Today’s Lecture

• Introduction and overview of Imitation Learning
• What does imitation mean?
• Inverse Reinforcement Learning vs Behavioral Cloning
• Efficient representation for motor policies
• Case study: Learning motor skills in table tennis and ball in a cup
Motivation

From hand engineered solutions

To autonomous learning systems

Learning is necessary to allow for adaptive systems
Types of Learning

**Supervised Learning:**
Learner infers data structure from a finite list of input and output pairs. The correctly labeled outputs provide an error signal for the learner.

**Unsupervised Learning:**
Infer data structure from data consisting only from input data without any output label given. There is no error signal to evaluate the solution. E.g., data clustering.

**Reinforcement Learning:**
Agent explores the space of strategies by interacting with its environment and receiving reward of its actions. From this reward the agent infers the optimal strategy.
Markov Decision Processes

A world consisting of states, actions, and rewards

Goal is to find an optimal policy $\pi^*$, i.e., the best solution to our problem
Learning: the Solution?

How we perceive humans and animals learn:

- **Reinforcement Learning:** Learning from trial and error
- **Imitation Learning:** Learning from Demonstration

Easy! Right?
Reinforcement Learning

Unknown stochastic environment, i.e., reward matrix and transition probabilities are not known

Objective of Reinforcement Learning:

$$\max_{\pi} J^\pi = E \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \bigg| \pi \right]$$

Learn through experience and interactions with the world to make the optimal sequence of decision under uncertainty
Model Based Reinforcement Learning

Experience: \( \langle s_t, a_t, r_t, s_{t+1} \rangle \)

Policy Search

Value function approaches

Model based RL

\[ \pi(s) \]

\[ \text{arg max}_a Q(s, a) \]

Solve Bellman Equation

\[ T, R \]
Reinforcement Learning

What are the challenges with respect to robotics?

- Curse of dimensionality
- No accurate models for robot and its environment available
- Exploring the real world
- Reward function definition:
  - Desired behavior is defined by reward function
  - How to define the reward function?
Learning: the Solution?

How we perceive humans and animals learn:

• **Reinforcement Learning:** Learning from trial and error
  
  Not easy

• **Imitation Learning:** Learning from Demonstration

Easy! Right?
Reinforcement Learning

But how is it then possible to use Reinforcement Learning

• Use more effective representation
  • More compact state-action representation
  • Focus learning on those parts of the state-action space that are relevant
• Structure the task
• Use prior knowledge

Imitation Learning
Imitation Learning

Concept Imitation Learning:
• Reproduce the observed behavior!

Formal problem statement:
• Given a set of labeled training data (demonstrations), learn a function that maps the (observed) state to an action.
Imitation Learning

Concept Imitation Learning:
• Reproduce the observed behavior!

Formal problem statement:
• Given a set of labeled training data (demonstrations), learn a function that maps the (observed) state to an action.
Learning from Demonstration

• Initialize robot learning approaches
• Reduce the learning time significantly
• Allows robots to acquire human-like behavior
Imitation Learning

Examples: Learning motor tasks from demonstrations

Imitation Learning

Demonstrated Behavior

Novel Scene

Images: Drew Bagnell
Imitation Learning

Demonstrated Behavior

Learned Behavior

Images: Drew Bagnell
Imitation Learning

How do we imitate?
Learning from Demonstration

How to imitate depends on what to imitate!

What to imitate?
- Object target location
- Object movement
- Force interaction
- End-effector movement
- Joint configuration over time
Learning from Demonstration

Types of Imitation Learning
Learning from Demonstration

Learning manipulation from demonstration

What do we need to learn to learn to pour soda into a glass?
Learning from Demonstration

How to imitate depends on what to imitate!

Correspondence Problem

Agents might not share the same morphology and/or the same affordances!

How can we transfer the observed state-action sequences of the teacher to state-action sequences of the learner?
Learning from Demonstration

How to imitate depends on what to imitate!

Agents might not share the same morphology and/or the same affordances!

How can we transfer the observed state-action sequences of the teacher to state-action sequences of the learner?
Learning from Demonstration

How to imitate depends on what to imitate!

How to demonstrate?

• Observing a teacher e.g., with camera, kinect, motion capture
  – Detection error
  – Sensor noise
  – Post processing time
  – Correspondence problem

• Teleoperating a robot, e.g., with joystick
  – Out of body experience

• Kinesthetic teach-in

• Verbal instruction
  – Hard for lower level control
  – More suited for higher level instructions
How to Learn from Demonstrations

- Reward $R$
- Reinforcement Learning, Optimal Control
- Dynamical Model $T$
- Inverse Reinforcement Learning
- Control Policy $\pi$
- Behavioral Cloning
- Expert Demonstration
Learning from Demonstration

Case Study: Learning motor skills from demonstration
Scenario – Robot Table Tennis

What makes it an interesting case study?

• Uncertainty about the ball state
• Adaptation to new targets
• Different time critical hitting movements
How can we learn a manipulation tasks?

State $s$ $\rightarrow$ Policy $\rightarrow$ Joint Values $\theta, \dot{\theta}, \ddot{\theta}$ $\rightarrow$ Execution $\rightarrow$ Action $u$
Structure the Problem

Example driving

- Drive route
  - Stop at $a=0$
  - Observe until free $a=-1$
  - Pass Intersection $a=-5$
Structure the Problem

Table Tennis

Win Table Tennis
- Left corner
  - $\ddot{\theta}_t$
- Low right corner
  - $\ddot{\theta}_{t+1}$
- Smash left corner
  - $\ddot{\theta}_{t+2}$
How can we learn a manipulation tasks?

Learning Strategies:
- Learning strategic elements from demonstrations using Inverse Reinforcement Learning

Learning Hitting Movements:
- Learning motor skills from demonstration
- Learning how to select and generalize motor primitives
How can we learn a manipulation task?

1) Learn Motor Policies!

Why not just use planning?
How can we learn a motor policy?

Step 1: Representation and Initialization of Behavior
Representation of Behavior

Movement as state-action pairs?
• Limits us to one specific movement (not adaptable)
• Stuck with policy

Learning what the movement optimizes (reward function)?
• Unclear if we can apply this on a robot

Need a representation that allows us to encode the movement depending on the goal such that we can adapt it to new goals and length.
Representation of Motor Policies

Dynamic System Motor Primitives:

**General Idea**: Encode the shape of the movement!

- Straight
- curved
- circle

---

*A* a
Dynamical System Motor Primitives

Canonical system – action synchronization
\[ \dot{z} = b(z) \]
- phase of movement

Transformed system – desired state
\[ \ddot{\theta} = h(\theta, g, z, w) \]
- goal state
- current state
- shape parameters

Ispeert Model
- start state
- external force
- current state
- spring damper system
- goal state

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003
Dynamical System Motor Primitives

Canonical system – action synchronization
\[ \dot{z} = z \]

Transformed system – desired state
\[ \dot{v} = y(y(g - y)v) + f(z) \]
\[ \dot{y} = v \]

Transformation function
\[ f = \sum_{j=1}^{N} w_j j(z)z \]

Ispeert Model
- start state
- external force
- current state
- spring damper system
- goal state

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003
Dynamical System Motor Primitives

 Canonical system – action synchronization

\[ \tau \dot{z} = -\alpha z \dot{z} \]

Ispeert Model

- start state
- external force
- current state
- spring damper system
- goal state

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003

K. Muelling (National Robotics Engineering Center, CMU)
Dynamical System Motor Primitives

Canonical system – action synchronization
\[ \dot{z} = z \]

Transformed system – desired state
\[ \begin{align*}
\dot{v} &= \dot{y}(g - y) - v + f(z) \\
\dot{y} &= v
\end{align*} \]

Transformation function
\[ f = \frac{\sum_{j=1}^{N} w_j j(z)z}{\sum_{j=1}^{N} j(z)} \]

Ispeert Model
- start state
- external force
- current state
- spring damper system
- goal state

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003
Dynamical System Motor Primitives

Spring damper system

Ispeert Model

- start state
- external force
- current state

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003
Dynamical System Motor Primitives

Canonical system – action synchronization
\[ \dot{z} = z \]

Transformed system – desired state
\[ \dot{v} = y(g - y) - v + f(z) \]
\[ \dot{y} = v \]

Transformation function
\[ f(z) = \sum_{j=1}^{N} \frac{w_j j(z)}{\sum_{j=1}^{N} j(z)} \]

Ispeert Model
- start state
- external force
- current state
- spring damper system
- goal state

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003
Dynamical System Motor Primitives

Ispeert Model

- Start state
- External force
- Current state
- Spring damper system
- Goal state

\[ f = \frac{\sum_{j=1}^{N} w_j j(z)}{N} \]

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003
Dynamical System Motor Primitives

\[ f = \sum_{j=1}^{N} \frac{w_j}{N} j(z)z \]

Ispeert Model
- start state
- external force
- current state
- spring damper system
- goal state

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003
Dynamical System Motor Primitives

Canonical system – action synchronization
\[ \dot{z} = z \]

Transformed system – desired state
\[ \dot{v} = \dot{y} \left( y(g, y) v \right) + f(z) \]
\[ \dot{y} = v \]

Transformation function
\[ f = \sum_{j=1}^{N} w_j j(z) z \]

Ispeert Model
- start state
- external force
- current state
- spring damper system
- goal state

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003
Dynamical System Motor Primitives

\[ \dot{v} = y \left( y(g - y) - v \right) + h f(z) \]

\[ \dot{y} = v \]

Final Trajectory

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003
Why do we love them?

- Arbitrarily shaped smooth movements
- It is possible to include object avoidance
- Simple to adapt
- Stable and robust
- Linear in parameters $\mathbf{w}$
  - straightforward to learn by imitation learning
  - well suited for reinforcement learning

**Ispeert Model**

- start state
- external force
- current state
- spring damper system
- goal state

S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert
Learning Motor Primitives, ISRR, 2003
Dynamical System Motor Primitives

What do we gain from this representation?

Motor policy representation that performs an automatic mapping of states to actions over time

\[ \pi_w \begin{pmatrix} g \\ \theta_t \\ \dot{\theta}_t \\ t \end{pmatrix} = a_{t+1} \]

Mapping depends on shape parameters $w$
Dynamical System Primitives

How to learn in multi-dimensional spaces?

Task Space Control

Canonical System

Transformed System

\[ p_x \]  \[ p_y \]  \[ p_z \]
Dynamical System Primitives

How to learn in multi-dimensional spaces?

Task Space Control
Dynamical System Primitives

Video: Peter Pastor

Pastor et al.: Learning and generalization of motor skills by learning from demonstration
How can we learn a manipulation tasks?

Step 1: Representation and Initialization of Behavior

Step 2: Learning from demonstration
Learning movements from demonstration

... using DMPs

**Goal**: Get data set of state-action pairs from teacher such that we can learn a hitting movement!

First: we need a demonstration – how do we do this?
- Observing human performing task: Correspondence problem.
- Teleoperation of end-effector: Hard for human to hit ball.
- Kinesthetic teach-in: needs a little exercise.
- Verbal: Very difficult.
Second: How do we learn it?

- Reward $R$
- Reinforcement Learning, Optimal Control
- Dynamical Model $T$
- Inverse Reinforcement Learning
- Control Policy $\pi$
- Behavioral Cloning
- Expert Demonstration
Initialization of Behavior

Learning from Imitation

- Starting point for robot learning
- Goal: reproduce the demonstrated behavior
- Record a movement from demonstration
- Parameter can be estimated with Locally Weighted Regression (LWR)

A. Ijspeert, J. Nakanishi, S. Schaal
Movement imitation with nonlinear dynamical systems in humanoid robot, ICRA 2002
Learning from Demonstration

**Input:** \([\theta_t, \dot{\theta}_t, \ddot{\theta}_t], t \in \{1, ..., T_f\}\)

For all DoF \(i\) and each parameter \(w_n\) do

Set \(g = \theta_{T_f}\)

Calculate \(z_t\) by integrating \(\tau \dot{z} = \alpha_z z\) for all \(t\)

Calculate \(\psi^m_t = \exp(\rho_n(z_t - \mu_n)^2)\)

Calculate reference value \(f_{t}^{\text{ref}}\) from demonstration

\[f_{t}^{\text{ref}} = (\tau^2 \ddot{\theta}_t - \alpha_y (\beta_y (g - \theta_t) - \tau \dot{\theta}_t))/\eta\]

Create matrices \(z = [z_1, ..., z_{T_f}]'\), \(\Psi = \text{diag}(\psi_1^n, ..., \psi_{T_f}^n)\), \(f_{\text{ref}} = [f_{1}^{\text{ref}}, ..., f_{T_f}^{\text{ref}}]\)

Compute weights via locally weighted regression

\(w_n = (Z^T \Psi Z)^{-1} Z^T \Psi f_{\text{ref}}\)
Initialization of Behavior

Learning from Imitation

• Starting point for robot learning
• Goal: reproduce the demonstrated behavior
• Record a movement from demonstration
• Parameter can be estimated with Locally Weighted Regression (LWR)

A. Ijspeert, J. Nakanishi, S. Schaal
Movement imitation with nonlinear dynamical systems in humanoid robot, ICRA 2002
Why is this not enough?

• Variety in hitting points
• Variety in approaching directions
• Variety in return directions
Initialization of Behavior

Learning from Imitation

Limitations:
• Generalization of DMP is limited
• Undemonstrated situations
• Quality of demonstrations
• Rigid policy

Overcoming the limitations:
• Collect demonstrations for different situations
• Learn mapping from situation to primitive
• Generate new skills from existing ones
How can we learn a manipulation task?

Step 1: Representation and Initialization of Behavior

Step 2: Learning from demonstration

Step 3: Selecting and Generalization of Behavior
Mixture of Motor Primitives

- Select and generalize among motor primitives
- Each MP is associated with an external stimulus

\[ \pi(\tilde{s}) = \sum_{i=1}^{C} \alpha_i(\tilde{s}) \pi_i(\tilde{s}) \]

K. Muelling, J. Kober, O. Kroemer, J. Peters
Learning to select and generalize striking movements in robot table tennis, IJRR, 2013
Selecting and Generalization of a Skill

Learn to associate situation with primitive

- Start with all having the same probability
- Update probability after observing $\langle s, a, r \rangle$
Reinforcement Learning for Adaptation

Idea  weight the contribution of the single MPs

Learn weights $\alpha_i$ of MP $i$

$w_i(s) = Z \exp \left\{ T \sum_i \alpha_i(s) \right\}$

Initiate: $\alpha_i = 1$

Update:

$\lambda_i = \left( \Phi^T W \Phi + \beta I \right)^{-1} \Phi^T W r$

K. Muelling, J. Kober, O. Kroemer, J. Peters
Learning to select and generalize striking movements in robot table tennis, IJRR, 2013
What is the reward?
From Imitation Learning we obtain 25 Movement Primitives
Playing Table Tennis

Evaluation on the real system

• Recorded 25 hitting motions on the real system

• High variation in their overall performance

• Using the MoMP, we could reduce the performance error

• Increased performance with increasing number of MPs

K. Muelling, J.Kober, O. Kroemer, J. Peters
Learning to select and generalize striking movements in robot table tennis, IJRR, 2013
Playing Table Tennis

Distribution of Primitives

K. Muelling, J. Kober, O. Kroemer, J. Peters
Learning to select and generalize striking movements in robot table tennis, IJRR, 2013
Summary

Imitation Learning:

• Learning from demonstration is a great tool to initiate learning and to make learning on real robots possible.
• Representing movements with DMPs allow to efficiently learn movements from demonstration and through self improvement.
• When learning from demonstration keep in mind:
  – What you want to learn.
  – Is it possible to map human demonstration to robot learner?
  – Does it make sense to map human demonstration to the robot?
  – There are different ways to learn from demonstration.
Summary

Inverse Reinforcement Learning vs Behavioral Cloning

Can we directly learn the policy?

Formulated as supervised learning problem:
1) Fix policy class
2) Find suitable ML
3) Learn policy directly from demonstrations
Reinforcement Learning

Reinforcement learning offers to robotics a framework and set of tools for the design of sophisticated and hard-to-engineer behaviors. Conversely, the challenges of robotic problems provide both inspiration, impact, and validation for developments in reinforcement learning. The relationship between disciplines has sufficient promise to be likened to that between physics and mathematics. In this article, we attempt to strengthen the links between the two research communities by providing a survey of work in reinforcement learning for behavior generation in robots. We highlight both key challenges in robot reinforcement learning as well as notable successes. We discuss how contributions tamed the complexity of the domain and study the role of algorithms, representations, and prior knowledge in achieving these successes. As a result, a provides feedback in terms of a scalar objective function that measures the one-step performance of the robot. Figure 1 illustrates the diverse set of robots that have learned tasks using reinforcement learning. Consider, for example, attempting to train a robot to return a table tennis ball over the net (Muelling et al., 2012). In this case, the robot might make an observations of dynamic variables specifying ball position and velocity and the internal dynamics of the joint position and velocity. This might in fact capture well the state $s$ of the system – providing a complete statistic for predicting future observations. The actions $a$ available to the robot might be the torque sent to motors or the desired accelerations sent to an inverse dynamics control system. A function $a$ that generates the motor commands (i.e., the actions) based on the
