

Today's lecture:

- What is AI
- History

Administrative announcements:

- Lectures will move to Mondays/Wednesdays

What is AI?

What is intelligence? What is artificial intelligence? What is an agent? What attributes does an agent have?

There are 2 different “camps” in defining AI.

- (A) the “neats”: Logic, set theory and things like that
- (B) the “scruffies”: Experimental sciences

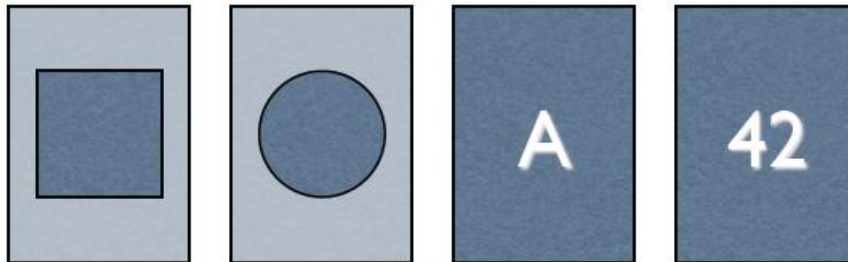
They take different approaches: top-down vs. bottom-up.

Say we want to mimic human behavior using a computer program, say – play chess like a human. (A) would define the problem and come up with an algorithm to solve it – from the bottom up. (B) would create some heuristics, like a program that tries to play chess, and tweak it until it becomes good without understanding what’s below. Then they will dissect the program and understand why is it intelligent. That’s more a top-bottom approach.

Comment: Intro AI covers the neat AI methodology – the (A) approach.

Advanced AI will get into probabilistic methods, which came from the scruffies community (B), and pattern recognition falls into this aspect.

(Philosophical) Exercise:



Which cards need to be flipped to prove whether every card with a letter on one side has a square on the other side? **2 and 3** – that’s easily proven logically. Notice that flipping card 1 is irrelevant – whether it has a letter or a number on the other side, it is ok.

There’s another example in the slides.

What is AI?

- Thinking like a human
- Acting like a human
- Thinking rationally
- Acting rationally

If an entity thinks / acts rationally, would it be considered intelligent? As it would not necessarily be **human**.

Why study AI?

There are a lot of useful AI algorithms. There's a synergy between the things humans and computers are good at, which are somewhat mutually exclusive.

Agent

- One that acts or has the power or authority to act
- Another definition in the slides...

But in the scope of AI, an agent is another way to design a system that is well-suited to certain types of problems. A lot of the problems we can solve using an agent paradigm, happen to be AI problems.

An agent has to be embedded in some (not necessarily physical) environment, it can sense from the environment and act in it. For instance:

- A robot in a physical environment
- A stock agent – a software entity in the stock market realm, that senses the market and acts (e.g. execute trades)

Classification of agents:

- Simple reflex agent – responsive agents: has a lookup table for all actions and reactions that can occur. A DFA is a way to implement such an agent. For instance, a smoke detector.
- Goal-directed agents: have a specific goal in mind and try to achieve that. Like a chess-simulating AI.
- Autonomous agents: like stock-trading agents.
- Social agents: for instance, an agent that initiates a chat between two humans in online chat rooms.

For agents to interact with each other there's a need for a protocol and trading/currency mechanism.

PEAS

One way to model agents is through the PEAS model:

- Performance measure
- Environment
- Actuators
- Sensors

For instance: the vacuum cleaner robot RUMBA will be measured by how much dirt from the environment it picks up, the actuators are the vacuum and wheels and sensors for dust, walls...

Another example is autonomous cars. Consider an automated taxi driver – detailed in the slides.

Comment: Back in the 90s there was development of algorithms to determine depth from an intersection of close yet distant images. It's a computationally hard problem. Lidar is this type of sensor that is used on the DARPA grand-challenge (competition to create autonomous vehicles), Google-cars.

Agent or Program

What is the difference? Agent is a type of program; a software agent is a program that is situated in some environment, has some autonomy, senses the environment and acts in it.

AI applications:

- Google search – like semantic analysis of your search words for suggestions
- Autonomous vehicles
- Netflix recommendation system – learning someone’s taste in movies from his likes and dislikes
- Automated trading
- ...

What is CS510?

There will be a quiz about the readings (published on the class website). The reading portion will account for 5% of the grade.

Assignments:

- The first assignment will be about search and will be written in LISP
- The second assignment will be on game-planning and will be written in Java. There will be a game interface in which we need to plug in the AI engine
- Logic assignment written in Prolog
- Planning problems

Non-programming assignments submitted in Latex will earn up to 5% points (if we lose any).

AI TopicsSearch:

- Intractability of exhaustive search – problems can be hard and long to solve in brute force
- The A* search algorithm
- Local search “satisficing” – when a problem is too hard to solve exactly, you can find a solution that satisfies some of the goals and is optimal under the circumstances

Open problems:

- Distributed search – for instance a set of robots that need to communicate between themselves to not collide with each other
- Dynamic search – the problem is changing as you’re solving it

Constraint reasoning:

- A way of representing knowledge and structure on a problem
- These problems are modeled by a set of variables that need to be assigned values, the domain is the set of possible values, and there’s a set of constraints that represent desired relations between

the variables. For instance: the coloring problem (color the map of US states with 3 colors s.t. adjacent states are colored differently)

Applications:

- Supply chain management
- Scheduling problems
- Resource and task allocation – what job should be done when
- Multiagent coordination

Open problems in CR:

- How to easily express problems as constraint problems (and then use known techniques on them)
- Dynamic / uncertain domain
- How to measure performance in distributed systems

Games / Adversarial Search:

We will examine only deterministic games (without chance)

Mechanism Design:

- Create an environment for agents to incentivize them to a certain type of behavior. For instance: rules for auctions such that the auction will be fair; traffic systems – how to time the traffic lights to maximize the flow of traffic

Knowledge representation:

- How to encode stuff so the computer can “understand” them

Planning:

- Given a set of actions, goal state and present state, and a knowledge base (some representation of the world) – how can we get from the present state to the goal state.
- Example: a utility-based agent – an agent has a model for some amount of profit for getting different types of goals; we try to maximize the profit for a specific state.

Planning problems:

- Highly dynamic environments
- Uncertain information

Example: Hanoi towers – where you want to figure out how to move the “donuts” between the pegs without having a large donut above a small one.

Learning

- Supervised: the computer gets some training data, evidence, and the computer needs to learn to model from that to be used for classification of unlabeled data.
- Unsupervised learning – given a set of data instances, the computer is required to cluster them into different groups.

Examples:

- The Netflix problem
- Email filtering in organizations – for instance Gmail classification of high-priority email
- Automated investing services
- Authorship recognition - learning who wrote a document by linguistic features