Vorlesung
Grundlagen adaptiver Wissenssysteme

Prof. Dr. Thomas Gabel
Frankfurt University of Applied Sciences
Faculty of Computer Science and Engineering
tgabel@fb2.fra-uas.de

Vorlesungseinheit 3

Markov'sche Entscheidungsprozesse

Markov'sche Entscheidungsprozesse

Lernziele

- Definition Markov'scher Entscheidungsprozesse (MDPs)
- Kennenlernen grundlegender Begrifflichkeiten
- Überblick über Kategorien des optimierenden Lernens
- Einführung in Dynamisches Programmieren (DP)

Danksagung

Ein Teil der Folien basiert auf einem Foliensatz von Martin Riedmiller, einige Abbildungen stammen von David Silver (beide Google Deepmind).

Markov'sche Entscheidungsprozesse

Überblick

1. Agenten und ihre Umgebung

Markov’sche Entscheidungsprozesse

Überblick

1. Agenten und ihre Umgebung

2. Definition von MDPs

Markov'sche Entscheidungsprozesse

Überblick

1. Agenten und ihre Umgebung
2. Definition von MDPs
3. Handlungsstratregien

Markov'sche Entscheidungsprozesse

Überblick

1. Agenten und ihre Umgebung
2. Definition von MDPs
3. Handlungsstratregien
4. Wertfunktionen und Pfadkostenvektoren

Markov’sche Entscheidungsprozesse

Überblick

Markov’sche Entscheidungsprozesse

Überblick

1. Agenten und ihre Umgebung
2. Definition von MDPs
3. Handlungsstratregien
4. Wertfunktionen und Pfadkostenvektoren
5. Horizont und Modell
6. Kategorisierung des Themengebiets RL

Markov'sche Entscheidungsprozesse

Überblick

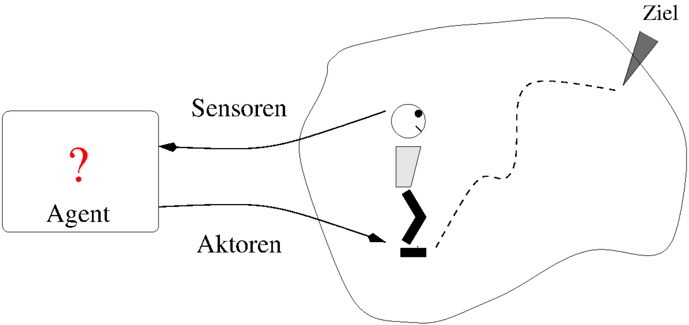
1. Agenten und ihre Umgebung
2. Definition von MDPs
3. Handlungsstratregien
4. Wertfunktionen und Pfadkostenvektoren
5. Horizont und Modell
6. Kategorisierung des Themengebiets RL

Rückblick

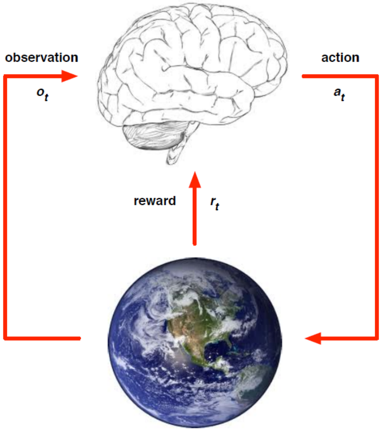
Typ von Problemen, die wir betrachten:

- Prozess, durch Aktionen beeinflussbar
- Agent: Sensor-Eingabe, Aktions-Ausgabe
- Rückkopplung
- OL (RL): lediglich Bewertung als Trainingsinformation
- Verzögertes Reinforcement Learning:
  Entscheidung, Entscheidung, Entscheidung, . . . Bewertung
- mehrstufiges Entscheidungsproblem
- Optimierung

Das Agenten-Konzept



Der Agent und seine Umgebung



In jedem Zeitschritt t wird der Agent

- Beobachtungen o_t aus seiner Umgebung empfangen
- eine Aktion a_t ausführen
- eine skalare Bewertung erfahren
  - in Form einer Belohnung r_t
  - oder als Kosten c_t

Die Historie des Agenten

- Die Historie des Agenten ist die Sequenz aller Beobachtungen, Aktionen und Bewertungen

$$h_t = o_1, a_1, r_1, \ldots, o_t, a_t, r_t$$

- d.h. alle wahrgenommenen Variablen bis zur Zeit t

Die Historie des Agenten

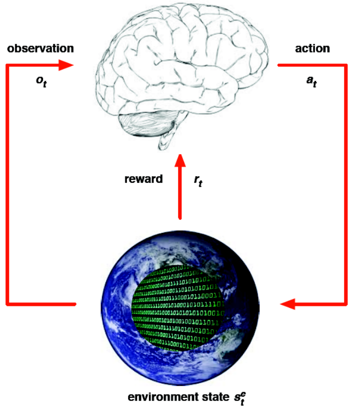
- Die Historie des Agenten ist die Sequenz aller Beobachtungen, Aktionen und Bewertungen

$$h_t = o_1, a_1, r_1, \ldots, o_t, a_t, r_t$$

- d.h. alle wahrgenommenen Variablen bis zur Zeit t
- Was als nächstes passiert, hängt von der Historie ab:
  - Der Agent wählt seine nächste Aktion.
  - Die Umgebung teilt dem Agenten die nächste Bewertung mit.
  - Die Umgebung teilt dem Agenten die nächste Beobachtung mit.
- Unter dem Zustand verstehen wir alle Informationen, die der Agent nutzt, um seine nächste Aktion zu wählen.
- Formal ist der Zustand eine Funktion der Historie:

$$s_t = f(h_t)$$

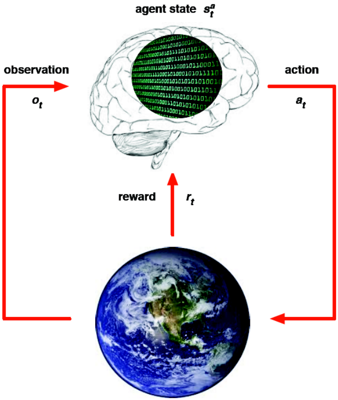
Der Zustand der Umgebung



Der Zustand des Umgebung s_t^e ist die, möglicherweise nicht öffentlich verfügbare, Repräsentation der Umwelt.

- enthält alle Informationen, die die Umgebung für die Verteilung der nächsten Beobachtungen bzw. Bewertungen braucht
- In der Praxis ist der Zustand der Umgebung für den Agenten oftmals nicht voll beobachtbar.
- kann (aus Sicht des Agenten) überflüssige und irrelevante Informationen enthalten

Der Zustand eines Agenten

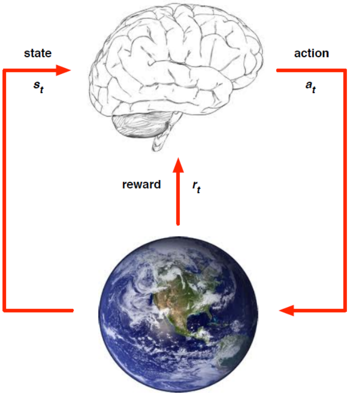


Der Zustand des Agenten s_t^a ist die interne Repräsentation, die der Agent von der ihn umgebenden Welt hat.

- enthält alle Informationen, die der Agent für die Auswahl seiner nächsten Aktion braucht
- ist die Information, mit der RL-Algorithmen arbeiten
- kann eine beliebige Funktion der Historie sein

$$s_t^a = f(h_t)$$

Voll beobachtbare Umgebungen



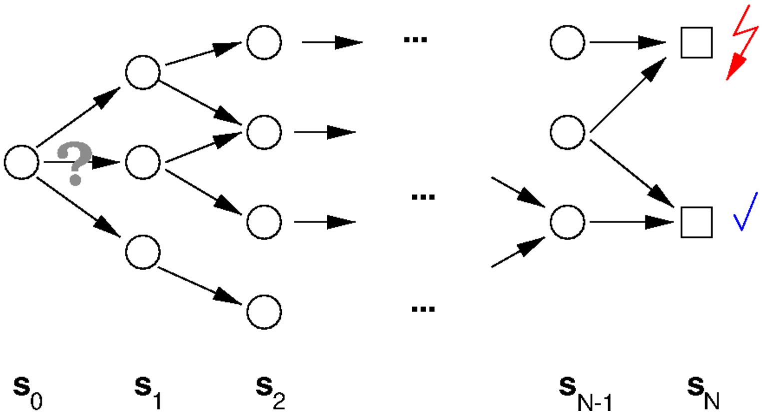
Volle Beobachtbarkeit bedeutet, dass der Agent den Zustand der Umgebung vollständig wahrnehmen kann:

$$o_t = s_t^a = s_t^e =: s_t$$

- Zustand des Agenten = Zustand der Umgebung
- Formal führt dies zu Markov'schen Entscheidungsprozessen (MDP)
- im Gegensatz zu partiell beobachtbaren Umgebungen (POMDP)

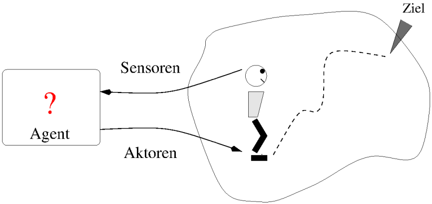
Mehrstufige Entscheidungsprobleme

Zur Erinnerung:
Mehrstufige, d.h. sequentielle Entscheidungsprobleme



Drei “Baustellen”

- System, Prozess
  - Umwelt bzw. Umgebung, Zustandsübergänge
- Bewertungen (Kosten bzw. Belohungen)
  - Rückmeldungen an den Agenten, Trainingssignal
- Strategie
  - Verhalten des Agenten, Aktionswahl



Markov'sche Entscheidungsprozesse

Überblick

1. Agenten und ihre Umgebung
2. Definition von MDPs
3. Handlungsstratregien
4. Wertfunktionen und Pfadkostenvektoren
5. Horizont und Modell
6. Kategorisierung des Themengebiets RL

Anforderungen an die Beschreibung der Umgebung

Ziel: Beschreibung des Systemverhaltens

- wobei mit System auch bezeichnet wird: Prozess, Umwelt, Welt, Umgebung

Anforderungen an solch ein formales Modell:

Anforderungen an die Beschreibung der Umgebung

Ziel: Beschreibung des Systemverhaltens

- wobei mit System auch bezeichnet wird: Prozess, Umwelt, Welt, Umgebung

Anforderungen an solch ein formales Modell:

- aktuelle Situation kann durch Einflussnahme des Agenten geändert werden (Bsp: Schach, Skifahrer, ...)
- Eingriffe des Agenten sind zu bestimmten diskreten Zeitpunkten möglich
- Zielspezifikation: mittels der Definition von Kosten
  - alternativ: mittels der Definition von Belohnungen, die mit negative Kosten gleichgesetzt werden können
- Störungen, Rauschen müssen berücksichtigt werden

Systembeschreibung

diskrete Entscheidungszeitpunkte (Stufen, *Stages*)	$t \in T = \{0, 1, \ldots, N\}$ bzw. $T = \{0, 1, \ldots\}$

Systembeschreibung

diskrete Entscheidungszeitpunkte (Stufen, *Stages*)	$t \in T = \{0, 1, \ldots, N\}$ bzw. $T = \{0, 1, \ldots\}$
Zustand (Situation)	$s_t \in S$ hier: S endlich

Systembeschreibung

diskrete Entscheidungszeitpunkte (Stufen, *Stages*)	$t \in T = \{0, 1, \dots, N\}$ bzw. $T = \{0, 1, \dots\}$
Zustand (Situation)	$s_t \in S$ hier: S endlich
Aktionen	$a_t \in A$ hier: A endlich

Systembeschreibung

diskrete Entscheidungszeitpunkte (Stufen, *Stages*)	$t \in T = \{0, 1, \dots, N\}$ bzw. $T = \{0, 1, \dots\}$
Zustand (Situation)	$s_t \in S$ hier: S endlich
Aktionen	$a_t \in A$ hier: A endlich
Übergangsfunktion "Reaktion des Systems"	$s_{t+1} = f(s_t, a_t)$

Zielformulierung: Einführung von Kosten (1)

Bewertungssignale

- In der Literatur wird wechselweise von Belohnungen, Bestrafungen bzw. Kosten (im Sinne negativer Belohnungen) sowie von Bewertungen (im neutralen Sinne) gesprochen.
- Alle drei Formalisierungen sind äquivalent und können leicht ineinander überführt werden.
- Im Folgenden verwenden wir bevorzugt die Formalisierung mit Hilfe von Kosten.

Zielformulierung: Einführung von Kosten (2)

Bei jeder Entscheidung (= in jeder Stufe) fallen direkte Kosten an

direkte Kosten	$c : S \to \mathbb{R}$
Verfeinerung: abhängig von Zustand und Aktion	$c : S \times A \to \mathbb{R}$
weitere Verfeinerung: abhängig von Zustand, Aktion und Folgezustand	$c : S \times A \times S \to \mathbb{R}$

Zielformulierung: Einführung von Kosten (2)

Bei jeder Entscheidung (= in jeder Stufe) fallen direkte Kosten an

direkte Kosten	$c : S \to \mathbb{R}$
Verfeinerung: abhängig von Zustand und Aktion	$c : S \times A \to \mathbb{R}$
weitere Verfeinerung: abhängig von Zustand, Aktion und Folgezustand	$c : S \times A \times S \to \mathbb{R}$

Erinnerung: Optimierungsziel über mehrere Stufen
$\Rightarrow$ Bewertung der Gesamtsequenz
(Erinnerung: Entscheidung, Entscheidung, ..., Bewertung)
$\Rightarrow$ Betrachte additive Gesamtkosten:

$$\sum_t c(s_t, a_t)$$

Zusammenfassung: deterministische Systeme

diskrete Entscheidungszeitpunkte	$t \in T = \{0, 1, \dots, N\}$ bzw.
"Stufen" (stages)	$T = \{0, 1, \dots\}$
Systemzustand (Situation)	$s_t \in S$
Aktionen	$a_t \in A$
Übergangsfunktion	$s_{t+1} = f(s_t, a_t)$
direkte Kosten	$c : S \times A \to \mathbb{R}$
alternativ zu Kosten: Belohnungen	$r : S \times A \to \mathbb{R}$ mit $r = -c$

$\Rightarrow$ Beschreibung der Umwelt mit einem 5-Tupel (T, S, A, f, c)

Beispiel

Kürzester-Pfad-Probleme

Suche den “kürzesten” Pfad von Anfangsknoten zu Endknoten.

- ist ein deterministisches Problem
- Frage: Wie lassen sich Kosten hier adäquat repräsentieren?

Beispiel

Kürzester-Pfad-Probleme

Suche den “kürzesten” Pfad von Anfangsknoten zu Endknoten.

- ist ein deterministisches Problem
- Frage: Wie lassen sich Kosten hier adäquat repräsentieren?
- Antwort: Jede Kante ist mit Kosten behaftet, die als “Länge” interpretiert werden.

Stochastische Systeme

Erinnerung: Anforderungen an ein Modell der Umgebung

- aktuelle Situation kann durch Einflussnahme geändert werden (Bsp: Schach, Skifahrer, ...)
- Eingriffe zu bestimmten diskreten Zeitpunkten möglich
- Definition von Kosten
  - alternativ: Belohnungen
- bislang noch nicht berücksichtigt:

Stochastische Systeme

Erinnerung: Anforderungen an ein Modell der Umgebung

- aktuelle Situation kann durch Einflussnahme geändert werden (Bsp: Schach, Skifahrer, ...)
- Eingriffe zu bestimmten diskreten Zeitpunkten möglich
- Definition von Kosten
  - alternativ: Belohnungen
- bislang noch nicht berücksichtigt: Störungen, Rauschen

Im Folgenden erfolgt Betrachtung stochastischer Umgebungen (in denen Störungen oder Rauschen auftreten können):
$\Rightarrow$ Markov'scher Entscheidungsprozess (MDP)

Markov'sche Entscheidungsprozesse

Deterministisches System: 5-Tupel (T, S, A, f, c)

Kernidee zur Erweiterung für stochastische Systeme:
Die deterministische Übergangsfunktion f wird ersetzt durch eine bedingte Wahrscheinlichkeitsverteilung.
Wir betrachten im folgenden eine *endliche* Zustandsmenge $S = (1, 2, \ldots, N)$. Es seien $i, j \in S$ zwei Zustände.

Schreibweise:

$$P(s_{t+1} = j | s_t = i, a_t = a) = p_{ij}(a)$$

$\Rightarrow$ Dies führt zur Definition eines

Markov'schen Entscheidungsprozesses (MDP)

Definition MDP

Definition (Markov'scher Entscheidungsprozess (MDP))

Ein Markov'scher Entscheidungsprozess (MDP) ist definiert als 5-Tupel (T, S, A, p, c) mit

- einer Menge T diskreter Entscheidungszeitpunkte
- einer Menge S der Zustände
- einer Menge A der durch den Agenten ausführbaren Aktionen
- der Funktion $p : S \times A \times S \to [0, 1]$ der Zustandsübergangswahrscheinlichkeiten mit $p(i, a, j) = p_{ij}(a)$
- der Funktion der direkten Kosten $c : S \times A \to \mathbb{R}$

Die Markov-Eigenschaft

Es gilt:

$$P(s_{t+1} = j | s_t, a_t) = P(s_{t+1} = j | s_t, s_{t-1}, \ldots, a_t, a_{t-1}, \ldots)$$

Die Wahrscheinlichkeitsverteilung des Folgezustands s_{t+1} ist durch Kenntnis des aktuellen Zustands s_t und der Aktion a_t eindeutig bestimmt. Sie hängt insbesondere nicht *von der gesamten bisherigen Historie h_t des Systems ab.*

Bemerkungen (1)

- Ein deterministisches System ist ein Sonderfall eines MDPs.
- Hier gilt vereinfachend:

Bemerkungen (1)

- Ein deterministisches System ist ein Sonderfall eines MDPs.
- Hier gilt vereinfachend:

$$P(s_{t+1}|s_t, a_t) = \begin{cases} 1 & , \quad s_{t+1} = f(s_t, a_t) \\ 0 & , \quad \text{sonst} \end{cases}$$

Bemerkungen (2)

Eine äquivalente Darstellung mit deterministischer Übergangsfunktion f ist möglich

- Idee: zusätzliches Argument einführen: Zufallsvariable w_t (Rauschen):

$$s_{t+1} = f(s_t, a_t, w_t)$$

- dabei ist w_t Zufalllsvariable mit gegebener Wahrscheinlichkeitsverteilung $P(w_t|s_t, a_t)$

Bemerkungen (2)

Eine äquivalente Darstellung mit deterministischer Übergangsfunktion f ist möglich

- Idee: zusätzliches Argument einführen: Zufallsvariable w_t (Rauschen):

$$s_{t+1} = f(s_t, a_t, w_t)$$

- dabei ist w_t Zufalllsvariable mit gegebener Wahrscheinlichkeitsverteilung $P(w_t|s_t, a_t)$

Transformation in vorige Darstellungsform:

- Sei $W(i, a, j) = \{w | j = f(i, a, w)\}$ die Menge aller Werte von w, für die das System aus Zustand i bei Eingabe von a in Zustand j übergeht.
- Dann gilt:

$$p_{ij}(a) = P(w \in W(i, a, j))$$

Zusammenfassung: MDPs (1)

diskrete Entscheidungszeitpunkte "Stufen"	$t \in T = \{0, 1, \dots, N\}$ bzw. $T = \{0, 1, \dots\}$
Systemzustand (Situation)	$s_t \in S$
Aktionen	$a_t \in A$
Übergangswahrscheinlichkeit $p_{ij}(a)$	$P(s_{t+1} = j \mid s_t = i, a_t = a) = p_{ij}(a)$
Alternativ: Übergangsfunktion mit Zufallsvariable w_t	$s_{t+1} = f(s_t, a_t, w_t)$
direkte Kosten	$c : S \times A \to \mathbb{R}$

$\Rightarrow$ 5-Tupel $(T, S, A, p_{ij}(a), c(s, a))$

Zusammenfassung: MDPs (2)

- Modell der Interaktion
  - Zustände, Aktionen, Folgezustände
- deterministische und stochastische Übergangsfunktion
- Information über “Historie” im aktuellen Zustand zusammengefasst
  - Markov-Eigenschaft

Zusammenfassung: MDPs (2)

- Modell der Interaktion
  - Zustände, Aktionen, Folgezustände
- deterministische und stochastische Übergangsfunktion
- Information über "Historie" im aktuellen Zustand zusammengefasst
  - Markov-Eigenschaft
- sehr allgemeine Beschreibungsform
  - nutzbar in Regelungstechnik, Operations Research, Spiele, ...
- diverse Verallgemeinerungen existieren (nicht Bestandteil der VL)
  - partiell beobachtbare Umgebungen (POMDPs)
  - Übergangsfunktion nicht stationär $p_{ij,t}(a)$
  - Kosten nicht stationär $c_t(i, a)$
  - Umgebungen mit mehr als einem Agenten
    - Multi-Agenten-Systeme (DEC-MDPs und DEC-POMDPs)

Beispiel

Lagerhaltung

Zustand: Anzahl Autos	s_t
Aktion: Bestelle Autos beim Werk	a_t
“Störung”: Verkaufte Autos	w_t
$\Rightarrow$ Systemgleichung:	$s_{t+1} = s_t + a_t - w_t$

Beispiel

Lagerhaltung

Zustand: Anzahl Autos s_t
Aktion: Bestelle Autos beim Werk a_t
"Störung": Verkaufte Autos w_t
$\Rightarrow$ Systemgleichung: $s_{t+1} = s_t + a_t - w_t$

Kostenarten:

- Kosten für Autos am Lager
- zuzüglich Anschaffungskosten für jedes gekaufte Auto
- abzüglich Gewinn für verkaufte Autos, ergibt:

$$c(s,a) = c_1(s) + c_2(a) - Gewinn$$

Beispiel

Lagerhaltung

Zustand: Anzahl Autos	s_t
Aktion: Bestelle Autos beim Werk	a_t
“Störung”: Verkaufte Autos	w_t
$\Rightarrow$ Systemgleichung:	$s_{t+1} = s_t + a_t - w_t$

Kostenarten:

- Kosten für Autos am Lager
- zuzüglich Anschaffungskosten für jedes gekaufte Auto
- abzüglich Gewinn für verkaufte Autos, ergibt:

$$c(s, a) = c_1(s) + c_2(a) - Gewinn$$

Außerdem fallen Terminalkosten $g(s)$ an, wenn nach Ablauf der N Zeitschritte noch unverkaufte Autos am Lager sind.

Markov’sche Entscheidungsprozesse

Überblick

Strategie und Auswahlfunktion (1)

Definition (Auswahlfunktion)

Die Auswahlfunktion $\pi_t : S \to A, \pi_t(s) = a$ wählt zum Zeitpunkt t eine Aktion $a \in A$ als Funktion des aktuellen Zustands $s \in S$.

- Eine Auswahlfunktion lässt den Agenten agieren und wählt dazu eine Aktion in Abhängigkeit der Situation (vgl. Skizze “Agent”).

Strategie und Auswahlfunktion (1)

Definition (Auswahlfunktion)

Die Auswahlfunktion $\pi_t : S \to A, \pi_t(s) = a$ wählt zum Zeitpunkt t eine Aktion $a \in A$ als Funktion des aktuellen Zustands $s \in S$.

- Eine Auswahlfunktion lässt den Agenten agieren und wählt dazu eine Aktion in Abhängigkeit der Situation (vgl. Skizze “Agent”).
- Verfeinerung im Hinblick auf sich verändernde (zustandsabhängige) Aktionsmengen ist möglich.
  - z.B. im Schach

Definition (Situationsabhängige Auswahlfunktion)

Eine situationsabhängige Auswahlfunktion ist definiert als $\pi_t : S \to A, \pi_t(s) = a,$ mit $a \in A(s)$.

Strategie und Auswahlfunktion (2)

Definition (Strategie)

Eine Strategie $\hat{\pi}$ besteht aus N Auswahlfunktionen (mit N als Anzahl der Entscheidungszeitpunkte)

$$\hat{\pi} = (\pi_0, \pi_1, \ldots, \pi_t, \ldots)$$

Namensgebung:

- englische Bezeichnung: policy
- daher übliche weitere Bezeichnungen: Politik, Taktik

Nichtstationäre Strategien

Zeitabhängige Entscheidungen

- Die Auswahlfunktion π_t kann vom Zeitpunkt der Entscheidung abhängen.
- Bedeutung: Dieselbe Situation kann zu unterschiedlichen Zeitpunkten eine unterschiedliche Entscheidung des Agenten hervorrufen.

Erinnerung

$$\hat{\pi} = (\pi_0, \pi_1, \ldots, \pi_t, \ldots)$$

Definition (Nichtstationäre Strategie)

Wenn sich die Auswahlfunktionen für einzelne Zeitpunkte unterscheiden, spricht man von nicht-stationären Strategien.

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Beispiel

Fußballspiel

Situation s: Mittelfeldspieler hat den Ball.
Sinnvolle Aktion in der ersten Minute: $\pi_1(s)$ = Rückpass
Sinnvolle Aktion in der letzten Minute: $\pi_{90}(s)$ = Torschuss

Erkenntnis:
Der begrenzte Optimierungszeitraum (d.h. der "endliche Horizont", der vom Agenten betrachtet wird) erfordert im Allgemeinen eine nichtstationäre Strategie!

Stationäre Strategien (1)

Definition (Stationäre Strategie)

Wenn die Auswahlfunktionen für alle Zeitpunkte identisch sind, spricht man von stationären Strategien.

Eigenschaften

- Es gilt dann $\pi_0 = \pi_1 = \ldots \pi_t \ldots =: \pi$ und

$$\hat{\pi} = (\pi, \pi, \ldots, \pi, \ldots)$$

- Bei stationären Strategien fallen also die Begriffe "Strategie" und "Auswahlfunktion" zusammen.
- Wir werden – wie in der Literatur im allgemeinen üblich – die Auswahlfunktion "π" als Strategie bezeichnen.

Stationäre Strategien (2)

Bemerkungen

- Wir werden uns im weiteren Verlauf der Vorlesung hauptsächlich mit stationären Strategien beschäftigen.
    - Ausnahme: Backward Dynamic Programming

Stationäre Strategien (2)

Bemerkungen

- Wir werden uns im weiteren Verlauf der Vorlesung hauptsächlich mit stationären Strategien beschäftigen.
  - Ausnahme: Backward Dynamic Programming
- Außerdem kommen im weiteren Verlauf ausschließlich deterministische Auswahlfunktionen zum Einsatz.
  - Im Gegensatz zu deterministischen Auswahlfunktionen wählen stochastische Auswahlfunktionen die nächste Aktion des Agenten mit einer gewissen Wahrscheinlichkeit.
  - Dann liefert $\pi_t : S \times A \to \mathbb{R}$ mit $\pi_t(s, a) \in [0, 1]$ eine Wahrscheinlichkeitsverteilung über die Aktionen.

Ziel der Strategie

Die Strategie des Agenten bestimmt dessen Handeln. Und damit implizit auch die Belohnungen bzw. Kosten, die er erfährt.

> *Aber: Das Optimierungsziel soll über mehrere Stufen (also mittels einer Sequenz von Entscheidungen) erreicht werden.*

Ziel der Strategie

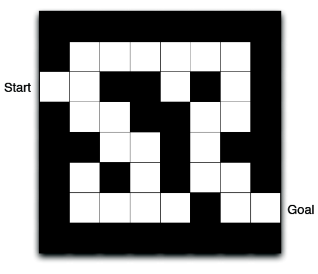
Die Strategie des Agenten bestimmt dessen Handeln. Und damit implizit auch die Belohnungen bzw. Kosten, die er erfährt.

Aber: Das Optimierungsziel soll über mehrere Stufen (also mittels einer Sequenz von Entscheidungen) erreicht werden.

- **Erkenntnis 1:** Das Bestimmen der idealen Strategie des Agenten ist ein Optimierungsproblem.
- **Erkenntnis 2:** Unser Ziel ist die Lösung dieses dynamischen Optimierungsproblems.
  - Daher auch der Terminus “optimierendes Lernen”

Beispiel (1)

Labyrinth

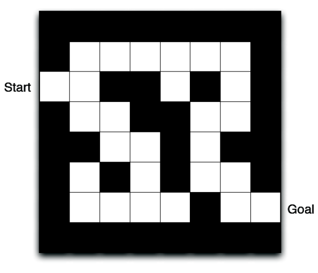


Charakterisierung der Umgebung mit Kosten, Aktionen und Zuständen

- Kosten pro Zeitschritt:

Beispiel (1)

Labyrinth

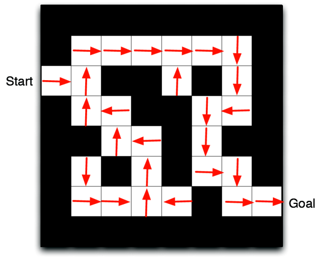


Charakterisierung der Umgebung mit Kosten, Aktionen und Zuständen

- Kosten pro Zeitschritt: 1
- Aktionen: N, S, E, W
- Zustände: Felder des Labyrinths, in denen sich der Agent befinden kann

Beispiel (2)

Labyrinth



Beispiel einer Strategie

- Jeder Pfeil deutet an, welche Aktion der Agent in welchem Zustand auswählen würde: $\pi(s)$

Markov'sche Entscheidungsprozesse

Überblick

1. Agenten und ihre Umgebung
2. Definition von MDPs
3. Handlungsstratregien
4. Wertfunktionen und Pfadkostenvektoren
5. Horizont und Modell
6. Kategorisierung des Themengebiets RL

Wertfunktionen und Pfadkosten

Definition (Wertfunktion)

Eine Wertfunktion $V : S \rightarrow \mathbb{R}$ wird zur Bewertung der Güte von Zuständen verwendet. Sie liefert eine Abschätzung zukünftiger zu erhaltender Belohnungen.

$$V(s) = \mathbb{E}[r_t + r_{t+1} + r_{t+2} + \ldots | s_t = s]$$

Bezeichnung: Pfadkostenvektor vs. Wertfunktion

Der Begriff der Pfadkosten wird in der Literatur äquilavent zum Begriff der Wertfunktion gebraucht. Der Unterschied besteht darin, dass sich Wertfunktionen auf erwartete Belohnungen beziehen, wohingegen der Pfadkostenvektor auf der Beschreibung mit erwarteten Kosten basiert.

Pfadkosten

Weitere Bezeichnungen: Pfadkosten, Cumulated Costs, Costs-To-Go

Zustands- und Strategiebezogenheit:

Pfadkosten beziehen sich auf einen gegebenen Zustand s bei fester Strategie π:

$$V^{\pi}(s) = \sum_{t \in T} c(s_t, \pi(s_t)), \quad s_0 = s$$

Pfadkosten

Weitere Bezeichnungen: Pfadkosten, Cumulated Costs, Costs-To-Go

Zustands- und Strategiebezogenheit:
Pfadkosten beziehen sich auf einen gegebenen Zustand s bei fester Strategie π:

$$V^\pi(s) = \sum_{t \in T} c(s_t, \pi(s_t)), \quad s_0 = s$$

Gesucht:
Die optimale Strategie π^*, so dass für alle s gilt:

$$V^{\pi^*}(s) =$$

Pfadkosten

Weitere Bezeichnungen: Pfadkosten, Cumulated Costs, Costs-To-Go

Zustands- und Strategiebezogenheit:
Pfadkosten beziehen sich auf einen gegebenen Zustand s bei fester Strategie π:

$$V^{\pi}(s) = \sum_{t \in T} c(s_t, \pi(s_t)), \quad s_0 = s$$

Gesucht:
Die optimale Strategie π^*, so dass für alle s gilt:

$$V^{\pi^*}(s) = \min_{\pi \in \Pi} \sum_{t \in T} c(s_t, \pi(s_t)), \quad s_0 = s$$

unter der Nebenbedingung $s_{t+1} = f(s_t, \pi(s_t))$

Pfadkosten im MDP

MDP modelliert aber ein stochastisches System

- Betrachte erwartete Pfadkosten für einen gegebenen Zustand s bei fester Strategie π:

$$V^{\pi}(s) = \mathbb{E}_w \sum_{t \in T} c(s_t, \pi(s_t)), \quad s_0 = s$$

- Gesucht: Die optimale Strategie π^* so dass für alle s gilt:

$$V^{\pi^*}(s) =$$

Pfadkosten im MDP

MDP modelliert aber ein stochastisches System

- Betrachte erwartete Pfadkosten für einen gegebenen Zustand s bei fester Strategie π:

$$V^\pi(s) = \mathbb{E}_w \sum_{t \in T} c(s_t, \pi(s_t)), \quad s_0 = s$$

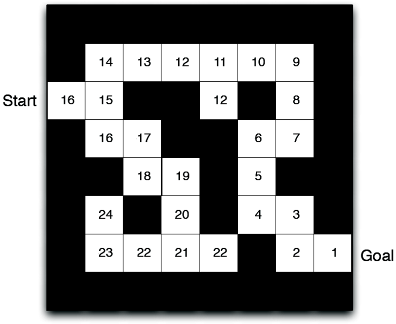
- Gesucht: Die optimale Strategie π^* so dass für alle s gilt:

$$V^{\pi^*}(s) = \min_{\pi \in \Pi} \mathbb{E}_w \sum_{t \in T} c(s_t, \pi(s_t)), \quad s_0 = s$$

  unter der Nebenbedingung $s_{t+1} = f(s_t, \pi(s_t), w_t)$ bzw. bei gegebener Wahrscheinlichkeitsverteilung $P(s_{t+1} = j | s_t = i, \pi(s_t) = a) = p_{ij}(a)$

Beispiel

Labyrinth



Beispiel einer Wertfunktion

- Jeder Eintrag gibt die Pfadkosten $V(s)$ eines Zustandes
- konkret: die Pfadkosten, die unter der optimalen Strategie π^* entstehen würden

Markov'sche Entscheidungsprozesse

Überblick

1. Agenten und ihre Umgebung
2. Definition von MDPs
3. Handlungsstratregien
4. Wertfunktionen und Pfadkostenvektoren
5. Horizont und Modell
6. Kategorisierung des Themengebiets RL

Problemtypen

Definition (Horizont)

Der Horizont N eines Problems bezeichnet die Anzahl der zu durchlaufenden Entscheidungsstufen.

Unterscheidung nach Horizonttypen:

- endlicher Horizont: Probleme mit vorgegebenem Abbruchzeitpunkt
- unendlicher Horizont: Approximation für sehr lange dauernde Vorgänge bzw. Vorgänge mit unbestimmtem Ende (z.B. Regelungsaufgaben)

Endlicher Horizont

Eigenschaften:

- N-stufiges Entscheidungsproblem
- Jeder Zustand hat Terminalkosten $g(i)$, die anfallen, wenn das System nach N Stufen in i endet.
- Kosten für eine Strategie π

$$V_N^\pi(s) =$$

Endlicher Horizont

Eigenschaften:

- N-stufiges Entscheidungsproblem
- Jeder Zustand hat Terminalkosten $g(i)$, die anfallen, wenn das System nach N Stufen in i endet.
- Kosten für eine Strategie π

$$V_N^\pi(s) = \mathbb{E}[g(s_N) + \sum_{t=0}^{N-1} c(s_t, \pi_t(s_t))|s_0 = s]$$

- typisch: nicht-stationäre Strategien werden angewendet
  - Weil das “Ende absehbar” ist, kann bzw. sollte in jedem Zeitschritt eine andere Auswahlfunktion zur Aktionswahl zum Einsatz kommen.

Unendlicher Horizont

Numerische Probleme:

- Kosten für eine Strategie π im Fall eines unendlichen Horizonts

$$V^\pi(s) = \lim_{N\to\infty} \mathbb{E}[\sum_{t=0}^{N} c(s_t, \pi_t(s_t))|s_0 = s]$$

- Frage: Ergeben sich hier endliche Kosten?

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Unendlicher Horizont

Numerische Probleme:

- Kosten für eine Strategie π im Fall eines unendlichen Horizonts

$$V^\pi(s) = \lim_{N\to\infty} \mathbb{E}[\sum_{t=0}^{N} c(s_t, \pi_t(s_t))|s_0 = s]$$

- Frage: Ergeben sich hier endliche Kosten? $\Rightarrow$ I.A. nein!
- Abhilfe: Diskontierung mit $\gamma < 1$

$$V^\pi(s) = \lim_{N\to\infty} \mathbb{E}[\sum_{t=0}^{N} \gamma^t c(s_t, \pi_t(s_t))|s_0 = s]$$

  - Umgangssprachlich: Kosten, die mich erst in der Zukunft treffen, wiegen “weniger schwer”.

Modell

Definition (Modell)

In einem MDP $M = [T, S, A, p, c]$ bezeichnen wir die Übergangswahrscheinlichkeiten p sowie die Kostenfunktion c (oder alternativ die Belohnungsfunktion r) als Modell der Umgebung.

Bemerkungen:

Modell

Definition (Modell)

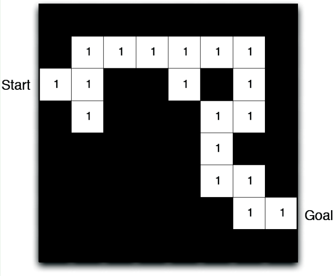
In einem MDP $M = [T, S, A, p, c]$ bezeichnen wir die Übergangswahrscheinlichkeiten p sowie die Kostenfunktion c (oder alternativ die Belohnungsfunktion r) als Modell der Umgebung.

Bemerkungen:

- Das Modell sagt vorher, was die Umgebung als nächstes tun wird.
- $p_{ij}(a)$ sagt voraus, mit welcher Wahrscheinlichkeit welcher Folgezustand eintreten wird
- $c(s, a)$ sagt die nächsten direkten Kosten voraus
- Das Modell kann unvollständig oder ungenau sein.

Beispiel

Labyrinth



Das Modell im Labyrinthbeispiel

- Dynamik: Wie ändert sich der Zustand infolge von Aktionen.
- Gittermuster repräsentiert das Zustandsübergangsmodell p.
- Kosten: Wieviele Kosten erbringt jeder Zustand(sübergang).
- Zahlen (in der Abbildung) geben direkte Kosten an (die für alle Aktionen gleich sind).

Beispiel: Modellierung

Typische Problemstellung: Ich möchte eine bestimmte Entscheidungsaufgabe lösen.
Zu beantwortende Frage:
Wie kann ich die Aufgabenstellung als MDP formulieren, d.h.

Beispiel: Modellierung

Typische Problemstellung: Ich möchte eine bestimmte Entscheidungsaufgabe lösen.

Zu beantwortende Frage:

Wie kann ich die Aufgabenstellung als MDP formulieren, d.h. was sind meine

- Zustände
- Aktionen
- Übergangswahrscheinlichkeiten (Modell)
- Entscheidungskosten
- Horizont/ Diskontierungsparameter

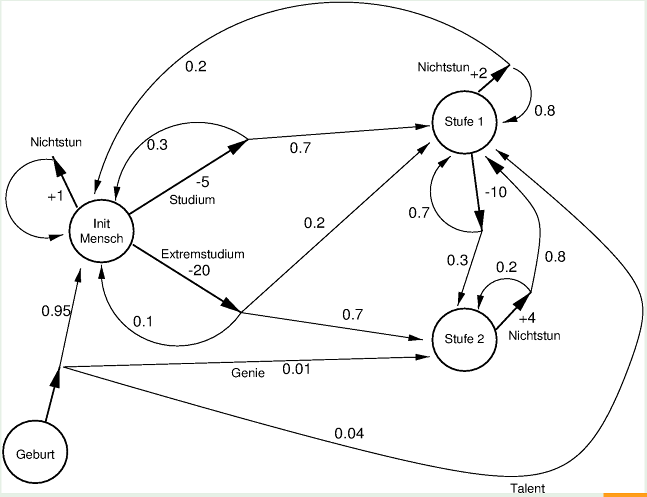
Beispiel

MDP "Leben"

- Zustände: Start (Geburt), Stufe Init, Stufe 1, Stufe 2
- Aktionen: faul sein (a), normal sein (b), fleissig sein (c)
- Man benötigt: Übergangswahrscheinlichkeiten
- Kosten, ggf. Diskontierung
- Ziel: Maximiere das Wohlbefinden oder Minimiere das Unglück

Beispiel

MDP “Leben”



Markov'sche Entscheidungsprozesse

Überblick

Überblick über RL-Verfahren (1)

Kategorien von RL-Agenten

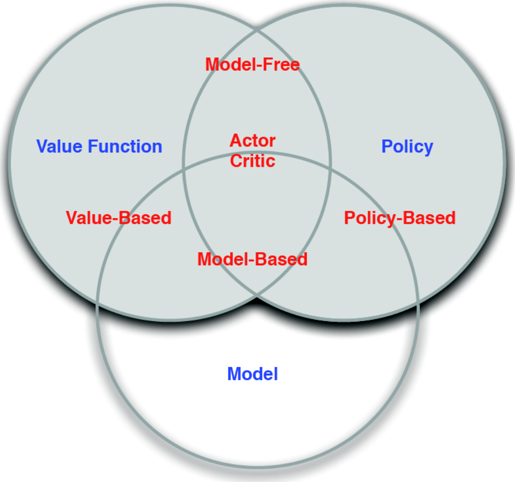
1. wertfunktionsbasierte / pfadkostenbasierte Methoden
   - verwenden eine Wertfunktion bzw. Pfadkostenvektor
   - die Strategie dieser Agenten ergibt sich implizit aus der Wertfunktion
2. strategiebasierte Methoden
   - verwenden eine explizite Strategie
   - also eine explizite Repräsentation der Funktion π
   - kommen ohne Wertfunktion / Pfadkostenvektor aus
3. Actor-Critic-Methoden
   - Kombination aus 1. und 2.
   - verwenden sowohl Wertfunktion als auch explizite Strategie

Überblick über RL-Verfahren (2)

Kategorien von RL-Agenten bzgl. Modellverwendung

1. modellfreie Methoden
   - verwenden eine Wertfunktion (bzw. Pfadkostenvektor) und/oder Strategie
   - verwenden kein Modell der Umgebung
2. modellbasierte Methoden
   - verwenden eine Wertfunktion (bzw. Pfadkostenvektor) und/oder Strategie
   - verwenden ein Modell der Umgebung

Überblick über RL-Verfahren (3)



1

[1] Bildquelle: D. Silver, University College London

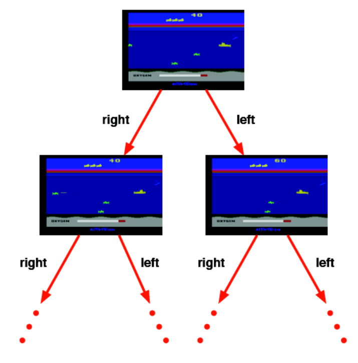
Planen versus Lernen

In mehrstufigen Entscheidungsproblemen unterscheidet man ganz grundsätzlich zwei Problemtypen:

1. Planung
   - Ein Modell der Umgebung ist bekannt.
   - Der Agent führt Berechnungen mit Hilfe dieses Modells durch (ohne externe Interaktion mit der Umgebung).
   - Der Agent verbessert seine Strategie.
   - gängige Begriffe dafür: Deliberation, Schließen, Suche, Reasoning, Intrspection
2. Optimierendes Lernen (RL)
   - Die Umgebung ist zu Beginn unbekannt.
   - Der Agent interagiert mit der Umgebung.
   - Der Agent verbessert seine Strategie.

Beispiel

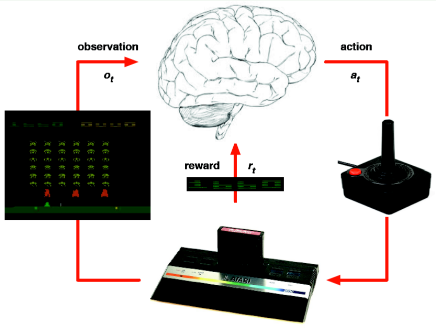
Videospiele: Planung



- Regeln des Spiels sind bekannt
- Agent kann einen Emulator (= Modell) befragen
  - perfektes Modell der Umgebung existiert innerhalb des Agenten
- Bekannt: Wenn ich Aktion *a* im Zustand *s* ausführe,
  - welcher Folgezustand tritt ein?
  - welcher Spielstand ergibt sich?
- Vorausplanen, um die optimale Aktion zu finden
  - z.B. mittels Baumsuche

Beispiel

Videospiele: Optimierendes Lernen



- Regeln des Spiels sind unbekannt
- Lernen erfolgt direkt durch interaktives Spielen
- Agent wählt Aktionen für seinen Joystick und nimmt den Spielstand sowie die Pixel auf dem Bildschirm wahr

Ausbeutung und Exploration (1)

Wie neugierig sollte ein RL-Agent sein?

- Optimierendes Lernen bedeutet Lernen auf Basis von Versuch und Irrtum (Trial and Error).
- Der Agent soll eine gute Strategie finden.
- Er soll dies schaffen auf Basis seiner Erfahrungen, die er in Interaktion mit der Umgebung gesammelt hat.
- Während er das tut, soll er gleichzeitig so wenig Kosten wie möglich erfahren.
  - z.B. sich nicht ernsthaft beschädigen

Ausbeutung und Exploration (2)

Definition (Ausbeutung und Exploration)

- Unter Ausbeutung (Exploitation) bzw. ausbeutendem Verhalten versteht man die Ausnutzung vorhandener, bereits gelernter Informationen mit dem Ziel, entstehende Kosten zu minimieren (bzw. erhaltbare Belohnungen zu maximieren).

Ausbeutung und Exploration (2)

Definition (Ausbeutung und Exploration)

- Unter Ausbeutung (Exploitation) bzw. ausbeutendem Verhalten versteht man die Ausnutzung vorhandener, bereits gelernter Informationen mit dem Ziel, entstehende Kosten zu minimieren (bzw. erhaltbare Belohnungen zu maximieren).
- Unter Exploration bzw. explorierendem Verhalten versteht man die Auswahl von unbekannten oder wenig bekannten Aktionen mit dem Ziel, möglichst viele neue Informationen über die Umgebung zu erlangen.
- Anmerkung: Typischerweise braucht ein Agent beides, ausbeutendes sowie explorierendes Verhalten.

Beispiel

Beispiele für Exploration vs. Exploitation

- Restaurantauswahl
    - Exploitation:

Beispiel

Beispiele für Exploration vs. Exploitation

- Restaurantauswahl
  - Exploitation: Lieblingsrestaurant besuchen
  - Exploration: ein neues Restaurant ausprobieren
- Online-Banner-Werbung
  - Exploitation:

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Beispiel

Beispiele für Exploration vs. Exploitation

- Restaurantauswahl
  - Exploitation: Lieblingsrestaurant besuchen
  - Exploration: ein neues Restaurant ausprobieren
- Online-Banner-Werbung
  - Exploitation: erfolgreichstes Banner anzeigen
  - Exploration: ein anderes, kreatives Banner anzeigen
- Ölbohrung
  - Exploitation:

Beispiel

Beispiele für Exploration vs. Exploitation

- Restaurantauswahl
  - Exploitation: Lieblingsrestaurant besuchen
  - Exploration: ein neues Restaurant ausprobieren
- Online-Banner-Werbung
  - Exploitation: erfolgreichstes Banner anzeigen
  - Exploration: ein anderes, kreatives Banner anzeigen
- Ölbohrung
  - Exploitation: an der besten bekannten Stelle bohren
  - Exploration: an einer neuen Stelle bohren
- Brettspiele
  - Exploitation:

Beispiel

Beispiele für Exploration vs. Exploitation

- Restaurantauswahl
  - Exploitation: Lieblingsrestaurant besuchen
  - Exploration: ein neues Restaurant ausprobieren
- Online-Banner-Werbung
  - Exploitation: erfolgreichstes Banner anzeigen
  - Exploration: ein anderes, kreatives Banner anzeigen
- Ölbohrung
  - Exploitation: an der besten bekannten Stelle bohren
  - Exploration: an einer neuen Stelle bohren
- Brettspiele
  - Exploitation: Zug machen, von dem man denkt, er sei der beste
  - Exploration: einen experimentellen Zug machen

Ausblick

- Diese Vorlesungseinheit hat (sehr viele) Grundbegriffe eingeführt, auf denen der Rest der Lehrveranstaltung aufbaut.
  - MDPs, Strategien, Horizont, Pfadkosten / Werfunktionen, Modell, Exploration, etc.
- In den folgenden Einheiten werden wir konkrete Verfahren (also Methoden und Algorithmen) kennenlernen, die auf der soweit kennengelernten Formalisierung aufsetzen (und mit denen man zum Teil sehr spannende Ergebnisse erzielen kann).
- Erster (nächster) Schritt: Ein Lernalgorithmus für Probleme mit endlichem (Entscheidungs-)Horizont.