Robot Learning from Demonstration by Constructing Skill Trees

Seminar on Computational Intelligence

Manuel Zellhöfer

Machine Learning and Robotics Lab - Institut für Informatik

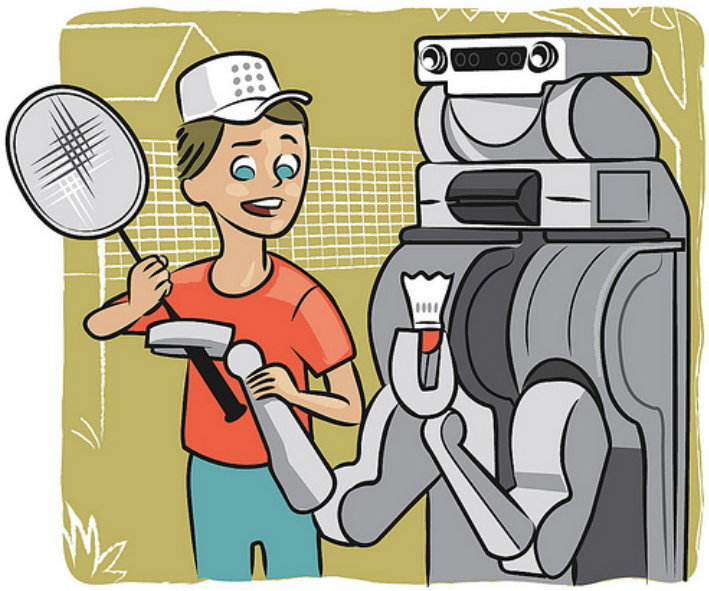
2. Juli 2012

Gliederung

Worum geht's?

Learning from Demonstration!

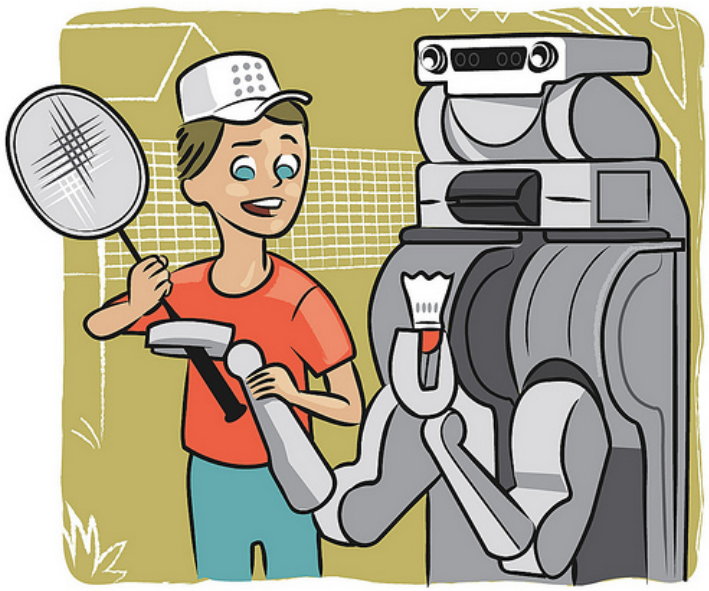
- Bringe einem Agenten anhand von Beispieltrajektorie(n) gewünschtes Verhalten bei
- Intuitives Programmieren von Robotern, auch für Laien...!
- *Imitation Learning*, eine Form von Reinforcement Learning
- Vermeidung von unnötiger Exploration



Worum geht's?

Learning from Demonstration!

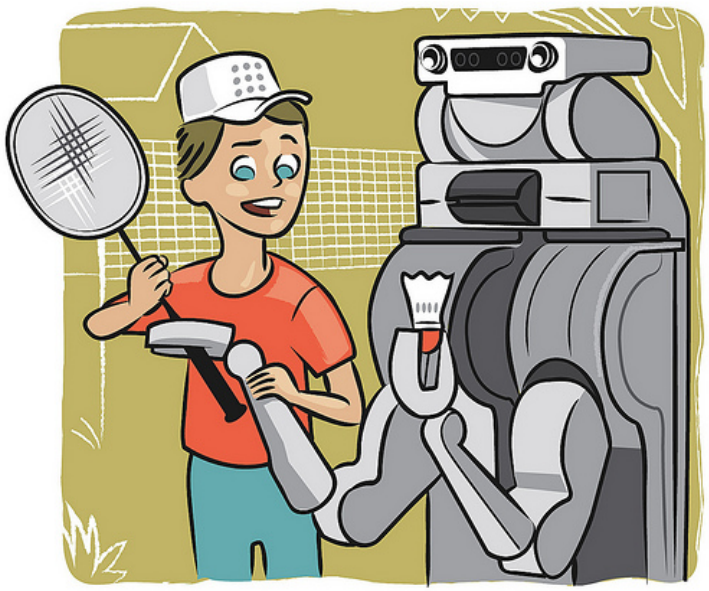
- Bringe einem Agenten anhand von Beispieltrajektorie(n) gewünschtes Verhalten bei
- Intuitives Programmieren von Robotern, auch für Laien...!
- *Imitation Learning*, eine Form von Reinforcement Learning
- Vermeidung von unnötiger Exploration



Worum geht's?

Learning from Demonstration!

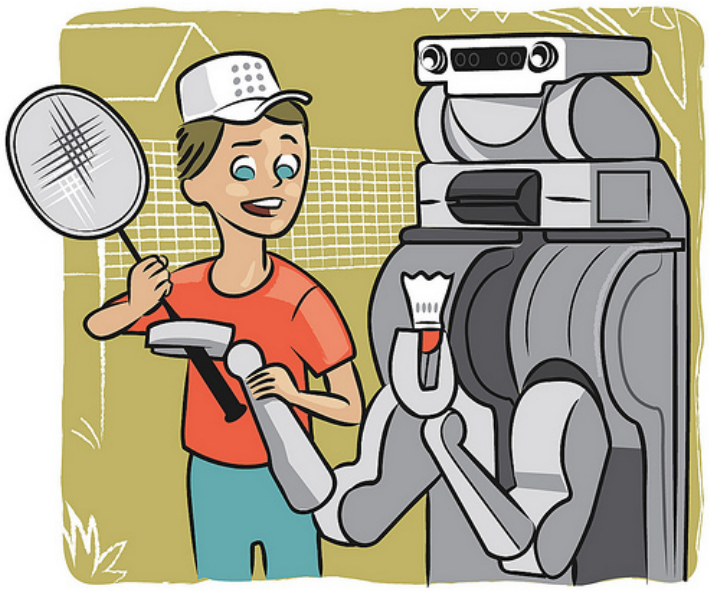
- Bringe einem Agenten anhand von Beispieltrajektorie(n) gewünschtes Verhalten bei
- Intuitives Programmieren von Robotern, auch für Laien...!
- *Imitation Learning*, eine Form von Reinforcement Learning
- Vermeidung von unnötiger Exploration



Worum geht's?

Learning from Demonstration!

- Bringe einem Agenten anhand von Beispieltrajektorie(n) gewünschtes Verhalten bei
- Intuitives Programmieren von Robotern, auch für Laien...!
- *Imitation Learning*, eine Form von Reinforcement Learning
- Vermeidung von unnötiger Exploration



Reinforcement Learning

- Löst Markov Decision Process...

  Zustände $S \subset \mathbb{R}^d$

  Aktionen $A \subset \mathbb{R}^n$

  Returns $R : S \times A \to \mathbb{R}$ (usw.)

  ... durch lernen einer Strategie (Policy)

  $$\pi : S \to A$$

- Maximieren des Erwarteten Returns
- Value-Function $V(s)$ bewertet jeden Zustand anhand der von ihm erreichbaren Returns $\rightarrow$ Formalismus um π zu berechnen
- Approximierung durch ein lineares Modell

$$\bar{V}(s) = \mathbf{w} \cdot \mathbf{\Phi}(s) = \sum_i w_i \phi_i(s)$$

Reinforcement Learning

- Löst Markov Decision Process...

  Zustände $S \subset \mathbb{R}^d$

  Aktionen $A \subset \mathbb{R}^n$

  Returns $R : S \times A \to \mathbb{R}$ (usw.)

  ... durch lernen einer Strategie (Policy)

  $$\pi : S \to A$$

- Maximieren des Erwarteten Returns
- Value-Function $V(s)$ bewertet jeden Zustand anhand der von ihm erreichbaren Returns $\rightarrow$ Formalismus um π zu berechnen
- Approximierung durch ein lineares Modell

$$\bar{V}(s) = \mathbf{w} \cdot \mathbf{\Phi}(s) = \sum_i w_i \phi_i(s)$$

Reinforcement Learning

- Löst Markov Decision Process...

  Zustände $S \subset \mathbb{R}^d$

  Aktionen $A \subset \mathbb{R}^n$

  Returns $R : S \times A \to \mathbb{R}$ (usw.)

  ... durch lernen einer Strategie (Policy)

  $$\pi : S \to A$$

- Maximieren des Erwarteten Returns
- Value-Function $V(s)$ bewertet jeden Zustand anhand der von ihm erreichbaren Returns $\rightarrow$ Formalismus um π zu berechnen
- Approximierung durch ein lineares Modell

$$\bar{V}(s) = \mathbf{w} \cdot \mathbf{\Phi}(s) = \sum_i w_i \phi_i(s)$$

Reinforcement Learning

- Löst Markov Decision Process...

  Zustände $S \subset \mathbb{R}^d$

  Aktionen $A \subset \mathbb{R}^n$

  Returns $R : S \times A \to \mathbb{R}$ (usw.)

  ... durch lernen einer Strategie (Policy)

  $$\pi : S \to A$$

- Maximieren des Erwarteten Returns
- Value-Function $V(s)$ bewertet jeden Zustand anhand der von ihm erreichbaren Returns $\rightarrow$ Formalismus um π zu berechnen
- Approximierung durch ein lineares Modell

$$\bar{V}(s) = \mathbf{w} \cdot \mathbf{\Phi}(s) = \sum_i w_i \phi_i(s)$$

Reinforcement Learning

- Löst Markov Decision Process...

  Zustände $S \subset \mathbb{R}^d$

  Aktionen $A \subset \mathbb{R}^n$

  Returns $R : S \times A \to \mathbb{R}$ (usw.)

  ... durch lernen einer Strategie (Policy)

  $$\pi : S \to A$$

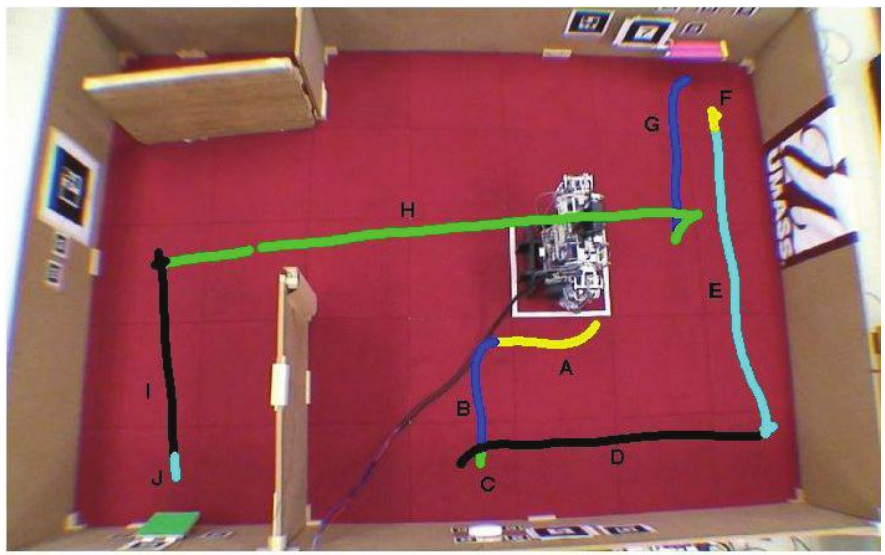
- Maximieren des Erwarteten Returns
- Value-Function $V(s)$ bewertet jeden Zustand anhand der von ihm erreichbaren Returns $\rightarrow$ Formalismus um π zu berechnen
- Approximierung durch ein lineares Modell

$$\bar{V}(s) = \mathbf{w} \cdot \mathbf{\Phi}(s) = \sum_i w_i \phi_i(s)$$

Reinforcement Learning

Schwierigkeiten und Herausforderungen

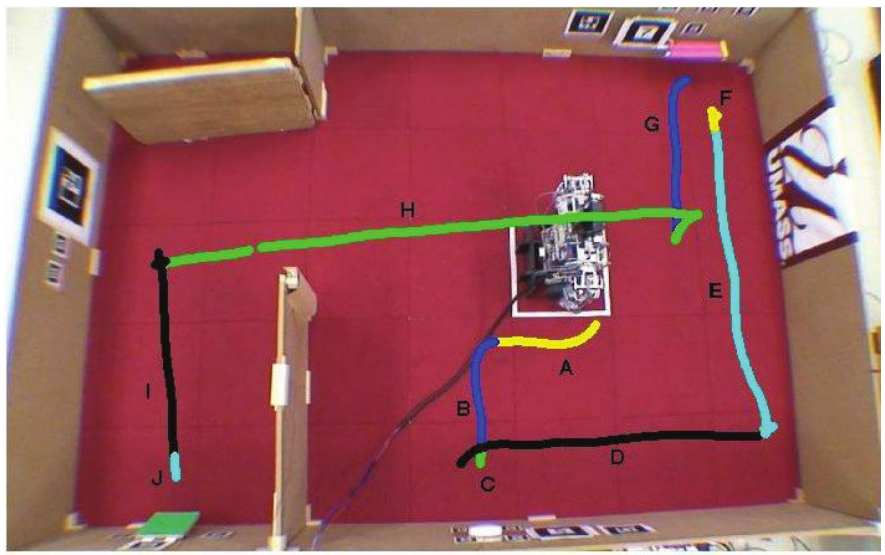
- kontinuierliche, hochdimensionale Zustände und Aktionen
- Komplexe Strategien
- Wiederverwenden und Verbessern von bereits Gelerntem



Reinforcement Learning

Schwierigkeiten und Herausforderungen

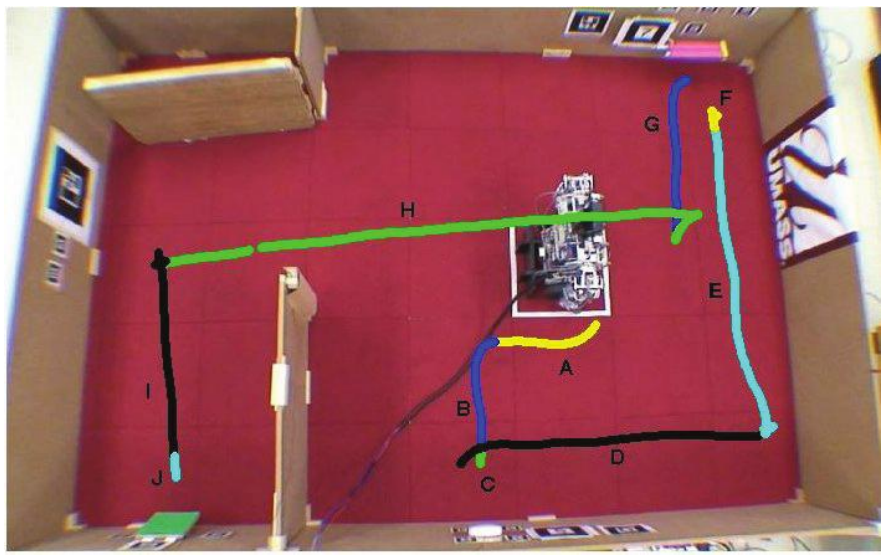
- kontinuierliche, hochdimensionale Zustände und Aktionen
- Komplexe Strategien
- Wiederverwenden und Verbessern von bereits Gelerntem



Reinforcement Learning

Schwierigkeiten und Herausforderungen

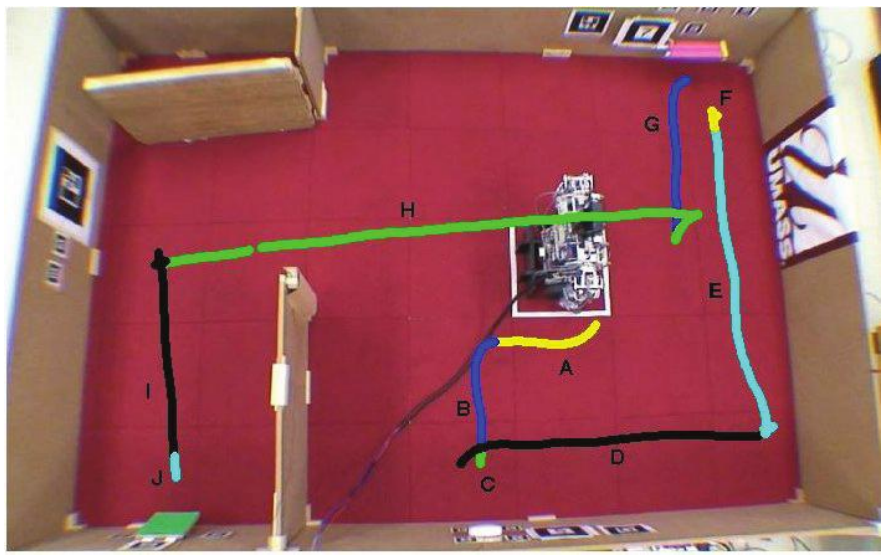
- kontinuierliche, hochdimensionale Zustände und Aktionen
- Komplexe Strategien
- Wiederverwenden und Verbessern von bereits Gelerntem



Reinforcement Learning

Schwierigkeiten und Herausforderungen

- kontinuierliche, hochdimensionale Zustände und Aktionen
- Komplexe Strategien
- Wiederverwenden und Verbessern von bereits Gelerntem



Gliederung

Vorgehen

- Komplexe RL Probleme lösen
- Dimensionalität reduzieren
- Beispieltrajektorien verwenden

Hierarchisches RL: Options

Was ist das?

- Aktionen mit eigener Strategie

Policy	π_o
Initiation Set	$I_o \subseteq S$
Termination Condition	$\beta_o(s)$

- Eigenständiges RL Problem
- Verwendet der Agent als zusätzliche Aktion

Was kann das?

- Aufteilen komplexer RL Probleme in mehrere, einfache
- Zeitlich fortdauernde Aktionen Beschreiben

Hierarchisches RL: Options

Was ist das?

- Aktionen mit eigener Strategie

Policy	π_o
Initiation Set	$I_o \subseteq S$
Termination Condition	$\beta_o(s)$

- Eigenständiges RL Problem
- Verwendet der Agent als zusätzliche Aktion

Was kann das?

- Aufteilen komplexer RL Probleme in mehrere, einfache
- Zeitlich fortdauernde Aktionen Beschreiben

Hierarchisches RL: Options

Was ist das?

- Aktionen mit eigener Strategie

Policy	π_o
Initiation Set	$I_o \subseteq S$
Termination Condition	$\beta_o(s)$

- Eigenständiges RL Problem
- Verwendet der Agent als zusätzliche Aktion

Was kann das?

- Aufteilen komplexer RL Probleme in mehrere, einfache
- Zeitlich fortdauernde Aktionen Beschreiben

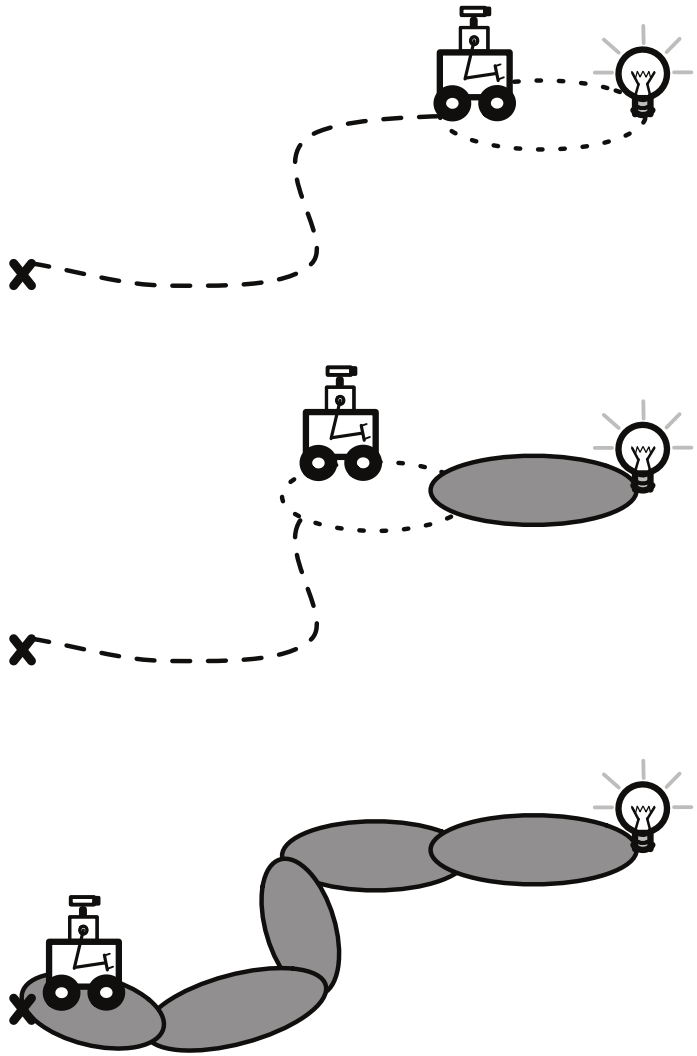
Skill Chaining

Löst ein RL Problem durch aufteilen in eine Kette von Skills:

1. Finde Ziel (hier die Lampe)
2. Erstelle Skill mit $\beta_o \widehat{=}$Ziel
3. Lerne I_o durch Trial & Error und Klassifikator

Bei erreichen des Initiation Sets (I_o) eines vorhandenen Skills:

- Setze Ziel für neuen Skill auf I_o des vorhandenen Skills
- Sehr viele Episoden nötig



Abstraction Selection

Abstraction

Abstraction M sind zwei Abbildungen (σ_M, τ_M)

$$\sigma_M : S \to S_M$$

$$\tau_M : A \to A_M$$

und Basisfunktionen $\mathbf{\Phi}_M$ um $V(s)$ zu approximieren.

- Geeignete Abstractions werden vorher definiert
- *in der Regel:* $S_M \subset S$ bzw. $A_M \subset A$
- *Idee:* Jeder Skill hat seine eigene Abstraction

Abstraction Selection

- Welche Abstraction passt zu welchem Skill?

Abstraction Selection

Abstraction

Abstraction M sind zwei Abbildungen (σ_M, τ_M)

$$\sigma_M : S \to S_M$$

$$\tau_M : A \to A_M$$

und Basisfunktionen $\mathbf{\Phi}_M$ um $V(s)$ zu approximieren.

- Geeignete Abstractions werden vorher definiert
- *in der Regel:* $S_M \subset S$ bzw. $A_M \subset A$
- *Idee:* Jeder Skill hat seine eigene Abstraction

Abstraction Selection

- Welche Abstraction passt zu welchem Skill?

Abstraction Selection

Abstraction

Abstraction M sind zwei Abbildungen (σ_M, τ_M)

$$\sigma_M : S \to S_M$$

$$\tau_M : A \to A_M$$

und Basisfunktionen $\mathbf{\Phi}_M$ um $V(s)$ zu approximieren.

- Geeignete Abstractions werden vorher definiert
- *in der Regel:* $S_M \subset S$ bzw. $A_M \subset A$
- *Idee:* Jeder Skill hat seine eigene Abstraction

Abstraction Selection

- Welche Abstraction passt zu welchem Skill?

Abstraction Selection

Abstraction

Abstraction M sind zwei Abbildungen (σ_M, τ_M)

$$\sigma_M : S \to S_M$$

$$\tau_M : A \to A_M$$

und Basisfunktionen $\mathbf{\Phi}_M$ um $V(s)$ zu approximieren.

- Geeignete Abstractions werden vorher definiert
- *in der Regel:* $S_M \subset S$ bzw. $A_M \subset A$
- *Idee:* Jeder Skill hat seine eigene Abstraction

Abstraction Selection

- Welche Abstraction passt zu welchem Skill?

Abstraction Selection

Abstraction

Abstraction M sind zwei Abbildungen (σ_M, τ_M)

$$\sigma_M : S \to S_M$$

$$\tau_M : A \to A_M$$

und Basisfunktionen $\mathbf{\Phi}_M$ um $V(s)$ zu approximieren.

- Geeignete Abstractions werden vorher definiert
- *in der Regel:* $S_M \subset S$ bzw. $A_M \subset A$
- *Idee:* Jeder Skill hat seine eigene Abstraction

Abstraction Selection

- Welche Abstraction passt zu welchem Skill?

... from Demonstration?

- Anstelle Skills mühsam zu erlernen werden Beispieltrajektorien segmentiert
- „Changepoints" in der Trajektorie stellen Start- / Zielmengen einzelner Skills dar

Verwenden eines Hidden Markov Models

- Versteckte Zustände entsprechen den linearen Modellen (Φ_M) der Abstractions
- Übergangswahrscheinlichkeiten hängen ab von
  - der Länge des aktuellen Segments
  - der a-priori Wahrscheinlichkeit für das Modell
- Ausgabewahrscheinlichkeiten durch Fitten der Modelle (Regression)

... from Demonstration?

- Anstelle Skills mühsam zu erlernen werden Beispieltrajektorien segmentiert
- „Changepoints" in der Trajektorie stellen Start- / Zielmengen einzelner Skills dar

Verwenden eines Hidden Markov Models

- Versteckte Zustände entsprechen den linearen Modellen (Φ_M) der Abstractions
- Übergangswahrscheinlichkeiten hängen ab von
  - der Länge des aktuellen Segments
  - der a-priori Wahrscheinlichkeit für das Modell
- Ausgabewahrscheinlichkeiten durch Fitten der Modelle (Regression)

... from Demonstration?

- Anstelle Skills mühsam zu erlernen werden Beispieltrajektorien segmentiert
- ,,Changepoints" in der Trajektorie stellen Start- / Zielmengen einzelner Skills dar

Verwenden eines Hidden Markov Models

- Versteckte Zustände entsprechen den linearen Modellen (Φ_M) der Abstractions
- Übergangswahrscheinlichkeiten hängen ab von
  - der Länge des aktuellen Segments
  - der a-priori Wahrscheinlichkeit für das Modell
- Ausgabewahrscheinlichkeiten durch Fitten der Modelle (Regression)

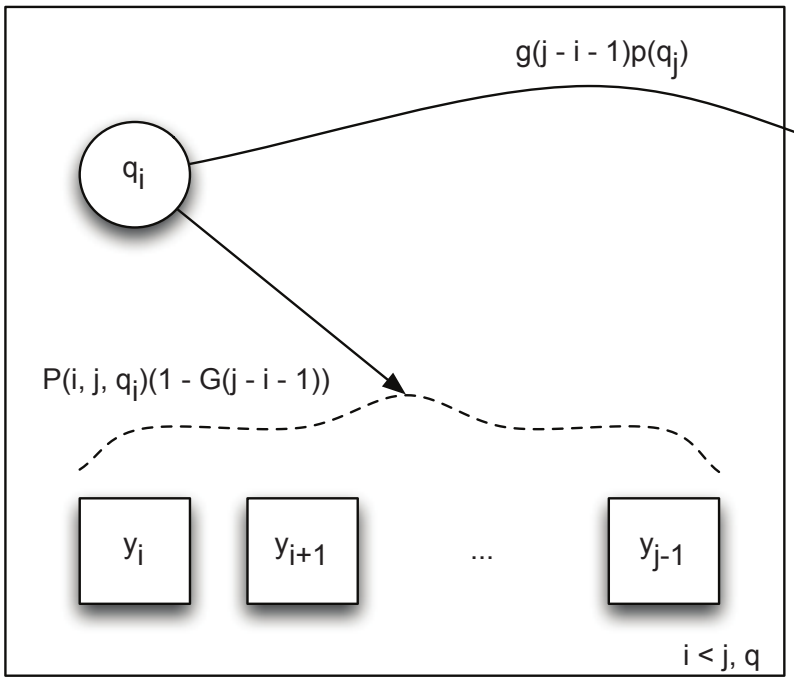
... from Demonstration?

- Anstelle Skills mühsam zu erlernen werden Beispieltrajektorien segmentiert
- ,,Changepoints" in der Trajektorie stellen Start- / Zielmengen einzelner Skills dar

Verwenden eines Hidden Markov Models

- Versteckte Zustände entsprechen den linearen Modellen ($\mathbf{\Phi}_M$) der Abstractions
- Übergangswahrscheinlichkeiten hängen ab von
  - der Länge des aktuellen Segments
  - der a-priori Wahrscheinlichkeit für das Modell
- Ausgabewahrscheinlichkeiten durch Fitten der Modelle (Regression)

HMM zur Changepoint Detection

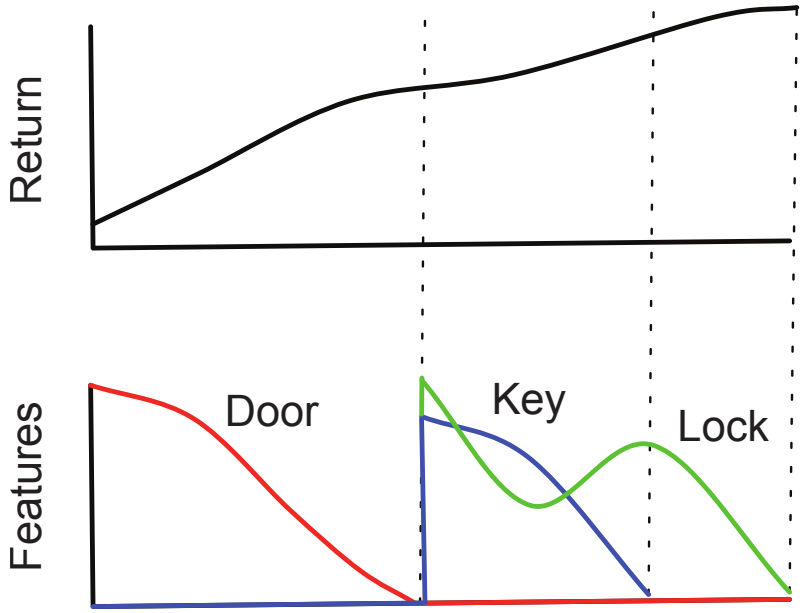


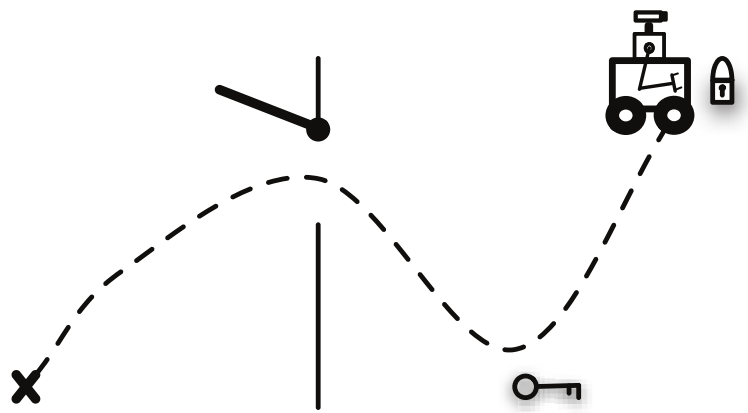
- $p(q)$ = a-priori Wsk. für q
- $g(l) = P(\text{,,Länge} = l\text{''})$
- $G(l) = P(\text{,,Länge} \leq l\text{''})$
- $P(i, j, q)$ = Fit-Wsk. für Modell q

On-Line Viterbi Algorithmus:

- Wsk. für Changepoint zum Zeitpunkt $j < t$ mit Modell q
- Viele Kombinationen j,$q \rightarrow$ Partikelfilter
- Iterativer Algorithmus

Beispiel – Changepoint Detection & Abstraction Selection





- $y_i = R_t$

3 Abstractions:

- $d(\text{Roboter,Tür})$
- $d(\text{Roboter,Schlüssel})$
- $d(\text{Roboter,Schloss})$

Approximieren des Returns mit einer einzelnen Abstraction je Segment

Ein ausgeklügelter Ansatz

Options Framework / Skill-Chaining

- Hierarchisches Reinforcement Learning
- Ermöglicht komplexe Strategien

Abstraction Selection

- „Feature Selection"
- Verringert die berücksichtigten Zustände/Aktionen

On-Line Changepoint Detection

- „Live" Skill-Chaining
- Beschleunigt lernen (LfD) deutlich

Ein ausgeklügelter Ansatz

Options Framework / Skill-Chaining

- Hierarchisches Reinforcement Learning
- Ermöglicht komplexe Strategien

Abstraction Selection

- „Feature Selection"
- Verringert die berücksichtigten Zustände/Aktionen

On-Line Changepoint Detection

- „Live" Skill-Chaining
- Beschleunigt lernen (LfD) deutlich

Ein ausgeklügelter Ansatz

Options Framework / Skill-Chaining

- Hierarchisches Reinforcement Learning
- Ermöglicht komplexe Strategien

Abstraction Selection

- „Feature Selection"
- Verringert die berücksichtigten Zustände/Aktionen

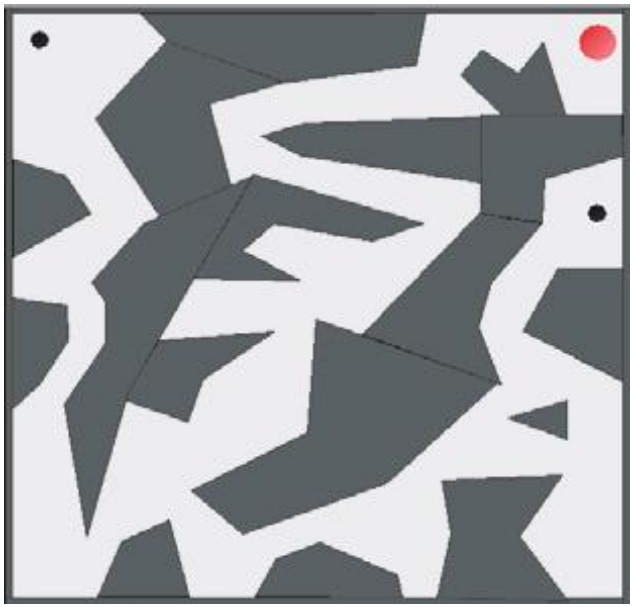
On-Line Changepoint Detection

- „Live" Skill-Chaining
- Beschleunigt lernen (LfD) deutlich

Gliederung

Pinball Domain

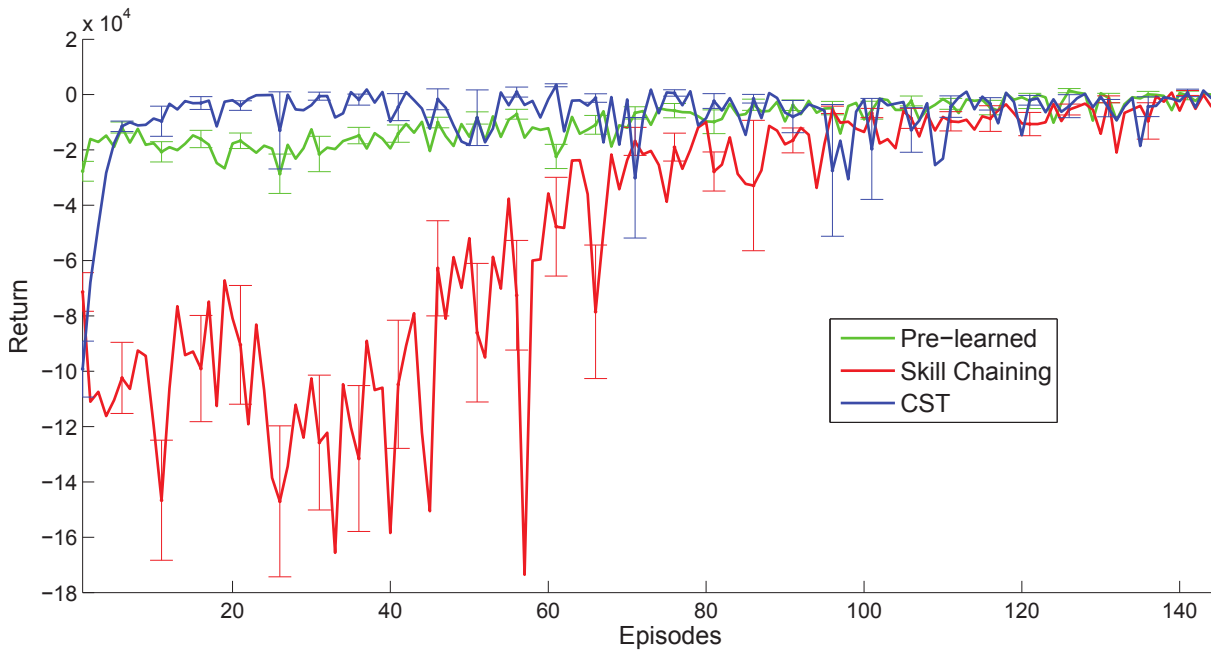
- 4 Zustände $x, y, \dot{x}, \dot{y}$
- 5 Aktionen
  - $\dot{x}, \dot{y}$ erhöhen, verringern $(r = -5)$
  - nichts tun $(r = -1)$
- Ziel erreichen: $r = 10000$
- 2 Demonstrationen



Vergleich:

- einfaches Skill Chaining
- Agent mit vorgegebenen Skills
- Constructing Skill Trees

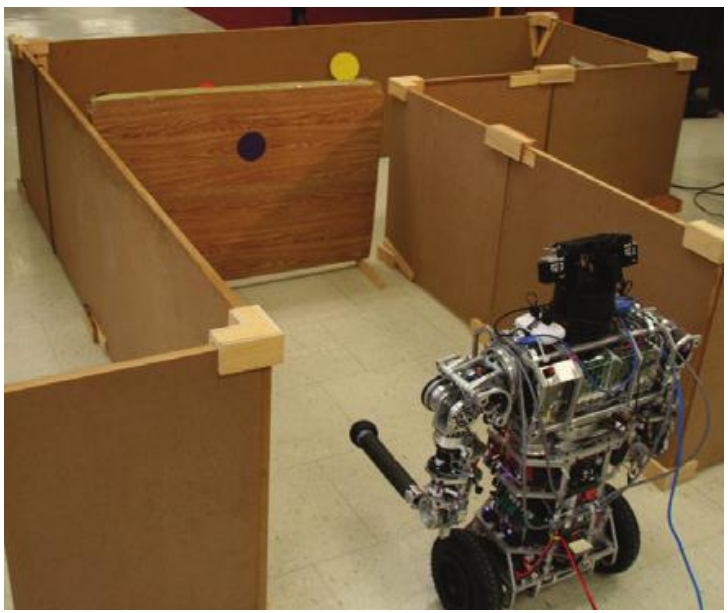
Pinball Domain, Ergebnis



Pinball Domain, Video

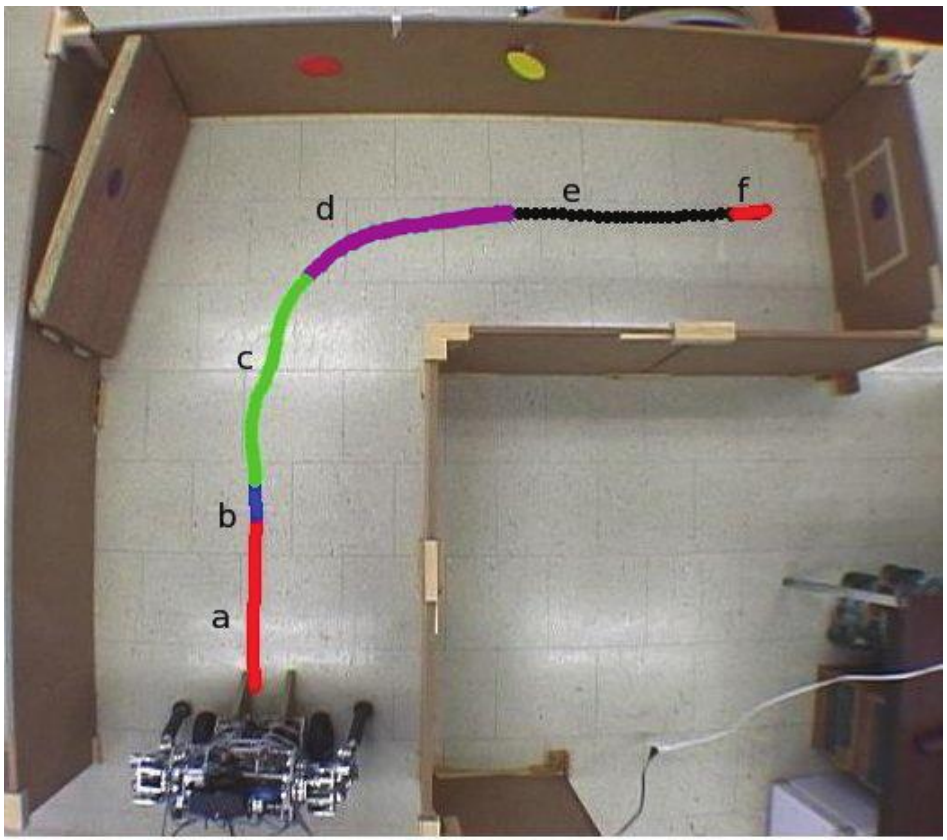
Mobile Robot Manipulation

- 6 Abstractions:
- Zustände:
  a) d(Hand,3 Marker) $\in \mathbb{R}^3$
  b) d(Torso,3 Marker) $\in \mathbb{R}^2$
- Aktionen:
  a) Endeffektor Position ($\in \mathbb{R}^3$)
  b) Vorwärtsgeschwindigkeit und Winkel ($\in \mathbb{R}^2$)
     - $r = -1$
- 12 Demonstrationen



Mobile Robot Manipulation, Ergebnis

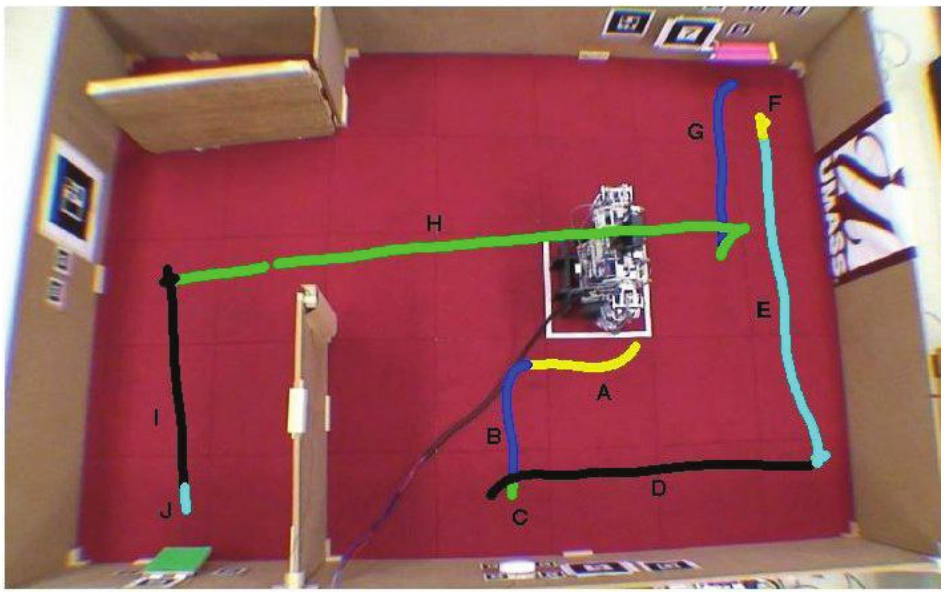
- 6 gelernte Skills
- 2-3 Demonstrationen nötig um jeweiligen Skill fehlerfrei auszuführen



Mobile Robot Manipulation, Video

Mobile Robot Manipulation, cont.

- Komplexere Aufgabe
  1. Knopf drücken um Hebel zu aktivieren
  2. Hebel drücken um Tür zu öffnen
  3. Schalter drücken
- Vorgegebene Aktionen
- RL um optimale Strategie zu finden



Dann:

- Trajektorien extrahiert und CST angewendet
- Vergleich der Vorgegeben Aktionen mit den gelernten Skills

Fazit

- Komplexes Gesamtsystem, Robotik auf hohem Niveau
- Viele moderne Konzepte aus maschinellem Lernen
- Faszinierend!
- Noch ein Weiter weg...

Fazit

- Komplexes Gesamtsystem, Robotik auf hohem Niveau
- Viele moderne Konzepte aus maschinellem Lernen
- Faszinierend!
- Noch ein Weiter weg...

Fazit

- Komplexes Gesamtsystem, Robotik auf hohem Niveau
- Viele moderne Konzepte aus maschinellem Lernen
- Faszinierend!
- Noch ein Weiter weg...

Fazit

- Komplexes Gesamtsystem, Robotik auf hohem Niveau
- Viele moderne Konzepte aus maschinellem Lernen
- Faszinierend!
- Noch ein Weiter weg...

Vielen Dank!

Quellen

G.D. Konidaris, S.R. Kuindersma, R.A. Grupen and A.G. Barto. *Robot Learning from Demonstration by Constructing Skill Trees*. The International Journal of Robotics Research 31(3), pages 360-375, March 2012.

`http://people.csail.mit.edu/gdk/arsa.html`

`http://flickr.com/photos/willowgarage/sets/72157624356302313/`