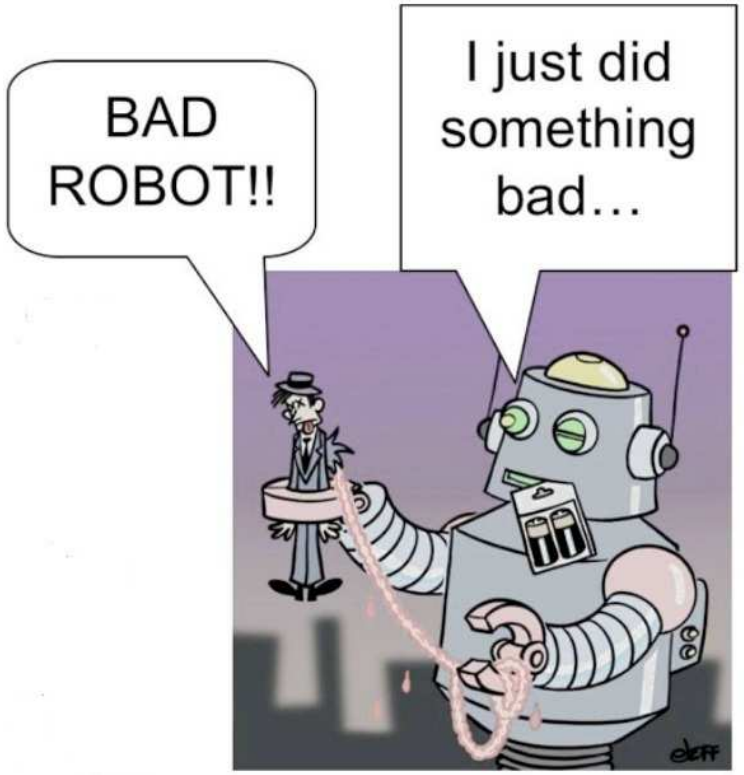
Combining Manual Feedback with Subsequent MDP Reward Signals for Reinforcement Learning

W. Bradley Knox und Peter Stone

TECHNISCHE UNIVERSITÄT DARMSTADT



Inhalt

TECHNISCHE UNIVERSITÄT DARMSTADT

- Problemstellung
- MDP
- TAMER
- Kombinationsmethoden
- Experimente
- Ergebnisse
- Fazit
- Quellen

Problemstellung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Lernende Agenten gehen mehr und mehr von der Forschung ins reale Leben über

Probleme:

1. Nicht alle Endbenutzer haben Fachkenntnisse
2. Manche Trainingsdaten sind kostspielig, erbringen aber keine gute Leistung

Nicht alle Endbenutzer haben Fachkenntnisse

TECHNISCHE UNIVERSITÄT DARMSTADT

- In realer Welt kann nicht jeder Mensch programmieren
- Soll dennoch in der Lage sein, einem Agenten ein gewünschtes Verhalten beizubringen

Kostspielige Trainingsdaten mit geringer Leistung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Sie sollen reduziert werden
- Wenn möglich komplett weglassen

Wichtige Begriffe

TECHNISCHE UNIVERSITÄT DARMSTADT

- Markov Decision Process
- TAMER

Wichtige Begriffe

TECHNISCHE UNIVERSITÄT DARMSTADT

- Markov Decision Process
- TAMER

Markov Decision Prozess

TECHNISCHE UNIVERSITÄT DARMSTADT

- MDP – Wofür ?
- MDP – Funktionsweise
- MDP Reward
- Eigenschaften des MDP

MDP – Wofür ?

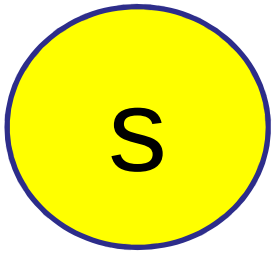
TECHNISCHE UNIVERSITÄT DARMSTADT

- Wahrscheinlichkeitsberechnung
- Schlägt nächsten Zustand vor

MDP – Funktionsweise

TECHNISCHE UNIVERSITÄT DARMSTADT

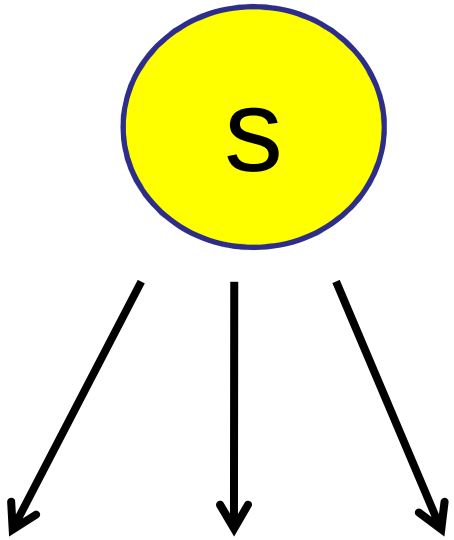
1. Zu jedem Zeitpunkt befindet sich Prozess in einem Zustand *s*



MDP – Funktionsweise

TECHNISCHE UNIVERSITÄT DARMSTADT

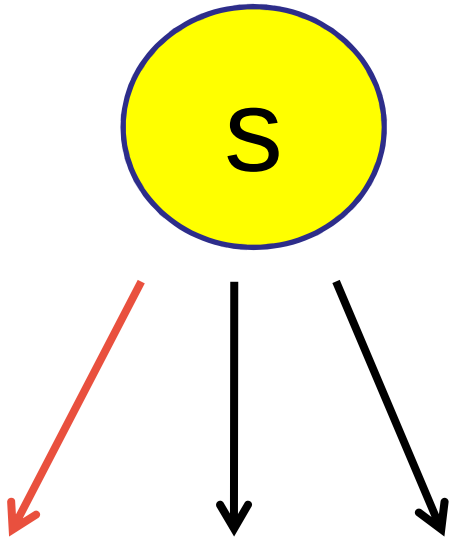
1. Zu jedem Zeitpunkt befindet sich Prozess in einem Zustand *s*
2. In Zustand *s* sind ein oder mehrere Aktionen verfügbar



MDP – Funktionsweise

TECHNISCHE UNIVERSITÄT DARMSTADT

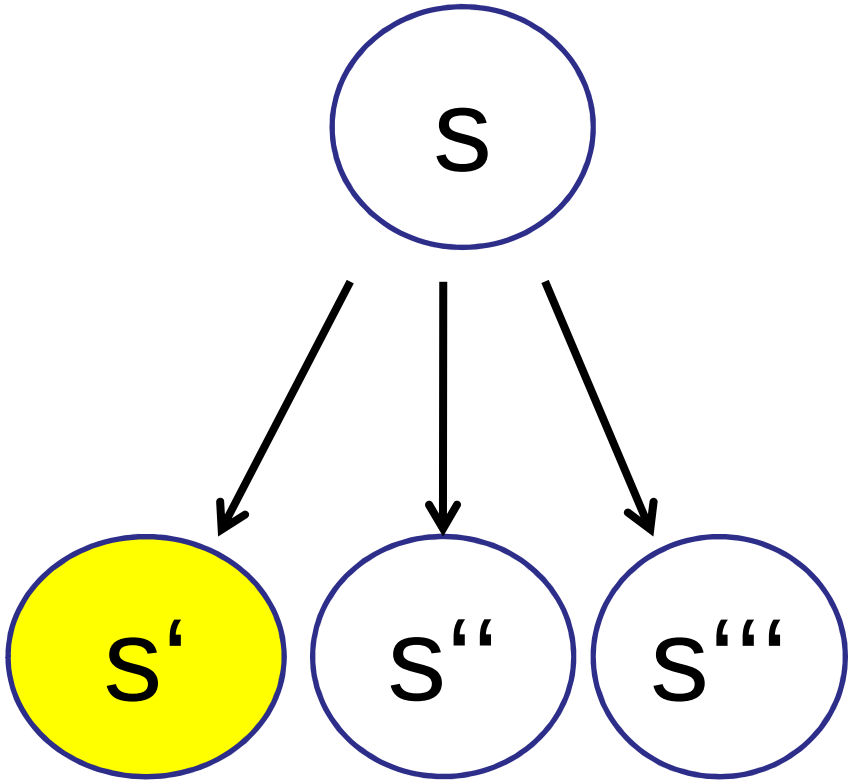
1. Zu jedem Zeitpunkt befindet sich Prozess in einem Zustand *s*
2. In Zustand *s* sind ein oder mehrere Aktionen verfügbar
3. Eine Aktion *a* wird gewählt



MDP – Funktionsweise

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Zu jedem Zeitpunkt befindet sich Prozess in einem Zustand *s*
2. In Zustand *s* sind ein oder mehrere Aktionen verfügbar
3. Eine Aktion *a* wird gewählt
4. Nachfolgende Zeitpunkt bewegt sich Prozess in Zustand *s‘* und liefert den dazugehörigen Reward



MDP Reward

TECHNISCHE UNIVERSITÄT DARMSTADT

- Als Signal genutzt
- Drückt Belohnung, von Zustand *s* in Zustand *s‘*, zu gelangen aus
- Fehlerlos
- Informationsarm
- z.T. Verzögert

Eigenschaften des MDP

TECHNISCHE UNIVERSITÄT DARMSTADT

- *s‘* abhängig von aktuellem Zustand *s* und Aktion *a*
- Unabhängig von davor gewählten Aktionen

Wichtige Begriffe

TECHNISCHE UNIVERSITÄT DARMSTADT

- Markov Decision Process
- TAMER

TAMER

TECHNISCHE UNIVERSITÄT DARMSTADT

- Was ist TAMER ?
- Wie funktioniert TAMER ?
- Bisherige Ergebnisse von TAMER
- Schlussfolgerungen

Was ist TAMER ?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Teaching Agents Manually via Evaluative Reinforcement
- Framework
- Agenten können interaktiv geformt werden

Wie funktioniert TAMER ?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Menschlicher Trainer gibt positives oder negatives Feedback Signal
- An Hand der Signale wird Human Reinforcement Function $\hat{H}$ erstellt
- TAMER wählt Aktion auf Grund von $\hat{H}$

Bisherige Beobachtungen von TAMER

TECHNISCHE UNIVERSITÄT DARMSTADT

- Sample Komplexität wird reduziert → Gewährleistung einer guten Policy
- Laien können Agenten zu gewünschtem Verhalten trainieren
- Agenten lernen in einem MDP ohne Reward Function

Überlegungen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Human Reinforcement Signal: **informationsreich** und **fehleranfällig**
- MDP Reward: **informationsarm** und **fehlerlos**

→Kombination des Human Reinforcement Signals und des MDP Reward

Problem bei der Umsetzung

TECHNISCHE UNIVERSITÄT DARMSTADT

Kombination von Reinforcement Algorithmen mit TAMER

→ TAMER ist mit Reward Function nicht kompatibel !

Kombinationsmethoden

TECHNISCHE UNIVERSITÄT DARMSTADT

Ziel:

- Etablierung von Kombinationsmethoden für TAMER und RL Algorithmen
- Erforschung verschiedener Wege, um sie umzusetzen

Kombinationsmethoden

TECHNISCHE UNIVERSITÄT DARMSTADT

- SARSA(λ) als Reinforcement Learning Algorithmus
- *Q* Function wird erstellt
- *Q* Function wird aktualisiert
- Q Function betrachtet zuvorgegangen Wege (Aktionssequenzen)

Kombinationsmethoden

TECHNISCHE UNIVERSITÄT DARMSTADT

- Aktion *a*
- Zustand *s*
- $\hat{H}(s,a)$
- $R(s,a)$
- $Q(s,a)$
- Parameter
  - *weight* – wird dekrementiert
  - *constant* – bleibt konstant

Methode 1

TECHNISCHE UNIVERSITÄT DARMSTADT

$$R'(s,a) = R(s,a) + (weight * \hat{H}(s,a))$$

- Reward Signal wird durch die Summe von sich und einem Anteil der Human Reinforcement Function ersetzt
- Anteil der Reinforcement Function nimmt ab → somit auch ihr Einfluss

→ Vorhersage des Menschen wird mit der Zeit unbedeutender

Methode 2

TECHNISCHE UNIVERSITÄT DARMSTADT

$$f' = f.append(\hat{H}(s,a))$$

- Annahme, dass die *Q* Function über einen Feature Vector erlernt wird
- Fügt zu sich selbst zusätzlich noch die Information der Human Reinforcement Function hinzu
- Information kann vom Reinforcement Learning Algorithmus, je nach Bedarf, mehr oder weniger genutzt werden

Methode 3

TECHNISCHE UNIVERSITÄT DARMSTADT

$$Q(s,a) \rightarrow (constant * \hat{H}(s,a))$$

- *Q* Function so trainieren, dass sie sich an einen Anteil von $\hat{H}(s,a)$ annähert
- *Q* wird mit 0 initialisiert
- $\hat{H}$ wird als *Q* Function behandelt

Methode 4

TECHNISCHE UNIVERSITÄT DARMSTADT

$$Q'(s,a) = Q(s,a) + constant * \hat{H}(s,a)$$

- Ähnlich wie Methode 1
- Statt MDP Reward wird *Q* Function benutzt
- $\hat{H}(s,a)$ nimmt nicht ab

Methode 5

TECHNISCHE UNIVERSITÄT DARMSTADT

$$A' = A \cup \operatorname{argmax}_a[\hat{H}(s,a)]$$

- Fügt zu der Menge der möglichen Aktionen die Aktion hinzu, welche der TAMER Agent wählen würde
- Mindestens zwei optimale Aktionen stehen zur Verfügung

Methode 6

TECHNISCHE UNIVERSITÄT DARMSTADT

$$a = argmax_a[Q(s,a) + weight * \hat{H}(s,a)]$$

- Ein Anteil der Human Reinforcement Function $\hat{H}$ wird zur Q Function während der Aktionsauswahl hinzugefügt
- Der Anteil der Human Reinforcement Function $\hat{H}$ nimmt mit der Zeit ab

Methode 7

TECHNISCHE UNIVERSITÄT DARMSTADT

$$P\,(a = argmax_a\,[\hat{H}(s,a)]) = p$$

- p ist eine fixe Wahrscheinlichkeit
- Entweder wählt Reinforcement Learning Algorithmus die nächste Aktion aus oder
- TAMER Agent wählt die nächste Aktion aus, während der Reinforcement Learning Algorithmus überwacht und aktualisiert
- Anfangs wird die Aktion von $\hat{H}$ gewählt
- Gegen Ende vom RL-Algorithmus

Methode 8

TECHNISCHE UNIVERSITÄT DARMSTADT

$$R'(s_t, a) = R(s,a) + constant * (\Phi(s_t) - \Phi(s_{t-1}))$$

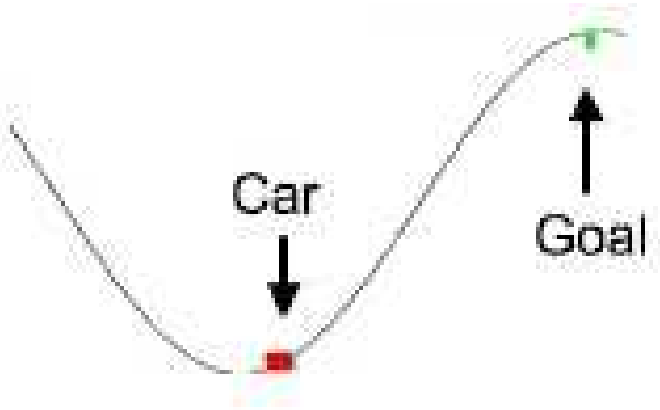
$$\Phi(s) = max_a H(s,a)$$

- *H* wird über Potentialfunktion *Φ* genutzt
- Potentialfunktion *Φ* gibt an, welches die optimale Aktion *a* im Zustand *s* ist
- Aktueller Zustand und Zustand aus Zeitschritt davor wird betrachtet
- Je größer die Abweichung desto größer der Einfluss

Experimente - Randbedingungen

TECHNISCHE UNIVERSITÄT DARMSTADT

- SARSA(λ) als Reinforcement Algorithmus
- Mountain Car Umgebung
- Zwei Human Reinforcement Functions
  - $\hat{H}_1$: durchschnittliche Funktion
  - $\hat{H}_2$: beste Funktion
- Zwei Parameter Sets
  - Optimistisch: Initialisierung von *Q = 0*
  - Pessimistisch: Initialisierung von *Q = -120*
- *constant* und *weight*
  - Auswahl der besten Parameter



Erfolgskriterien

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Höheres Level der Endperformanz der Kombinationstechnik ist größer als RL Algorithmus oder TAMER alleine

2. MDP Reward ist nach mehreren Lernepisoden (> 500) größer als RL Algorithmus oder TAMER alleine

1. Höhere Endperformanz

TECHNISCHE UNIVERSITÄT DARMSTADT

Optimistische Initialisierung

Nur mit Q=0

1. Höhere Endperformanz

TECHNISCHE UNIVERSITÄT DARMSTADT

Optimistische Initialisierung

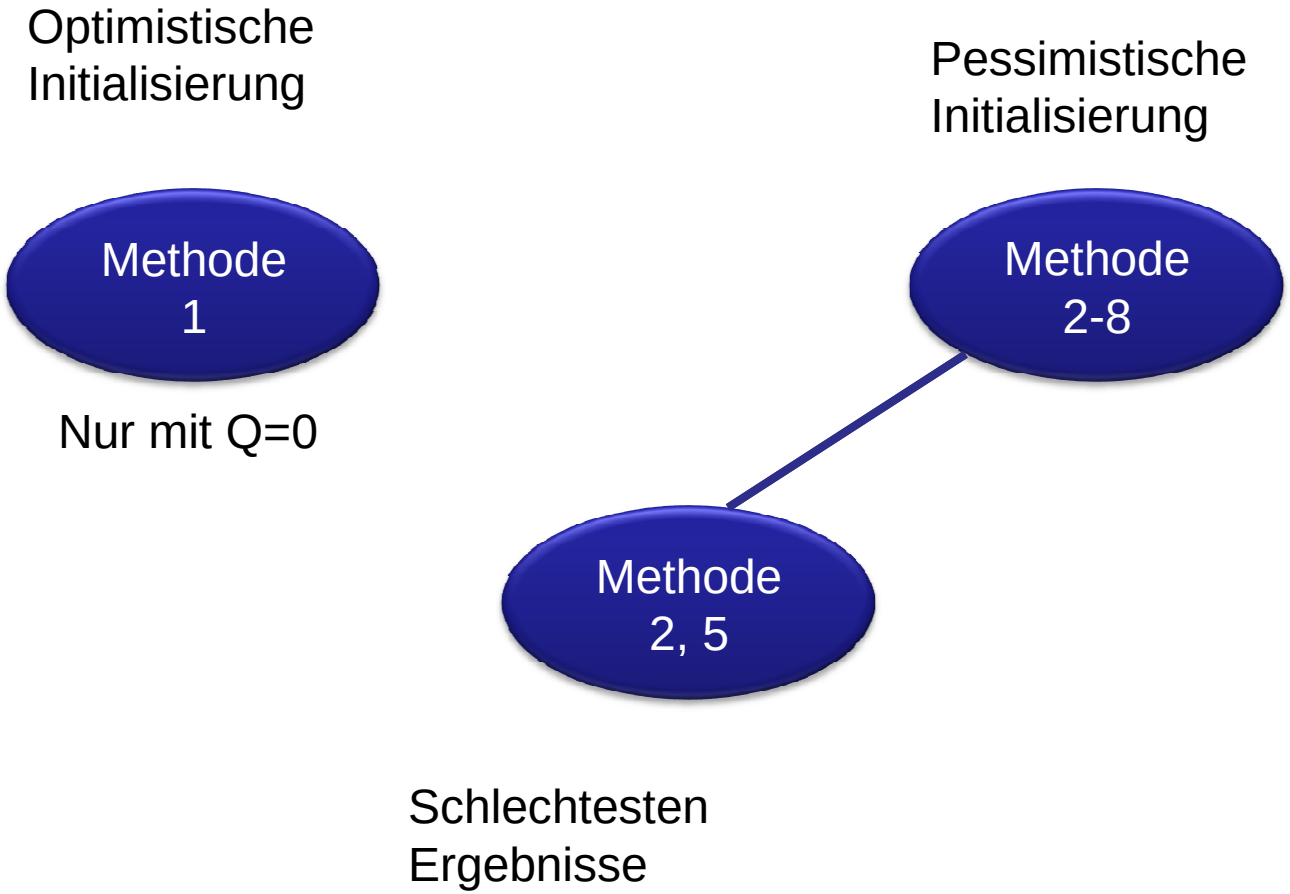
Methode 1

Nur mit Q=0

Pessimistische Initialisierung

Methode 2-8

1. Höhere Endperformanz



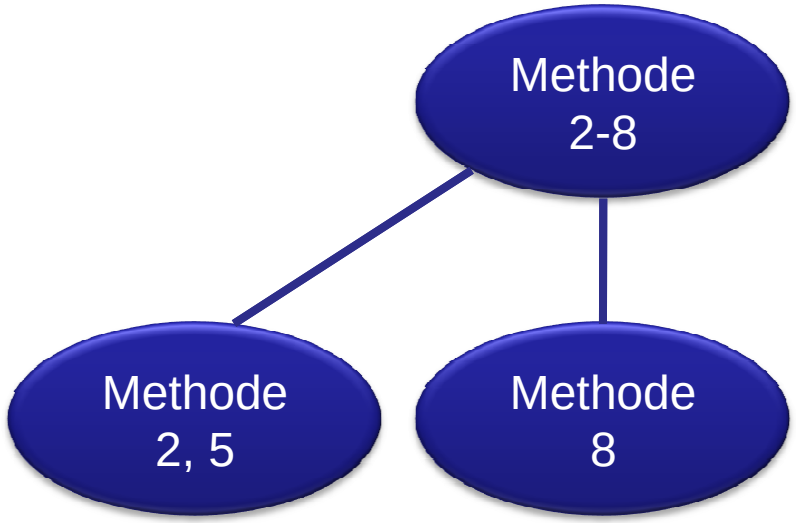
1. Höhere Endperformanz

TECHNISCHE UNIVERSITÄT DARMSTADT

Optimistische Initialisierung

Nur mit Q=0

Pessimistische Initialisierung



Schlechtesten Ergebnisse

Etwas verbessert

1. Höhere Endperformanz

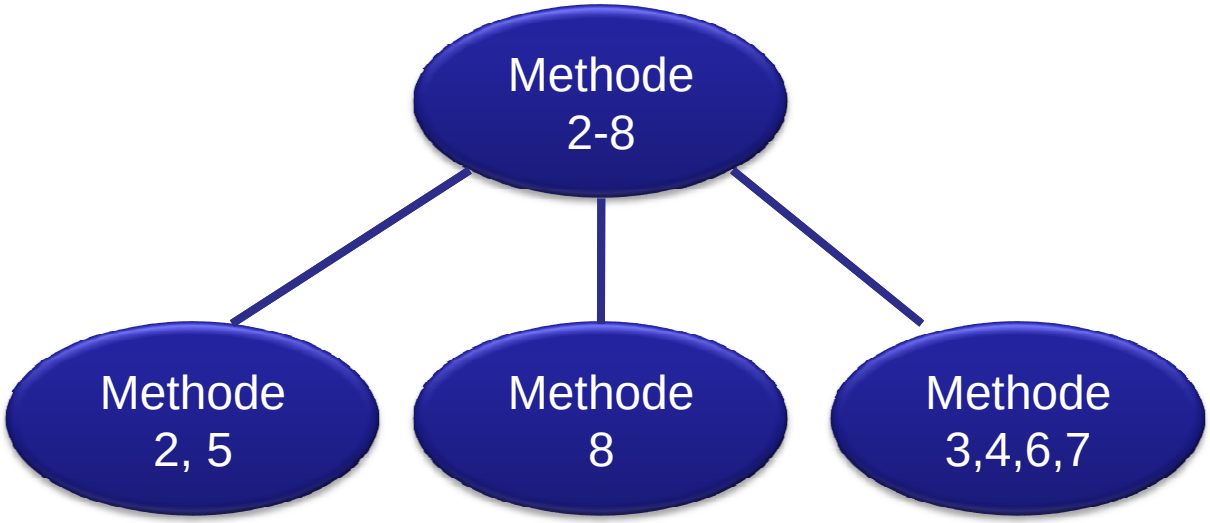
TECHNISCHE UNIVERSITÄT DARMSTADT

Optimistische Initialisierung

Methode 1

Nur mit Q=0

Pessimistische Initialisierung



Schlechtesten Ergebnisse

Etwas verbessert

Besten Ergebnisse

2. Höherer Reward

TECHNISCHE UNIVERSITÄT DARMSTADT

Verbesserungen mit beiden Initialisierungen

2. Höherer Reward

TECHNISCHE UNIVERSITÄT DARMSTADT

Verbesserungen mit beiden Initialisierungen

Verbessert sich auch etwas

2. Höherer Reward

TECHNISCHE UNIVERSITÄT DARMSTADT

Verbesserungen mit beiden Initialisierungen

Methode 1,6,7

Verbessert sich auch etwas

Methode 4

Verschlechterungen

Methode 3

Methode 8

Methode 5

Methode 2

- Nicht unbedingt besser als SARSA(λ)
- Geringe Verbesserung unter einer der beiden H Funktionen
- Schlechter als SARSA(λ) alleine

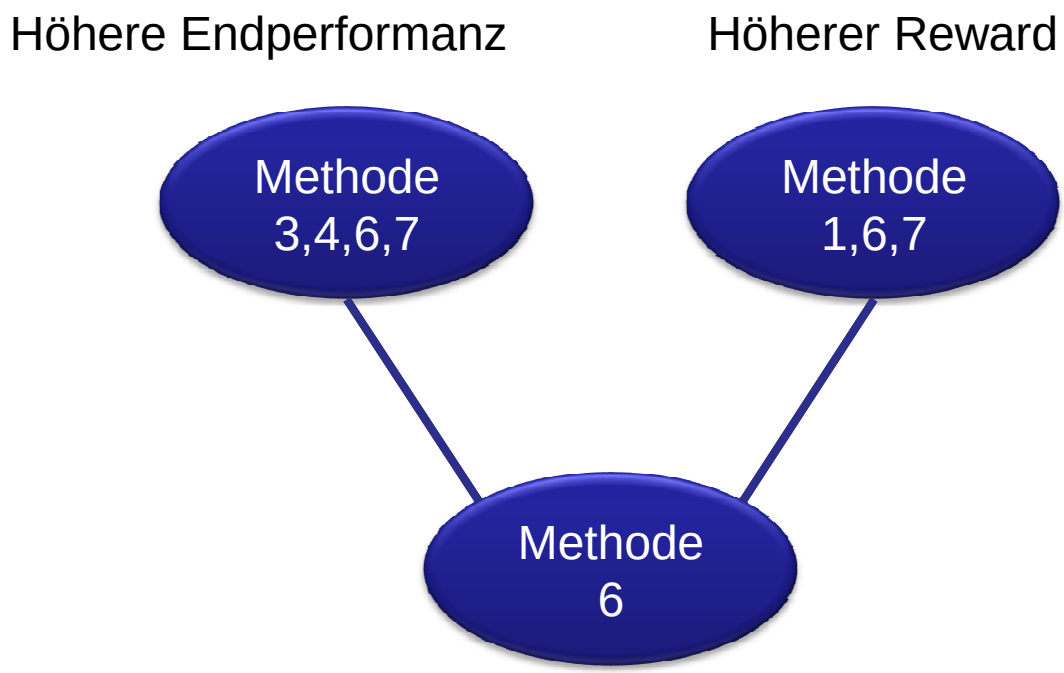
Beste Methode

TECHNISCHE UNIVERSITÄT DARMSTADT

Höhere Endperformanz

Höherer Reward

Beste Methode



Fazit

TECHNISCHE UNIVERSITÄT DARMSTADT

Zwei erkennbare Muster

1. Wird die Q Funktion am Anfang manipuliert, führt das zu einer schlechteren Performanz

Fazit

TECHNISCHE UNIVERSITÄT DARMSTADT

Zwei erkennbare Muster

1. Wird die Q Funktion am Anfang manipuliert, führt das zu einer schlechteren Performanz
2. Wenn der Agent sanft in eine Richtung geleitet wird und der Einfluss von $\hat{H}$ mit der Zeit reduziert wird, führt das zu einer guten Performanz

Ziel

TECHNISCHE UNIVERSITÄT DARMSTADT

- Etablierung von Kombinationsmethoden für TAMER und RL Algorithmen
- Erforschung verschiedener Wege, um sie umzusetzen

Frage

TECHNISCHE UNIVERSITÄT DARMSTADT

Ziel:

- Etablierung von Kombinationsmethoden für TAMER und RL Algorithmen
- Erforschung verschiedener Wege, um sie umzusetzen

Erreicht, ja oder nein ?

Erreicht, ja oder nein ?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Jein
- Guter Ansatz

Erreicht, ja oder nein ?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Jein
- Guter Ansatz
- Spiele (Schach, Backgammon, etc.) → ja

Erreicht, ja oder nein ?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Jein
- Guter Ansatz
- Spiele (Schach, Backgammon, etc.) → ja
- Erkennung (Gesichtsdetektion) → ja

Erreicht, ja oder nein ?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Jein
- Guter Ansatz
- Spiele (Schach, Backgammon, etc.) → ja
- Erkennung (Gesichtsdetektion) → ja
- Roboter (Operationen) → nein

Erreicht, ja oder nein ?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Jein
- Guter Ansatz
- Spiele (Schach, Backgammon, etc.) → ja
- Erkennung (Gesichtsdetektion) → ja
- Roboter (Operationen) → nein
- Für komplexe Anwendungen noch nicht genügend erforscht

Quellen

TECHNISCHE UNIVERSITÄT DARMSTADT

- http://en.wikipedia.org/wiki/Markov_decision_process
- http://en.wikipedia.org/wiki/Markov_property
- http://en.wikipedia.org/wiki/SARSA
- http://webdocs.cs.ualberta.ca/~sutton/book/ebook/node77.html
- http://www.cs.utexas.edu/~bradknox/papers/aamas10poster-knox.pdf