferenzverteilung analysiert, um die Auswirkungen der gewählten Vereinfachungen zu dokumentieren.

Züge der Spielfindungsphase

Die Züge der Spielfindungsphase können in unterschiedlichen Detailgraden ausgewertet werden. Prinzipiell kann der Ablauf der gesamten Spielfindung als Zug betrachtet werden; das schließt den exakten Reizvorgang und die Spielansage zusammen mit etwaigen Modifikatoren wie Hand oder Schneider ein.

Ein hoher Detailgrad führt jedoch wieder zu einem unnötig großen Zustandsraum, weshalb der Spielansagezug in dieser Arbeit auf den Spieltyp (Grand, ♣, ♠, ♥, ♦ oder Null) beschränkt wird. Die Ansage eines Spiels erfordert eine überlegene Hand für den Spieltyp und vermittelt damit viel Information über das Alleinspielerblatt.

Merkmalsklassen der Spielfindungsphase

Bei der Wahl von Merkmalsklassen sind mehrere Faktoren ausschlaggebend. Einerseits muss sich die beobachtete Häufigkeit der Merkmale im Gegnermodell deutlich von den Häufigkeiten unter der Tischgleichverteilung abheben, um die Wahrscheinlichkeitsverteilung messbar zu beeinflussen. Andererseits muss sichergestellt werden, dass die Menge aktiver Merkmalskombinationen in der Simulation hinreichend gut reproduziert werden kann — hier können sich die großzügigen Unabhängigkeitsannahmen (3.2), (3.3) und (3.4) sowie die Vereinfachung (3.13) negativ auswirken.

Um eine adäquate Wahl der Merkmalsklassen für die Spielansage zu gewährleisten, werden sie deshalb mit dem Reproduktionsmaß validiert, das am Ende dieses Abschnitts eingeführt wird.

Die gewählten Merkmale basieren grob auf denen aus Long (2011), befassen sich aber mit Farbklassen statt aufgedruckter Kartenfarben. Ergänzt wurden sie durch die Be-

schreibung der Farbfreiheit mit der Klasse *VoidSuitCount*, das Erkennen alleinstehender Karten mit *LoneRankPresence* und die Betrachtung des abgelegten Skats mit *AceSwapCardsCount* und *TrumpSwapCardsCount*. Konkret werden folgende Merkmalsklassen auf den Alleinspieler- und Skatkarten der Spielsituationen ausgewertet:

- **VoidSuitCount:** $\mathbb{T} \rightarrow \{0, \dots, n\}$: Anzahl der Farbklassen des Spieltyps ($n = 5$ bei Grand, $n = 4$ bei allen anderen Spieltypen), in denen der Alleinspieler keine Karte hält. Farbfreiheit erlaubt dem Alleinspieler, Stiche der Gegenpartei zu übernehmen.
- **TrumpCardCount:** $\mathbb{T} \rightarrow \{0, \dots, 10\}$: Anzahl der Trumpfkarten.
- **JackConfiguration:** $\mathbb{T} \rightarrow \{0, \dots, 15\}$: Die exakte Konfiguration der Buben als Summe der Werte 8, 4, 2 und 1 für den Kreuz-, Pik-, Herz- und Karobuben. Sie ist beim Grand entscheidend, da dort die vier Buben die einzigen Trümpfe darstellen und die bloße Trumpfanzahl nicht mehr aussagekräftig ist.
- **AceCount:** $\mathbb{T} \rightarrow \{0, \dots, 4\}$: Anzahl der Asse im Beiblatt.
- **JackCount:** $\mathbb{T} \rightarrow \{0, \dots, 4\}$: Anzahl der Buben. Ein größeres Maß als *JackConfiguration*, das für Farbspiele geeignet ist.
- **TrumpSwapCardsCount:** $\mathbb{T} \rightarrow \{0, \dots, 2\}$: Anzahl der Trumpfkarten im Skat nach dem Drücken. Bis auf exotische Situationen ist es für den Alleinspieler immer von Nachteil, Trumpf zu drücken.
- **AceSwapCardsCount:** $\mathbb{T} \rightarrow \{0, \dots, 2\}$: Anzahl der Asse im Skat nach dem Drücken. Das Ablegen eines Asses ist im Grand schwer zu rechtfertigen.
- **KingQueenCount:** $\mathbb{T} \rightarrow \{0, \dots, 8\}$: Anzahl der Damen und Könige im Beiblatt.
- **LowCardCount:** $\mathbb{T} \rightarrow \{0, \dots, 12\}$: Anzahl der Luschen (7, 8, 9) im Beiblatt. Diese Karten sind beim Nullspiel vermehrt vertreten.
- **LoneRankPresence:** $(r_1, r_2, \dots) : \mathbb{T} \rightarrow \mathbb{N}_0$: Anzahl der blanken Karten im Beiblatt, deren Kartenrang in $\{r_1, r_2, \dots\}$ liegt. Eine blanken Karte ist die einzige ihrer Farbkategorie in der betrachteten Hand. Diese Merkmalsklasse wird für den jeweiligen Anwendungsfall nach r_1, \dots parametrisiert.

Abhängig vom Spieltyp bringen gewisse blanken Karten einen starken Nachteil: Bei Trumpfspiele wird eine blanken Zehn fast sicher gestochen; bei einem Nullspiel stellt eine blanken hohe Karte (B, D, K, A) ein starkes Verlustrisiko dar.

Wie in der Aufzählung bereits erwähnt, haben die Merkmale unterschiedliche Signifikanz für die verschiedenen Spieltypen. Da die Merkmalsklassen über die Aufteilung nach Farbklassen bereits vom Spieltyp abhängig sind, kann auch die Auswahl zwischen ihnen in Abhängigkeit von der Spielansage erfolgen. Dieser Schritt findet in der verwendeten Literatur noch keine Anwendung. Tabelle 3.1 listet auf, welche Klassen bei welchen Spieltypen zum Einsatz kommen.

Spieltypen	Merkmalsklassen
Kreuz	<i>TrumpCardCount</i>
Pik	<i>VoidSuitCount</i>
Herz	<i>JackCount</i>
Karo	<i>TrumpSwapCardsCount</i> <i>LoneRankPresence(10)</i>
Grand	<i>VoidSuitCount</i> <i>JackConfiguration</i> <i>AceCount</i> <i>TrumpSwapCardsCount</i> <i>AceSwapCardsCount</i> <i>LoneRankPresence(10)</i>
Null	<i>VoidSuitCount</i> <i>AceCount</i> <i>KingQueenCount</i> <i>LowCardCount</i> <i>LoneRankPresence(B, D, K, A)</i>

Tabelle 3.1: Gewählte Merkmalsklassen in Abhängigkeit vom angesagten Spieltyp

Auswirkungen der Unabhängigkeitsannahme

Wie oben bereits angemerkt, sind die Unabhängigkeitsannahmen zwischen den Merkmalsklassen aus Satz 3.1 lediglich eine Vereinfachung. Auch geht die Abschätzung (3.13) in die konstruierte Inferenzverteilung ein. Die Frage, ob diese Näherungen gerechtfertigt sind, lässt sich anhand der Reproduktionsgenauigkeit der Merkmale in der simulierten Wahrscheinlichkeitsverteilung beantworten.

Dazu wurde in dieser Arbeit ein Gegnermodell für die Spielansageinferenz basierend auf 100.000 zufällig ausgewählten Spielen der ISS-Datenbank (siehe Abschnitt 1.7) aufgezeichnet. Handspiele wurden ausgeschlossen, um einer gesonderten Behandlung der Merkmalsklassen *AceSwapCardsCount* und *TrumpSwapCardsCount* aus dem Weg zu gehen. Overtspiele wurden ebenfalls nicht betrachtet.

Anschließend wurden für jeden Spieltyp 10.000 Spiele der Datenbank ausgewertet, indem für jedes Spiel 100 Stichproben der geschätzten Wahrscheinlichkeitsverteilung gezogen und die relative Häufigkeit der Merkmale bestimmt wurde.

Abbildungen 3.1, 3.2 und 3.3 zeigen die Merkmalshäufigkeiten der verschiedenen Wahrscheinlichkeitsverteilungen¹. Die Datenreihen zum „Gegnermodell“ beschreiben die beobachteten Merkmale bei Erzeugung des Modells, „Gleichverteilung“ die Merkmale in den Stichproben der Tischgleichverteilung und „Reproduktion“ die Merkmale in den gezogenen Stichproben der Inferenzverteilung.

Eindeutige Merkmale wie *TrumpSwapCardsCount* werden hierbei nahezu perfekt reproduziert. Das kann dadurch erklärt werden, dass das Gegnermodell für diese Klassen Merkmalswahrscheinlichkeiten nahe 0 und 1 vergibt, wodurch die Simulation effektiv Stichproben mit Wahrscheinlichkeit 0 verwirft und jene mit Wahrscheinlichkeit 1 unberührt lässt.

Komplexere Klassen wie *VoidSuitCount* weisen dagegen eine sichtbare Abweichung vom Modell auf. Trotzdem zeigen die Merkmalswahrscheinlichkeiten der Inferenzverteilung deutlich mehr Gemeinsamkeiten mit dem Gegnermodell als die der Gleichverteilung. Dieses Ergebnis zeigt, dass das Verfahren trotz der erheblichen Vereinfachungen akzeptable Ergebnisse erzeugen kann — vorausgesetzt die Merkmalsklassen werden mit Bedacht gewählt.

¹Die Wahl eines Liniendiagramms soll hier keine Kontinuität suggerieren, sondern dient lediglich der besseren visuellen Unterscheidbarkeit der Daten.

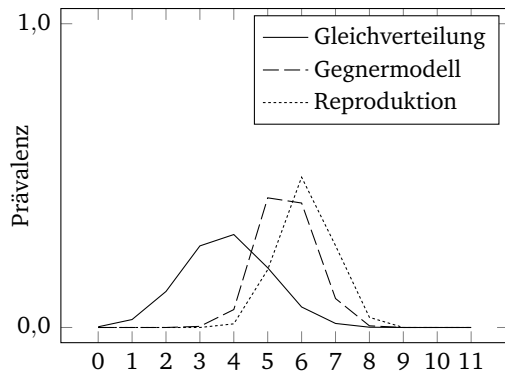
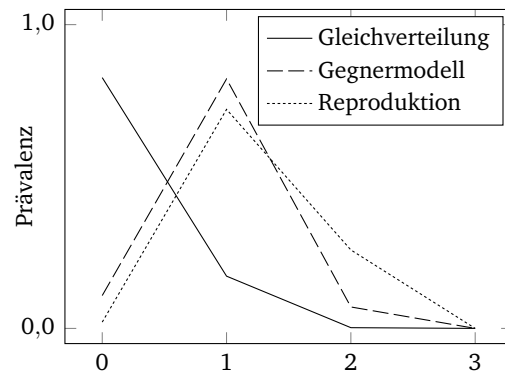
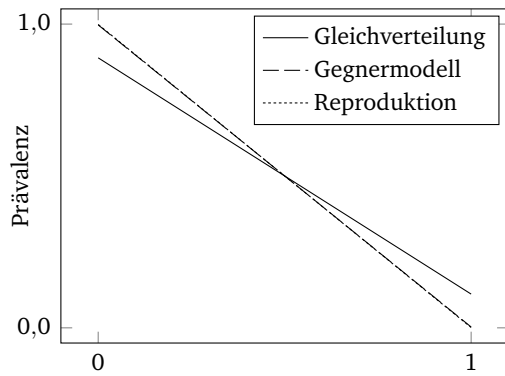
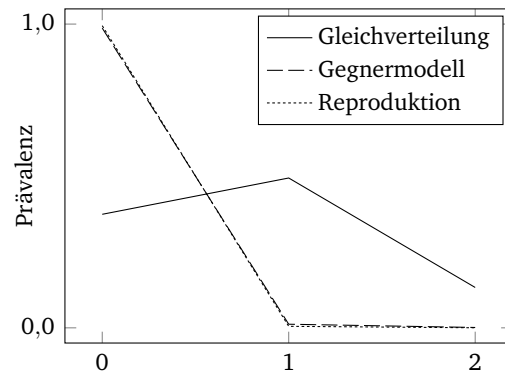
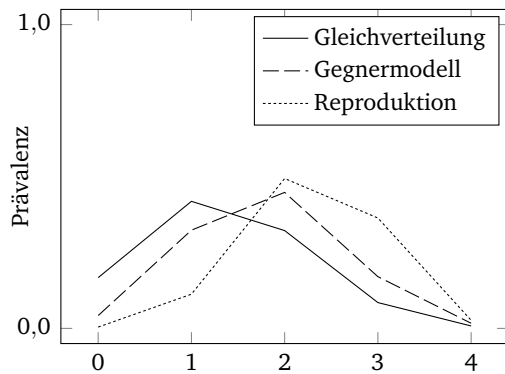
(a) *TrumpCardCount*(b) *VoidSuitCount*(c) *LoneRankPresence(10)*. Gegnermodell und Reproduktion sind identisch(d) *TrumpSwapCardsCount*. Gegnermodell und Reproduktion sind nahezu identisch(e) *JackCount*

Abbildung 3.1: Merkmalsreproduktion im Farbspiel. Eindeutige Merkmale wie *TrumpSwapCardsCount* in (d) werden perfekt reproduziert; komplexere Klassen wie *VoidSuitCount* in (b) weichen sichtbar vom Gegnermodell ab.

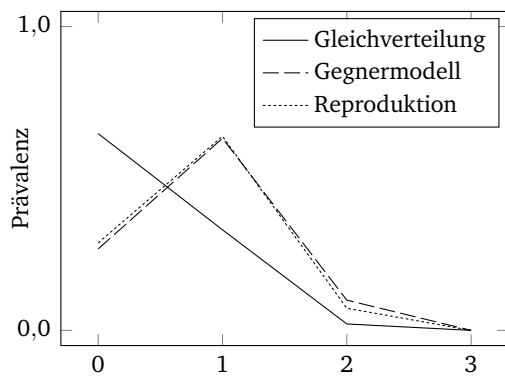
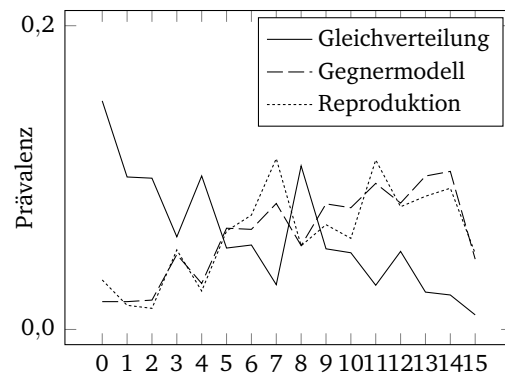
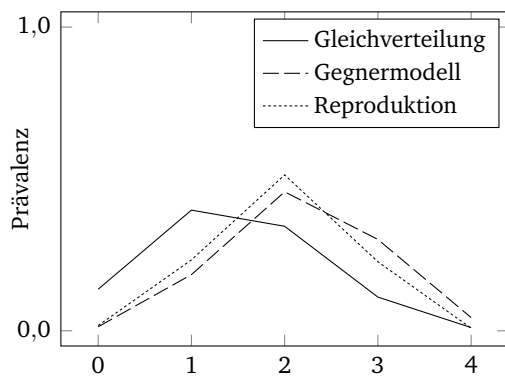
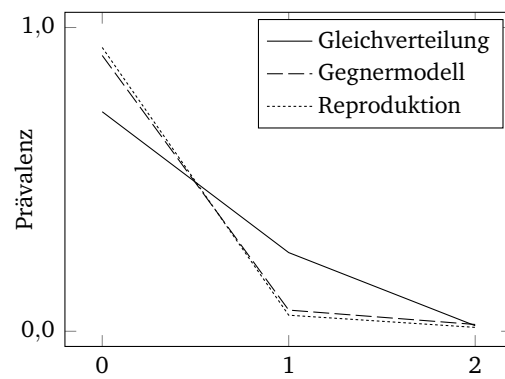
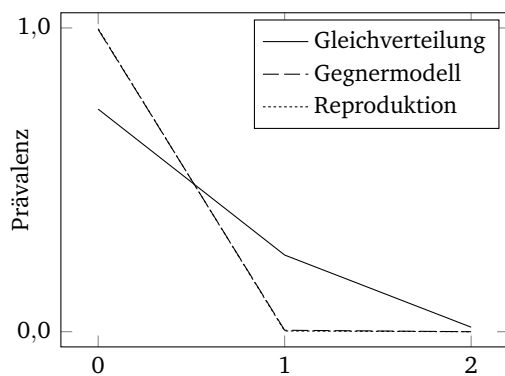
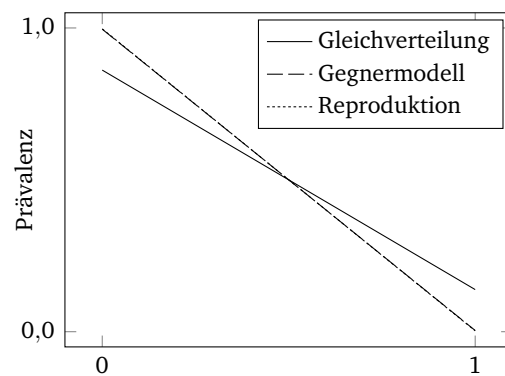
(a) *VoidSuitCount*(b) *JackConfiguration*(c) *AceCount*(d) *AceSwapCardsCount*(e) *TrumpSwapCardsCount*. Gegnermodell und Reproduktion sind nahezu identisch(f) *LoneRankPresence(10)*. Gegnermodell und Reproduktion sind identisch

Abbildung 3.2: Merkmalsreproduktion im Grand. Eindeutige Merkmale wie *TrumpSwapCardsCount* in (e) werden perfekt reproduziert; komplexere Klassen wie *JackConfiguration* in (b) weichen sichtbar vom Modell ab.

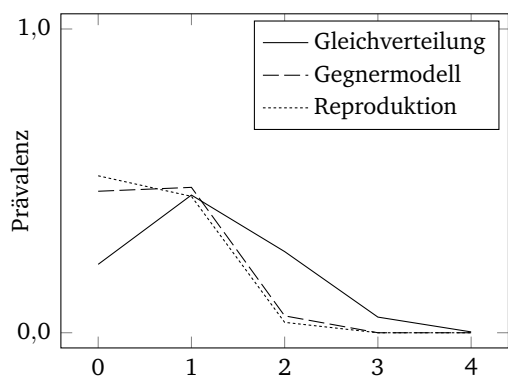
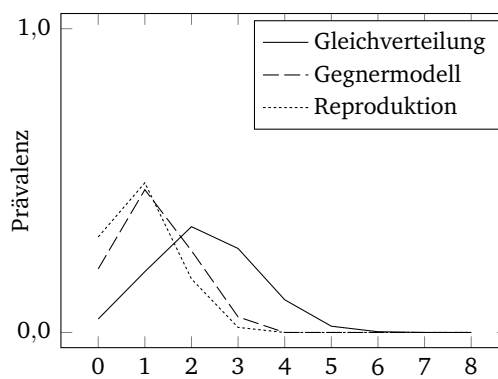
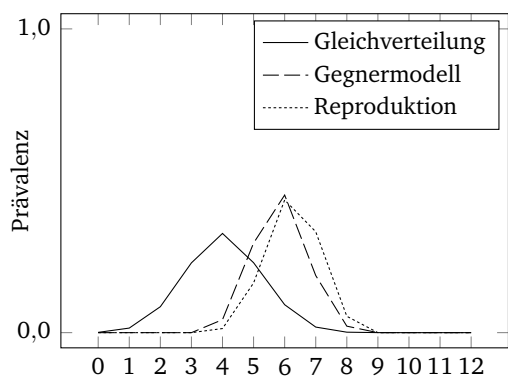
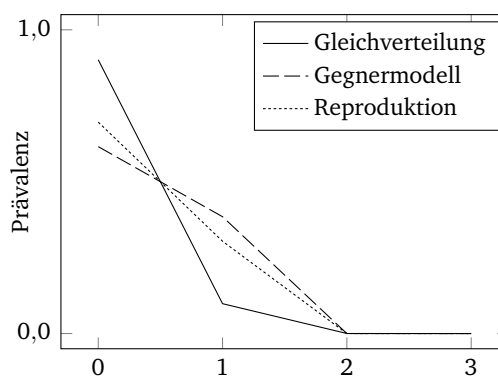
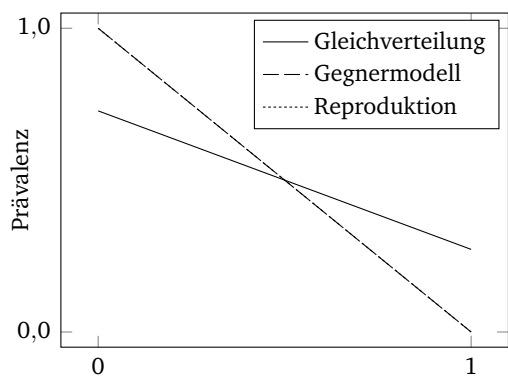
(a) *AceCount*(b) *KingQueenCount*(c) *LowCardCount*(d) *VoidSuitCount*(e) *LoneRankPresence(B, D, K, A)*. Gegnermodell und Reproduktion sind identisch

Abbildung 3.3: Merkmalsreproduktion im Nullspiel. Eindeutige Merkmale wie *LoneRankPresence* in (e) werden perfekt reproduziert; komplexere Klassen wie *VoidSuitCount* in (d) weichen sichtbar vom Modell ab.

3.5 Inferenz in der Ausspielphase

Das Grundkonzept der Zustandsinferenz in der Spielfindungsphase, das in Abschnitt 3.4 ausführlich besprochen wurde, wird in dieser Arbeit zum ersten Mal auf die Ausspielphase übertragen. Wie das Ansagen eines Spiels lässt auch das Ausspielen einer Karte Schlüsse auf die Hand zu, die einem guten Skatspieler nicht verborgen bleiben dürfen.

Ein menschlicher Spieler kann anhand einstudierter Verhaltensweisen die Menge der Karten einschränken, die dem beobachteten Mitspieler bei seinem Zug zur Auswahl standen. Muss ein Spieler etwa Farbe bedienen und wirft dabei eine teure 10 in einen sicher verlorenen Stich ab, so kann mit hoher Wahrscheinlichkeit davon ausgegangen werden, dass er keine billigere Karte dieser Farbe mehr auf der Hand hatte.

Analog zu Abschnitt 3.4 werden diese heuristischen Annahmen nun in ein merkmalsbasiertes Inferenzmodul übersetzt. Das grundlegende Vorgehen ist dem der Spielansageinferenz ähnlich, unterscheidet sich aber durch eine deutlich höhere Komplexität in der Definition eines Zuges.

Da in der Ausspielphase das Verhalten der Gegner beim Spielen von Stichen beobachtet wird, wird im Folgenden diese Form der Inferenz auch als *Stichinferenz* bezeichnet. Um die Zustandsinferenz aus Allein- und Gegenspielerpersicht logisch trennen zu können, wird die Stichinferenz in dieser Arbeit nur zur Bewertung der Gegenspielerkarten aus Alleinspielerpersicht benutzt.

Züge der Ausspielphase

Die Züge der Ausspielphase unterscheiden sich von denen der Spielfindungsphase fundamental darin, dass ihnen eine nicht zu vernachlässigende Spielhistorie vorausgeht. Zum Zeitpunkt der Spielansage kann zwar der Reizvorgang als Historie betrachtet werden, der wegen seines untergeordneten Einflusses auf die Wahl des Spieltyps in dieser Arbeit aber ignoriert wird.

In der Ausspielphase hat das Legen einer bestimmten Karte hingegen allein wenig Aussagekraft. Ein Zug lässt erst durch den Kontext, in dem er ausgespielt wird, eine Strategie erkennen. So kann das Trumpf-Ass etwa in Hinterhand einen Stich übernehmen oder durch Farbzwang in einen verlorenen Stich abgegeben werden.

Um die wichtigsten Aspekte der Spielsituation und -historie abzubilden, ohne den gesamten Spielverlauf einzubeziehen, werden sie im *Stichkontext* zusammengefasst. Er übernimmt in der Ausspielphase die Rolle des Zuges (siehe Definition 3.1) für die Zustandsinferenz.

Definition 3.7. Ein *Stichkontext* m ist das Tupel (Y, C, P, F, I, A) aus

- **Spieltyp** $Y \in \{\text{Grand}, \clubsuit, \spadesuit, \heartsuit, \diamondsuit, \text{Null}\}$: Das Ergebnis der Spielansage ist zwingend erforderlich, um Karten in Farbklassen (siehe Definition 1.1) einteilen zu können.
- **Ausgespielte Karte** C : Die exakte Kartenidentität mit Kartenrang und -farbe. Eine Verallgemeinerung wäre vom Spieltyp abhängig, weswegen sie hier nicht erfolgt.
- **Stichposition** des Ausspielers $P \in \{\text{Vorhand}, \text{Mittelhand}, \text{Hinterhand}\}$. Abhängig von der Position sind unterschiedliche Informationen über den Stich bekannt und es bieten sich verschiedene Strategien an.
- **Bedienen der angespielten Farbe** $F \in \{0,1\}$. Es gilt $F = 1$, falls die Farbklassse von C der der ersten Karte im Stich entspricht, ansonsten ist Farbfreiheit des Spielers bekannt und $F = 0$. Farbfreiheit hat einen großen Einfluss auf die Menge der spielbaren Karten. Dieser Wert wird nur betrachtet, wenn $P \in \{\text{Mittelhand}, \text{Hinterhand}\}$.
- **Intention** des Zuges $I \in \{\text{Übernehmen}, \text{Schmieren}, \text{Abwerfen}\}$. Dieser Wert beschreibt den Effekt der Karte auf den bisherigen Stich: Durch „Übernehmen“ wechselt der Stich zur Partei des Ausspielers, beim „Schmieren“ wird eine Karte in den Stich des Teampartners zugegeben und „Abwerfen“ bezeichnet das Legen einer Karte in den Stich der gegnerischen Partei, ohne dass der Stich dabei den Gewinner wechselt. Dieser Wert wird ebenso nur für $P \in \{\text{Mittelhand}, \text{Hinterhand}\}$ betrachtet.
- **Stichposition des Alleinspielers** $A \in \{\text{Vorhand}, \text{Mittelhand}, \text{Hinterhand}\}$. Dieser Aspekt beeinflusst etwa, ob ein Gegenspieler in Mittelhand befürchten muss, dass der Stich durch den Alleinspieler in Hinterhand übernommen wird.

Damit sind in der Ausspielphase 7.488 unterschiedliche Züge möglich, was im Kontrast zu den nur 6 möglichen Zügen in der Spielfindungsphase steht. Dies hat einen negativen Einfluss auf die Genauigkeit des Gegnermodells, da jeder einzelne Zug in den Trainingsdaten seltener beobachtet wird. Dieser Umstand kann nur teilweise durch die größere Häufigkeit von 30 Ausspielzügen gegenüber einem Spielansagezug pro Historie relativiert werden.

Merkmale der Ausspielphase

Unabhängig von der Phase des Spiels beziehen sich Merkmale immer auf die Kartenverteilung zum Zeitpunkt des beobachteten Zuges. Während für die Spielansage relevant ist, welche Karten für den Gewinn eines gesamten Spiels ausreichend sind, ist eine ausgespielte Karte oft lokal motiviert: Häufig ist das Ziel eines Zuges lediglich, die Punkte des aktuellen Stichs zu sichern oder sich für den nächsten Stich in eine vorteilhafte Position zu bringen.

Um die lokale Entscheidungsgrundlage eines Zuges zu beschreiben, werden Merkmalsklassen so gewählt, dass sie sich immer auf die Farbklasse der ausgespielten Karte beziehen. Dazu werden sie mit der Farbklasse a parametrisiert. Genau wie in der Spielfindungsphase werden die betrachteten Merkmalsklassen anhand des Zuges vorausgewählt: Interessant ist für das Modell, welche *anderen* Karten in dieser Situation spielbar gewesen wären, gegen die sich der Spieler aber aktiv entschieden hat.

Die Merkmalsklassen dazu stammen nicht aus der Literatur, sondern wurden hier neu entwickelt.

- ***LastCardInSuitClass***(a) $\in \{0, 1\}$: Gleich 1, falls sich zum Zeitpunkt des Zuges genau eine Karte der Farbklasse a beim Spieler befindet. Bildet das Freispielen zum Erreichen von Farbfreiheit ab, bei dem mit dem Zug m die letzte Karte einer Farbklasse abgeworfen oder zugegeben wird.
- ***MinCardPointsInSuitClass***(a) $\in \{0, 2, 3, 4, 10, 11\}$: Die niedrigste Augenzahl einer Karte in der Farbklasse a zum Zeitpunkt des Zuges, die Karte aus m eingeschlossen. Interessant beim Abwerfen einer Karte in einen verlorenen Stich: Geht eine Karte durch Farbzwang sicher verloren, so ist es meist sinnvoll, die billigste ihrer Klasse abzuwerfen.
- ***MaxCardPointsInSuitClass***(a) $\in \{0, 2, 3, 4, 10, 11\}$: Die höchste Augenzahl einer Karte in a einschließlich der des Zuges m — das Gegenstück zu *MinCardPointsInSuitClass*. Interessant etwa beim Schmieren einer teuren Karten in einen Stich des Partners: Ist ein Stich sicher gewonnen, so ist es oft günstig, die Karte mit der höchsten Augenzahl zuzugeben.

Zusammenfassen mehrerer Züge

Eine Neuerung der hier entwickelten Stichinferenz ist das potentielle Vorhandensein mehrerer Züge. Während es nur eine Spielansage gibt, können pro Spiel mehrere ausgespielte Karten beobachtet werden. Die Information aus diesen Zügen muss angemessen kombiniert werden, um das volle Potential der Zustandsinferenz auszuschöpfen.

Nicht jeder Zug lässt Schlüsse auf jeden Aspekt der Kartenverteilung zu. In der obigen Modellierung beeinflusst die Beobachtung einer ausgespielten Karte die Wahrscheinlichkeitsverteilung nur in Form der Merkmale aus derselben Farbkategorie. Dies ermöglicht ein Zusammenfassen von Zügen innerhalb der Farbkategorien:

Bezeichne $A := \{T, \clubsuit, \spadesuit, \heartsuit, \diamondsuit\}$ die Menge aller Farbkategorien. Für die vorliegende Spielsituation sei weiter M_a die Menge der beobachteten Züge in Farbkategorie a und \mathbb{F}_a die Menge aller mit a parametrisierten Merkmalskategorien. Dann werden für jede Wahl von Farbkategorien $a, b \in A$, $a \neq b$, jedes Merkmal $F_a \in \mathbb{F}_a$ und alle Züge $m_a \in M_a$ und $m_b \in M_b$ die folgenden Unabhängigkeiten angenommen:

$$P(F_a \circ T = F_a(t) \mid m_b) = P(F_a \circ T = F_a(t)) \quad (3.17)$$

$$P(F_a \circ T = F_a(t) \mid m_a \cap m_b) = P(F_a \circ T = F_a(t) \mid m_a) \quad (3.18)$$

Die einzelnen Merkmalswahrscheinlichkeiten $P(F \circ T = F(t) \mid m)$ bilden für jeden Zug $m \in M_a$ einen Schätzer für alle Merkmale $F \in \mathbb{F}_a$ der selben Farbkategorie. Diese Einzelschätzungen können durch Mittelwertbildung zusammengefasst werden:

$$P\left(F \circ T = F(t) \mid \bigcap_{m \in M_a} m\right) \approx \frac{1}{\#M_a} \sum_{m \in M_a} P(F \circ T = F(t) \mid m) \quad (3.19)$$

Das Einsetzen in die allgemeine Inferenzgleichung (3.5) ergibt die folgende Formel zur Ermittlung der vollständigen Stichprobenwahrscheinlichkeit:

$$P\left(T = t \mid \bigcap_{m \in M_a} m\right) \approx P(T = t) \cdot \prod_{a \in A} \prod_{F \in \mathbb{F}_a} \frac{\sum_{m \in M_a} P(F \circ T = F(t) \mid m)}{\#M_a P(F \circ T = F(t))} \quad (3.20)$$

Auch diese Formalisierung ist wieder eine Näherung, die die Komplexität des Zugraums auf Kosten der Exaktheit reduziert. Sie ist aber zumindest dem naiven Mitteln einzeln bestimmter Wahrscheinlichkeiten $P(T = t \mid m_1)$, $P(T = t \mid m_2)$, ... theoretisch überlegen, da sie nach Farbkategorien trennt und durch den Zug unberührte Merkmalskategorien in der Mittelwertbildung nicht beeinflusst.

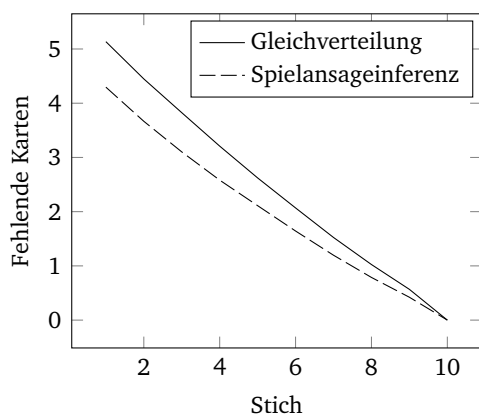
3.6 Evaluation

Um die Effektivität der vorgestellten Inferenzmethoden beziffern zu können, wird ein neues, passendes Fehlermaß für die Qualität eines Schätzers eingeführt.

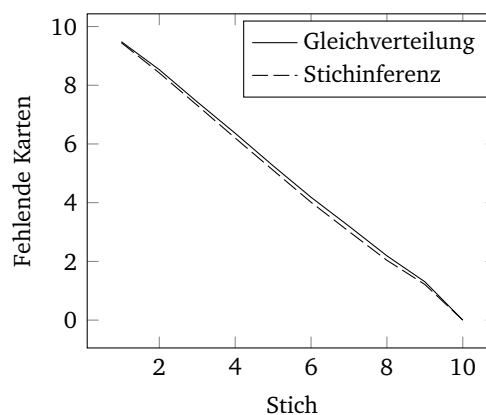
Definition 3.8. Sei $t = (H_j)_{j \in J} \in \mathbb{T}$ eine Kartenverteilung, $\{t^{(i)} = (H_j^{(i)})_{j \in J} \mid i \in I\} \subset \mathbb{T}$ eine Menge von Stichproben für die Indexmenge I und $J' \subseteq J$ eine Teilmenge von Tischpositionen. Das Maß der mittleren **fehlenden Karten pro Stichprobe** ist definiert als

$$X(t, t^{(\cdot)}, J') := \frac{1}{\#I} \sum_{i \in I} \sum_{j \in J'} \#\{H_j \setminus H_j^{(i)}\}$$

Das Maß zählt, wie viele Karten an vorgegebenen Positionen im Mittel falsch zugeordnet wurden. Die Parametrisierung nach J' ist notwendig, da die beiden Inferenzansätze unterschiedliche Positionen betreffen: Die Spielansageinferenz bewertet ausschließlich die Alleinspielerkarten, während sich die Stichinferenz auf die Hand der beiden Gegenspieler beschränkt.



(a) Alleinspielerkarten aus Gegenspieler-sicht. Es sind 10 Karten zu bestimmen.



(b) Gegenspielerkarten aus Alleinspielersicht. Es sind 20 Karten zu bestimmen.

Abbildung 3.4: Mittlere fehlende Karten pro Stichprobe (Definition 3.8) mit und ohne Zustandsinferenz. Die Spielansageinferenz kann die Einschätzung der realen Wahrscheinlichkeitsverteilung deutlich verbessern.

Für Abbildung 3.4 wurde das Fehlermaß X anhand von 1.000 zufällig ausgewählten Spielhistorien der ISS-Datenbank (siehe Abschnitt 1.7) mit 100 Stichproben pro Situation ausgewertet. Die Spielansageinferenz kann dabei den Fehler in den Alleinspielerkarten deutlich reduzieren, da der angesagte Spieltyp wie erwartet ein verlässlicher Indikator für die Hand des Alleinspielers ist.

Die Stichinferenz bringt dagegen nur einen marginalen Gewinn gegenüber der Tischgleichverteilung. Offensichtlich sind die Zusammenhänge zwischen Zügen und Merkmalen zu komplex für den vorgestellten einfachen Apparat. Es ist zu erwarten, dass die geringe Reduktion der fehlenden Karten nicht zu einer messbaren Verbesserung der Spielstärke führt (siehe auch Abschnitt 5.1).

Um die ermittelten Werte theoretisch nachzuvollziehen, wird exemplarisch der erste Datenpunkt der Tischgleichverteilung aus Abbildung 3.4a anhand eines Urnenmodells ermittelt. Zu Beginn des ersten Stichs kann dabei die Hand des Alleinspielers durch Ziehen von 10 Karten ohne Zurücklegen aus der Menge der 22 unbekanntes modelliert werden. $10 - X$, die Anzahl korrekt zugeordneter Karten, ist dann hypergeometrisch verteilt zu

$$10 - X \sim h(k | 22; 10; 10), \quad 0 \leq k \leq 10 \quad (3.21)$$

Das Maß der falsch zugeordneten Karten pro Stichprobe X hat in dieser Situation damit den Erwartungswert

$$E(X) = 10 - \sum_{k=0}^{10} h(k | 22; 10; 10) \cdot k \approx 5.45 \quad (3.22)$$

was sich mit dem ersten Datenpunkt in Abbildung 3.4a deckt.

In den nachfolgenden Stichen fällt der Fehler X auch ohne Zustandsinferenz superlinear, da durch beobachtete Farbfreiheit Informationen über noch nicht ausgespielte Karten bekannt werden.

3.7 Zusammenfassung

In diesem Kapitel wurde gezeigt, dass sich mit merkmalsbasierter Zustandsinferenz ein deutlich verbesserter Schätzer für die reale Wahrscheinlichkeitsverteilung einer Situation konstruieren lässt. Dazu wurde der Ansatz von Long (2011) und Buro u. a. (2009) zur Inferenz in der Spielfindungsphase präzisiert, um neue Merkmalsklassen erweitert und die Zulässigkeit seiner Grundannahmen überprüft.

Anschließend wurde die Methode auf den Kontext der Ausspielphase verallgemeinert, um einen Schätzer auf Basis ausgespielter Stiche zu beschreiben. Dazu wurden geeignete Merkmale und Züge definiert, die eine partielle Spielhistorie in Form des Stichkontextes berücksichtigen.

Schließlich wurde ein Fehlermaß entwickelt, um die Qualität der vorgestellten Inferenzverfahren zu beziffern. Durch dieses Maß wird sichtbar, dass die Zustandsinferenz anhand der Spielansage die Schätzung der Alleinspielerkarten deutlich verbessern kann. Im Gegensatz dazu ermöglicht die Inferenz anhand ausgespielter Karten nur kleine Gewinne bei der Herleitung der Gegenspielerkarten.

Da aber erfahrene Spieler oft bereits nach wenigen Stichen die Kartenverteilung mit hoher Sicherheit kennen, bleibt die Stichinferenz hier weit hinter ihrem Potential zurück. Eine mögliche Erklärung dafür ist die Komplexität der beobachteten Züge, die große Mengen an Kontextinformation enthalten. Die Relevanz der einzelnen Komponenten des Stichkontextes wurde hier nicht weiter studiert. Diese Arbeit liefert aber das Werkzeug, um durch eine ausgefeiltere Definition von Zügen und Merkmalsklassen eine starke Zustandsinferenz zu realisieren.

4 Spielstärkemaße

Um die Spielstärke des Computers an seine Gegner anzupassen, ist es zunächst notwendig, deren Fähigkeiten korrekt einzuschätzen. Zu diesem Zweck werden im Folgenden mehrere Maße für die Spielstärke entwickelt.

Definition 4.1. Sei P eine Menge von Spielern. Ein **Spielstärkemaß** ist eine Abbildung $B : P \rightarrow \mathbb{R}$, die einem Akteur anhand seiner Spielhistorie eine Spielstärke zuordnet.

Ein solches Maß kann sowohl auf Spielebene das Reiz- und Gewinnverhalten betrachten als auch auf Stichebene einzelne Züge analysieren. Dabei ist wünschenswert, dass die gemessene Spielstärke unterschiedlich starke Spieler scharf trennt, gleichzeitig aber schnell konvergiert, um die Einschätzung nach einer überschaubaren Anzahl beobachteter Spiele zu ermöglichen.

Unter Turnierbedingungen werden Skatspieler traditionell an *Listenpunkten* gemessen. In Abschnitt 4.1 wird gezeigt, wie dieses Maß zu bestimmen ist und welche Schwächen es aufweist.

In Abschnitt 4.2 wird ein alternatives Spielstärkemaß aus der Literatur vorgestellt, das auf dem Nachvollziehen beobachteter Züge in offenen Situationen basiert (siehe Abschnitt 1.4). Die existierenden Ergebnisse werden reproduziert und Vergleichbarkeit sowie Nachteile des Maßes aufgezeigt.

Daraufhin wird in Abschnitt 4.3 ein neues Spielstärkemaß entwickelt, das beobachtete Züge mit Perfect-Information-Monte-Carlo-Suche (kurz PIMC-Suche, siehe Kapitel 2) nachvollzieht. Es wird mit den existierenden Verfahren verglichen und demonstriert, dass es ihnen in Exaktheit und Konvergenzgeschwindigkeit deutlich überlegen ist.

In Abschnitt 4.4 wird eine Metrik zur Bestimmung der Konvergenzgeschwindigkeit der Spielstärke eingeführt und daran die beschriebenen Maße verglichen. Schließlich werden die vorgestellten Verfahren in Abschnitt 4.5 mit den Daten eines Skatexperten verifiziert und aufgezeigt, in welchen Situationen sie unzureichend sind.

Um belastbare Aussagen über die entwickelten Maße treffen zu können, müssen sie auf einer großen Zahl von Spielen unterschiedlicher Akteure ausgewertet werden. Da es genügt, diese Analysen nach Abschluss eines Spiels stattfinden zu lassen, kann diese Bewertung anhand von Offline-Daten erfolgen. Dazu wird wieder auf die ISS-Datenbank zurückgegriffen (siehe Abschnitt 1.7).

4.1 Mittelung der Listenpunkte

Eine *Serie* oder *Liste* von 36 Spielen ist der Rahmen, in dem sich Spieler in offiziellen Skatturnieren messen. Die Punktesumme aus mehreren Listen entscheidet dort über das Ergebnis des Wettbewerbs (vgl. DSkV (2010)). Es ist daher naheliegend, die Fähigkeiten eines Spielers anhand der mittleren Listenpunkte zu messen, die er erreicht.

Im Skat existieren mehrere Abrechnungsmethoden, um den Punktwert eines Spiels zu bestimmen. In dieser Arbeit wird die *Seeger-Fabian-Abrechnungsmethode* verwendet, die auch der Abrechnungsregel in Turnieren entspricht (vgl. DSkV (2010)). Den präsentierten Zahlen nach zu urteilen deckt sich diese Entscheidung mit existierender Literatur (vgl. Long und Buro (2011)).

Definition 4.2. Sei v der Spielwert eines Skatspiels und p ein teilnehmender Spieler. Dann ist der **Seeger-Fabian-Wert** $w(v, p)$ definiert als

$$w(v, p) := \begin{cases} v + 50, & p \text{ ist Alleinspieler und hat gewonnen} \\ -(2 \cdot v + 50), & p \text{ ist Alleinspieler und hat verloren} \\ 40, & p \text{ ist Gegenspieler und hat gewonnen} \\ 0, & \text{sonst} \end{cases} \quad (4.1)$$

Die Listenpunkte eines Spielers ergeben sich dann aus der Summe seines Seeger-Fabian-Werts über 36 Spiele. Um die Beschränkung auf diese exakte Spielzahl zu umgehen, wird das Maß wie folgt formalisiert:

Definition 4.3. Seien $(v_i)_{i \in I}$ Spielwerte einer Folge von Skatspielen I und p ein Spieler. Dann ist das Maß der **mittleren Listenpunkte** V definiert als

$$V := \frac{36}{\#I} \sum_{i \in I} w(v_i, p) \quad (4.2)$$

Skatspieler können nun über ihren erreichten Wert für V miteinander verglichen werden. Dieser Wert ist jedoch nicht allein von den Fähigkeiten des Spielers abhängig. Die zufällige Kartenverteilung entscheidet in jedem Spiel darüber, wer die Rolle des Alleinspielers übernehmen kann und somit in die Chance zu einem signifikanten Punktegewinn hat. Auch das Niveau der Gegenspieler hat einen starken Einfluss auf die Gewinnwahrscheinlichkeit des Alleinspielers.

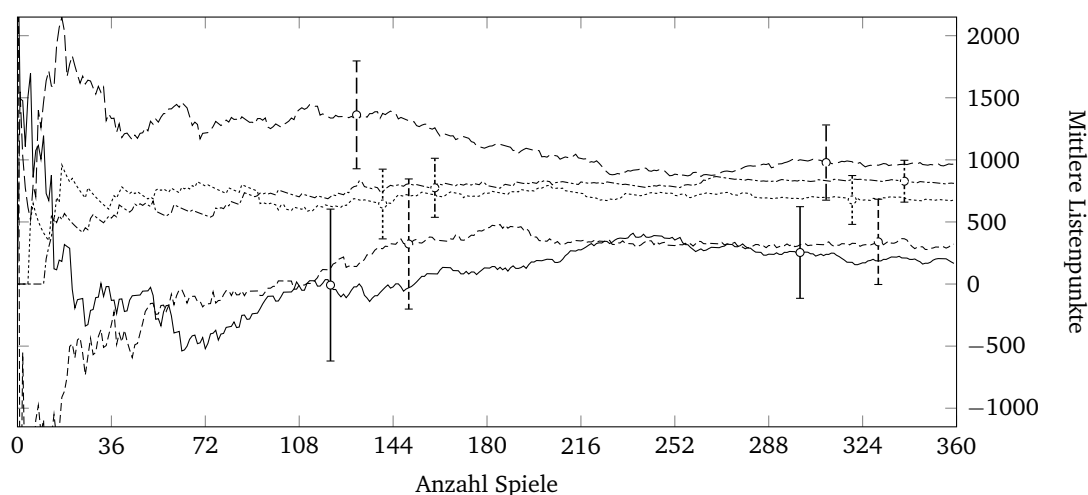


Abbildung 4.1: Exemplarische Entwicklung des Listenpunktemaßes für fünf Spieler zusammen mit ihren 5%-Konfidenzintervallen. Alle Spieler treten gegen das gleiche Paar identischer Computergegner an, um Vergleichbarkeit der Werte zu gewährleisten. Die meisten Datenreihen sind nach 10 Listen (360 Spielen) nur schlecht trennbar.

Abbildung 4.1 zeigt exemplarisch die Entwicklung der mittleren Listenpunkte von fünf zufällig ausgewählten Spielern der ISS-Datenbank zusammen mit ihren 5%-Konfidenzintervallen. Um variierende Gegenspieler als Einflussfaktor auszuschließen, wurden nur solche Spiele ausgewertet, in denen ein menschlicher Spieler gegen zwei Computergegner aus der *Kermit*-Familie antritt (siehe Abschnitt 1.7). Den Konfidenzintervallen nach zu urteilen ist die Rangordnung zwischen benachbarten Spielern nach 10 Listen noch sehr unsicher.

Um den Einfluss der Gegenspieler auf das Listenpunktemaß aufzuzeigen, wurden anschließend alle Spieler ausgewählt, die sowohl gegen zwei Instanzen des starken PIMC-Spielers *Kermit* als auch gegen zwei Instanzen des schwachen heuristischen Spielers *XSkat* (siehe Abschnitt 1.7) in mindestens 300 Spielen antraten. Die mittleren Listen-

punkte dieser Spiele sind in Abbildung 4.2 aufgetragen. Ein ideales, vom Gegenspiel unabhängiges Maß würde für beide Gegner identische Spielstärken liefern. Da Kermit jedoch ein deutlich stärkeres Gegenspiel bietet, werden bei gleicher Spielerfähigkeit weniger Punkte erzielt.

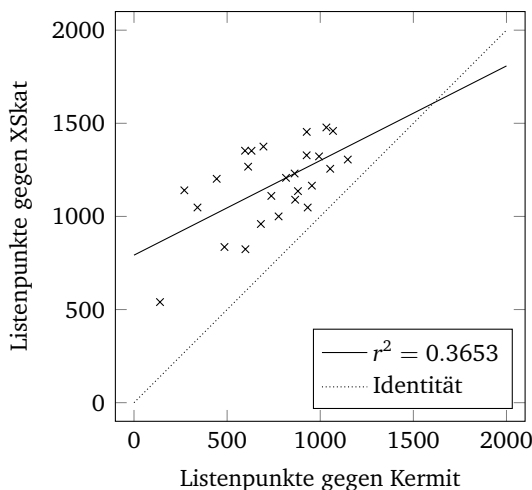


Abbildung 4.2: Vergleichbarkeit des Listenpunktemaßes zwischen starkem (Kermit) und schwachem Gegenspiel (XSkat). Jeder Datenpunkt repräsentiert einen Spieler, der gegen beide Gegner in mindestens 300 Spielen angetreten ist. Die Werte weichen stark von der gewünschten Identität ab.

Treten jedoch mehrere Spieler gegen die gleichen Kontrahenten an, so wird der Einflussfaktor des Gegenspiels zwischen ihnen eliminiert. Wurde außerdem eine hinreichend große Anzahl von Spielen beobachtet, so wird auch der Zufallsaspekt der Kartenverteilungen vernachlässigbar. Unter diesen Voraussetzungen sind die mittleren Listenpunkte das ideale Spielstärkemaß, da sie genau die Größe repräsentieren, die ein guter Skatspieler für sein Listenergebnis maximiert.

4.2 Perfect Information Post Mortem Analysis

Long und Buro (2011) stellen ein alternatives Spielstärkemaß namens **Perfect Information Post Mortem Analysis** (kurz PIPMA) vor. Dieses Verfahren vollzieht die Ausspielphase der beobachteten Spiele nach deren Vollendung als offene Situation nach, in der alle Karten aufgedeckt werden (siehe Abschnitt 1.4). Würde ein beobachteter Zug das offene Spiel verlieren, so wird er als Spielfehler markiert. Eine hohe Spielstärke manifestiert sich dann durch eine geringe Fehlerrate.

Definition 4.4. Sei s eine Spielsituation in der Ausspielphase und p der Ausspieler in s .

- a) Stehen p sowohl gewinnende als auch verlierende Züge zur Auswahl, so wird die Situation s als **Perfect-Information-Gelegenheit** für p bezeichnet.
- b) Ist s eine Perfect-Information-Gelegenheit und p wählt einen verlierenden Zug, so heißt die Situation s **Perfect-Information-Fehler** für p .

Die Unterscheidung zwischen gewinnenden und verlierenden Zügen ist dabei wieder im Kontext eines offenen Spiels zu verstehen: Bei Kenntnis aller Karten steht die optimale Spielweise und das Spielergebnis für jeden Zug von Anfang an fest. Damit ist ein gewinnender Zug ein solcher, der unter Annahme perfekter Folgezüge das Spiel gewinnt, und ein verlierender einer, der das Spiel unter diesen Voraussetzungen verliert.

Definition 4.5. Das Spielstärkemaß der **Perfect Information Mistake Rate (PIMR)**, auch **Perfect-Information-Fehlerrate**, ist der Quotient aus Anzahl der Perfect-Information-Fehler eines Spielers und Anzahl seiner Perfect-Information-Gelegenheiten.

Das Verfahren verdankt seinen Namen der Tatsache, dass die Kartenverteilung zu Spielbeginn nach Ausspiel der letzten Karte allen Parteien bekannt ist. Dadurch kann das Verhalten eines Spielers nach dem Ende der Partie, also *Post Mortem*, aus seiner Perspektive begutachtet werden.

Eine Motivation für dieses Maß ist die isolierte Betrachtung einzelner Züge, die potentiell spezifischere Informationen über einen Spieler liefern als das Listenpunktemaß. Auch liefert die Unterteilung eines Spiels in mehrere Situationen möglicherweise eine höhere Datenrate. Auf diesen Punkt wird in Abschnitt 4.4 näher eingegangen.

Eine Beispielimplementierung von PIPMA ist in Algorithmus 4.1 gegeben.

Abbildung 4.3 zeigt die Entwicklung der Perfect-Information-Fehlerrate über zehn Listen für die selben fünf Spieler wie Abbildung 4.1. Die Qualität des Ergebnisses ist mit der des Listenpunktemaßes vergleichbar, was andeutet, dass PIPMA nicht wesentlich schneller Informationen über einen Spieler sammeln kann als das Listenpunktemaß.

Abbildung 4.4 zeigt die Korrelation zwischen Perfect Information Mistake Rate und den mittleren Listenpunkten eines Spielers. Dazu wurden aus der ISS-Datenbank alle Spieler ausgewählt, die mindestens 2000 Spiele gegen zwei Computergegner aus der *Kermit*-Familie absolviert haben (siehe Abschnitt 1.7). Da damit das Gegenspielverhalten über alle Partien vergleichbar ist, sollte ein gutes Spielstärkemaß die Ordnung nach Listenpunkten reproduzieren (siehe Abschnitt 4.1).

```

def pi_mistake(turns_won, actual_turn):
    has_winning_turns = any(won for t, won in turns_won)
    has_losing_turns = any(not won for t, won in turns_won)
    opportunity = has_winning_turns and has_losing_turns
    mistake = opportunity and not turns_won[actual_turn]
    return mistake, opportunity

def pipma(history, player):
    mistakes = 0
    opportunities = 0
    for situation, turn in history:
        if situation.current == player:
            turns_won = pi.evaluate(situation)
            mistake, opportunity = pi_mistake(turns_won, turn)
            mistakes += int(mistake)
            opportunities += int(opportunity)
    pimr = mistakes / opportunities
    return pimr

```

Algorithmus 4.1: Perfect Information Post Mortem Analysis (PIPMA). Bestimmt das Spielstärkemaß der Perfect Information Mistake Rate (PIMR) aus einer Spielhistorie.

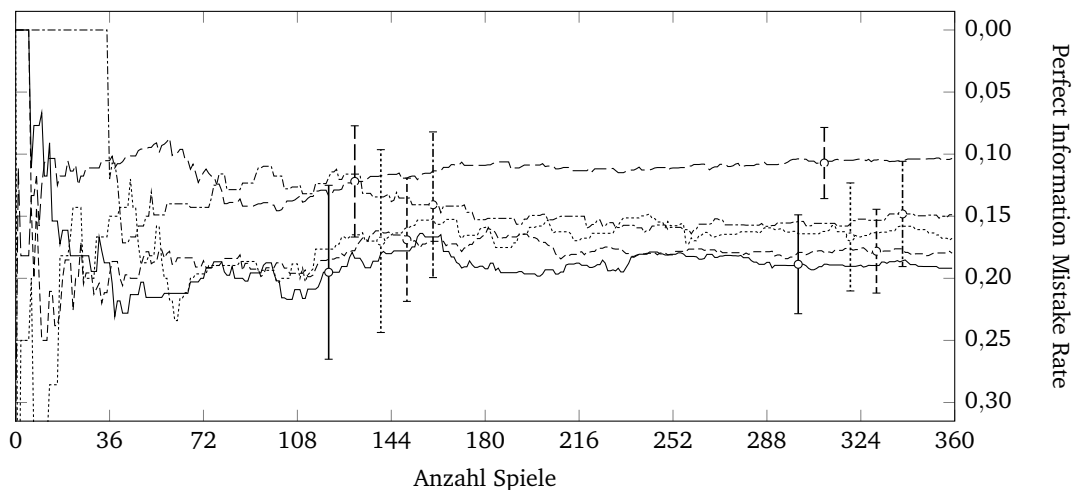


Abbildung 4.3: Entwicklung der Perfect-Information-Fehlerrate der fünf Spieler aus Abbildung 4.1 zusammen mit ihren 5%-Konfidenzintervallen. Die Konvergenzgeschwindigkeit ähnelt der aus Abbildung 4.1.

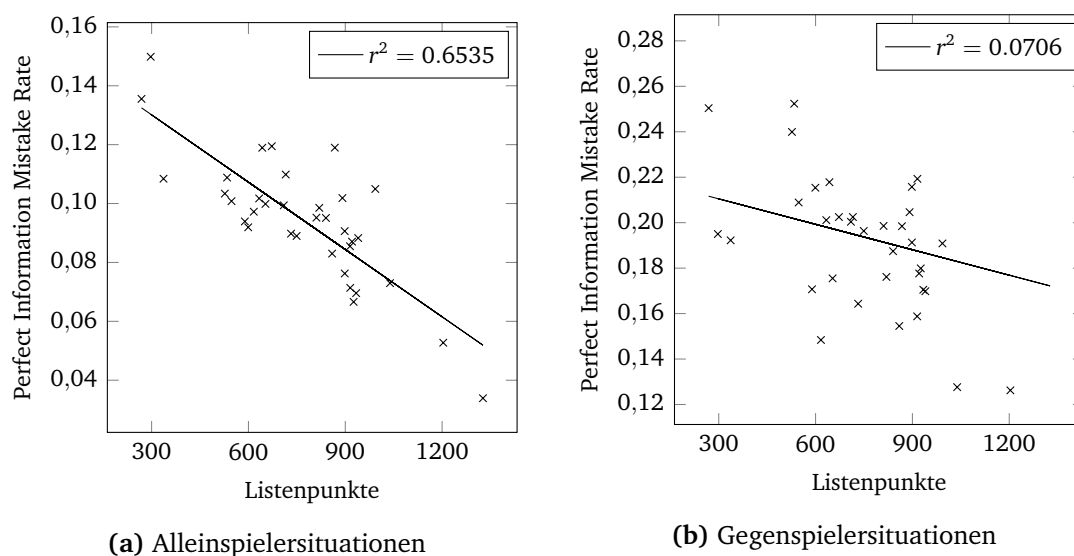


Abbildung 4.4: Korrelation zwischen Listenpunkten und Perfect Information Mistake Rate. Betrachtet wurden 35 Spieler, die gegen eine feste Auswahl vergleichbarer Gegner antreten. In Alleinspiersituationen (a) korrelieren die Spielstärkemaße deutlich besser als im Gegenspiel (b).

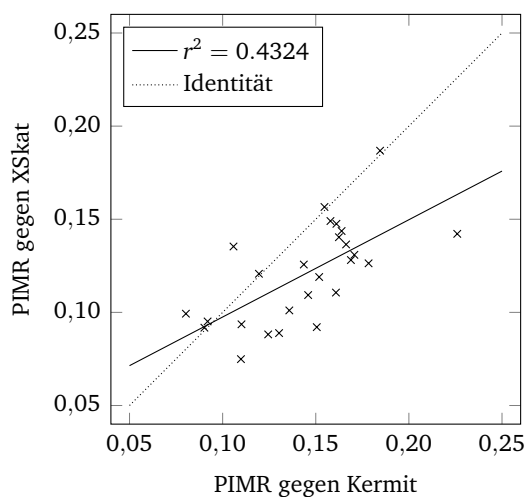


Abbildung 4.5: Vergleichbarkeit der Perfect-Information-Fehlerrate zwischen starkem (Kermit) und schwachem Gegenspiel (XSkat). Es sind dieselben Spieler abgebildet wie in Abbildung 4.2; die Werte sind einander deutlich ähnlicher, reproduzieren die gewünschte Identität jedoch noch nicht.

In Alleinspielersituationen korrelieren die Maße gut, was dafür spricht, dass die Fehlerrate im Alleinspiel ein guter Indikator für die erwarteten Listenergebnisse ist. In Gegenspielersituationen lässt sich dagegen nur mit Mühe ein Zusammenhang erkennen. Da ein erfolgreiches Gegenspiel in verdeckten Spielen aber stärker auf Informationsgewinnung und -austausch basiert und die Gegenpartei selten mit einer festen Strategie startet, erscheint dieses Ergebnis plausibel (vgl. Abschnitt 2.6).

Eine theoretisch und praktisch relevante Schwäche von PIPMA (und grundsätzlich aller derartiger Post-Mortem-Verfahren) entsteht aus dem Vergleichen beobachteter Züge mit einem Referenzspieler, der nicht Nash-optimal spielt (siehe Abschnitt 1.3). Auch wenn der PIPMA-Referenzspieler durch Kenntnis aller Karten jeden realen Spieler mit hoher Wahrscheinlichkeit schlagen könnte, ist er weit von der Spielweise eines optimalen Skatspielers entfernt.

Long und Buro (2011) erkennen, dass ein Nash-Spieler für normale Skatsituationen mit unvollständiger Information eine Perfect-Information-Fehlerrate größer 0 aufweisen muss: Ohne Kenntnis der realen Kartenverteilung ist er zu einer konservativen Strategie gezwungen, die in unvorhergesehenen Situationen zu Perfect-Information-Fehlern führen kann.

Widerspricht die Nash-optimale Spielweise der Strategie im äquivalenten offenen Spiel, so können zwei unterschiedlich gute Züge von PIPMA damit möglicherweise qualitativ nicht unterschieden werden. Dies ist auch eine plausible Begründung für die schlechte Korrelation der PI-Fehlerrate in Gegenspielersituationen mit den objektiv aussagekräftigeren Listenpunkten in Abbildung 4.4b.

Die Perfect-Information-Fehlerrate analysiert ausschließlich Spielzüge in der Ausspielphase und ignoriert die Spielfindung und die Rate der tatsächlich gewonnenen Spiele. Da letztere stark vom Reizverhalten und der Risikobereitschaft der Gegner abhängen, stellt sich die Frage, ob das Perfect-Information-Maß weniger Abhängigkeiten vom Verhalten der Gegenspieler zeigt als das Maß der mittleren Listenpunkte.

Abbildung 4.5 zeigt das Verhalten der Perfect Information Mistake Rate bei stark wechselnden Fähigkeiten der Gegenspieler. Es kommen die gleichen Testspiele zum Einsatz wie in Abbildung 4.2. Die Korrelation zwischen den Situationen verbessert sich durch PIMR gegenüber dem Listenpunktemaß stark, verfehlt die gewünschte Gleichheit aber noch deutlich.

4.3 Monte Carlo Post Mortem Analysis

Die Unsicherheit über die Kartenverteilung ist ein zentraler Aspekt des Skatspiels und Teil der Strategieplanung jedes Spielers. Ein gutes zugbasiertes Spielstärkemaß sollte also die Informationslage zum Zeitpunkt einer Spielentscheidung berücksichtigen.

Bei Perfect Information Post Mortem Analysis (PIPMA, siehe Abschnitt 4.2) wird diesem Umstand keine Beachtung geschenkt: Dem Referenzspieler wird die Kartenverteilung vollständig bekannt gemacht, sodass jede Fehlentscheidung aus seiner Perspektive gleich schwer wiegt. So kann die Wahl zwischen zwei Zügen, die selbst aus Sicht eines optimalen Spielers mit gleicher Wahrscheinlichkeit gewinnen, als Fehler wahrgenommen werden.

Um Züge aus der Sicht eines realen Spielers einschätzen zu können, der sich mit unvollständiger Information zufrieden geben muss, wird hier ein neues Verfahren auf Basis der Perfect-Information-Monte-Carlo-Suche (siehe Kapitel 2) entwickelt. Ähnlich wie PIPMA vergleicht es ausgespielte Züge nach Vollendung eines Spiels gegen einen Referenzspieler und wird damit **Monte Carlo Post Mortem Analysis** (MCPMA) genannt.

Ein Perfect-Information-Monte-Carlo-Spieler (kurz PIMC-Spieler) schätzt aus stichprobenhafter Auswertung plausibler Kartenverteilungen die Gewinnwahrscheinlichkeit jedes Zuges. Zeigen zwei Züge stark unterschiedliche Gewinnchancen, so ist aus Sicht des PIMC-Spielers einer der beiden klar vorzuziehen. Sind die Wahrscheinlichkeiten dagegen nahezu identisch, so lassen sich die Auswirkungen der Züge anhand der momentanen Informationslage nicht klar unterscheiden.

Mit dieser Beobachtung können die Konzepte der *Gelegenheit* und des *Fehlers*, die mit PIPMA in Definition 4.4 eingeführt wurden, im Kontext von MCPMA formuliert werden.

Definition 4.6. Sei s eine Spielsituation, M die Menge möglicher Züge in s , $h(s, m')$, $m' \in M$, die relative Häufigkeit des Gewinns in einer PIMC-Stichprobe und $m \in M$ der gewählte Zug des beobachteten Spielers.

Dann bezeichnet die **Monte-Carlo-Gelegenheit** $U_{\max}(s)$ die Größe

$$U_{\max}(s) = \max_{m' \in M} h(s, m') - \min_{m' \in M} h(s, m') \quad (4.3)$$

und der **Monte-Carlo-Fehler** $U(s, m)$ die Größe

$$U(s, m) = \max_{m' \in M} h(s, m') - h(s, m). \quad (4.4)$$

Der Monte-Carlo-Fehler ist anschaulich die eingebüßte Gewinnwahrscheinlichkeit durch die suboptimale Wahl des Zugs m . Die Monte-Carlo-Gelegenheit ist der größtmögliche Monte-Carlo-Fehler in einer Situation. Damit bilden Monte-Carlo-Gelegenheit und -fehler ein reellwertiges Gegenstück zu ihren Perfect-Information-Äquivalenten.

Definition 4.7. Sei S eine Menge von Spielsituationen und m_s der Zug des beobachteten Spielers in der Situation $s \in S$. Das Spielstärkemaß der **Monte Carlo Mistake Rate (MCMR)** ist definiert als

$$MCMR = \frac{\sum_{s \in S} U(s, m_s)}{\sum_{s \in S} U_{\max}(s)} \quad (4.5)$$

Damit ist analog zu Definition 4.5 die Monte-Carlo-Fehlerrate als Quotient der Summen von Fehler und Gelegenheiten definiert. Da schwer unterscheidbare Züge jedoch betragsmäßig kleine Gelegenheiten produzieren, leisten Fehlentscheidungen zwischen ihnen nur einen kleinen Beitrag zur Fehlerrate.

Eine Beispielimplementierung von MCPMA ist in Algorithmus 4.2 gegeben.

Abbildung 4.6 zeigt das Verhalten der Monte-Carlo-Fehlerrate für die fünf Spieler aus den Abbildungen 4.1 und 4.3. An den deutlich kleineren Konfidenzintervallen des Fehlermaßes ist zu sehen, dass die Fehlerrate die fünf Spieler nach 10 beobachteten Listen bis auf ein Spielerpaar mit hoher Sicherheit trennen kann. Das Konvergenzverhalten der verschiedenen Verfahren wird in Abschnitt 4.4 genauer analysiert.

Für Abbildung 4.7 wurde die Monte-Carlo-Fehlerrate anhand der selben Spiele mit dem Listenpunktemaß verglichen wie die Perfect-Information-Rate in Abbildung 4.4. Es wird sichtbar, dass die Monte-Carlo-Rate messbar stärker mit den Listenpunkten korreliert als ihr Perfect-Information-Äquivalent. Das suggeriert, dass MCPMA Informationen aus dem Spielverlauf extrahieren kann, die PIPMA verborgen bleiben, in den Listenpunkten aber sichtbar werden.

Auch wenn sich der Zusammenhang in Gegenspielersituationen deutlich besser erkennen lässt als bei PIPMA, ist die Fehlerrate in diesen Situationen trotzdem nur ein schwacher Indikator für das Turniermaß der Listenpunkte. Das passt zur Beobachtung, dass die einfache Optimierung der Monte-Carlo-Gewinnwahrscheinlichkeit aus Kapitel 2 keine optimale Gegenspielstrategie darstellt. Diese Schwäche wurde in Abschnitt 2.6 bereits diskutiert und wird in Abschnitt 4.5 genauer untersucht.

```

def mc_mistake(turn_samples, actual_turn):
    win_probabilities = {}
    for turn, samples in turn_samples.items():
        win_probabilities[turn] = sum(1 for s in samples if s.won) / len(samples)
    lowest_win_prob = min(win_probabilities.values())
    highest_win_prob = max(win_probabilities.values())
    actual_win_prob = win_probs[actual_turn]
    severity = highest_win_prob - actual_win_prob
    weight = highest_win_prob - lowest_win_prob
    return severity, weight

def mcpma(history, player):
    accumulator = 0
    total_weight = 0
    for situation, turn in history:
        if situation.current == player:
            turn_samples = pimc.evaluate(situation)
            severity, weight = mc_mistake(turn_samples, turn)
            accumulator += severity
            total_weight += weight
    mcmr = accumulator / total_weight
    return mcmr

```

Algorithmus 4.2: Monte Carlo Post Mortem Analysis (MCPMA). Bestimmt das Spielstärkemaß der Monte Carlo Mistake Rate (MCMR).

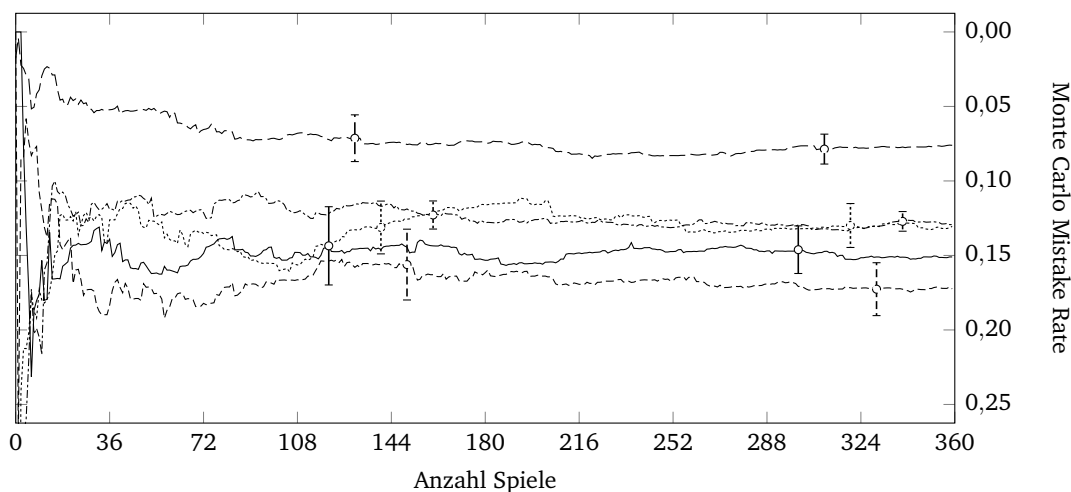


Abbildung 4.6: Entwicklung der Monte-Carlo-Fehlerrate für die fünf Spieler aus Abbildung 4.1 zusammen mit ihren 5%-Konfidenzintervallen. Bis auf ein Paar lassen sich alle Spieler nach 10 Listen zuverlässig trennen.

Wie PIPMA weist auch MCPMA theoretische und praktische Schwächen durch die Suboptimalität des verwendeten Referenzspielers auf. Der PIMC-Spieler, der bei MCPMA zum Einsatz kommt, arbeitet immerhin regelkonform auf unvollständiger Information arbeitet und zeigt gemäß Abbildung 4.7 einen stärkeren Zusammenhang zum Turniermaß der Listenpunkte. Damit scheinen die praktischen Nachteile geringer ausgeprägt zu sein als bei PIPMA.

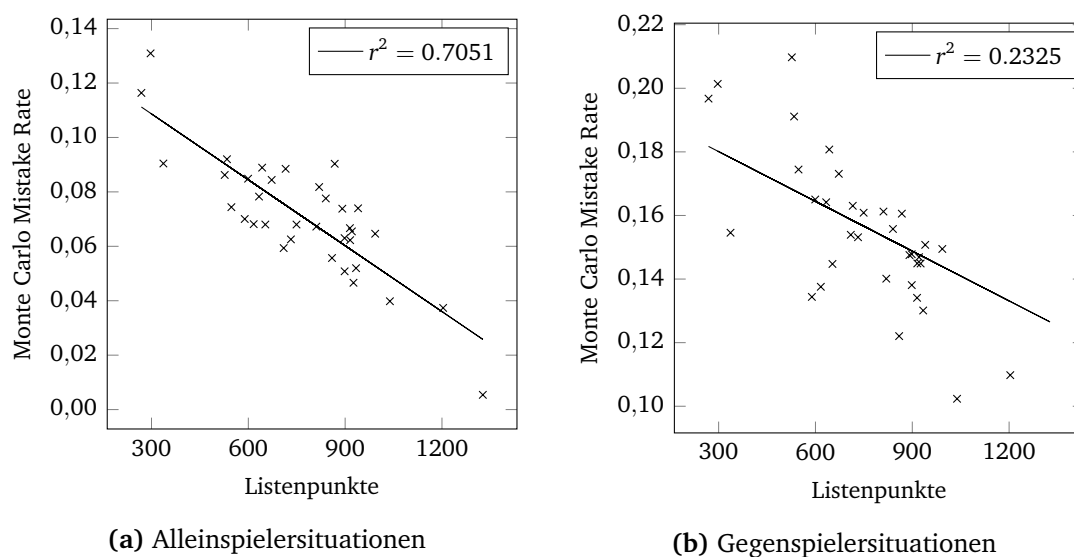


Abbildung 4.7: Korrelation zwischen Listenpunkten und Monte Carlo Mistake Rate. Es sind die selben Spieler abgebildet wie in Abbildung 4.4, die MCMR-Werte korrelieren jedoch deutlich besser mit den Listenpunkten.

Dafür entsteht durch die Regelkonformität eine neue potentielle Schwachstelle: Ein Skatexperte kann dem Referenzspieler durchaus in allen Spielsituationen überlegen sein. MCPMA wird trotzdem alle Abweichungen von der PIMC-Spielweise als positiven Monte-Carlo-Fehler aufzeichnen, was zum paradoxen Ergebnis führt, dass der Experte umso schlechter eingestuft wird, je weiter er dem PIMC-Spieler überlegen ist. Da aber in Abbildung 4.7 bei hohen mittleren Listenpunkten die Monte-Carlo-Fehlerrate nicht unerwartet zunimmt, manifestiert sich dieses Problem zumindest in den hier verwendeten Testdaten nicht.

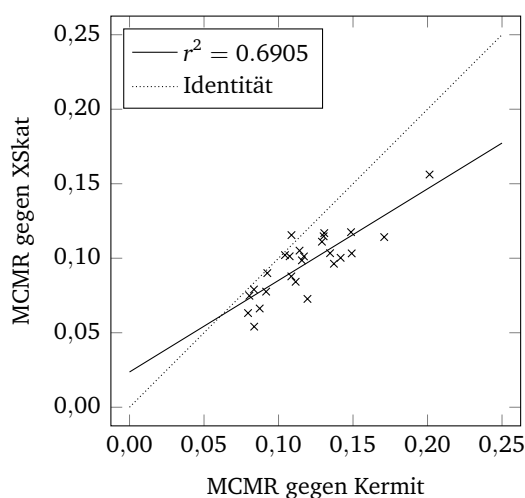


Abbildung 4.8: Vergleichbarkeit der Monte Carlo Mistake Rate zwischen starkem (Kermit) und schwachem Gegenspiel (XSkat). Es sind die selben Spieler abgebildet wie in den Abbildungen 4.2 und 4.5; von allen besprochenen Verfahren kommen die Werte der MCMR der gewünschten Identität am nächsten.

Wie Abbildung 4.8 zeigt, korreliert die Monte-Carlo-Fehlerrate zwischen den unterschiedlichen Computergegnern deutlich besser, als das bei Listenpunkten und Perfect-Information-Fehlerrate der Fall ist (siehe Abbildungen 4.2 und 4.5). Außerdem sind sich die Werte auf beiden Achsen ähnlicher, sodass die MCMR ohne Kenntnis der Mitspieler ein verlässlicheres Spielstärkemaß darstellt als die mittleren Listenpunkte.

Es bleibt zu vermuten, dass auch ein ideales stichbasiertes Spielstärkemaß, das Züge gegen einen Nash-Spieler vergleicht (siehe Abschnitt 4.2), nicht invariant gegenüber dem Gegenspiel ist. Da einerseits ein fähiger Spieler durch geschickte Züge seine Gegner zu Spielfehlern verleiten kann und andererseits ein primitiver heuristischer Spieler durch sein leicht vorhersehbares Zugverhalten bessere Möglichkeiten zur Zustandsinferenz (siehe Kapitel 3) bietet, kann die reale Spielstärke tatsächlich mit dem Gegner variieren. Es ist nicht klar, wie nahe MCPMA hier dem Optimum kommt.

4.4 Vergleich der Verfahren

Um ein solides Fundament für den Vergleich der vorgestellten Spielstärkemaße zu schaffen, wird nun eine Möglichkeit entwickelt, ihr Konvergenzverhalten zu beziffern.

Die konkreten Zahlenwerte eines Spielstärkemaßes sind für die Qualität des Verfahrens nebensächlich; viel wichtiger ist, mit wie vielen Datenpunkten ein Maß zwei Spieler sicher trennen kann. Um diesen Aspekt quantifizieren zu können, wird der Begriff des *Rangordnungsfehlers* eingeführt.

Definition 4.8. Sei P eine Menge von Spielern und $B : P \rightarrow \mathbb{R}$ ein Spielstärkemaß. Dann erzeugt B für jede Teilmenge $Q \subset P$ die **Rangordnung** $R_B : Q \times Q \rightarrow \{0,1\}$,

$$R_B(p_1, p_2) = \begin{cases} 1, & B(p_1) < B(p_2) \\ 0, & \text{sonst} \end{cases}$$

Interessant ist nun, nach wie vielen Spielen sich die paarweise Rangordnung einer Spielermenge nicht mehr ändert.

Definition 4.9. Sei Q eine Menge von Spielern und R_B, R_C Rangordnungen. Der **Rangordnungsfehler** $E_R \in [0,1]$ zwischen R_B und R_C ist definiert als

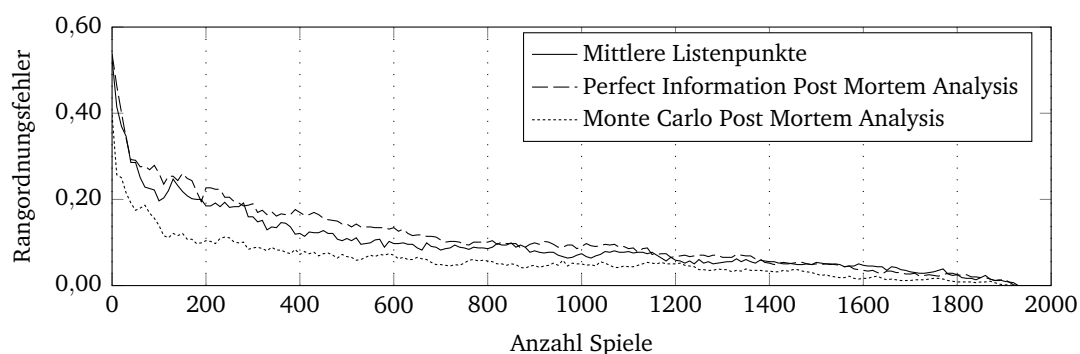
$$E_R(R_B, R_C) = \frac{1}{2(\#Q)(\#Q - 1)} \sum_{p_1, p_2 \in Q} |R_B(p_1, p_2) - R_C(p_1, p_2)|$$

Der Rangordnungsfehler bildet eine Metrik: Für zwei R_B und R_C ist $E_R(R_B, R_C) = 0$ genau dann, wenn die Rangordnungen identisch sind, und 1, wenn die R_C die Rangfolge R_B genau umkehrt. Ist mindestens eine der Rangordnungen rein zufällig, liegt der Erwartungswert von E_R nahe 0,5.

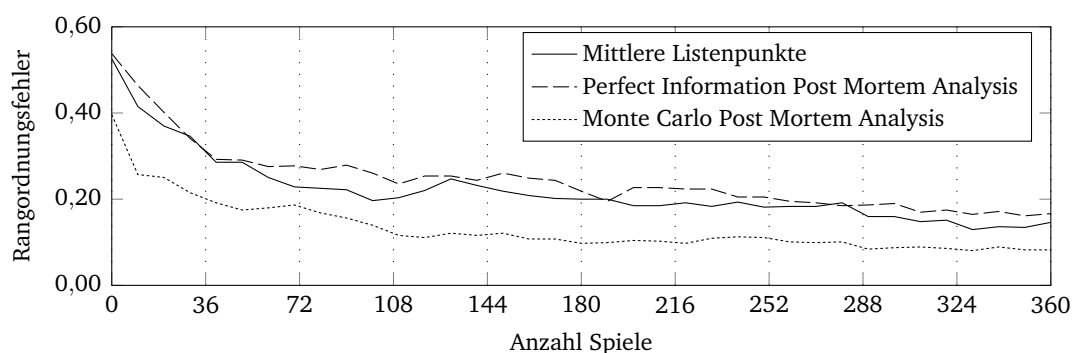
Die Konvergenzgeschwindigkeit der Spielstärkemaße wird nun anhand des Rangordnungsfehlers bestimmt. Dazu wird erneut die ISS-Datenbank aus Abschnitt 1.7 herangezogen. Betrachtet werden alle 35 Spieler, die je mindestens 2000 Spiele gegen ein Gegnerpaar aus der *Kermit*-Familie absolviert haben. Da aufgrund der Endlichkeit der Testdaten kein echter Grenzwert für die Rangordnung bestimmt werden kann, wird ersatzweise die Konvergenz gegen die Ordnung nach 2000 Spielen gemessen.

Wie in Abbildung 4.9 zu sehen ist, beginnt Monte Carlo Post Mortem Analysis mit einem deutlichen Vorsprung und liegt für die gesamte Datenreihe weit unter dem Rangordnungsfehler der anderen Verfahren.

Das überlegene Konvergenzverhalten kann durch die verbesserte Informationslage erklärt werden: Die Auswertung von circa 20.000 zufällig gewählten Spielen der ISS-Datenbank ergibt, dass MCPMA mehr als die dreifache Menge an Daten aus dem gleichen Spiel extrahieren kann als die beiden anderen Verfahren, wie Tabelle 4.1 zeigt. Für PIPMA wurde dabei jede Perfect-Information-Gelegenheit (siehe Definition 4.4) gezählt, für MCPMA jede Monte-Carlo-Gelegenheit $U_{\max} > 0$ (siehe Definition 4.6).



(a) Entwicklung bis zum Referenzwert bei 2000 Spielen



(b) Entwicklung in den ersten 10 Listen (360 Spielen)

Abbildung 4.9: Entwicklung des Rangordnungsfehlers gegen die Ordnung nach 2000 Spielen. Betrachtet wurden 35 Spieler, die gegen eine feste Auswahl vergleichbarer Gegner antreten. Monte Carlo Post Mortem Analysis konvergiert schneller als die anderen Verfahren.

Spielstärkemaß	Datenpunkte	
	pro Situation	pro Spieler und Spiel
Mittlere Listenpunkte		1
Perfect Information Post Mortem Analysis	0,158	1,58
Monte Carlo Post Mortem Analysis	0,512	5,12

Tabelle 4.1: Anzahl der nutzbaren Datenpunkte pro Spielstärkemaß. MCPMA erreicht eine deutlich höhere Datenrate als die anderen Verfahren, da mehr Situationen als potentielle Spielfehler ausgewertet werden können.

4.5 Vergleich mit einem professionellen Spieler

Um die Qualität der Fehlermaße einschätzen und ihre Grenzen aufzeigen zu können, werden sie zuletzt noch an einem Experten im Skatspiel evaluiert.

Dazu wurden von Skatprofi Daniel Schäfer Spielhistorien aus 16 Serien (576 individuelle Spiele) zur Verfügung gestellt (siehe Abschnitt 1.7). In diesen Spielen tritt er gegen einen PIMC-Spieler mit etwas schwächerem Profil an als *Kermit*, der in Abschnitt 4.1 als Gegenspieler benutzt wurde. Die dort vorgestellten Zahlen sind deshalb nicht uneingeschränkt auf diesen Abschnitt übertragbar, können aber einen groben Eindruck von den betrachteten Spielstärken vermitteln.

Spielstärkemaß	Daniel Schäfer			PIMC-Spieler		
	Allein	Gegen	Gesamt	Allein	Gegen	Gesamt
Listenpunkte			1191			768
PIMR	0,0475	0,2146	0,1070	0,0608	0,1686	0,1090
MCMR	0,0309	0,1145	0,0594_w	0,0322	0,0916	0,0595

Tabelle 4.2: Spielstärke des Skatexperten Daniel Schäfer im Spiel mit einem PIMC-Gegner. Die Wertung des besseren Spielers ist jeweils fett gedruckt. Die MCPMA-Wertung eines PIMC-Spielers ist nicht aussagekräftig.

Tabelle 4.2 zeigt die erreichten Punkte und Fehlerraten der Kontrahenten. An den Listenpunkten ist eindeutig zu sehen, dass Daniel Schäfer dem Computerspieler weit überlegen ist. Dabei ist wichtig, dass diese Überlegenheit nicht nur aus höherer Sicherheit während des Spiels, sondern aus mehr Geschicklichkeit beim Reizen und der Spielwahl stammt — ein Aspekt, der in dieser Arbeit ausgeklammert wurde.

Der Perfect-Information-Fehlerrate nach zu urteilen befindet sich das Ausspielverhalten beider Parteien auf vergleichbarem Niveau. Diese Schlussfolgerung ist aber mit Vorsicht zu genießen, da ein großer Teil der Spielstärke des Computers aus seiner vermeintlichen Überlegenheit im Gegenspiel stammt. Nach Einschätzung Experten (siehe Abschnitt 2.6) ist das Kräfteverhältnis in diesen Situationen genau umgekehrt.

Das passt zu der Beobachtung, dass die Perfect-Information-Fehlerrate im Gegenspiel wichtige Aspekte einer starken Spielstrategie ignoriert und nur schlecht mit dem Erfolg in der Liste korreliert (vgl. Abschnitt 4.2 und Abbildung 4.4). PIPMA überschätzt die

Stärke des Computers damit also deutlich — das Maß konzentriert sich auf unwichtige Details der Spielstrategie, denen auch der PIMC-Spieler folgt.

Die Monte-Carlo-Fehlerrate liefert ein etwas plausibleres Bild des Experten: Zusammen mit den Werten aus Abbildung 4.7 befindet er sich unter den Spielern mit der geringsten gemessenen Fehlerrate. Dass die Monte-Carlo-Fehlerrate auch deutlich besser mit dem Listenpunktemaß korreliert als die Perfect-Information-Rate in Abbildung 4.4, spricht zumindest dafür, dass menschliche Spieler mit MCPMA verlässlicher untereinander geordnet werden können als mit PIPMA. Der Computerspieler kann mit dieser Methode dagegen nicht sinnvoll bewertet werden, da sein Verhalten als PIMC-Spieler natürlicherweise dem MCPMA-Referenzspieler sehr ähnlich ist und er dadurch unrealistisch niedrige Fehlerraten zugewiesen bekommt.

Es lässt sich also festhalten, dass sich die Fehlermaße PIPMA und MCPMA gut zur Bewertung menschlicher Alleinspieler eignen. Im Gegenspiel sind sie unschärfer, wobei MCPMA noch deutlich verlässlichere Werte liefert als sein Vorgänger aus der Literatur.

4.6 Zusammenfassung

In diesem Kapitel wurde ein neues, präzises und schnelles Maß für die Spielstärke eines Skatspielers entwickelt — eine wichtige Voraussetzung für einen selbstkalibrierenden Computerspieler, der sich den Fähigkeiten seiner Gegner anzupassen vermag.

Mit den mittleren Listenpunkten und Perfect Information Post Mortem Analysis wurden zwei existierende Methoden vorgestellt und ihre Schwächen diskutiert. Mit Monte Carlo Post Mortem Analysis (MCPMA) wurde dann ein neues Spielstärkemaß auf Basis des Skatspielers aus Kapitel 3 entwickelt. Schließlich wurden die Verfahren anhand von Testdaten und dem Verhalten eines professionellen Skatspielers verglichen, wobei die Überlegenheit von MCPMA demonstriert werden konnte.

In Kapitel 5 kann das entwickelte Maß nun als Datenbasis für die Entwicklung einer adaptiven Spielweise herangezogen werden.

5 Adaptive Spielweise

In Kapitel 4 wurde gezeigt, wie die Fähigkeiten eines menschlichen Mitspielers anhand eines Spielstärkemaßes schnell und zuverlässig eingeschätzt werden können. Wie in Kapitel 1 skizziert, soll der Computergegner dem Menschen nun mit einer ähnlichen Leistung entgentreten, um ihm eine angemessene Herausforderung zu bieten.

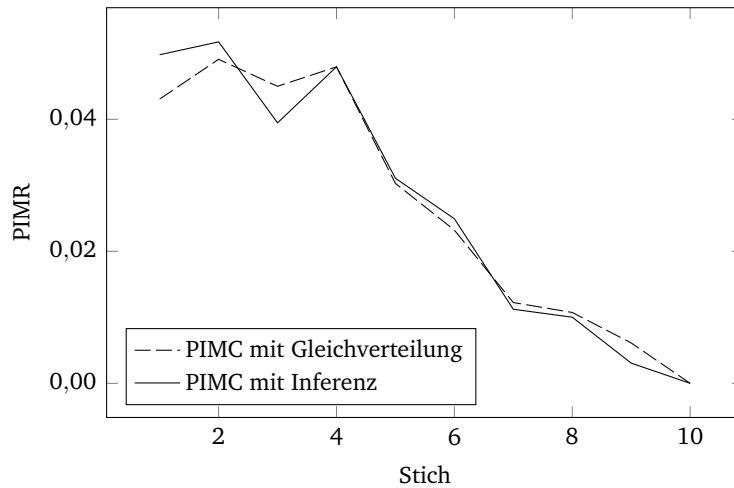
Liegt die Spielstärke eines Spielers in Form seiner Perfect-Information- (PIMR) oder Monte-Carlo-Fehlerrate (MCMR) vor, so ist das Ziel eines adaptiven Computerspielers nun, diese Fehlerrate zu imitieren. Die maximale Spielstärke des Computers ist dabei durch seine eigenen Fähigkeiten beschränkt; eine Reduktion kann durch das Einführen künstlicher Spielfehler realisiert werden.

Als Nebenbedingung ist wünschenswert, dass ein Absenken der Spielstärke gleichmäßig erfolgt, anstatt die vorgegebene Fehlerrate durch wenige seltene Ausreißer zu erreichen. Aus Spielersicht macht sich diese Eigenschaft dadurch bemerkbar, dass der Computer in schwierigen Situationen häufig eine leicht suboptimale Wahl trifft, anstatt zwischen perfektem Spiel und groben Schnitzern abzuwechseln.

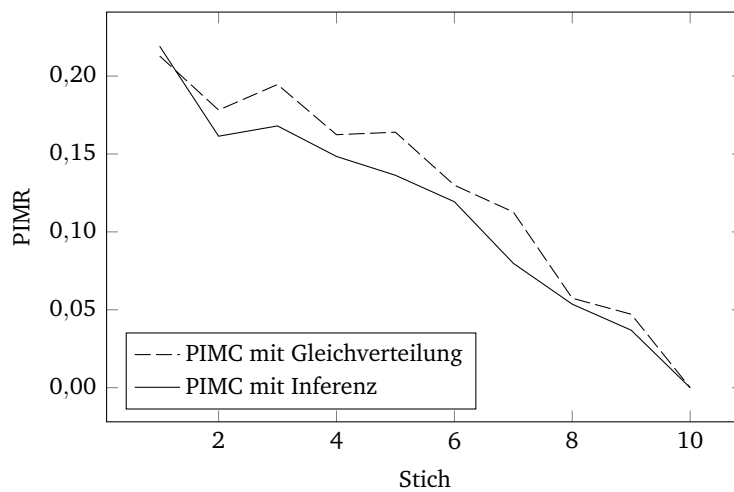
5.1 Maximale Spielstärke des Computers

Mit den Werkzeugen aus Kapitel 4 kann erstmals die Stärke des Perfect-Information-Monte-Carlo-Spielers (kurz PIMC-Spieler) aus Kapitel 2 und 3 bewertet werden.

Monte Carlo Post Mortem Analysis (siehe Abschnitt 4.3) ist nicht für die Bewertung des Computers geeignet, da sie intern den PIMC-Spieler selbst als Referenz benutzt und durch die natürliche Ähnlichkeit seiner Spielweise die Fähigkeiten des Computerspielers überschätzen würde. Stattdessen wird auf Perfect Information Post Mortem Analysis zurückgegriffen (siehe Abschnitt 4.2), ein Verfahren, welches das Spiel aus einem etwas anderen Blickwinkel betrachtet.



(a) Fehlerrate in Alleinspielerpositionen; Zustandsinferenz anhand der ausgespielten Karten



(b) Fehlerrate in Gegenspielerpositionen; Zustandsinferenz anhand der Spielansage

Abbildung 5.1: Vergleich der Perfect-Information-Fehlerrate zwischen einem PIMC-Spieler ohne und mit Zustandsinferenz. Es wurden 10.000 Spiele nachvollzogen. In Gegenspielerposition erzielt der inferierende Spieler mit Abstand das beste Ergebnis, in Alleinspielerposition sind die Spieler mit und ohne Inferenz etwa gleichauf.

Abbildung 5.1 vergleicht den PIMC-Spieler aus Kapitel 2 mit und ohne Verwendung der Zustandsinferenz aus Kapitel 3. Dazu wurden 10.000 Spiele der ISS-Datenbank (siehe Abschnitt 1.7) nachgespielt, indem jede Situation mit dem zu bewertenden Computerspieler nachempfunden und dessen Zugwahl gegen den optimalen Zug im offenen Spiel verglichen wurde. Im letzten PIMC-Schritt kam dabei die konservative Zugwahl (siehe Verfahren 2.5) zum Einsatz.

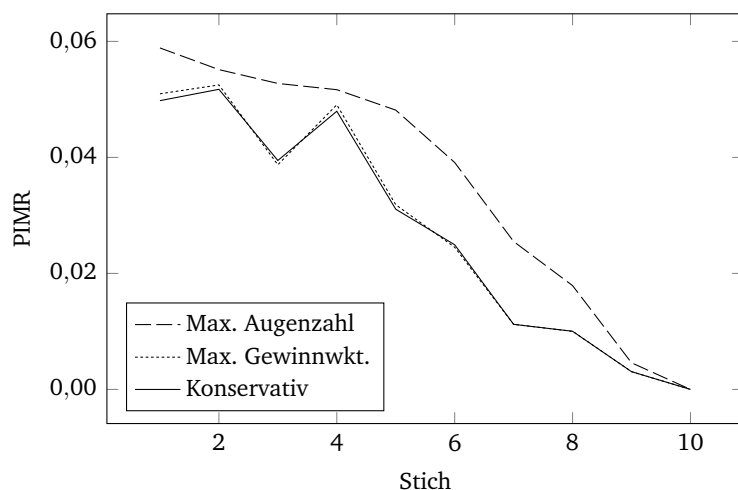
Aufgrund der stark unterschiedlichen Anforderungen von Allein- und Gegenspielerpositionen wurden diese getrennt betrachtet. Allgemein wird sichtbar, dass die beobachteten Fehlerraten im Alleinspiel signifikant unter denen im Gegenspiel liegen. Der Grund dafür liegt vermutlich gleichermaßen in den höheren Anforderungen des Gegenspiels, den Schwächen von PIMC in dieser Position (siehe Abschnitt 2.6) und den Unzulänglichkeiten der Perfect-Information-Fehlerrate (siehe Abschnitt 4.2).

Die Fehlerrate sinkt auf natürliche Weise mit jedem Stich, da über die Wahrheit (siehe Definition 2.6) mit jedem Zug neue Informationen über die Kartenverteilung bekannt werden. Da in Stich 10 keine Wahlmöglichkeit mehr herrscht, ist dort die Fehlerrate jedes Spielers gezwungenermaßen 0.

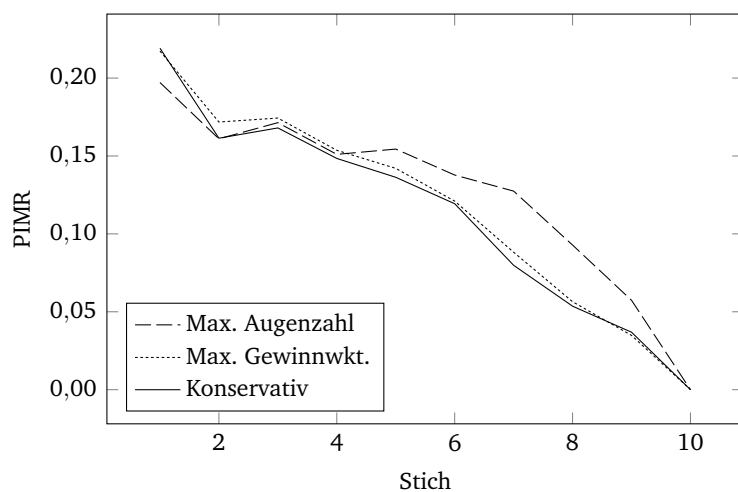
In Alleinspielerposition benutzt der Computerspieler die Stichinferenz aus Abschnitt 3.5 und bestätigt die dort geäußerte Vermutung, dass die Technik keinen messbaren Einfluss auf die Spielstärke hat. In Gegenspielerposition kann der Computer dagegen auf die mächtigere Spielansageinferenz aus Abschnitt 3.4 zurückgreifen, die eine wesentliche Verbesserung bewirkt.

Zugwahlverfahren	Alleinspieler		Gegenpartei		Gesamt	
	Gleichv.	Inferenz	Gleichv.	Inferenz	Gleichv.	Inferenz
Zufälliger Zug	0,3991	0,3991	0,4538	0,4538	0,4243	0,4243
Max. Augenzahl	0,0443	0,0446	0,1807	0,1535	0,1073	0,0949
Max. Gewinnwkt.	0,0349	0,0350	0,1670	0,1496	0,0959	0,0879
Konservativ	0,0341	0,0346	0,1601	0,1450	0,0923	0,0856

Tabelle 5.1: Perfect-Information-Fehlerrate der PIMC-Spieler aus Kapitel 2 („Gleichverteilung“) und Kapitel 3 („Inferenz“), ausgewertet an den Situationen von 10.000 Testspielen. Die optimale Konfiguration aus Inferenz und Zugwahlverfahren einer Position ist jeweils fett gedruckt.



(a) Alleinspietersituationen



(b) Gegenspietersituationen

Abbildung 5.2: Vergleich der Perfect-Information-Fehlerrate (PIMR) eines inferierenden PIMC-Spielers mit unterschiedlichen Zugwahlverfahren. Es wurden die Situationen aus 10.000 Spielen nachvollzogen. In nahezu allen Fällen ist die konservative Zugwahl die erfolgreichste Methode, dicht gefolgt vom Maximieren der Gewinnwahrscheinlichkeit.

Abbildung 5.2 vergleicht das Verhalten des Computerspielers mit Zustandsinferenz bei wechselndem Zugwahlverfahren (siehe Abschnitt 2.5). Im Alleinspiel liegt das Maximieren der Gewinnwahrscheinlichkeit (Verfahren 2.3) mit der konservativen Zugwahl (Verfahren 2.5) gleichauf, wogegen das Optimieren der erwarteten Augenzahl (Verfahren 2.4) deutlich schlechter abschneidet. In Gegenspielerposition ist die Trennung nicht so eindeutig; im ersten Zug liefert das Spiel nach Augenzahl das beste Ergebnis, während in den meisten anderen Fällen die konservative Zugwahl dominiert.

Tabelle 5.1 vergleicht alle Kombinationen aus Zustandsinferenz und Zugwahlverfahren. Um den maximal zu erwartenden Perfect-Information-Fehler darzustellen, wurde eine weitere Zugwahlmethode eingeführt, die unabhängig vom Stichprobenergebnis in jeder Situation eine zufällige Karte ausspielt. Es wird sichtbar, dass die konservative Zugwahl in beiden Positionen das beste Ergebnis liefert. Die Zustandsinferenz im Gegenspiel bringt ebenfalls einen deutlichen Gewinn, während im Alleinspiel kein signifikanter Unterschied zum naiven Spieler ohne Inferenz auszumachen ist.

5.2 Verrauschen der Stichprobenergebnisse

Als finale Komponente des adaptiven Computerspielers wird nun eine neue Methode zur kontrollierten Reduktion der Spielstärke des PIMC-Algorithmus entwickelt.

Je ungenauer die Ergebnisse aus den Stichproben (siehe Abschnitt 2.4) vor der Zugwahl (siehe Abschnitt 2.5) sind, desto höher wird intuitiv die mittlere Fehlerrate des Spielers. Dies kann durch Verminderung der Stichprobenanzahl oder Veränderung der Stichprobenergebnisse selbst geschehen. Letzterer Ansatz hat den Vorteil, dass er robuster gegenüber zufälligen Einflüssen ist und besser gesteuert werden kann.

Für alle normalen Farb- und Grandspiele ist der Gewinn oder Verlust des Spiels eine Funktion der erreichten Augenzahl. Um die Daten aus den Stichproben zu verwässern, kann durch Addition eines normalverteilten Fehlers diese Augenzahl in jeder Stichprobe verrauscht werden:

Definition 5.1. Sei (Ω, \mathcal{A}, P) ein Wahrscheinlichkeitsraum, $a \in \mathbb{N}_0$ die erreichte Augenzahl einer Stichprobe im offenen Spiel, $n \in \mathbb{N}$ die gesamte Stichprobenanzahl und $\sigma \geq 0$ die gewünschte Standardabweichung. Die **verrauschte Augenzahl** A ist verteilt zu

$$A : \Omega \rightarrow \mathbb{R}, \quad A \sim N(a, n\sigma^2) \quad (5.1)$$

Der Mittelwert über die verrauschten Augenzahlen A_1, \dots, A_n aller n Stichproben hat damit die vorgegebene Standardabweichung

$$\text{SD} \left(\frac{1}{n} \sum_{k=0}^n A_k \right) = \sqrt{\frac{n\sigma^2}{n}} = \sigma \quad (5.2)$$

Die Wahl von σ steuert dabei die Ungenauigkeit der Stichprobenergebnisse. Für $\sigma = 0$ wird das unmodifizierte Verfahren mit maximaler Spielstärke reproduziert. Der Grenzwert für $\sigma \rightarrow \infty$ entspricht einer zufälligen Zugwahl in jeder Situation, deren Fehlerrate in Tabelle 5.1 gegeben ist. Die verrauschten Daten können mit den bekannten, unveränderten Zugwahlverfahren aus Abschnitt 2.5 konsolidiert werden, sodass lediglich ein neuer Zwischenschritt nach der Stichprobenauswertung eingeführt werden muss.

Da für Nullspiele kein Äquivalent zur Augenzahl existiert, kann nur die Information über Gewinn oder Verlust des Spiels beeinflusst werden. Zu diesem Zweck wird die Gewinninformation jeder Stichprobe auf die Zahlen 0 und 1 abgebildet und abhängig vom Unschärfeparameter σ mit einer gewissen Wahrscheinlichkeit gekippt.

Definition 5.2. Sei (Ω, \mathcal{A}, P) ein Wahrscheinlichkeitsraum, $g \in \{0,1\}$ gleich 1 genau dann, wenn die ausspielende Partei in der betrachteten Stichprobe das offene Spiel gewinnt, $n \in \mathbb{N}$ die gesamte Stichprobenanzahl und $\sigma \geq 0$ der Unschärfeparameter. Die **verrauschte Gewinninformation** G ist verteilt zu

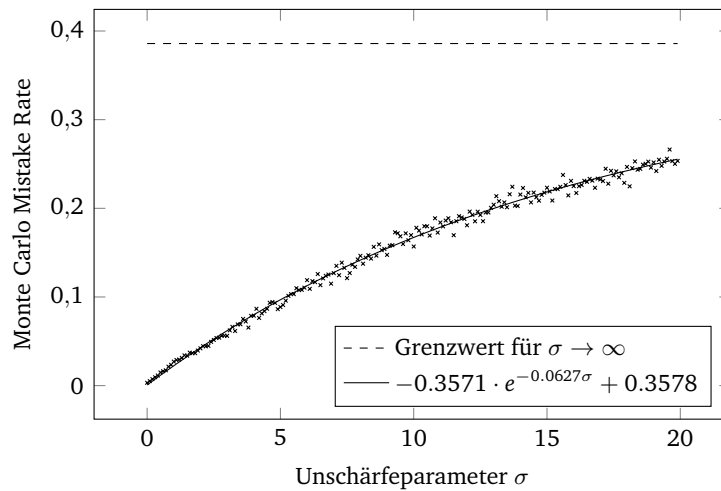
$$G : \Omega \rightarrow \{0,1\}, \quad G = \mathbf{1}_{>0,5}(\tilde{G}) \quad (5.3)$$

wobei $\mathbf{1}$ die charakteristische Funktion ist und die Hilfsvariable \tilde{G} verteilt ist zu

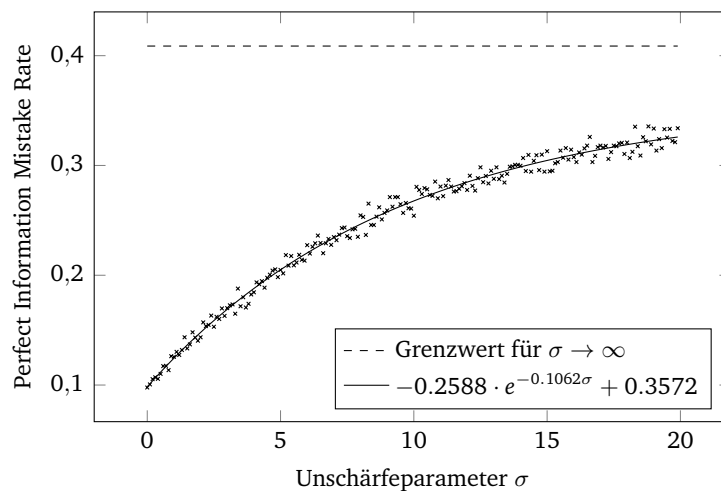
$$\tilde{G} : \Omega \rightarrow \mathbb{R}, \quad \tilde{G} \sim N(g, n\sigma^2) \quad (5.4)$$

Genau wie bei der verrauschten Augenzahl wird für $\sigma = 0$ das unmodifizierte Verfahren reproduziert und mit $\sigma \rightarrow \infty$ eine zufällige Zugwahl erreicht.

Um die Effektivität des Verfahrens zu testen, wurden für Abbildung 5.3 und 5.4 je 1.000 Spiele aus der ISS-Datenbank nachvollzogen und sowohl die Perfect-Information- als auch die Monte-Carlo-Fehlerrate des Spielers für unterschiedliche Werte von σ ermittelt. Der Monte-Carlo-Fehler beginnt wie erwartet nahe 0, da ein PIMC-Spieler aus MCPMA-Sicht optimal spielt. Der Perfect-Information-Fehler beginnt beim Wert des unmodifizierten Spielers; Als Grenzwert ist die Spielstärke bei Wahl zufälliger Züge aufgetragen (für beide Werte siehe Tabelle 5.1).

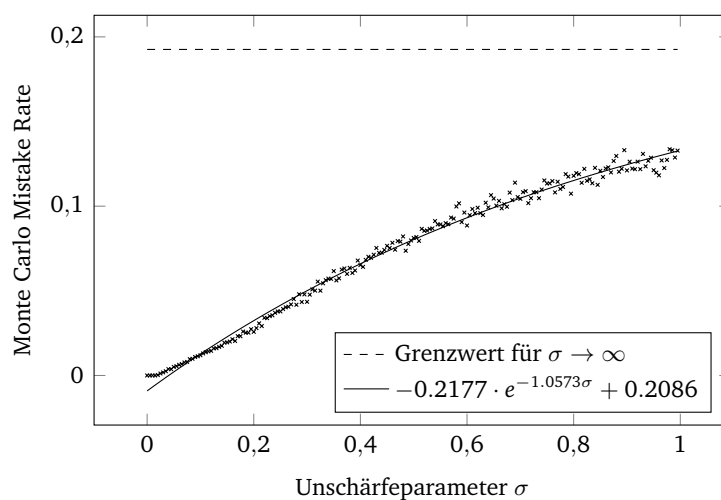


(a) Effekt von σ auf die Monte Carlo Mistake Rate. Der Wertebereich beginnt bei $MCMR = 0$, da das Verfahren für $\sigma = 0$ den Referenzspieler reproduziert.

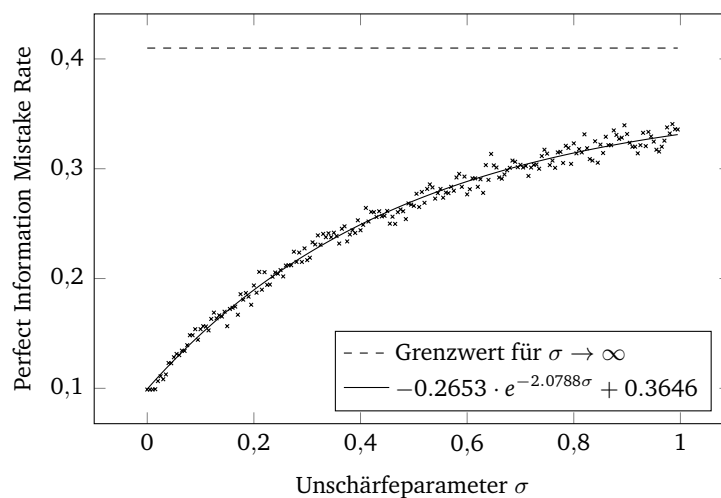


(b) Effekt von σ auf die Perfect Information Mistake Rate. Der Wertebereich beginnt bei $PIMR \approx 0,085$, der Fehlerrate des inferierenden Referenzspielers.

Abbildung 5.3: Adaptive Fehlerrate in Farb- und Grandspielen durch Verrauschen der Augenzahl pro Stichprobe, bestimmt an den Situationen von 1.000 Spielen. Die Fehlerrate lässt sich gut durch eine exponentielle Funktion in Abhängigkeit vom Unschärfeparameter σ vorhersagen.



(a) Effekt von σ auf die Monte Carlo Mistake Rate. Der Wertebereich beginnt bei $MCMR = 0$, da das Verfahren für $\sigma = 0$ den Referenzspieler reproduziert.



(b) Effekt von σ auf die Perfect Information Mistake Rate. Der Wertebereich beginnt bei $PIMR \approx 0,1$, der Fehlerrate des inferierenden Referenzspielers.

Abbildung 5.4: Adaptive Fehlerrate in Nullspielen durch Verrauschen der Gewinninformation je Stichprobe, bestimmt an den Situationen von 1.000 Spielen. Die Fehlerrate lässt sich gut durch eine exponentielle Funktion in Abhängigkeit vom Unschärfeparameter σ vorhersagen.

Es wird sichtbar, dass die Fehlerrate E gut in Abhängigkeit vom Parameter σ vorhersagbar ist. Sie lässt sich durch einen exponentiellen Term der folgenden Form annähern:

$$E \approx be^{a\sigma} + c \quad (5.5)$$

Parameter a , b und c sind dabei Eigenschaften des Fehlermaßes. Aus den hier analysierten Daten ergibt sich für Farb- und Grandspiele die Gleichung

$$MCMR \approx -0,3571 \cdot e^{-0,0627\sigma} + 0,3578 \quad (5.6)$$

$$PIMR \approx -0,2588 \cdot e^{-0,1062\sigma} + 0,3572 \quad (5.7)$$

und für Nullspiele der Zusammenhang

$$MCMR \approx -0,2177 \cdot e^{-1,0573\sigma} + 0,2086 \quad (5.8)$$

$$PIMR \approx -0,2653 \cdot e^{-2,0788\sigma} + 0,3646 \quad (5.9)$$

Aus der erwünschten Fehlerrate E mit $a < 0$, $b < 0$ und $c > E$ kann der Unschärfeparameter σ bestimmt werden zu

$$\sigma \approx \frac{1}{a} \ln \frac{E - c}{b} \quad (5.10)$$

Damit kann eine vorgegebene Spielstärke, quantifiziert durch eine mittlere Fehlerrate, durch einfache Anpassung von σ eingestellt werden.

Diese Umsetzung genügt auch der Anforderung, Fehler bevorzugt in unsicheren Situationen zu verursachen. Hebt sich ein Zug durch die erwartete Augenzahl stark von seinen Alternativen ab, so ist sein Vorteil wahrscheinlich auch von einem ungeübten Spieler zu erkennen. Eine leichte Unsicherheit in der verrauschten Augenzahl wird dann auch das Zugwahlverfahren nicht irritieren.

Lassen sich zwei Züge dagegen nicht verlässlich anhand der erreichten Augen trennen, so sind sie in ihrem Ausgang möglicherweise nur von einem erfahrenen Spieler zu unterscheiden. Eine Erhöhung der Unsicherheit über die erreichbare Augenzahl wird dann im Zugwahlschritt keine eindeutige Entscheidung mehr ermöglichen, und ein Spielfehler wird wahrscheinlicher.

5.3 Schrittweises Anpassen der Spielstärke

Beginnt das Spiel gegen einen unbekanntem menschlichen Spieler, so liegen noch keine Historien vor, aus denen sich seine Spielstärke bestimmen lässt. Auch können sich seine Fähigkeiten nach einigen Partien verbessern, sobald er die Spielweise des Computers ausreichend einschätzen kann. Daher gilt es, eine vernünftige Anfangsstrategie zu entwerfen, die fließend in das Spiel mit der gemessenen Stärke übergeht.

Solange keine andere Information vorliegt, tritt dazu der Computerspieler aus Kapitel 3 mit vorgegebener, idealerweise durchschnittlicher, Spielstärke an. Sind kleine Mengen an Spielerdaten gesammelt, so beginnt der Computer, sich seinen Gegnern anzupassen. Dies darf jedoch nicht zu schnell geschehen, um noch unsicheren Daten nicht blind zu vertrauen. Ist schließlich eine ausreichend große Datenbasis vorhanden, so sollten neuere Spielbeobachtungen einen größeren Einfluss auf das Verhalten haben als alte, um eine sich wandelnde Spielweise abzubilden.

Diese ist ein bekanntes Problem der Entscheidungsfindung auf beobachtetem Spielverhalten und keine Eigenart von Skat. Die kanonische Lösung ist die Verwendung von *Decaying Histories*: In einer langen Historie werden neue Datenpunkte stärker gewichtet als alte und die ältesten Werte bei Überschreiten einer Höchstzahl gegebenenfalls ganz verworfen (vgl. Billings u. a. (2004)).

Mit diesem Ansatz und der Vorgabe einer durchschnittlichen Spielhistorie zur Entwicklung der Anfangsstrategie kann nun ein Computerspieler realisiert werden, der mit definiertem Spielverhalten beginnt und sich an die Spielstärke seiner Gegner anpasst, sobald ausreichend Daten für deren Bewertung vorliegen.

5.4 Zusammenfassung

In diesem Kapitel wurde ein Verfahren aufgezeigt um eine Spielstärke zu imitieren, die mit den Maßen aus Kapitel 4 bestimmt wurde. Dazu wurde die maximal erreichbare Stärke des Computerspielers aus Kapitel 2 und 3 bestimmt und eine einfache, probabilistische Methode eingeführt, um sie auf einen vorgegebenen Wert abzusinken. Zuletzt wurde eine Möglichkeit skizziert, um ein definiertes Anfangsverhalten trotz mangelhafter Datenlage zu garantieren.

Damit wurde die Arbeit aller vorausgegangenen Kapitel zusammengeführt und das Ziel eines Skatspielers mit selbstkalibrierender Spielstärke realisiert.

6 Ausblick

Zu guter Letzt sollen die Entwicklungen und Ergebnisse dieser Arbeit zusammengefasst und ein Ausblick auf weitere, mögliche Forschung auf dem Gebiet gegeben werden.

6.1 Zusammenfassung

In dieser Arbeit wurde unter Verwendung etablierter und Entwicklung neuer Verfahren ein Computerspieler für Skat implementiert, der in der Lage ist, sich selbstständig den Fähigkeiten seiner Mitspieler anzupassen.

Dazu wurde zunächst in Kapitel 2 eine effiziente Methode zur Simulation von Wahrscheinlichkeitsverteilungen am Tisch entworfen, mit der das gängige Verfahren der Perfect-Information-Monte-Carlo-Suche für Skat umgesetzt werden kann. Um anhand der Stichprobenergebnisse eine auszuspielende Karte festlegen zu können, wurden anschließend mehrere Zugwahlverfahren entwickelt und umgesetzt.

In Kapitel 3 wurden Theorie und Praxis der Zustandsinferenz eingeführt, dem Schätzen der Kartenverteilung auf Basis gegnerischer Züge. Aufbauend darauf wurde eine Inferenzmethode auf Grundlage der Spielansage weiterentwickelt und ein neuer Ansatz zur Inferenz anhand ausgespielter Karten entwickelt. Um den Effekt der implementierten Verfahren bewerten zu können, wurde schließlich ein passendes Fehlermaß definiert und ausgewertet.

Um Gegner verlässlich einschätzen zu können, wurden in Kapitel 4 verschiedene Maße zur Bestimmung ihrer Spielstärke diskutiert. Neben dem klassischen Turniermaß und einem existierenden Verfahren aus der Literatur wurde die neue Methode der Monte Carlo Post Mortem Analysis entwickelt, die bereits nach wenigen Spielen eine solide Bewertung der Spielerfähigkeiten liefert. Um das Konvergenzverhalten der Verfahren zu quantifizieren, wurde wieder ein passendes Fehlermaß definiert und ausgewertet. Zuletzt wurden Annahmen über die Maße mit Daten eines Experten verifiziert.

In Kapitel 5 wurden die gewonnenen Erkenntnisse schließlich zusammengefasst, um eine adaptive Spielweise zu realisieren. Dazu wurde die maximale Spielstärke des Computerspielers ermittelt und ein neues Verfahren zur Absenkung der Fähigkeiten auf das Niveau der Gegner entwickelt.

6.2 Weitere Forschung

Während der Entwicklung des adaptiven Skatspielers sind Fragestellungen aufgetreten, die zwar nicht mehr in den Rahmen dieser Arbeit passen, aber dennoch für künftige Weiterentwicklungen interessant sein könnten.

Zustandsinferenz anhand ausgespielter Karten

Die Stichinferenz aus Abschnitt 3.5 hat großes theoretisches Potential. Die Implementierung in dieser Arbeit ist eher als *Proof of Concept* zu verstehen und bedarf noch weiterer Entwicklungen, um signifikante Gewinne gegenüber der naiven Gleichverteilungsannahme zu erreichen. Dabei ist etwa eine verbesserte Definition des Stichkontextes (siehe Definition 3.7) denkbar, der die Merkmale der Spielerhände zuverlässiger trennt.

Selbstinferenz und Bluffing

Lassen sich die Auswirkungen eines Spielzugs auf die geschätzte Kartenverteilung durch Zustandsinferenz hinreichend gut beziffern, so eröffnet dies die Möglichkeit zur *Selbstinferenz*: Abhängig davon, wie gut die Mitspieler nach einem Zug die Hand des Ausspielers einschätzen können, kann dem Zug neben der erwarteten Gewinnwahrscheinlichkeit auch ein Informationsgehalt zugewiesen werden.

Diese Größe kann als Grundlage dienen, um die Unzulänglichkeiten der PIMC-Suche im Zusammenspiel mit einem Partner (siehe Abschnitt 2.6) zu minimieren. Nach diesem Gesichtspunkt könnte ein günstiger Zug etwa die Informationslage des Teampartners verbessern, gleichzeitig aber die des Alleinspielers unberührt lassen.

Als Erweiterung ist es sogar denkbar, durch unerwartete Züge eine unzutreffende Hand zu suggerieren und damit die Zustandsinferenz des Gegners zu verwirren — eine Technik, die als *Bluffing* bekannt ist. Das Konzept von Bluffing auf Basis von Selbstinferenz wird in Whitehouse (2014) beschrieben, wenn auch nicht im Kontext der PIMC-Suche.

Abkürzungsverzeichnis

KI Künstliche Intelligenz, bezieht sich auf den Skatspieler aus Kapitel 2 und 3

ISS International Skat Server, siehe Abschnitt 1.7

PI Perfect Information, Präfix für offene Skatspiele, siehe Abschnitt 1.4

PIMC Perfect Information Monte Carlo, siehe Kapitel 2

PIPMA Perfect Information Post Mortem Analysis, siehe Abschnitt 4.2

PIMR Perfect Information Mistake Rate, siehe Definition 4.5

MCPMA Monte Carlo Post Mortem Analysis, siehe Abschnitt 4.3

MCMR Monte Carlo Mistake Rate, siehe Definition 4.7

Abbildungsverzeichnis

1.1	Spielfluss	7
2.1	Strategiefindung mit Perfect Information Monte Carlo Search	11
3.1	Merkmalsreproduktion bei Spielansageinferenz im Farbspiel	38
3.2	Merkmalsreproduktion bei Spielansageinferenz im Grand	39
3.3	Merkmalsreproduktion bei Spielansageinferenz im Nullspiel	40
3.4	Mittlere fehlende Karten pro Stichprobe	45
4.1	Entwicklung des Listenpunktemaßes	51
4.2	Abhängigkeit des Listenpunktemaßes vom Gegenspiel	52
4.3	Entwicklung der Perfect-Information-Fehlerrate	54
4.4	Korrelation zwischen Listenpunkten und PIMR	55
4.5	Abhängigkeit der Perfect-Information-Fehlerrate vom Gegenspiel	55
4.6	Entwicklung der Monte-Carlo-Fehlerrate	59
4.7	Korrelation zwischen Listenpunkten und Monte Carlo Mistake Rate	60
4.8	Abhängigkeit der Monte-Carlo-Fehlerrate vom Gegenspiel	61
4.9	Entwicklung des Rangordnungsfehlers	63
5.1	PI-Fehlerrate des Computers mit und ohne Inferenz	68
5.2	PI-Fehlerrate des Computers bei unterschiedlichen Zugwahlverfahren	70
5.3	Adaptive Fehlerrate in Farb- und Grandspielen	73
5.4	Adaptive Fehlerrate in Nullspielen	74

Literaturverzeichnis

- [Billings u. a. 2004] Billings, Darse ; Davidson, Aaron ; Schauenberg, Terence ; Burch, Neil ; Bowling, Michael ; Holte, Robert ; Schaeffer, Jonathan ; Szafron, Duane: Game-tree search with adaptation in stochastic imperfect-information games. In: *International Conference on Computers and Games* Springer (Veranst.), 2004, S. 21–34
- [Buro u. a. 2009] Buro, Michael ; Long, Jeffrey R. ; Furtak, Timothy ; Sturtevant, Nathan R.: Improving State Evaluation, Inference, and Search in Trick-Based Card Games. In: *IJCAI*, 2009, S. 1407–1413
- [DSkV 2010] DSkV: *Internationale Skatordnung*. http://www.dskv.de/upload_user/skatgericht/PDF/ISk0.pdf. 2010. – Abgerufen am 07.09.2017
- [Frank und Basin 1998] Frank, Ian ; Basin, David: Search in games with incomplete information: A case study using bridge card play. In: *Artificial Intelligence* 100 (1998), Nr. 1-2, S. 87–123
- [Georgii 2009] Georgii, Hans-Otto: *Stochastik: Einführung in die Wahrscheinlichkeitstheorie und Statistik*. Walter de Gruyter, 2009
- [Ginsberg 1999] Ginsberg, Matthew L.: GIB: Steps toward an expert-level bridge-playing program. In: *IJCAI*, 1999, S. 584–593
- [Kapp 2012] Kapp, Karl M.: *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012
- [Knuth und Moore 1975] Knuth, Donald E. ; Moore, Ronald W.: An analysis of alpha-beta pruning. In: *Artificial intelligence* 6 (1975), Nr. 4, S. 293–326
- [Kupferschmid und Helmert 2006] Kupferschmid, Sebastian ; Helmert, Malte: A skat player based on Monte-Carlo simulation. In: *International Conference on Computers and Games* Springer (Veranst.), 2006, S. 135–147
- [Long 2011] Long, Jeffrey R.: *Search, inference and opponent modelling in an expert-caliber skat player*, Dissertation, 2011

- [Long und Buro 2011] Long, Jeffrey R. ; Buro, Michael: Real-Time Opponent Modeling in Trick-Taking Card Games. In: *IJCAI Proceedings-International Joint Conference on Artificial Intelligence* Bd. 22, 2011, S. 617
- [Sauer 2017] Sauer, Tomas: *Spieltheorie*. Logos Verlag Berlin GmbH, 2017
- [Whitehouse 2014] Whitehouse, Daniel: *Monte Carlo Tree Search for games with Hidden Information and Uncertainty*, University of York, Dissertation, 2014