Artificial Intelligence

Introduction

Marc Toussaint
University of Stuttgart
Winter 2018/19

(some slides based on Stuart Russell's AI course)

What is intelligence?

What is intelligence?

- Maybe it is easier to first ask what systems we actually talk about:
  - Decision making
  - Interacting with an environment

What is intelligence?

- Maybe it is easier to first ask what systems we actually talk about:
  - Decision making
  - Interacting with an environment

- Then define objectives!
  - Quantify what you consider good or successful
  - Intelligence means to optimize...

Intelligence as Optimization?

- A cognitive scientist or psychologist: “Why are you AI people always so obsessed with optimization? Humans are not optimal!”

Intelligence as Optimization?

- A cognitive scientist or psychologist: “Why are you AI people always so obsessed with optimization? Humans are not optimal!”

- That’s a total misunderstanding of what “being optimal” means.
- Optimization principles are a means to describe systems:
  - Feynman’s “unworldliness measure” objective function
  - Everything can be cast optimal – under *some* objective
  - Optimality principles are just a scientific means of formally describing systems and their behaviors (esp. in physics, economy, ... and AI)
  - Toussaint, Ritter & Brock: *The Optimization Route to Robotics – and Alternatives*. Künstliche Intelligenz, 2015

Intelligence as Optimization?

- A cognitive scientist or psychologist: “Why are you AI people always so obsessed with optimization? Humans are not optimal!”

- That’s a total misunderstanding of what “being optimal” means.
- Optimization principles are a means to describe systems:
  - Feynman’s “unworldliness measure” objective function
  - Everything can be cast optimal – under *some* objective
  - Optimality principles are just a scientific means of formally describing systems and their behaviors (esp. in physics, economy, ... and AI)
  - Toussaint, Ritter & Brock: *The Optimization Route to Robotics – and Alternatives*. Künstliche Intelligenz, 2015

- Generally, I would roughly distinguish three basic types of problems:
  - Optimization
  - Logical/categorial Inference (CSP, find feasible solutions)
  - Probabilistic Inference

What are interesting objectives?

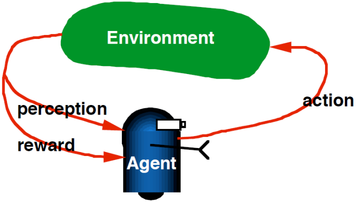
- Learn to control all degrees of freedom of the environment that are controllable
  - DOFs are mechanical/kinematics DOFs, objects, light/temperature, mood of humans
  - This objective is generic: no preferences, not limits
  - Implies to actively go exploring and finding controllable DOFs
  - Acting to Learning (instead of 'Learning to Act' for a fixed task)
  - Related notions in other fields: *(Bayesian) Experimental Design*, *Active Learning*, curiosity, intrinsic motivation

- At time T, the system will be given a random task (e.g., random goal configuration of DOFs); the objective then is to reach it as quickly as possible

More on objectives

- The value alignment dilemma
- What are objectives that describe things like “creativity”, “empathy”, etc?
- Coming up with objective functions that imply desired behavior is a core part of AI research

Interactive domains



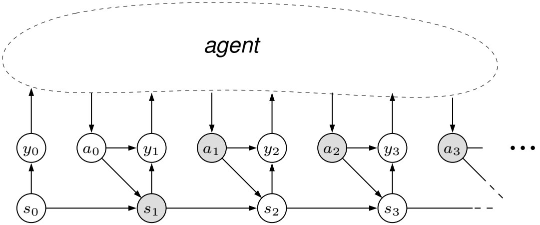
- We assume the agent is in *interaction* with a domain.
  - The world is in a state $s_t \in \mathcal{S}$ (see below on what that means)
  - The agent senses observations $y_t \in \mathcal{O}$
  - The agent decides on an action $a_t \in \mathcal{A}$
  - The world transitions to a new state s_{t+1}

- The *observation* y_t describes all information received by the agent (sensors, also rewards, feedback, etc) if not explicitly stated otherwise

(The technical term for this is a POMDP)

State

- The notion of *state* is often used imprecisely
- At any time t, we assume the world is in a state $s_t \in \mathcal{S}$
- s_t is a *state description* of a domain iff future observations $y_{t^+}, t^+ > t$ are conditionally independent of all history observations $y_{t^-}, t^- < t$ given s_t and future actions $a_{t:t^+}$:



- Notes:
  - Intuitively, s_t describes everything about the world that is "relevant"
  - Worlds do not have additional latent (hidden) variables to the state s_t

Examples

- What is a sufficient definition of *state* of a computer that you interact with?
- What is a sufficient definition of *state* for a thermostat scenario? (First, assume the 'room' is an isolated chamber.)
- What is a sufficient definition of *state* in an autonomous car case?

Examples

- What is a sufficient definition of *state* of a computer that you interact with?

- What is a sufficient definition of *state* for a thermostat scenario? (First, assume the 'room' is an isolated chamber.)

- What is a sufficient definition of *state* in an autonomous car case?

$\rightarrow$ in real worlds, the exact *state* is practically not representable
$\rightarrow$ all models of domains will have to make approximating assumptions (e.g., about independencies)

How can agents be formally described?

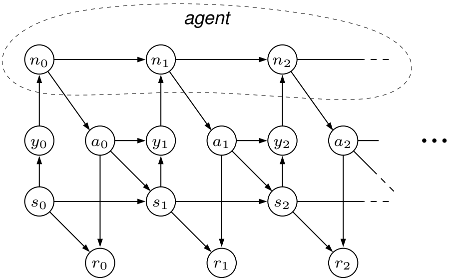
...or, what formal classes of agents do exist?

- Basic alternative agent models:
  - The agent maps $y_t \mapsto a_t$
    (**stimulus-response** mapping.. non-optimal)
  - The agent stores all previous observations and maps

    $$f: \ y_{0:t}, a_{0:t\text{-}1} \mapsto a_t$$

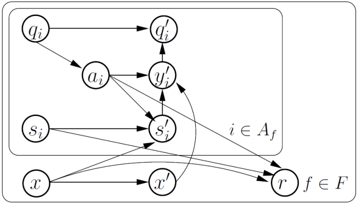
    f is called **agent function**. This is the most general model, including the others as special cases.
  - The agent stores only the recent history and maps
    $y_{t-k:t}, a_{t-k:t\text{-}1} \mapsto a_t$ (crude, but may be a good heuristic)
  - The agent is some machine with its own **internal state** n_t, e.g., a computer, a finite state machine, a brain... The agent maps $(n_{t\text{-}1}, y_t) \mapsto n_t$ (internal state update) and $n_t \mapsto a_t$
  - The agent maintains a full probability distribution (**belief**) $b_t(s_t)$ over the state, maps $(b_{t\text{-}1}, y_t) \mapsto b_t$ (Bayesian belief update), and $b_t \mapsto a_t$

POMDP coupled to a state machine agent



Multi-agent domain models

(The technical term for this is a Decentralized POMDPs)



(from Kumar et al., IJCAI 2011)

- This is a special type (simplification) of a general DEC-POMDP
- Generally, this level of description is very general, but NEXP-hard
  Approximate methods can yield very good results, though

Summary – AI is about:

- Systems that interact with the environment
  - We distinguish between ’system’ and ’environment’ (cf. embodiment)
  - We just introduced basic models of interaction
  - A core part of AI research is to develop formal models for interaction

- Systems that aim to manipulate their invironment towards ’desired’ states (optimality)
  - Optimality principles are a standard way to describe desired behaviors
  - We sketched some interesting objectives
  - Coming up with objective functions that imply desired behavior is a core part of AI research

Organisation

Vorlesungen der Abteilung MLR

- Bachelor:
  - Grundlagen der Künstlichen Intelligenz (3+1 SWS)
- Master:
  - Vertiefungslinie Intelligente Systeme (gemeinsam mit Andres Bruhn)
  - WS: Maths for Intelligent Systems
  - WS: Introduction to Robotics
  - SS: Machine Learning
  - (SS: Optimization)
  - (Reinforcement Learning), (Advanced Robotics)
  - Practical Course Robotics (SS)
  - (Hauptseminare: Machine Learning (WS), Robotics (SS))

Andres Bruhn's Vorlesungen in der Vertiefungslinie

- WS: Computer Vision
- SS: Correspondence Problems in Computer Vision
- Hauptseminar: Recent Advances in Computer Vision

Vorraussetzungen für die KI Vorlesung

- Mathematik für Informatiker und Softwaretechniker
- außerdem hilfreich:
  - Algorithmen und Datenstrukturen
  - Theoretische Informatik

Vorlesungsmaterial

- Webseite zur Vorlesung:
  `https://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/`
  die Folien und Übungsaufgaben werden dort online gestellt

- Alle Materialien des letzten Jahres sind online – bitte machen Sie sich einen Eindruck

- Hauptliteratur:
  Stuart Russell & Peter Norvig: Artificial Intelligence A Modern Approach
  - Many slides are adopted from Stuart

Prüfung

- Schriftliche Prüfung, 90 Minuten
- Termin zentral organisiert
- keine Hilfsmittel erlaubt
- Anmeldung: Im LSF / beim Prüfungsamt
- Prüfungszulassung:
  - 50% der Punkte der Programmieraufgaben
  - UND 50% der Votieraufgaben

Übungen

- 8 Übungsgruppen (4 Tutoren)
- 2 Arten von Aufgaben: Coding- und Votier-Übungen
- Coding-Aufgaben: Teams von bis zu 3 Studenten geben die Coding-Aufgaben zusammen ab
- Votier-Aufgaben:
  - Zu Beginn der Übung eintragen, welche Aufgaben bearbeiten wurden/präsentiert werden können
  - Zufällige Auswahl
- Schein-Kriterium:
  - 50% der Punkte der Programmieraufgaben
  - UND 50% der Votieraufgaben

- **Registrierung**

  `https://ipvs.informatik.uni-stuttgart.de/mlr/teaching/course-registration/`