Eine Übersicht über den Aufbau und die aktuellen Möglichkeiten von Multi-Agenten Systemen

Studienarbeit

im Rahmen der Prüfung zum

Bachelor of Science (B.Sc.)

des Studienganges Informatik

an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

Johannes Quast

Abgabedatum:	? 2021
Bearbeitungszeitraum:	01.10.2017 - 31.01.2018
Matrikelnummer, Kurs:	0000000, TINF19B2
Ausbildungsfirma:	SAP SE Dietmar-Hopp-Allee 16 69190 Walldorf, Deutschland
Betreuer der Ausbildungsfirma:	B-Vorname B-Nachname
Gutachter der Dualen Hochschule:	Prof. Dr. Ralph Lausen

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema:

> *Eine Übersicht über den Aufbau und die aktuellen Möglichkeiten von Multi-Agenten Systemen*

gemäß § 5 der „Studien- und Prüfungsordnung DHBW Technik“ vom 29. September 2017 selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, den 15. Januar 2022

Quast, Johannes

Sperrvermerk

Die nachfolgende Arbeit enthält vertrauliche Daten der:

SAP SE
Dietmar-Hopp-Allee 16
69190 Walldorf, Deutschland

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anderslautende Genehmigung vom Dualen Partner vorliegt.

Abstract

- English -

This is the starting point of the Abstract. For the final bachelor thesis, there must be an abstract included in your document. So, start now writing it in German and English. The abstract is a short summary with around 200 to 250 words.

Try to include in this abstract the main question of your work, the methods you used or the main results of your work.

Abstract

- Deutsch -

Dies ist der Beginn des Abstracts. Für die finale Bachelorarbeit musst du ein Abstract in deinem Dokument mit einbauen. So, schreibe es am besten jetzt in Deutsch und Englisch. Das Abstract ist eine kurze Zusammenfassung mit ca. 200 bis 250 Wörtern.

Versuche in das Abstract folgende Punkte aufzunehmen: Fragestellung der Arbeit, methodische Vorgehensweise oder die Hauptergebnisse deiner Arbeit.

Inhaltsverzeichnis

Formelverzeichnis

A	mm^2	Fläche
D	mm	Werkstückdurchmesser
$d_{\mathbf{min}}$	mm	kleinster Schaftdurchmesser
$L_{\mathbf{1}}$	mm	Länge des Werkstückes Nr. 1
	Grad	Freiwinkel
	Grad	Keilwinkel

Abkürzungsverzeichnis

MDP Markov Decision Process

MAL Multi-Agent Learning

MARL Multi-Agent Reinforcement Learning

MAS Multi-Agent System

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

1 Einleitung

1.1 Motivation

1.2 Zielsetzung

1.3 Struktur der Arbeit

1.4 Literaturübersicht

2 Grundlagen

In diesem Kapitel werden zunächst die wichtigsten Konzepte und Methode erläutert, die für das Verständnis von Systemen benötigt werden, an denen nur ein einzelner Agent beteiligt ist. Diese Systeme stellen einen sehr guten Einstiegspunkt für das weitere Verständnis dar und sollen die zugrundeliegenden Konzepte einfach und verständlich erläutern. Diese werden anschließend vertieft und erweitert, sodass sie auf Systeme mit mehreren Agenten anwendbar werden.
Besonderer Fokus wird dabei auf das mathematische Verständnis gelegt, da im weiteren Verlauf der Arbeit vermehrt auf diese zurückgegriffen werden wird.

2.1 Agenten

In der Literatur gibt es keine eindeutige Definition für einen Agenten, allerdings existiert eine generelle Überschneidung dahingehend, dass ein intelligenter Agent alles sein kann, was seine Umgebung über Sensoren wahrnimmt und diese über Aktoren beeinflussen kann. Um diese sehr generelle Beschreibung in ein konkretes Beispiel zu überführen, kann beispielhaft ein Roboter benutzt werden, dessen Aufgabe es ist, einen bestimmten Gegenstand anzuheben. Der Roboter nimmt seine Umgebung über eine Vielzahl von Sensoren wie z. B. Kameras, Abstandsmesser oder ähnliches war. Diese Umgebung kann er z. B. über einen oder mehrere Greifarme (Aktoren) beeinflussen. Der Roboter muss u.a. anhand der empfangenen Sensordaten die nächste Entscheidung treffen, um sein Ziel zu erreichen. Abbildung 2.1 veranschaulicht das generelle Prinzip erneut anhand eines Flussdiagramms. Im Verlauf dieser Arbeit wird neben der Anzahl und Komplexität der verschiedenen Aktoren auch die Größe und Wahrnehmbarkeit der Umgebung eine immer wichtigere Rolle bekommen,

- da diese in ihrer Größe und Form nicht beschränkt ist. Die Umgebung kann z. B. von der Größe eines Raumes bis hin zum gesamten Universum reichen.
- da die Umgebung häufig nicht vollständig von den Sensoren erfasst werden kann und der Agent so nur einen eingeschränkten Teil der Umgebung wahrnimmt.

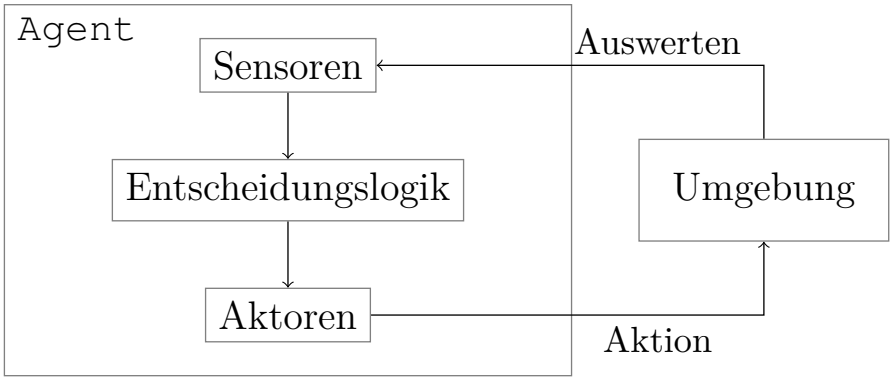

Abbildung 2.1: Die Interaktion eines Agenten mit seiner Umgebung.

Entscheidu
richtig?

Wie in Abbildung 2.1 dargestellt ist, nimmt der Agent seine Übergebung über seine Sensoren war und versucht anhand dieser eine Handlungsentscheidung zu treffen. Diese Entscheidung wird anschließend in konkrete Aktionen übersetzt, die die Umgebung zum Vorteil des Agenten verändern sollen. Welche konkrete Ausprägung die Entscheidungslogik dabei annimmt ist abhängig vom Aufbau und Typ des Agenten.

Quellen,
Grafik

Verweis
auf wei-
tere
Quellen

2.2 Verhalten eines Agenten

Aufbauend auf der Definition eines Agenten und seinen Möglichkeiten, über Sensoren die Umgebung wahrzunehmen und durch Aktoren zu verändern, kann nun das Verhalten festgelegt werden. Durch jede Aktion, die der Agent durchführt, wird der Zustand seiner Umgebung auf eine bestimmte, eventuell unerwünschte Weise verändert. Anders als bei Menschen oder anderen Lebewesen, die ihre eigenen Ansichten und Vorstellung von einem erwünschten Endergebnis haben, ist dies bei Agenten nicht der Fall.

Erweitern

Aus diesem Grund muss derjenige, der den Agenten und damit dessen eigentliches Ziel entwirft, besonders auf die korrekte Messung des Erfolges acht geben, denn dies kann mitunter eine große Herausforderung darstellen. Ob der Agent einen gewünschten Zustand durch eine Aktion erreicht hat, wird ihm entweder durch eine Belohnung (Reward) oder eine Bestrafung (Penalty) mitgeteilt. Im allgemeinen wird diese Bewertung auch als *Performance Measure* bezeichnet und gibt dem Agenten ein entsprechendes Feedback, ob er durch sein Handeln einen gewünschten Zustand erreicht hat. Ziel eines *rationalen* Agenten ist es stets, die Summe seiner Belohnungen zu maximieren.

Ein Agent, welcher z. B. Schach spielt, könnte am Ende der Partie eine Belohnung von einem Punkt bekommen, bei einer Niederlage entsprechend einen Punkt abgezogen. Das Ziel des Agenten ist es in diesem Kontext, möglichst viele Partien zu gewinnen, um seine Belohnungen zu maximieren. Bei der Formulierung einen solchen Zieles kann mitunter der Fehler passieren, dass der Designer über die Belohnungen versucht, einen bestimmten *Weg* vorzugeben, wie der Agent gewinnen soll. In diesem einfachen Beispiel kann es z. B. eine zusätzliche Belohnung für das eliminieren der gegnerischen Dame geben, allerdings könnten bei diesem Vorgehen Probleme auftreten. Der Agent würde unter Umständen Wege finden, die Dame zu eliminieren, um eine bestimmte Belohnung zu bekommen, allerdings das übergeordnete Ziel - den Sieg der Partie - vernachlässigen.

Erweitern: Agent entscheidet nur aufgrund der Sensoren oder auch mit vergangenem Wissen

2.3 Reinforcement Learning

Das Reinforcement Learning stellt eine Reihe von Methoden zur Verfügung, damit ein Agent selbstständig lernen kann, wie er seinen akkumulierten Belohnungen maximiert. Dies ist besonders dann von großem Vorteil, wenn der Agent einer sich stetig ändernden, oder für ihn völlig unbekannten Umgebung ausgesetzt ist, da er sich selbstständig an die neuen Bedingungen anpassen kann. Die sehr abstrakte Belohnungs bzw. Bestrafungsmechanik aus Sektion 2.2 wird im folgenden konkretisiert und mathematisch beschrieben. Die algorithmische Betrachtung des *Lernens* für Agenten wird hingegen in Kapitel 5 stark vertieft.

2.3.1 Interaktion eines Agenten mit seiner Umgebung

Die generelle Funktionsweise der Interaktion zwischen einem Agenten und seiner Umgebung ist in Abbildung 2.2 dargestellt. Die **Umgebung (Environment)** liefert bestimmte Signale an den Agenten, auf dessen Basis dieser Entscheidungen bzw. *Actions* durchführt. Zu den Signalen gehören u.a. eines für den aktuellen **Zustand (State)**, welches die aktuelle Lage der Umgebung darstellt. Ein weiteres Signal ist das **Belohnungssignal** bzw. der Reward und dient dem Agenten als *Feedback*, wie gut die von ihm ausgewählte Aktion im aktuellen Zustand war. Wie in der vorherigen Sektion bereits beschrieben, versucht der Agent diesen Reward auf Dauer zu **maximieren**. Generell finden diese Interaktionen zwischen Agent und Umgebung zu diskreten Zeitpunkten $t \in \mathbb{N}$ statt. Zu jedem dieser

Zeitpunkt führt der Agent eine Aktion aus, worauf die Umgebung als Konsequenz in einen neuen Zustand übergeht und gleichzeitig ein Belohnungssignal sendet.

Diese informelle Beschreibung der Interaktion wird im Folgenden anhand Abbildung 2.2 formalisiert und mathematisch beschrieben.

- S_t beschreibt mit $S_t \in \mathcal{S}$ den Zustand, in dem sich die Umgebung aktuell befindet. Die Menge $\mathcal{S}$ beinhaltet alle möglichen Zustände.
- Die Aktion $A_t \in \mathcal{A}(S_t)$, wobei $\mathcal{A}(S_t)$ alle möglichen Aktionen beinhaltet, die in Zustand S_t möglich sind, ist die Aktion, die als Reaktion auf den aktuellen Zustand ausgeführt wird.
- Als Konsequenz dieser Aktion reagiert die Umgebung mit einem Belohnungssignal $R_{t+1} \in \mathcal{R}$, welches dem Agenten als Feedback dient.

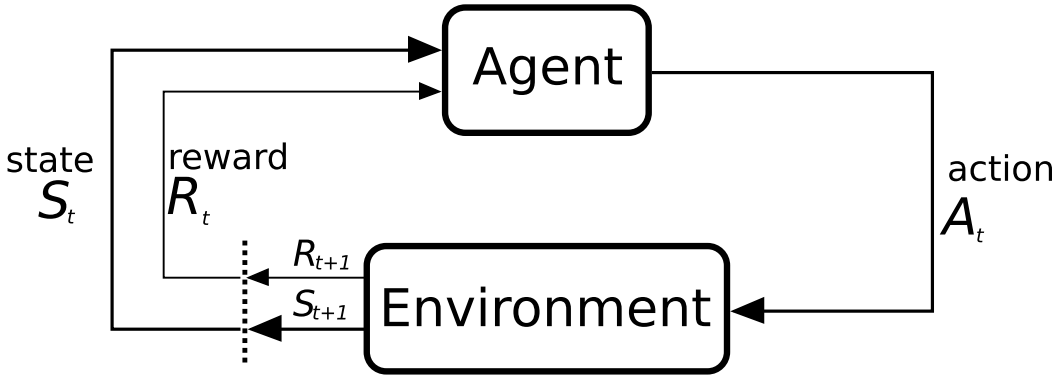

Abbildung 2.2: Interaktion zwischen einem Agenten und seiner Umgebung im Reinforcement Learning.

Die Entscheidung, welche Aktion in welchem Zustand ausgewählt wird, ist über die *Policy* Π_t des Agenten realisiert. $\Pi_t(a|s)$ ist dabei die Wahrscheinlichkeit, dass $A_t = a$, unter der Voraussetzung $S_t = s$. Vereinfacht ausgedrückt beschreibt $\Pi_t(a|s)$ damit die Wahrscheinlichkeit, dass zum Zeitpunkt t im Zustand s die Aktion a gewählt wird.

Ferner ist eine Historie $h_t = (S_0, A_0, R_1, S_1, A_1, ..., R_t, S_t)$ definiert, welche alle vergangenen Zustände, getroffenen Aktionen und erhaltene Belohnungssignale bis zum Zeitpunkt t aufzeichnet.

2.3.2 Markov Decision Process (MDP)

Mithilfe des Markov-Entcheidungsprozesses ist es möglich, die in Sektion 2.3.1 beschriebene Interaktion für viele verschiedene Probleme des Reinforcement-Learning zu modellieren und mathematisch zu beschreiben. Aus Sicht eines Agenten lässt sich dadurch die Umgebung formal erfassen.

Für *endliche MDP* besitzen die Zufallsvariablen S_t und R_t wohldefinierte, diskrete Zufallsverteilungen, die nur vom vorherigen Zustand und der gewählten Aktion abhängen. Die Wahrscheinlichkeit, dass zu einem bestimmten Zeitpunkt t die beiden Zufallsvariablen die konkreten Werte $s' \in \mathcal{S}$ und $r \in \mathcal{R}$ annehmen, ist über die Funktion $p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ definiert.

Endliche MDP Definieren

$$p(s', r|s, a) = Pr[S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a] \tag{2.1}$$

Aus dieser kompakten Funktion ist es nun u.a möglich, weiteren benötigte Funktionen zu bestimmen bzw. herzuleiten.

Erweitern

Markov-Eigenschaft

Die Markov-Eigenschaft ist eine wichtige Bedingung für viele Algorithmen und beschreibt, dass für die Vorhersage, welchen Zustand S_{t+1} die Umgebung als Reaktion auf die Aktion A_t annimmt, der geghenwärtige Zustand S_t ausreichend ist. Die Übergangswahrscheinlichkeit vom gegenwärtigen Zustand zum nachgolfenen ist somit unabhängig von der Vergangenheit, die in der Historie erfasst ist. Dies bedeutet, dass jeder Zustand *implizit* bereits alle nötigen Informationen mitführt, die für die Zukunft relevant sind. In Formel 2.2 ist diese informelle Beschreibung noch einmal mathematisch formuliert.

$$Pr[S_{t+1}|S_t = s, A_t = a] = Pr[S_{t+1}|h_t, A_t] \tag{2.2}$$

Jede Umgebung, die diese Eigenschaft erfüllt, kann als MDP modelliert werden.

2.4 Multi-Agent Systems

Für Systeme, in denen mehrere Agenten die Umgebung durch ihre Aktionen beiinflussen, kann die Agent-Umgebung Interaktion aus Abbildung 2.1 entsprechend erweitert werden.

2.4.1

2.4.2 Herausforderungen

3 Organisation von Agenten

TODO: Wie sind Agenten untereinander organisiert? Ist diese Origanisation statisch, oder kann sie sich verändern?

4 Kommunikationsstrategien

5 Lernalgorithmen

Dieses Kapitel legt den Fokus primär auf die verschiedenen Lernalgorithmen und die zugrundeliegenden mathematischen Konzepte. Dazu werden zunächst in Sektion 5.1 zwei wichtige Reinforcement Learning Algorithmen besprochen, die auf Umgebungen mit nur einem Agenten anwendbar sind. Zusätzlich wird eine Betrachtung durchgeführt, warum diese in der Theorie nur schlecht in Umgebungen mit mehreren Agenten einsetzbar sind. Anschließend werden entsprechend modifizierte Algorithmen vorgestellt, die versuchen die benannten Probleme zu lösen bzw. entsprechend zu umgehen.

5.1 Single-Agent Anwendungsfall

5.1.1 Q-Learning

5.1.2 Sarsa

5.1.3 Probleme für Multi-Agenten Systeme

5.2 Q-Learning

6 MAS Frameworks

7 Implementation einer Beispielanwendung

8 Fazit und Ausblick