at 12/2010

Algorithmen zum Automatischen Erlernen von Motorfähigkeiten

Policy Learning Algorithms for Motor Skills

Jan Peters, Jens Kober, Max Planck Institut für biologische Kybernetik, Tübingen,
Stefan Schaal, University of Southern California (USC)

Zusammenfassung Das automatische Erlernen von Motorfähigkeiten würde es einem autonomen Roboter ermöglichen, sich an neuartige Situationen anzupassen. Dieses Ziel ist seit langem eine Vision der Robotik, Künstlichen Intelligenz und Kognitionswissenschaft. Allerdings kann dieses Vorhaben mit heutigen Techniken nicht erreicht werden, da nur wenige der Methoden in der Lage sind, mit der Komplexität moderner Manipulatoren oder von humanoiden Robotern umzugehen. Dieser Artikel beschreibt allgemeine Ansätze für Policy Learning mit einem Schwerpunkt auf Motorstrategien. Das Ziel ist es, die Grundlagen zu schaffen, um motorische Fähigkeiten zu erlernen und sich selber automatisch zu verbessern. Hierfür werden zwei bedeutende Bereiche für einen solchen Ansatz betrachtet: einerseits Policy-Learning-Algorithmen, die den Anforderungen des Erwerbens motorischer Fähigkeiten gerecht werden, andererseits theoretisch fundierte, allgemeine Strukturen, welche für Aufgabendarstellung und -ausführung benötigt werden.

►►► **Summary** Robot learning methods which allow autonomous robots to adapt to novel situations have been a long standing vision of robotics, artificial intelligence, and cognitive sciences. However, to date, learning techniques have yet to fulfill this promise as only few methods manage to scale into the high-dimensional domains of manipulator robotics, or even the new upcoming trend of humanoid robotics. If possible, scaling was usually only achieved in precisely pre-structured domains. In this paper, we investigate the ingredients for a general approach policy learning with the goal of an application to motor skill refinement in order to get one step closer towards human-like performance. For doing so, we study two major components for such an approach, i. e., firstly, we study policy learning algorithms which can be applied in the general setting of motor skill learning, and, secondly, we study a theoretically well-founded general approach to representing the required control structures for task representation and execution.

Schlagwörter Maschinelles Lernen, Humanoide Robotik, Roboterlernen ►►► **Keywords** Robot learning, policy search, reinforcement learning

1 Einleitung

Trotz der zunehmenden Anzahl motorischer Fähigkeiten heutiger anthropomorpher und humanoider Roboter, hat sich der grundsätzliche Ansatz zur Erzeugung motorischen Verhaltens in den letzten Jahrzehnten kaum verändert [14]. Der Ingenieur modelliert die Aufgabe präzise und verwendet das menschliche Verständnis der motorischen Aufgabe, um das gewünschte Verhalten des Roboters zu erhalten. Um dieses zu erreichen, müssen meist jegliche Unsicherheiten in der Umgebung beseitigt werden, welche das starre, resultierende Programm behindern könnten. In den meisten Fällen wird dieser Vorgang darauf reduziert, einen gewünschten Bewegungsablauf in einer vorgefertigten Umgebung mit präzise platzierten Objekten aufzuzeichnen. Verbleibende Ungenauigkeiten behandelt der Ingenieur mit Hilfe seines Verständnisses der vorliegenden Aufgabe in Form von gut durchdachten Ausnahmen.

Obwohl solche Ansätze für strukturierte Industrie- oder Forschungsumgebungen häufig ausreichen, ist es offensichtlich, dass die Roboter der Zukunft flexibler instruierbar und adaptiver sein müssen. Zum Beispiel sind die Roboterprogrammierkosten der Industrie heutzutage ein elementarer limitierender Faktor für die Produk-

tion in Deutschland geworden, und viele zukünftige Anwendungen liegen in unstrukturierten Umgebungen außerhalb von Fabrikhallen (z. B. in der Servicerobotik). Daher ist es essentiell, diese starke Abhängigkeit von handgefertigten Modellen der Umgebung und des Roboters zu reduzieren. Stattdessen wird daher ein Ansatz benötigt, der es erlaubt, Roboter zu verwenden, die für die Interaktion mit weniger strukturierten und unsicheren Umgebungen entwickelt wurden. Ein derartiger Ansatz kann sich nicht allein auf menschliches Wissen verlassen, sondern die Fähigkeiten des Roboters müssen automatisch aus der Imitation von demonstrierten Fähigkeiten (im Sport würde man das Grobtraining nennen) und durch Versuch & Irrtum (engl. Trial & Error) des Roboters erzeugt werden.

Der enorme Fortschritt des maschinellen Lernens in den letzten Jahrzehnten stellt zahlreiche neue Ansätze für das Lernen motorischer Fähigkeiten zur Verfügung. Leider skalieren viele dieser Techniken nicht in die hochdimensionalen Arbeitsräume von Manipulatoren oder humanoiden Robotern. Daher können diese Verfahren nicht die alleinige Basis zum datengesteuerten Erwerb motorischer Fähigkeiten bieten. Wir benötigen einen Ansatz zum Roboterlernen, der auf dem Verständnis der motorischen Systeme basiert, anstelle eines unstrukturierten, monolithischen Ansatzes, der direkt dem maschinellen Lernen entnommen ist. Hierzu müssen separat adäquate Lösungen für die inhärenten Probleme der Aufgabenbeschreibung, des Lernens und der Ausführung bestimmt werden. Diese können für ein zusammenhängendes Rahmenkonzept verwendet werden, welches durch die Kombinationen von Imitation, Selbstverbesserung und dem Erlernen von Modellen in der Lage ist, komplexe motorische Fähigkeiten zu erlangen. Ein großer Vorteil eines solchen Ansatzes liegt in der Unterteilung der Hauptprobleme in initialen Fähigkeitserwerb, nachfolgende Verfeinerung und Ausführung. Da weder unstrukturierte, universelle Ansätze zum maschinellen Lernen noch handgefertigte Modelle mit vordefinierten Bewegungspfaden verwendet werden, wird es möglich, motorische Fähigkeiten in Form einer Policy von einer Demonstration zu erlernen und durch Trial & Error zu verfeinern.

2 Erlernen von motorischen Fähigkeiten

Hauptziel dieses Artikels ist es, die Grundlagen für ein allgemeines Rahmenkonzept zum Erlernen motorischer Fähigkeiten von Robotern zu schaffen. Die Notwendigkeit eines solchen Konzeptes lässt sich an einem Beispiel am besten erklären.

Beispiel. Wenn Menschen einen neuen Sport erlernen, z. B. Tennis oder Ping Pong, wird der Lehrer den Schüler bei der Hand nehmen und ihm Konzepte von Vorhand, Rückhand und Aufschlag vermitteln. Der Schüler wird zwar in der Lage sein, eine Grobform dieser elementaren Bewegungen zu reproduzieren, aber nicht eine perfekte Bewegung. Hierzu verlässt sich der Schüler darauf, dass sein Motorsystem nur sinnvolle Klassen von Bewegungen repräsentieren kann. Durch viel Übung wird der Schüler in der Lage sein, eine Feinform zu erlangen. Dabei verlässt der Schüler sich darauf, dass die Hand-Auge-Koordination und Bewegungssteuerung dabei gezielt verbessert werden, um die neue Aufgabe möglich zu machen.

Diesem Beispiel kann man die drei separaten Komponenten unseres Rahmenkonzepts entnehmen. Zum automatischen Fähigkeitserwerb sind nötig: (i) adäquate Darstellungen von Bewegungen, (ii) Lernalgorithmen, welche auf diese Bewegungsdarstellungen angewendet werden können und (iii) eine Transformation, welche die Ausführung im geeigneten Arbeitsraum des Roboters ermöglicht.

2.1 Erforderliche Komponenten

Dieses Kapitel beschreibt die wesentlichen Konzepte, die den drei erforderlichen Komponenten, d. h. Darstellung, Lernen und Ausführung, zu Grunde liegen.

Darstellung. Für die Darstellung motorischer Fähigkeiten kann die Erkenntnis herangezogen werden, dass der Mensch zwar in der Lage ist, eine große Anzahl komplizierter Bewegungen auszuführen, sich allerdings dabei auf eine kleinere Anzahl einfacher Bewegungsprimitive beschränkt [13]. Wie von Ijspeert et al. gezeigt [4], lassen sich solche Bewegungsprimitive durch nichtlineare dynamische Systeme repräsentieren. Diese dynamischen Systeme erzeugen Gelenkpositionen, Geschwindigkeiten und Beschleunigungen basierend auf externen Signalen. Zu beachten ist, dass diese dynamischen Systeme sowohl Bewegungen im Operationsraum als auch im Gelenkraum des Roboters beschreiben können.

Lernen. Das Erlernen einfacher motorischer Fähigkeiten wird durch Anpassung der Systemparameter $\boldsymbol{\theta}_i$ einer Motor-Primitiven i erreicht. Da nicht alle möglichen Bewegungen ausprobiert werden können, ist die unstrukturierte Verwendung der Techniken des maschinellen Lernens praktisch unmöglich. Stattdessen muss eine Kombination aus Überwachtem Lernen (engl. Supervised Learning) und Bestärkendem Lernen (engl. Reinforcement Learning) verwendet werden, um motorische Fähigkeiten zu erlernen. Durch Supervised Learning werden die motorischen Fähigkeiten initialisiert und danach durch Reinforcement Learning verbessert. Demnach besteht das Erlernen einer neuen motorischen Aufgabe aus zwei Phasen, d. h. der ‚lernende Roboter' versucht, die Fähigkeit, die er aus einer Demonstration gewinnt, zu imitieren und verbessert diese mit den Ergebnissen aus Trial & Error, d. h. durch Reinforcement Learning.

Ausführung. Das Ausführen motorischer Fähigkeiten fügt eine weitere Ebene der Komplexität hinzu. Es verlangt, dass ein mechanisches System dazu gebracht wird, eine solche Bewegungsprimitive auszuführen. Die

Bewegungsprimitive kann hierbei als mechanische Nebenbedingung des Systems betrachtet werden, welche mittels akkurater Vorhersage der benötigten Kräfte (anhand analytischer Modelle) eingehalten wird. Wenn man zugleich niedrige Reglerverstärkung und hohe Genauigkeit benötigt (wie z. B. Roboter in von Menschen bewohnten Umgebungen), reicht die Genauigkeit von traditionellen Modellen nicht aus und gelernte Systeme gewinnen auch in der Ausführung an Bedeutung.

In diesem Artikel wird solch ein Ansatz als Grundlage für das Verständnis motorischen Fähigkeitserwerbs betrachtet.

2.2 Resultierender Ansatz

Wie in der Diskussion um Ziele und essentielle Komponenten dargelegt, wird ein adäquates Rahmenkonzept zum Erwerb motorischer Fähigkeiten benötigt. In unserem Fall werden die gewünschten Bewegungen oft auch im Arbeitsraum erzeugt, durch Bewegungsprimitive repräsentiert und von der Bewegungsausführung in den Aktionsraum übersetzt. Ausgehend von der analytischen Betrachtungsweise der Robotik auf diesen Wandel stellen wir ein solches Lernkonzept für die Aufgabenausführung im Operationsraum vor. Hierfür müssen zwei Komponenten beachtet werden: das Erlernen des gewünschten Verhaltens in Form von Primitiven und die Ausführung durch Transformation dieser Primitiven in motorische Anweisungen. Es müssen skalierbare Lernalgorithmen entwickelt werden, die adäquat sowie effizient bezüglich der gewählten Architektur zum Erlernen grundlegender motorischer Fähigkeiten sind. Darüber hinaus werden schnelle Algorithmen für die sofortige Umsetzung des Policy-Learning für die Bewegungsausführung aus unmittelbar observablen Erfolgen benötigt, damit das System sich während der Ausführung verbessern kann. Das Erlernen der Aufgabe selbst hingegen benötigt das Erlernen einer Policy, die einen längerfristigen Fortschritt der Aufgabe definiert, indem Bewegungsprimitive mittels Demonstration durch einen Lehrer erlernt und durch Reinforcement Learning verbessert werden. Der daraus resultierende Ansatz dieser Arbeit wird in Bild 1 dargestellt.

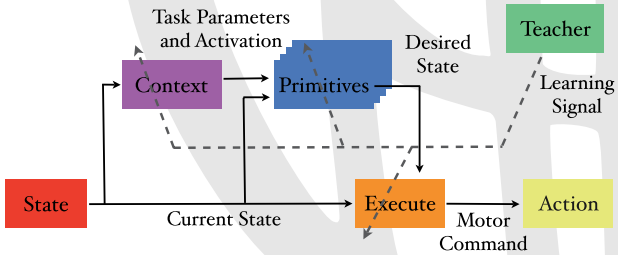

Bild 1 Dieses Bild zeigt die wichtigsten Komponenten unseres Rahmenkonzeptes zum automatischen Erlernen von Motorfähigkeiten. Ein Supervisor entscheidet, welche Motor Primitive (Bewegungsprimitive) von der Execution ausgeführt wird. Von einem Lehrer stammt ein Lernsignal, d. h. entweder ein Ausführungsfehler, eine Demonstration oder ein Reinforcement (Belohnung oder Bestrafung).

3 Ansätze zum Erlernen von Policies

Wie zuvor dargelegt, werden zwei verschiedene Arten von Policy-Learning-Algorithmen benötigt, um langfristige sowie sofortige Verbesserung zu kombinieren. Diese Verfahren basieren auf einer Erfolgsfunktion, welche wie folgt dargestellt werden kann:

$$J(\boldsymbol{\theta}) = \int_{\mathbb{T}} p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) r(\boldsymbol{\tau}) d\boldsymbol{\tau} \,. \tag{1}$$

Hier ist $\boldsymbol{\tau} = [\mathbf{x}_{1:n}, \mathbf{u}_{1:n}]$ eine Episode (engl. *Roll-out*) mit Zuständen $\mathbf{x}_{1:n}$ und Aktionen $\mathbf{u}_{1:n}$, und $r(\boldsymbol{\tau})$ die Belohnung entlang des Pfades, meist in der Form

$$r(\boldsymbol{\tau}) = \sum_{t=1}^{n} \gamma^t r_t \,.$$

Die Wahrscheinlichkeitsdichte $p_{\boldsymbol{\theta}}(d\boldsymbol{\tau})$ der Roll-outs kann meist durch

$$p_{\boldsymbol{\theta}}(d\boldsymbol{\tau}) = p(\mathbf{x}_1) \prod_{t=1}^{n-1} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) \pi(\mathbf{u}_t|\mathbf{x}_t; \boldsymbol{\theta})$$

dargestellt werden, mittels einer Ausgangszustandsverteilung $p(\mathbf{x}_1)$, einer Zustandsübergangsverteilung $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$ und einer Policy $\pi(\mathbf{u}_t|\mathbf{x}_t; \boldsymbol{\theta})$. Es ist zu beachten, dass $p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) r(\boldsymbol{\tau})$ auch als Verteilung gesehen werden kann, allerdings als eine nicht normalisierte Verteilung (engl. improper distribution). Die Policy $\pi(\mathbf{u}_t|\mathbf{x}_t; \boldsymbol{\theta})$ ist die Funktion, die durch Optimieren ihrer Parameter $\boldsymbol{\theta} \in \mathbb{R}^N$ erlernt werden soll. Viele Algorithmen für Policy-Learning sind entworfen worden, um eine solche Kostenfunktion zu optimieren, z. B. Policy-Gradients-Methoden [1], Actor-Critic-Methoden [7; 15], der Natural-Actor-Critic-Algorithmus [9; 10; 12] sowie der Reward-Weighted-Regression-Algorithmus [8]. Im folgenden wird ein einheitlicher Ansatz der Policy-Optimierung skizziert, der die Herleitung aller oben genannten Methoden aus einer einzelnen Kostenfunktion ermöglicht. Dieser Abschnitt mag abstrakt wirken verglichen mit dem Rest dieser Arbeit, allerdings beinhaltet er grundlegende Neuerungen, da er eine einheitliche Ansicht auf viele frühere und zukünftige Ansätze ermöglicht.

3.1 Schranken der Updates

In diesem Abschnitt werden obere und untere Schranken der Kostenfunktion betrachtet für das Erstellen von Verbesserungsschritten für die Policy. Aus der oberen Schranke kann man erkennen, dass ein Greedy-Operator keine praktikable Lösung ist, während die untere Schranke nützliche Regeln zum Verbessern von Policies liefert.

Obere Schranke zum Verbessern von Policies

Aus dem Bereich der stochastischen Programmierung ist bekannt, dass Greedy-Ansätze einer Verbesserung der Policy den großen Nachteil haben, eine verzerrte Lösung (engl. biassed solution) zu liefern. Dieser Nachteil lässt

sich ohne Probleme herleiten: Wenn wir $J(\boldsymbol{\theta})$ optimieren und durch Daten approximieren, beispielsweise durch

$$\hat{J}_S(\boldsymbol{\theta}) = \sum_{s=1}^{S} p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}_s\right) r\left(\boldsymbol{\tau}_s\right) \approx J(\boldsymbol{\theta})\,,$$

erhalten wir eine grundlegende Beziehung

$$E\{\max_{\boldsymbol{\theta}} \hat{J}_S(\boldsymbol{\theta})\} \geq \max_{\boldsymbol{\theta}} E\{\hat{J}_S(\boldsymbol{\theta})\}\,. \tag{2}$$

Diese folgt direkt daraus, dass das Maximum stets größer als jeder einzelne Eintrag eines Datenpunktes ist und somit eine einzelne Überschätzung sofort das Ergebnis verzerrt. Durch das Bilden des Erwartungswertes kann dieses Problem nicht beseitigt werden. Eine Policy, die durch den Schritt eines Greedy-Algorithmuses im Parameterspektrum optimiert wird, weist damit garantiert eine systematische Abweichung von $b_S(\boldsymbol{\theta}) = E\{\max_{\boldsymbol{\theta}} \hat{J}_S(\boldsymbol{\theta})\} - \max_{\boldsymbol{\theta}} E\{\hat{J}_S(\boldsymbol{\theta})\} \geq 0$ auf, sofern Fehler vorhanden sind. Es kann jedoch auch gezeigt werden, dass die Abweichung mit zunehmender Anzahl an Datenpunkten abnimmt, d. h. $b_S(\boldsymbol{\theta}) \geq b_{S+1}(\boldsymbol{\theta})$, und für unendlich viele Datenpunkte gegen Null konvergiert, d. h. $\lim_{S\to\infty} b_S(\boldsymbol{\theta}) = 0$ [11]. Diese Optimierungsabweichung zeigt die Mängel des Greedy-Operators: für eine endliche Anzahl an Datenpunkten ist jede Aktualisierung der Policy problematisch und kann in Form von Oszillation, Abweichung, usw. einen unbeständigen Lernprozess verursachen, wie häufig im Bereich des Reinforcement Learning beobachtet worden ist [1; 2].

Untere Schranke für Verbesserungen der Policy

In den meisten anderen Bereichen des maschinellen Lernens lag der Fokus stets auf unteren Schranken, beispielsweise zum Herleiten von Expectation-Maximization-Algorithmen (EM). Der Grund für diese Präferenz trifft auch auf das Policy Learning zu: Wenn die untere Schranke als Gleichung einer Erfolgsfunktion maximiert wird, kann garantiert werden, dass die Policy in diesem Schritt verbessert wird. Überraschenderweise lassen sich die unteren Schranken des Supervised Learning leicht übertragen. Dafür wird der Fall betrachtet (wie bereits vorgeschlagen in [3]), dass es eine Policy $\boldsymbol{\theta}'$ gibt und die Pfadverteilung, die von dieser Policy generiert wird, mit der erfolgsgewichteten Pfadverteilung übereinstimmen soll, um anschließend die Distanz der beiden Pfadverteilungen zu minimieren, d. h. $D(p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})||p_{\boldsymbol{\theta}}(\boldsymbol{\tau})r(\boldsymbol{\tau}))$. Erstaunlicherweise liefert dies eine untere Schranke basierend auf der Jensenschen Ungleichung und der Konvexität der logarithmischen Funktion. Damit erhalten wir

$$\log J(\boldsymbol{\theta}') = \log \int \frac{p_{\boldsymbol{\theta}}(\boldsymbol{\tau})}{p_{\boldsymbol{\theta}}(\boldsymbol{\tau})} p_{\boldsymbol{\theta}'}(\boldsymbol{\tau}) r(\boldsymbol{\tau}) d\boldsymbol{\tau}\,, \tag{3}$$

$$\geq \int p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) r(\boldsymbol{\tau}) \log \frac{p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})}{p_{\boldsymbol{\theta}}(\boldsymbol{\tau})} d\boldsymbol{\tau} \tag{4}$$

$$\propto - D\left(p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})||p_{\boldsymbol{\theta}}(\boldsymbol{\tau})r(\boldsymbol{\tau})\right), \tag{5}$$

wobei

$$D\left(p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})||p_{\boldsymbol{\theta}}(\boldsymbol{\tau})\right) = \int p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) \log(p_{\boldsymbol{\theta}}(\boldsymbol{\tau})/p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})) d\boldsymbol{\tau}$$

die Kullback-Leibler-Divergenz ist, d. h. ein Abstandsmaß für Wahrscheinlichkeitsverteilungen. Mit anderen Worten, es gibt die untere Schranke, deren Maximierung der Minimierung von

$$J_{\mathrm{KL}} = D\left(p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})||p_{\boldsymbol{\theta}}(\boldsymbol{\tau})r(\boldsymbol{\tau})\right) \tag{6}$$

$$= \int p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) r(\boldsymbol{\tau}) \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) r(\boldsymbol{\tau})}{p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})} d\boldsymbol{\tau} \tag{7}$$

entspricht. Da die Policy garantiert verbessert wird, können nicht die Probleme auftreten, die durch das Optimieren der oberen Schranke auftreten und für das traditionelle Reinforcement Learning so problematisch waren. In vielen Fällen jedoch könnte das zusätzliche Bestrafen der Abweichung von der vorherigen Lösung gewünscht sein, um das Erforschen neuer Lösungen zu reduzieren. In diesem Fall soll die Distanz, mit der man sich von der vorherigen Policy entfernt, zusätzlich bestraft werden, z. B. indem der Ausdruck $J_+ = D\left(p_{\boldsymbol{\theta}}(\boldsymbol{\tau})||p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})\right)$ minimiert wird. Sei $\lambda \in \mathbb{R}^+$ ein positiver Faktor zur Bestrafung mit $0 \leq \lambda \leq J(\boldsymbol{\theta})$, so erhalten wir die vereinte Kostenfunktion

$$J_{\mathrm{KL}+} = J_{\mathrm{KL}} + \lambda J_+\,. \tag{8}$$

Es ist anzumerken, dass die Argumente auf Grund der Asymmetrie der Kullback-Leibler-Divergenz vertauscht werden. Diese Kostenfunktion wird eine wichtige Rolle spielen, da sowohl die sogenannte *Baseline* als auch Natural-Policy-Gradients ein direktes Resultat daraus sind. Die richtige Bestimmung von λ ist nicht trivial und hängt von der gewählten Methode ab. Bei Policy-Gradients beispielsweise wird dieser Parameter zur Baseline.

3.2 Resultierende Ansätze für Policy-Learning

Als Nächstes werden aus dieser Betrachtungsweise drei verschiedene Ansätze für die Optimierung der unteren Schranke abgeleitet: Policy-Gradients, Natural-Actor-Critic- und Reward-Weighted-Regression-Algorithmus.

Policy-Gradient-Ansätze

Da bekanntermaßen Policy-Gradienten-Methoden [1; 2] nicht von den Nachteilen des Greedy-Operators betroffen sind, erlebten diese in den letzten Jahren einen deutlichen Aufschwung. Policy-Gradient-Ansätze können direkt aus dieser Formulierung hergeleitet werden, indem der steilste Abstieg der Taylor-Entwicklung erster Ordnung verwendet wird:

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \alpha(\nabla J_{\mathrm{KL}} - \lambda \nabla J_+) \tag{9}$$

$$= \boldsymbol{\theta} + \alpha \int p_{\boldsymbol{\theta}}(\boldsymbol{\tau})(r(\boldsymbol{\tau}) - \lambda) \nabla \log p_{\boldsymbol{\theta}'}(\boldsymbol{\tau}) d\boldsymbol{\tau}\,, \tag{10}$$

wobei α eine Lernrate (engl. learning rate) ist. Dies folgt aus der Kommutativität der Kullback-Leibler-Divergenz.

Durch den Term, welcher die Abweichung von der bisherigen Policy bestraft, lässt sich die Baseline zur Schätzung mit niedriger Varianz einfügen. Aus

$$\nabla \log p_{\theta'}(\tau) = \sum_{t=1}^{n-1} \nabla \log \pi(\mathbf{u}_t|\mathbf{x}_t;\theta)$$

erhält man Policy-Gradient-Schätzer, welche als REINFORCE, Policy Gradients Theorem oder GPOMDP bekannt sind (eine Übersicht kann in [1] gefunden werden). Der Ausdruck für die Bestrafung beschränkt lediglich die Varianz des Policy-Gradient-Schätzers und verschwindet für eine unendliche Anzahl an Daten, da dann $\nabla J_{\mathrm{KL}+} = \nabla J_{\mathrm{KL}}$ gilt. In der Praxis resultiert diese Policy-Verbesserungsregel jedoch eher in eine langsame Konvergenz zu guten Lösungen [5; 9; 10; 12].

Natural-Policy-Gradient-Ansätze
Erstaunlicherweise kann die Geschwindigkeitsaktualisierung signifikant beschleunigt werden, wenn Ausdrücke höherer Ordnung von J_+ bestraft werden, z. B. liefert der Ausdruck der Taylor-Erweiterung zweiter Ordnung

$$\theta' = \mathrm{argmax}_{\theta'}(\theta' - \theta)^T(\nabla J_{\mathrm{KL}} - \lambda \nabla J_+) \quad (11)$$
$$- \frac{1}{2}\lambda(\theta' - \theta)^T \nabla^2 J_+(\theta' - \theta)$$
$$= \lambda \left(\nabla^2 J_+\right)^{-1} \left(\nabla J_{\mathrm{KL}} - \lambda \nabla J_+\right) = \lambda F^{-1} g_1\,, \quad (12)$$

wobei

$$F = \nabla^2 D\left(p_\theta(\tau)||p_{\theta'}(\tau)\right)$$
$$= \nabla^2 D\left(p_{\theta'}(\tau)||p_\theta(\tau)\right) = \nabla^2 J_+$$

auch als Fisher-Informationsmatrix und das resultierende Update der Policy $\mathbf{g}_2$ als Natural-Policy-Gradient bekannt sind. Überraschenderweise wurde der Ausdruck zweiter Ordnung noch nicht erweitert und es sind keine Natural-Gradient-Ansätze der zweiten Ordnung bekannt.

EM-Policy-Learning
Es gibt Sonderfälle, wo die im Sinne der unteren Schranke optimalen Policy-Parameter direkt bestimmt werden können. Sollten die Statistiken des Schätzers linear sein nach Anwendung von dem Logarithmus und nach Ableitung bezüglich der Parameter, d. h.

$$\nabla \log \pi(\mathbf{u}_t|\mathbf{x}_t;\theta) = \mathbf{A}(\mathbf{x}_t, \mathbf{u}_t)\theta + \mathbf{b}(\mathbf{x}_t, \mathbf{u}_t)\,, \quad (13)$$

ist es relativ unkompliziert, einen EM-Algorithmus herzuleiten, so dass

$$\theta' = \alpha^{-1}\beta\,, \quad (14)$$

$$\alpha = \int p_\theta(\tau)(r(\tau) - \lambda) \sum_{t=1}^{n} \mathbf{A}(\mathbf{x}_t, \mathbf{u}_t) d\tau, \quad (15)$$

$$\beta = \int p_\theta(\tau)(r(\tau) - \lambda) \sum_{t=1}^{n} b(\mathbf{x}_t, \mathbf{u}_t) d\tau\,. \quad (16)$$

Diese Art Algorithmen können, wenn anwendbar, sehr schnelle Policy-Verbesserungen erzielen. Eine Lernrate wird nicht mehr benötigt und sie nähern sich mindetens einer lokalen Optimallösung im Sinne von Gleichung (1) an.

3.3 Skizze der resultierenden Algorithmen

Auf diese Weise wurden zwei Arten Algorithmen entwickelt, der Natural-Actor-Critic und EM-artige Algorithmen wie der Reward-Weighted-Regression-Algorithmus sowie der PoWER-Algorithmus.

Natural-Actor-Critic-Algorithmus
Die Natural-Actor-Critic-Algorithmen [9; 10] sind Varianten des zuvor beschriebenen Natural-Policy-Gradient-Algorithmuses mit großem oder unendlichem Planungshorizont n. Sie gelten als derzeit schnellste Policy-Gradient-Methoden („the current method of choice“ laut [1]). Sie beruhen auf der Erkenntnis, dass wir die Belohnung maximieren, dabei aber den Erfahrungsverlust konstant halten müssen, d. h. dass wir die Distanz zwischen unserer gegenwärtigen Rollout-Verteilung und der durch die Policy neu erzeugten Rollout-Verteilung messen. Diese Distanz kann durch die Kullback-Leibler-Divergenz gemessen und unter Verwendung der Fisher-Informationsmetrik approximiert werden, was einem Natural-Policy-Gradient-Ansatz entspricht. Dieser steht in Verbindung zur kürzlich eingeführten Approximation kompatibler Funktionen, was den gewünschten Natural-Actor-Critic liefert. Interessanterweise können frühere Actor-Critic-Ansätze von diesem neuen Ansatz abgeleitet werden. In seiner Anwendung auf das Erlernen von Motor-Primitiven kann demonstriert werden, dass der Natural-Actor-Critic sowohl Gradient-Methoden mit endlicher Differenz als auch einfache Policy-Gradient-Methoden mit optimaler Baseline in puncto Lerngeschwindigkeit übertrifft.

Reward-Weighted-Regression & PoWER-Algorithmen
Im Gegensatz zum Natural-Actor-Critic-Algorithmus sind die Reward-Weighted-Regression [8] und PoWER [6] nämlich EM-Algorithmen für Reinforcement Learning. Es handelt sich also um ein hinsichtlich der Belohnung gewichtetes Regressionsproblem, d. h. es gibt eine eindeutige Lösung, die durch bekannte Regressionsanalysetechniken ermittelt werden kann. Obgleich wir eine intuitivere Erklärung für diesen Algorithmus geliefert haben, entspricht dieser einem richtig abgeleiteten Maximization-Maximization-Algorithmus (MM), der eine untere Schranke für sofortige Belohnung maximiert, ähnlich einem EM-Algorithmus. Unsere Anwendungen zeigen, dass er in die höherdimensionalen Räume skaliert und eine gute Policy ohne jede Nachahmung eines menschlichen Lehrers erlernt.

4 Anwendungen der Robotik

Die bisher diskutierten Algorithmen können sowohl zum Lernen von Bewegungsprimitiven als auch zum Lernen von Operational-Space-Control verwendet werden.

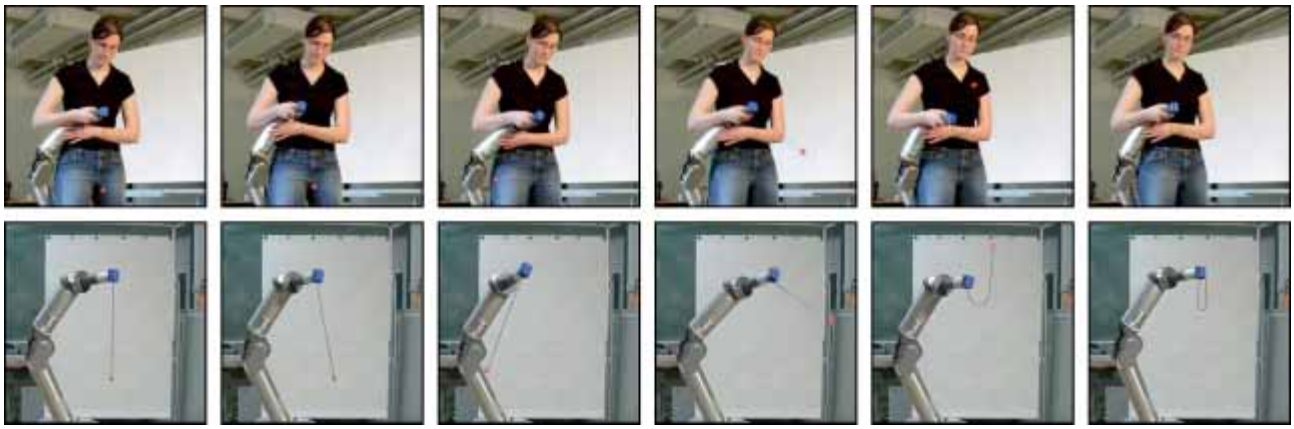

Bild 2 Diese Abbildung zeigt die beiden wichtigsten Schritte des Erlernens von Bewegungsprimitiven. Zuerst nimmt ein Mensch den Roboter bei der Hand, führt ihn entlang der gewünschten Trajektorie und demonstriert so die gewünschte Bewegung. Diese Bewegung soll den Ball durch einen Ruck nach oben schleudern und mit dem Becher auffangen. Sie soll zunächst vom Roboter per Imitationslernen nachgeahmt werden, aber eine Imitation reicht hier nicht aus und zusätzliches Reinforcement Learning ist nötig. Der PoWER-Algorithmus ist in der Lage innerhalb von weniger als 100 Trials diese Aufgabe so perfekt zu lernen, dass er jeden Ball mit dem Becher fängt.

4.1 Lernen von Bewegungsprimitiven

Die wichtigste Anwendung liegt im Erlernen von Bewegungsprimitiven. Hierzu wird zunächst die Bewegungsprimitive aus einer Demonstration durch Imitationslernen initialisiert. Danach wird diese durch das in diesem Artikel diskutierten Verfahren optimiert. Dieses zweistufige Vorgehen erlaubt es, den Suchraum zu reduzieren und ermöglicht eine Konvergenz innerhalb von einer realistischen Anzahl von Versuchen.

Der Natural Actor-Critic-Algorithmus wurde hier als erstes angewandt, um sowohl einen T-Ball-Swing zu lernen als auch einen Swing-Up mit Kraftlimitierung. Der PoWER-Algorithmus hat den Natural-Actor-Critic-Algorithmus in Bezug auf Lerngeschwindigkeit beim Erlernen des Swing-Up mit Kraftlimitierung geschlagen und es sogar erreicht komplizierte Aufgaben wie Ball-in-a-cup, siehe Bild 2, zu erlernen.

4.2 Lernen von Operational-Space-Control

Operational-Space-Control erlaubt es uns, im Aufgabenraum repräsentierte Bewegungsprimitive direkt auszuführen, anstatt sie umständlich in eine andere Repräsentation zu übersetzen. Wir haben ein Lernsystem für Operational-Space-Control vorgestellt, das eine neue Darlegung dieses Konzepts als grundlegendes, punktweise optimales Steuerungskonzept und Einblicke in Reinforcement Learning mit sofortiger Belohnung vereint. Da das allgemeine Erlernen dieses Konzepts mit mehrfachen Freiheitsgraden nicht-konvex ist und damit globale Techniken des Supervised Learning nicht direkt angewendet werden können, lassen sich daraus zwei Erkenntnisse gewinnen. Einerseits, dass das Problem ein lokal konvexes ist, und andererseits, dass eine punktweise Kostenfunktion eine globale Konsistenz der lokalen Lösungen gewährleistet. Es wurde gezeigt, dass dieses Verfahren die analytisch errechnete Optimallösung für einen simulierten Roboterarm mit drei Freiheitsgraden für jene Bereiche erzielt, für die genügend Datenpunkte im Zustandsraum ermittelt werden können.

5 Fazit

Im Rahmen dieses Artikels wurde ein allgemeines Rahmenkonzept für das Lernen motorischer Fähigkeiten vorgestellt, das auf einem vollständigen, analytischen Verständnis der Darstellung und Ausführung robotischer Aufgaben basiert. Es wurde eine allgemeine Herleitung für zahlreiche bekannte Policy-Learning-Algorithmen eingeführt, welche auch die Ableitung neuartiger Methoden wie des Natural-Actor-Critic, der Reward-Weighted-Regression und des PoWER-Algorithmuses erlaubt. Die Effizienz dieser Methoden haben wir in zahlreichen Anwendungen gezeigt.

Literatur

[1] D. Aberdeen. POMDPs and policy gradients. In *Proceedings of the Machine Learning Summer School (MLSS)*, Canberra, Australia, 2006.

[2] D. A. Aberdeen. *Policy-Gradient Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Australian National University, 2003.

[3] P. Dayan and G. E. Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.

[4] A. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volumne 15, pages 1547–1554, Cambridge, MA, 2003. MIT Press.

[5] S. A. Kakade. Natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, Vancouver, CA, 2002.

[6] J. Kober and J. Peters. Learning motor primitives for robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[7] V. Konda and J. Tsitsiklis. Actor-critic algorithms. *Advances in Neural Information Processing Systems 12*, 2000.

[8] J. Peters and S. Schaal. Learning operational space control. In *Proceedings of Robotics: Science and Systems (RSS)*, Philadelphia, PA, 2006.

[9] J. Peters, S. Vijayakumar, and S. Schaal. Reinforcement learning for humanoid robotics. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, Karlsruhe, Germany, September 2003.

[10] J. Peters, S. Vijayakumar, and S. Schaal. Natural actor-critic. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 280–291. Springer, 2005.

[11] J. Peters and S. Schaal. Policy learning for motor skills. *ICONIP*, 2:233–242, 2007.

[12] S. Richter, D. Aberdeen, and J. Yu. Natural actor-critic for road traffic optimisation. In B. Schoelkopf, J. C. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, volume 19, Cambridge, MA, 2007. MIT Press.

[13] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. In C. D. Frith and D. Wolpert, editors, *The Neuroscience of Social Interaction*, pages 199–218. Oxford University Press, Oxford, UK, 2004.

[14] L. Sciavicco and B. Siciliano. *Modeling and control of robot manipulators*. MacGraw-Hill, Heidelberg, Germany, 2007.

[15] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. A. Solla, T. K. Leen, and K.-R. Mueller, editors, *Advances in Neural Information Processing Systems (NIPS)*, Denver, CO, 2000. MIT Press.

Manuskripteingang: 30. September 2010

Dr. Jan Peters ist Leiter der Gruppe „Robot Learning“ in der Abteilung Empirische Inferenz am Max-Planck Institut für biologische Kybernetik in Tübingen. Hauptarbeitsgebiete: Robotik, Maschinelles Lernen.

Adresse: Max-Planck Institut für biologische Kybernetik, Robot Learning Laboratory, Abteilung Empirische Inferenz, Spemannstr. 38, 72076 Tübingen, Germany, Fax: +49-(0)7071-601-552, E-Mail: jan.peters@tuebingen.mpg.de

Dipl.-Ing. Jens Kober ist Doktorand in der Gruppe „Robot Learning“ in der Abteilung Empirische Inferenz am Max-Planck Institut für biologische Kybernetik in Tübingen. Hauptarbeitsgebiete: Robotik, Maschinelles Lernen.

Adresse: Max-Planck Institut für biologische Kybernetik, Robot Learning Laboratory, Abteilung Empirische Inferenz, Spemannstr. 38, 72076 Tübingen, Germany, Fax: +49-(0)7071-601-552, E-Mail: jens.kober@tuebingen.mpg.de

Prof. Dr.-Ing. Schaal ist Leiter des Fachgebietes „Computational Learning and Motor Control“ im Fachbereich Computer Science an der University of Southern California. Hauptarbeitsgebiete: Robotik, Maschinelles Lernen, Menschliche Bewegungsgeneration.

Adresse: University of Southern California, Computational Learning and Motor Control Lab, 3710 S. McClintock Ave Los Angeles, CA 90089-2905, USA, Fax: +1 (213) 740 1510, E-Mail: sschaal@usc.edu