Universität Bremen

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

7. Juni 2010

Vorschlag für eine Masterarbeit im Studiengang Systems Engineering

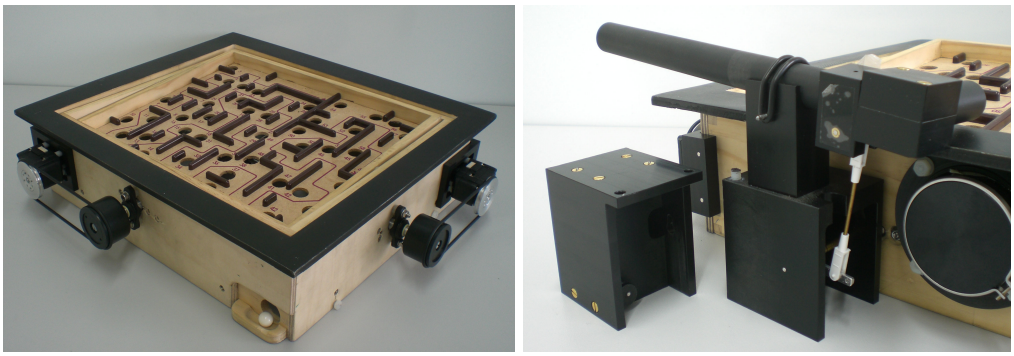
Reinforcement Learning mit Wissenstransfer am BRIO Labyrinth

Fabian Müller
mailto:fmueller@informatik.uni-bremen.de

Betreuer:
Dipl. Inf. Jan Hendrik Metzen
mailto:jhm@informatik.uni-bremen.de
M. Sc. Constantin Bergatt
mailto:constantin.bergatt@dfki.de

Gutachter:
Prof. Dr. rer. nat. Frank Kirchner
mailto:frank.kirchner@dfki.de

1 Motivation



(a) Servomotoren zur Steuerung der Brettneigung

(b) Magazin zur automatischen Kugelrückführung

Abbildung 1: modifiziertes BRIO Labyrinth als Testbed für autonome Lernverfahren

Das BRIO Labyrinth ist ein Geschicklichkeitsspiel, bei dem eine Kugel durch Neigen und Kippen des Spielbrettes durch ein Labyrinth aus Wänden und Löchern gerollt werden soll. Falls die Kugel auf dem Weg ins Ziel in eines der Löcher fällt, muss von vorne begonnen werden. Der Anstellwinkel des Spielbrettes (Brettwinkel) ist mit Hilfe zweier Drehknöpfe vom Spieler direkt einstellbar.

Das Spiel wurde, wie auch in [5] beschrieben, in vorangegangenen Arbeiten bereits häufiger am DFKI thematisiert. So hat Larbi Abdenebaoui im Rahmen seiner Diplom-

arbeit [1] bereits eine Simulation des Spiels entwickelt, um einen Reinforcement Learning Agenten darin zu trainieren. Der nächste Schritt in Richtung Automatisierung des realen Spiels wurde in dem Projekt *Labyrinth 2* (vgl. [4]) gemacht. Dort wurde ein Spielexemplar unter anderem mit einer Kamera zur Lokalisation der Kugel auf dem Spielbrett ausgestattet. Es wurden auch Servomotoren zur Steuerung der Brettwinkel und ein Ballmagazin zum automatischen Nachlegen einer neuen Kugel im Falle des Verlustes der alten integriert (vgl. Abbildung 1(a) und 1(b)). Dieser Aufbau – im Folgenden *Testbed* genannt – kann damit als Plattform für autonome Lernverfahren dienen.

Aufgrund der Größe des (kontinuierlichen) Zustandsraumes des Systems (Kugelposition auf Brett x, y, Kugelgeschwindigkeit $\dot{x}, \dot{y}$, Brettwinkel ϕ_x, ϕ_y), stellt es für einen autonom lernenden Agenten eine herausfordernde Zielsetzung dar, ein erfolgreiches Steuerungskonzept zu entwickeln. Nicht nur im vorliegenden Anwendungsfall gilt, dass ausführliche Explorationsphasen besonders auf dem realen Robotersystem extrem zeit- und ressourcenaufwändig sind (vgl. *curse of dimensionality*). Daher ist es erstrebenswert Teile der Lernarbeit durch Vorwissen zu ersetzen. Dieses Vorwissen kann zum Beispiel durch – dem Einsatz im Testbed vorgeschaltetes – Lernen in einer simulierten Umgebung erarbeitet und dann auf das reale System transferiert werden.

Dieses Vorhaben behindernd ist das sogenannte *Simulation-Reality-Gap*, die Constantin Bergatt in seiner Masterarbeit [3] behandelt hat. Da die Simulation die Gegegebenheiten des Testbed nicht in allen Details korrekt abbildet, lässt sich die vom Agenten erlernte Policy unter Umständen nicht ohne Weiteres auf das reale System übertragen. Bei endlicher Lerndauer lässt sich selbst in der Simulation nicht vermeiden, dass weite Teile des Zustandsraumes nicht ausreichend exploriert werden. Gelangt der Agent aufgrund der Abweichungen des Dynamikmodells des Testbeds doch in diese Bereiche, muss der Agent explorativ vorgehen um den richtigen Weg wiederzufinden. Sein Handeln bleibt dabei sub-optimal. Um dieses Problem anzugehen, wurde das der Simulation zugrunde liegende Modell weiter an das reale System angepasst.

2 Transfer Learning

Bisher nicht weiter untersucht wurde jedoch der eigentliche Transfer von erlerntem Wissen zwischen den beiden Systemen. Die Wiederverwendung des Wissens von Reinforcement Learning Agenten ist ein aktives Forschungsgebiet, das in den letzten Jahren viele verschiedene Methoden für unterschiedlichste Anwendungsfälle hervorgebracht hat. Eine umfangreiche Übersicht bietet [6].

In [2] wird Transfer Learning für einen Reinforcement Learning Agenten auf dem humanoiden Nao Roboter[1] eingesetzt. Darin soll der Agent den, an einer bestimmten Position liegenden Ball mit einer Hand in bestimmtem Winkel möglichst weit abstoßen. Bei einem der dort beschriebenen Wege, wird die in der Simulation gelernte Zustands-Aktionswertfunktion als Initialisierung auf den realen Roboter übertragen. Dort zeigt sich gegenüber dem einfachen Lernen auf dem realen System ein deutlicher Vorteil.

Neben dieser Methode sind auch viele weitere Ansätze des Transfers möglich. So ist zum Beispiel auch der Transfer von (s, a, r, s')-Trajektorien, (Teil-)Policy, Umgebungsmodell oder abstrakten „Richtlinien“ möglich. Darauf ausgerichtet ist die Anwendung von passenden Lernverfahren (Q-Learning, Sarsa(λ), Dyna-Q, Fitted R-max etc.) denkbar.

Auch gibt es im Bereich des Transfer Learning eigene Evaluationsmetriken, die speziell den durch den Transfer erzielten Gewinn quantifizieren sollen. Neben dem einfachen Qualitätsvergleich der Policies gegen die das jeweilige Verfahren konvergiert (*Asymptotic Performance*), ist auch die Zeit bis zum Erreichen eines bestimmten Performance-Schwellwerts (*Time-to-Threshold*) oder der Vorsprung in den ersten Episoden der Lernphase (*Jumpstart*) sehr relevant, wobei wirksamer Transfer oft deutliche Vorteile zeigt.

3 Ziel der Arbeit

Der Fokus der Arbeit soll auf der Anwendung und Evaluation von Techniken zum Transfer vom gelernten Wissen eines Reinforcement Learning Agenten liegen. Dies soll am Beispiel des BRIO Labyrinth Spiels geschehen und dem Zweck dienen, den Lernaufwand auf dem realen System *Testbed* zu minimieren. Hierzu soll zunächst ein Reinforcement Learning Agent in der Simulation des Spiels trainiert werden. Daran anschließend soll das Wissen, das dieser über die Umgebung gelernt hat, mit Hilfe von Transfer Learning in die Realität, bzw. auf den Lerner am Testbed übertragen werden. Beide Systeme bieten eine Schnittstelle zum MMLF[2]. Damit kann bereits auf dessen Funktionsumfang zurückgegriffen werden. Abbildung 2 zeigt die gesamte Prozedur schematisch.

Als Arbeitspakete ergeben sich folgende Punkte.

1. Mit dem MMLF vertraut machen, geeignete Lernverfahren auswählen und ggfs. implementieren. Dabei sollte schon auf die Anforderungen der einzusetzenden Transfertechniken geachtet werden (*was* soll transferiert werden). Eventuell ist auch eine

[1] `http://www.aldebaran-robotics.com`
[2] *Maja Machine Learning Framework*, `http://mmlf.sourcefourge.net`

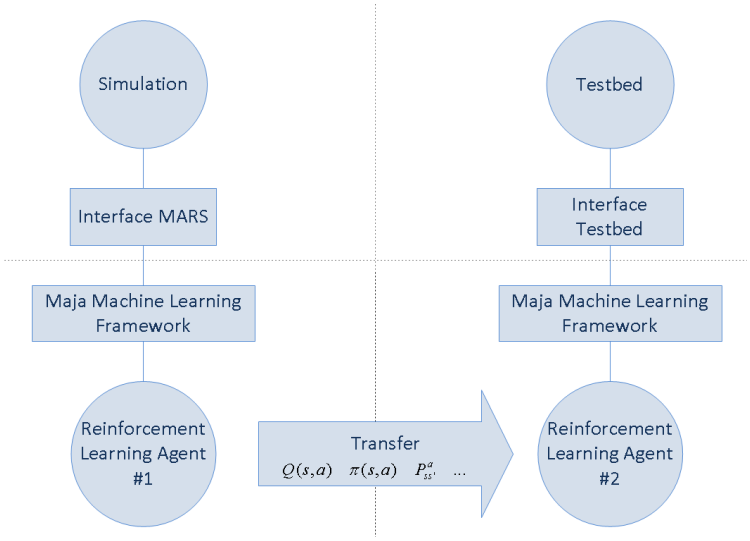


Abbildung 2: Schema Wissenstransfer

Kombination verschiedener Lerner sinnvoll (z.B. unterschiedliche Auswahl für Simulation/Testbed).

2. Auswahl und Implementation der anzuwendenden Transferverfahren. Zur Auswahl stehen verschiedene Verfahren. Zum Beispiel die einfache Verwendung der Zustands-Aktionswertfunktion um die Exploration des Lerners in die gewünschte Richtung zu lenken. Bis hin zu komplexeren Verfahren, die aus transferierten Trajektorien Handlungsrichtlinien ableiten, auf die dann im Verlauf der Exploration ebenfalls zurückgegriffen werden kann.

3. Mit den passenden statistischen Verfahren, bzw. Performance-Metriken für die durchzuführenden Experimente vertraut machen. Es sollen der Erfolg der Lerner in Simulation und Testbed mit und ohne Transfer verglichen werden.

4. Mit den Schnittstellen zu MARS und Testbed vertraut machen und Lernverfahren auf Simulation und Testbed anwenden. Die Ergebnisse dieser Versuche sollten zum objektiven Vergleich gemäß vorher definierter Metriken ausgewertet werden. Eventuell müssen die Simulation oder das Testbed erst wieder in einen funktionsfähigen Zustand gebracht werden.

5. Abschließend Transferversuche mit methodischer und strukturierter Auswertung gemäß den vorher definierten Metriken.

Literatur

[1] Larbi Abdenebaoui. Implementation and Evaluation of a Connectionist Learning Architecture in a Simulated "Brio Labyrinth Game". Master's thesis, Universität Bremen, 2007.

[2] Samuel Barrett, Matthew E. Taylor, and Peter Stone. Transfer learning for reinforcement learning on a physical robot. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS-10)*, May 2010.

[3] Constantin Bergatt. Minimierung und Untersuchung des Simulation-Reality-Gaps anhand eines BRIO-Labyrinth-Spiels. Master's thesis, Universität Rostock, 2009.

[4] E. Kirchner, L. Abdenebaoui, J. H. Metzen, M. Tabie, J. Teiwes, C. Bergatt, and F. Kirchner. BRIO Labyrinth 2 - Einrichtung als Testbed für Lernarchitekturen und EEG / fMRI Untersuchungen. Technical report, DFKI Bremen, 2009.

[5] J. H. Metzen, E. Kirchner, L. Abdenebaoui, and F. Kirchner. Learning to play the brio labyrinth game. *Künstliche Intelligenz*, 3, 2009.

[6] Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.