ICRA 2012 Tutorial on Reinforcement Learning

I. Introduction

Pieter Abbeel
UC Berkeley

Jan Peters
TU Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Motivational Example: Helicopter Control

- Unstable
- Nonlinear
- Complicated dynamics
  - Air flow
  - Coupling
  - Blade dynamics
- Noisy estimates of position, orientation, velocity, angular rate (and perhaps blade and engine speed)

Many success stories in hover and forward flight regime

- Just a few examples: Bagnell & Schneider, 2001; LaCivita, Papageorgiou, Messner & Kanade, 2002; Ng, Kim, Jordan & Sastry 2004a (2001); Roberts, Corke & Buskey, 2003; Saripalli, Montgomery & Sukhatme, 2003; Shim, Chung, Kim & Sastry, 2003; Doherty et al., 2004; Gavrilets, Martinos, Mettler and Feron, 2002; Ng et al., 2004b.
- Varying control techniques: inner/outer loop PID with hand or automatic tuning, $H\infty$, LQR, …

Using adaptation of state-of-the-art hover control techniques

Target trajectory: meticulously hand-engineered
Model: from (commonly used) frequency sweeps data

Stationary vs. aggressive flight

- Hover / stationary flight regimes:
  - Restrict attention to specific flight regime
  - Extensive data collection = collect control inputs, position, orientation, velocity, angular rate
  - Build model + model-based controller

→ Successful autonomous flight.

- Aggressive flight maneuvers --- additional challenges:
  - **Task description**: What is the target trajectory? [to regulate around]
  - **Dynamics model**: How to build a dynamics model sufficiently accurate to enable feedback control through non-stationary flight regimes?

Aggressive, non-stationary regimes

- Gavrilets, Martinos, Mettler and Feron, 2002
  - → 3 maneuvers: split-S, snap axial roll, stall-turn
  - → Took a PhD to get 3 maneuvers done.

Motivational Example 2: Robot Ping Pong

Motivational Example 2: Robot Ping Pong

- “Batman”
- Robot Ping-Pong world champion of 1993
- Took about 100 man years
  - more than 50 students worked on this from 1985 to 1997

Motivation

- Hand-engineering for a particular problem can make signficant headway on that problem

  ….

  but can be extremely laborious

- In this tutorial: Learning methods
  - general applicability
  - have already enabled robotic success stories of equal and higher quality with far less man-years

Outline of Tutorial and Dependencies

- Session I:
  - 1 Introduction (PA)
  - 2 Background: Supervised Learning (JP)
  - 3a Optimal Control: Foundations (PA)
- Session II:
  - 3b (requires: 2, 3a) Optimal Control: Advanced (JP)
  - 4 (requires: 3a) Value Function Methods (PA)
- Session III:
  - 5 Policy Search (JP)
  - 6 (requires: 4) Exploration (PA)
  - 7 Wrap-up (both)

Format

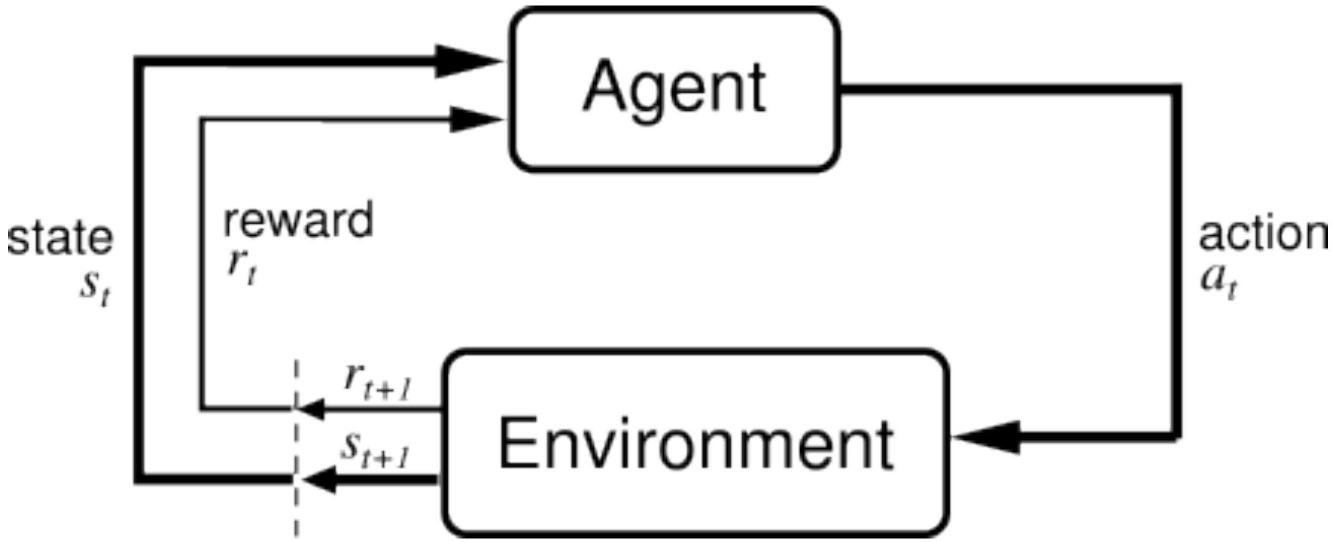
- Interleaving of some online exercises
  - Icra2012-rl.org

  Sign up now!

  Let's do Exercise 0 now!

- Optional: programming project over lunch break!

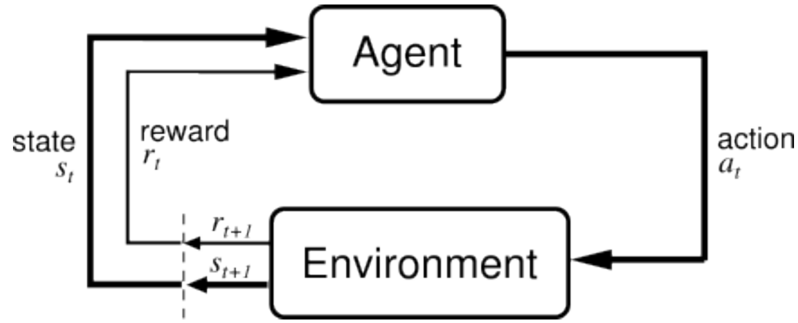
Markov Decision Process



Assumption: agent gets to observe the state

[Drawing from Sutton and Barto, Reinforcement Learning: An Introduction, 1998]

Markov Decision Process (X, U, T, R, γ, H)



Given

- X: set of states
- U: set of actions
- T: T(x,u,x') = P(x_{t+1} = x' | x_t = x, u_t =u)
- R: R(x,u) = reward for (x_t = x, u_t =u)
- $\gamma \in$ [0,1], discount factor
- H: horizon over which the agent will act

Goal:

- Find π : X x {0, 1, …, H} → U that maximizes expected sum of rewards, i.e.,

$$\pi^* = \arg\max_{\pi} \mathrm{E}[\sum_{t=0}^{H} R(X_t, U_t)|\pi]$$

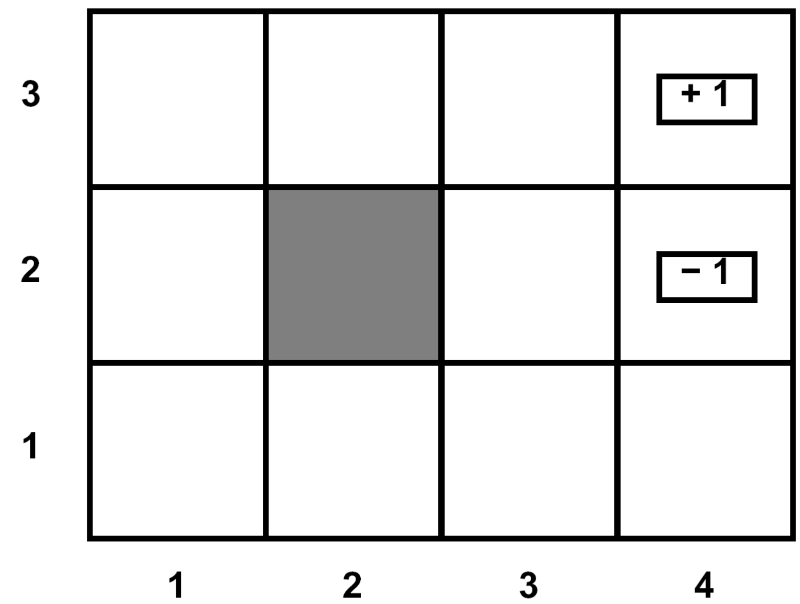
Examples

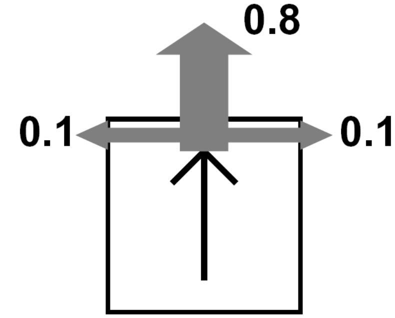
MDP (X, U, T, R, H), goal: $max_{\pi} \mathrm{E}[\sum_{t=0}^{H} R(X_t, U_t)|\pi]$

- Cleaning robot
- Walking robot
- Pole balancing
- Games: tetris, backgammon
- Server management
- Shortest path problems
- Models for animals, people

Canonical Example: Grid World

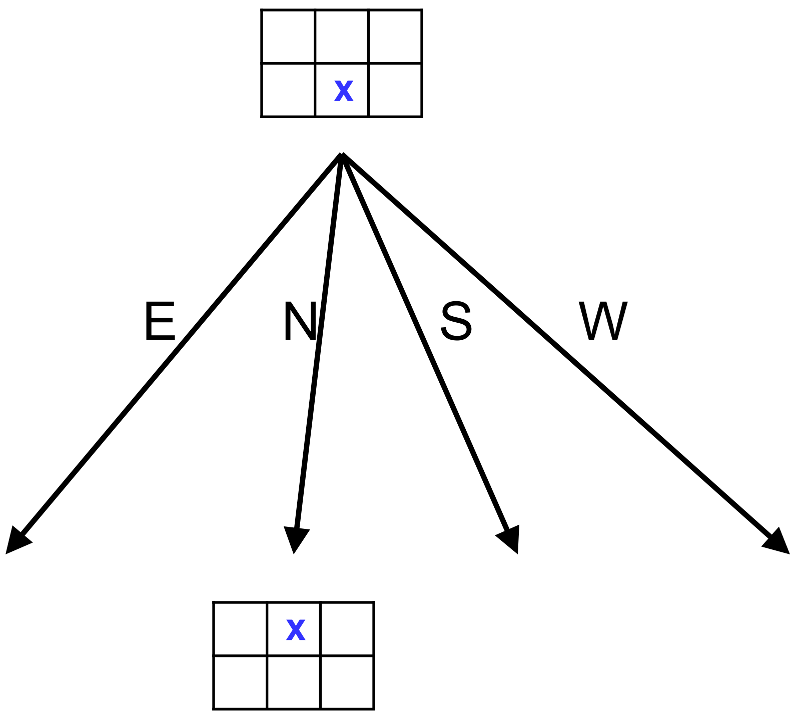
- The agent lives in a grid
- Walls block the agent's path
- The agent's actions do not always go as planned:
  - 80% of the time, the action North takes the agent North (if there is no wall there)
  - 10% of the time, North takes the agent West; 10% East
  - If there is a wall in the direction the agent would have been taken, the agent stays put
- Big rewards come at the end



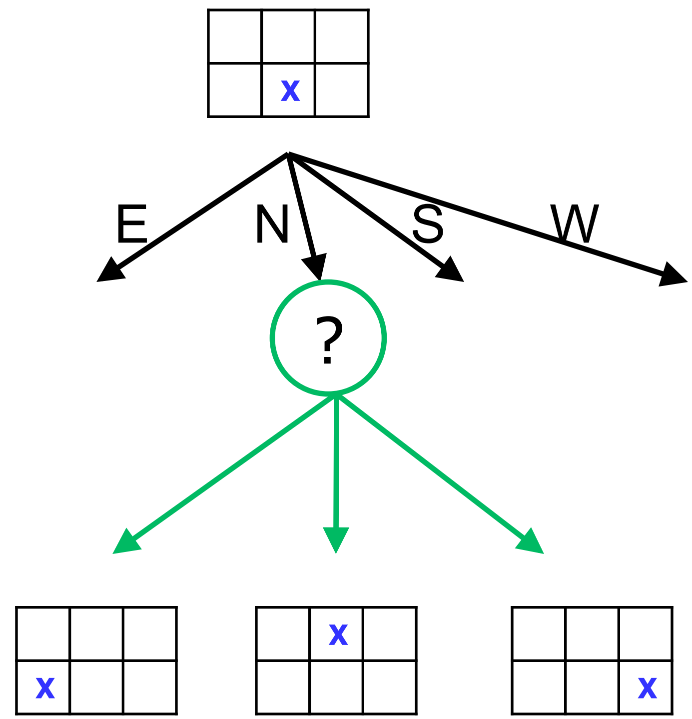


Grid Futures

Deterministic Grid World

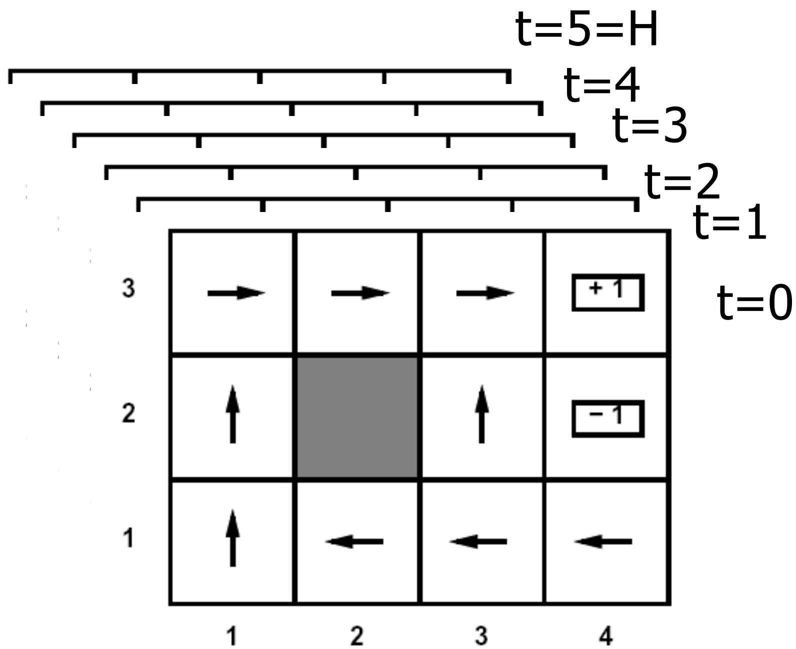


Stochastic Grid World



Solving MDPs

- In an MDP, we want an optimal policy $\pi^*: X \times 0{:}H \rightarrow U$
  - A policy π gives an action for each state for each time



  - An optimal policy maximizes expected sum of rewards
- If deterministic: want an optimal plan, or sequence of actions

Solving MDPs when H=∞

- When H=∞, at any given time there are infinitely many time steps left

→ Stationary optimal policy

  i.e., optimal policy does not depend on time

- In practice rarely truly H=∞, but still often used
  - If H sufficiently large, solution will be similar, and H=∞ solution is more compact
  - If H is unknown, H=∞ might be a reasonable choice
  - Some of the math for some solution methods happens to work nicely for H=∞

A Reinforcement Learning Ontology