Artificial Intelligence

Lesson01 Introduction

Acknowledgment: These materials are adopted from the slides created by © Dan Klein and Pieter Abbeel, University of California, Berkeley for CS188 Introduction to AI at UC Berkeley. All materials available at http://ai.berkeley.edu.
Course Information

- Introduction to Artificial Intelligence
- Workshop - 4-day online Tutorial MTWTh 10am-1pm
  - December 21-24 MTWTh 10am-1pm

- Day1 – Search as Problem Solving, Heuristics and A*, Maze game
- Day2 – Minimax Alpha-Beta Pruning, Game with Adversary, TicTacToe, ...
- Day3 – Constraint Satisfaction Problem, Logic and Inference, Puzzle, ...
- Day4 – Machine Learning and Data Mining, Weka, Decision Tree

- The course materials (ppts, videos, tools, programs, etc.) are from
  - http://aima.cs.berkeley.edu
  - http://ai.berkeley.edu
Textbook - Optional

  - [http://aima.cs.berkeley.edu/](http://aima.cs.berkeley.edu/)
  - [http://ai.berkeley.edu](http://ai.berkeley.edu)

  - The Textbook is available online: [https://artint.info/html/ArtInt.html](https://artint.info/html/ArtInt.html)
  - Video tutorial by Alan Mackworth: [https://www.youtube.com/watch?v=mo0gmLMC72E](https://www.youtube.com/watch?v=mo0gmLMC72E)
http://aima.cs.berkeley.edu/
Welcome to CS188!

Thank you for your interest in our materials developed for UC Berkeley's introductory artificial intelligence course, CS 188. In the navigation bar above, you will find the following:

- A sample course schedule from Spring 2014
- Complete sets of Lecture Slides and Videos
- Interface for Electronic Homework Assignments
- Section Handouts
- Specs for the Pacman Projects
- Source files and PDFs of past Berkeley CS188 exams
- Form to apply for edX hosted autograders for homeworks and projects (and more)
- Contact information
Today

- What is artificial intelligence?
- What can AI do?
- What is this tutorial?
Sci-Fi AI?
What is AI?

The science of making machines that:

- Think like people
- Act like people
- Think rationally
- Act rationally
Rational Decisions

We’ll use the term rational in a very specific, technical way:

- Rational: maximally achieving pre-defined goals
- Rationality only concerns what decisions are made (not the thought process behind them)
- Goals are expressed in terms of the utility of outcomes
- Being rational means maximizing your expected utility

A better title for this course would be:

Computational Rationality
What is Computer? Program?

Computer is a machine (that is, man-made, designed with purpose).

Modern Computer: Electronic, Programmable machine with stored program

What is a Program? or an Algorithm?

Algorithm is a finite sequence of well-defined instructions,
(usually computer-implementable)
typically to solve a class of problems or to perform a computation,
to produce output (solution), given input if any.

A Program is an implementation of An Algorithm using programming language
(e.g., Java, C++, Python, Javascript, etc.)
Intelligent Agents

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.

Intelligence – Does it further demand a moral or ethical dimension? Immoral, Unethical, Evil … vs Moral, Ethical, Good, …

To Know it? To Act upon it if it knows what to do (ought to do)?

3 Pillars of Philosophy: (1) Epistemology (2) Ontology (3) Ethics

Intelligent? (and not so stupid)

Natural Intelligence versus Man-made or Artificial Intelligence
What is Robot?

What is a Robot?

Consider a car-washing machine.

Programmable, Interacting with its environment, Problem-Solving

Basic Instructions (actions): Rinse, Soap, Brushing, Wax, Dry

(1) Basic Car Wash $5 – Rinse, Soap, Brush, Rinse.
(2) Deluxe Car Wash $10 – Rinse, Soap, Brush, Rinse, Wax, Dry
(3) Supreme Car Wash $15 – Rinse, RepeatTwice(Soap, Brush, Rinse), Wax, Dry
Structure of Intelligent Agents

Agents are built with programs running on some hardware. AI systems may be general-purpose computers, or special-purpose computers.

Agent programs are very complex, because intelligence is complex. Simple look up tables mapping percepts into actions will not work.

Example:
- A chess player would require $35^{100}$ entries ($10^{70-80}$)
- Go game would require $10^{300-400}$
Specifying the task environment

- Problem specification:
  Performance measure, Environment, Actuators, Sensors (PEAS)

Example: automated taxi driver:

1. Performance measure
   - Safe, fast, legal, comfortable trip, maximize profits
2. Environment
   - Roads, other traffic, pedestrians, customers
3. Actuators
   - Steering wheel, accelerator, brake, signal, horn
4. Sensors
   - Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard
**Example: Automated Taxi Driver**

<table>
<thead>
<tr>
<th>Agent Type</th>
<th>Performance Measure</th>
<th>Environment</th>
<th>Actuators</th>
<th>Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taxi driver</td>
<td>Safe, fast, legal, comfortable trip, maximize profits</td>
<td>Roads, other traffic,</td>
<td>Steering, accelerator, brake,</td>
<td>Cameras, sonar, speedometer, GPS, odometer,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pedestrians, customers</td>
<td>signal, horn, display</td>
<td>accelerometer, engine sensors, keyboard</td>
</tr>
</tbody>
</table>
### Agent Types and their Descriptions

<table>
<thead>
<tr>
<th>Agent Type</th>
<th>Performance Measure</th>
<th>Environment</th>
<th>Actuators</th>
<th>Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medical diagnosis system</td>
<td>Healthy patient, minimize costs, lawsuits</td>
<td>Patient, hospital, staff</td>
<td>Display questions, tests, diagnoses, treatments, referrals</td>
<td>Keyboard entry of symptoms, findings, patient’s answers</td>
</tr>
<tr>
<td>Satellite image analysis system</td>
<td>Correct image categorization</td>
<td>Downlink from orbiting satellite</td>
<td>Display categorization of scene</td>
<td>Color pixel arrays</td>
</tr>
<tr>
<td>Part-picking robot</td>
<td>Percentage of parts in correct bins</td>
<td>Conveyor belt with parts; bins</td>
<td>Jointed arm and hand</td>
<td>Camera, joint angle sensors</td>
</tr>
<tr>
<td>Refinery controller</td>
<td>Maximize purity, yield, safety</td>
<td>Refinery, operators</td>
<td>Valves, pumps, heaters, displays</td>
<td>Temperature, pressure, chemical sensors</td>
</tr>
<tr>
<td>Interactive English tutor</td>
<td>Maximize student’s score on test</td>
<td>Set of students, testing agency</td>
<td>Display exercises, suggestions, corrections</td>
<td>Keyboard entry</td>
</tr>
</tbody>
</table>
Environment types

- **Fully observable (vs. partially observable):** The agent's sensors give it access to the complete state of the environment at each point in time.

- **Deterministic (vs. stochastic):** The next state of the environment is completely determined by the current state and the agent’s action.
  - **Strategic:** the environment is deterministic except for the actions of other agents.

- **Episodic (vs. sequential):** The agent's experience is divided into atomic “episodes,” and the choice of action in each episode depends only on the episode itself.
Environment types

- **Static (vs. dynamic):** The environment is unchanged while an agent is deliberating
  - **Semidynamic:** the environment does not change with the passage of time, but the agent's performance score does

- **Discrete (vs. continuous):** The environment provides a fixed number of distinct percepts, actions, and environment states
  - Time can also evolve in a discrete or continuous fashion

- **Single agent (vs. multi-agent):** An agent operating by itself in an environment

- **Known (vs. unknown):** The agent knows the rules of the environment
## Properties of environments: examples

<table>
<thead>
<tr>
<th>Task Environment</th>
<th>Observable</th>
<th>Deterministic</th>
<th>Episodic</th>
<th>Static</th>
<th>Discrete</th>
<th>Agents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossword puzzle</td>
<td>Fully</td>
<td>Deterministic</td>
<td>Sequential</td>
<td>Static</td>
<td>Discrete</td>
<td>Single</td>
</tr>
<tr>
<td>Chess with a clock</td>
<td>Fully</td>
<td>Strategic</td>
<td>Sequential</td>
<td>Semi</td>
<td>Discrete</td>
<td>Multi</td>
</tr>
<tr>
<td>Poker</td>
<td>Partially</td>
<td>Strategic</td>
<td>Sequential</td>
<td>Static</td>
<td>Discrete</td>
<td>Multi</td>
</tr>
<tr>
<td>Backgammon</td>
<td>Fully</td>
<td>Stochastic</td>
<td>Sequential</td>
<td>Static</td>
<td>Discrete</td>
<td>Multi</td>
</tr>
<tr>
<td>Taxi driving</td>
<td>Partially</td>
<td>Stochastic</td>
<td>Sequential</td>
<td>Dynamic</td>
<td>Continuous</td>
<td>Multi</td>
</tr>
<tr>
<td>Medical diagnosis</td>
<td>Partially</td>
<td>Stochastic</td>
<td>Sequential</td>
<td>Dynamic</td>
<td>Continuous</td>
<td>Single</td>
</tr>
<tr>
<td>Image-analysis</td>
<td>Fully</td>
<td>Deterministic</td>
<td>Episodic</td>
<td>Semi</td>
<td>Continuous</td>
<td>Single</td>
</tr>
<tr>
<td>Part-picking robot</td>
<td>Partially</td>
<td>Stochastic</td>
<td>Episodic</td>
<td>Dynamic</td>
<td>Continuous</td>
<td>Single</td>
</tr>
<tr>
<td>Refinery controller</td>
<td>Partially</td>
<td>Stochastic</td>
<td>Sequential</td>
<td>Dynamic</td>
<td>Continuous</td>
<td>Single</td>
</tr>
<tr>
<td>Interactive English tutor</td>
<td>Partially</td>
<td>Stochastic</td>
<td>Sequential</td>
<td>Dynamic</td>
<td>Discrete</td>
<td>Multi</td>
</tr>
</tbody>
</table>
The Structure of Intelligent Agents

The question facing the designer is:
How to structure (architect) such an agent?

Four types of agents (in increasing complexity order)

1. Simple reflex agents
2. Agents that keep track of the world
3. Goal-based agents
4. Utility-based agents
Simple reflex agents

- These are implemented with condition-action rules.
  
  \[ \text{if condition then action} \]

  For example: \[ \text{if car-in-front-is-braking then initiate-braking} \]

- Rule conditions are compared with the current situation to determine which rule is applicable; once found, a rule is applied, meaning that some action is triggered.

- Environments need to be fully observable.
Simple reflex agents:
Agents that keep track of the world (agents with internal states)

A reflex agent with internal state.

It works by finding a rule whose condition matches the current situation (as defined by the percept and the stored internal state) and then doing the action associated with that rule.

```
function REFLEX-AGENT-WITH-STATE(percept) returns an action
  static: state, a description of the current world state
           rules, a set of condition-action rules
           action, the most recent action, initially none
  state ← UPDATE-STATE(state, action, percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION[rule]
  return action
```
Goal-based agents: Ability to do Searching

- Another enhancement is to give the agent a goal to look for.
- The agent actions constitute a sequence that leads to the goal.
  - Application examples: Searching, Planning.
- Goal information describes desirable situations

- The actions now are not provided by if-then rules, but they are selected such that will bring the system closer to the goal.
- States are still necessary.
- Note that the way actions are selected to get closer to the goal, may still use if-then rules (but not exclusively).
Goal-based agents: Ability to do Searching

Agent

- State
  - How the world evolves
  - What my actions do

Goals

Environment

- Sensors
  - What the world is like now
- Actuators
  - What it will be like if I do action A
  - What action I should do now
Utility-based agents: Ability to Assess Better

The idea is to provide a metric to assist the agent in moving toward the goal faster. A utility function maps the states into a performance number.
Learning Agents: Ability to Learn

- Ability to learn from
  - Experience
  - Reason

To gain Knowledge by
- Induction or Deduction
- Rational Knowledge

Data, Information, Knowledge, Wisdom, ...

Level of Knowledge
Maximize Your Expected Utility
What About the Brain?

- Brains (human minds) are very good at making rational decisions, but not perfect.
- Brains aren’t as modular as software, so hard to reverse engineer!
- “Brains are to intelligence as wings are to flight”
- Lessons learned from the brain: memory and simulation are key to decision making.

Do not undermine your memory. Your brain (memory) is more than a simple read-write device.
A (Short) History of AI

Demo: HISTORY – MT1950.wmv
A (Short) History of AI

- 1940-1950: Early days
  - 1943: McCulloch & Pitts: Boolean circuit model of brain
  - 1950: Turing’s “Computing Machinery and Intelligence”

- 1950—70: Excitement: Look, Ma, no hands!
  - 1950s: Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
  - 1956: Dartmouth meeting: “Artificial Intelligence” adopted
  - 1965: Robinson's complete algorithm for logical reasoning

- 1970—90: Knowledge-based approaches
  - 1969—79: Early development of knowledge-based systems
  - 1980—88: Expert systems industry booms

- 1990—: Statistical approaches
  - Resurgence of probability, focus on uncertainty
  - General increase in technical depth
  - Agents and learning systems... “AI Spring”?

- 2010—Deep Learning. Self-Driving Car, IBM Watson, Alpha-Go
- 2020—Where are we now?
What Can AI Do?

Quiz: Which of the following can be done at present?

- ✔ Play a decent game of table tennis?
- ✔ Play a decent game of Jeopardy?
- ✔ Drive safely along a curving mountain road?
- ✔ Drive safely along Telegraph Avenue?
- ✔ Buy a week's worth of groceries on the web?
- ✔ Buy a week's worth of groceries at Berkeley Bowl?
- 🔴 Discover and prove a new mathematical theorem?
- 🔴 Converse successfully with another person for an hour?
- 🔴 Perform a surgical operation?
- ✔ Put away the dishes and fold the laundry?
- ✔ Translate spoken Chinese into spoken English in real time?
- 🔴 Write an intentionally funny story?
Unintentionally Funny Stories

- One day Joe Bear was hungry. He asked his friend Irving Bird where some honey was. Irving told him there was a beehive in the oak tree. Joe walked to the oak tree. He ate the beehive. The End.

- Henry Squirrel was thirsty. He walked over to the river bank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. Gravity drowned. The End.

- Once upon a time there was a dishonest fox and a vain crow. One day the crow was sitting in his tree, holding a piece of cheese in his mouth. He noticed that he was holding the piece of cheese. He became hungry, and swallowed the cheese. The fox walked over to the crow. The End.

[Roger Shank, Tale-Spin System, 1984]
Natural Language

- Speech technologies (e.g. Siri)
  - Automatic speech recognition (ASR)
  - Text-to-speech synthesis (TTS)
  - Dialog systems

- Language processing technologies
  - Question answering
  - Machine translation
  - Web search
  - Text classification, spam filtering, etc...

IBM Watson
...
Amazon Alexa
Google, Microsoft …
Vision (Perception)

- Object and face recognition
- Scene segmentation
- Image classification

Images from Erik Sudderth (left), wikipedia (right)

Demo1: VISION – lec_1_t2_video.flv
Demo2: VISION – lec_1_obj_rec_0.mpg
Robotics

- Robotics
  - Part mech. eng.
  - Part AI
  - Reality much harder than simulations!

- Technologies
  - Vehicles
  - Rescue
  - Soccer!
  - Lots of automation...

- In this class:
  - We ignore mechanical aspects
  - Methods for planning
  - Methods for control

Images from UC Berkeley, Boston Dynamics, RoboCup, Google
Logic

- **Logical systems**
  - Theorem provers
  - NASA fault diagnosis
  - Question answering

- **Methods:**
  - Deduction systems
  - Constraint satisfaction
  - Satisfiability solvers (huge advances!)

Image from Bart Selman
Game Playing

- **Classic Moment: May, '97: Deep Blue vs. Kasparov**
  - First match won against world champion
  - “Intelligent creative” play
  - 200 million board positions per second
  - Humans understood 99.9 of Deep Blue's moves
  - Can do about the same now with a PC cluster

- **Open question:**
  - How does human cognition deal with the search space explosion of chess?
  - Or: how can humans compete with computers at all??

- **1996: Kasparov Beats Deep Blue**
  “I could feel --- I could smell --- a new kind of intelligence across the table.”

- **1997: Deep Blue Beats Kasparov**
  “Deep Blue hasn't proven anything.”

- **Huge game-playing advances recently, e.g. in Go!**

Text from Bart Selman, image from IBM's Deep Blue pages
Decision Making

- Applied AI involves many kinds of automation
  - Scheduling, e.g. airline routing, military
  - Route planning, e.g. Google maps
  - Medical diagnosis
  - Web search engines
  - Spam classifiers
  - Automated help desks
  - Fraud detection
  - Product recommendations
  - ... Lots more!
An **agent** is an entity that *perceives* and *acts*.

A **rational agent** selects actions that maximize its (expected) *utility*.

Characteristics of the **percepts, environment, and action space** dictate techniques for selecting rational actions.

**This course** is about:
- General AI techniques for a variety of problem types
- Learning to recognize when and how a new problem can be solved with an existing technique
Pac-Man as an Agent

Pac-Man is a registered trademark of Namco-Bandai Games, used here for educational purposes

Demo1: pacman-l1.mp4 or L1D2
Topics & Highlights

- Part I: Making Decisions
  - Search and Heuristics
  - Constraint satisfaction
  - Adversarial and uncertain search

- Part II: Reasoning with Logic
  - Propositional and First Order Logic
  - Automated Theorem Proving (ATP)
  - Planning and Knowledge Representation

- Part III: Reasoning under Uncertainty
  - Probability and Bayes’ nets
  - Decision theory, Machine learning

- Throughout: Applications
  - Search & problem solving, Games, CSP, ATP, natural language, vision, robotics, ...