CS440/ECE448: Intro to Artificial Intelligence

Lecture 2: Intelligent Agents

Prof. Julia Hockenmaier
juliahmr@illinois.edu

http://cs.illinois.edu/fa11/cs440

Key concepts from last lecture

Last time's key concepts

Content-wise:

- What is Artificial Intelligence?
  Why is it difficult?
- What is reasoning?
  Why does it require models?

Class admin:

- Can you log onto your Compass site?
- Did you do the survey on Compass?
- Do you have access to the textbook?

Compass survey

Java experience (26% no):
TA office hours next week will offer Java tutorials

Parisa's OH will be 3pm-5pm on Monday.
Yonatan's OH will be 11am-1pm on Wednesday

Compass survey

What is AI?

- How to make something behave like a human
- How to make something intelligently solve problems/reason.

Second answer is correct, first is not.

Today's lecture

Today's key questions

How can we design an "intelligent" **agent**
to solve a specific **task**
in a particular **environment**?

What is **intelligence**?

Today’s key concepts

Agents:

– Different kinds of agents
– The structure and components of agents

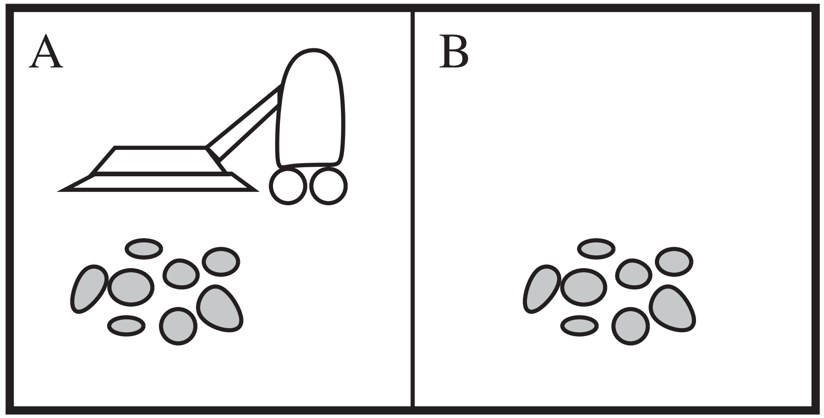
Describing and evaluating agents:

– Performance measures
– Task environments

Rationality:

– What makes an agent intelligent?

The vacuum world

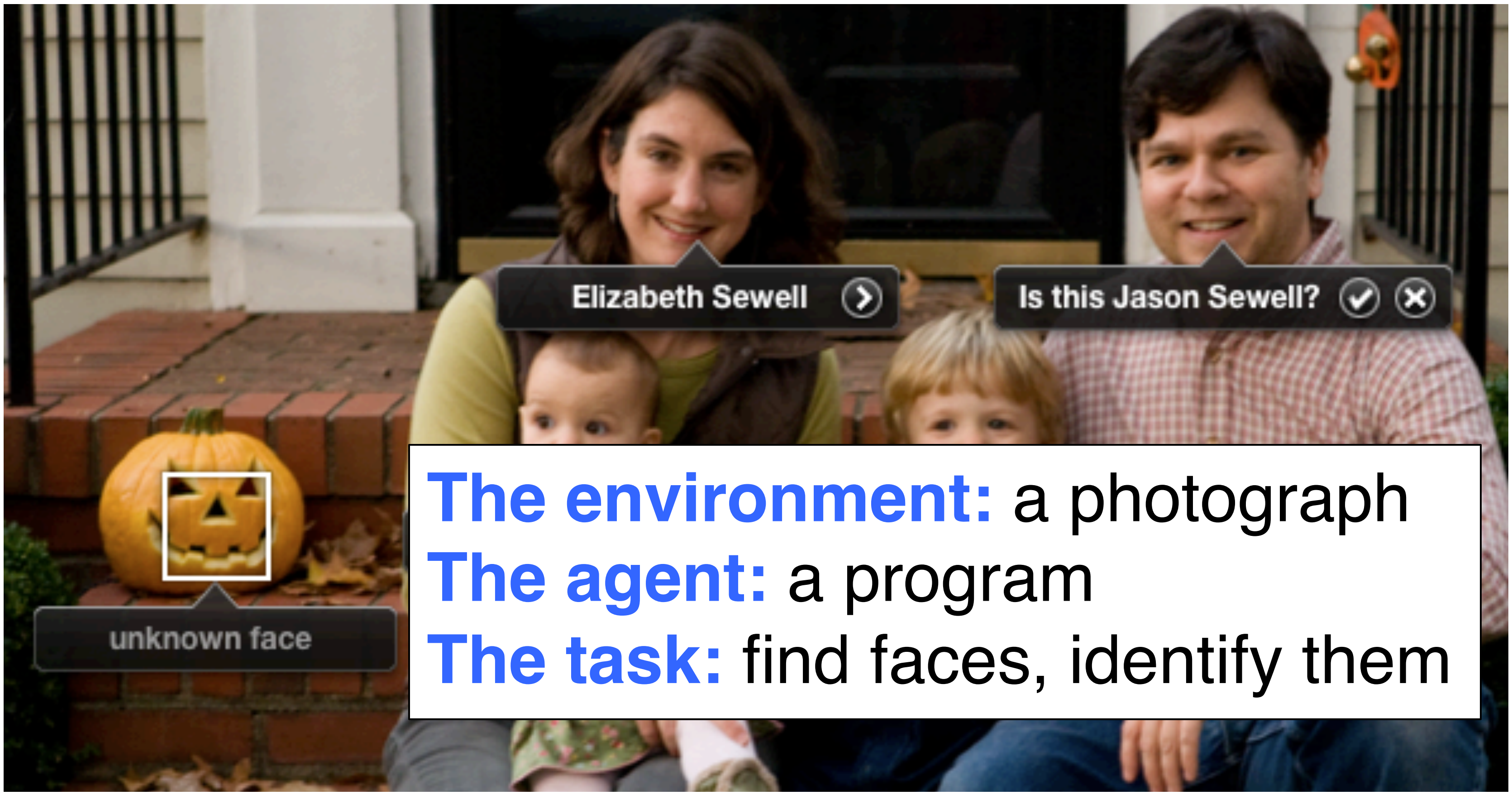


The environment: Location A and location B
Either can be clean or dirty

The agent: a vacuum cleaner

The task: clean both A and B

The face recognition world

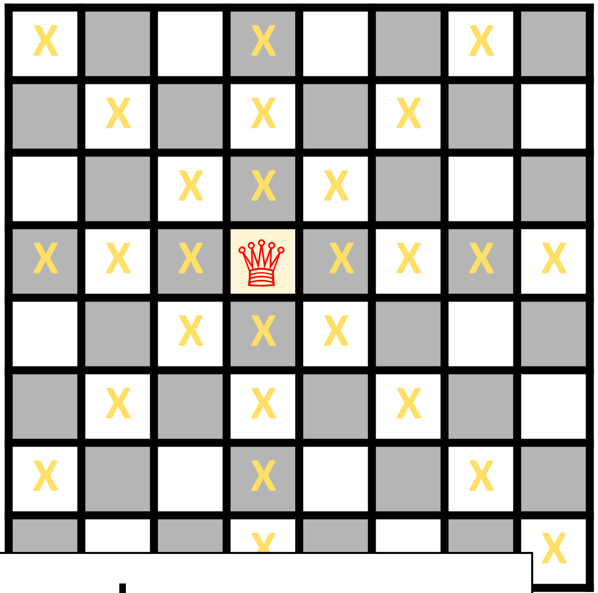


The environment: a photograph
The agent: a program
The task: find faces, identify them

Photo: Jason Sewell , on flickr.com

The chess world

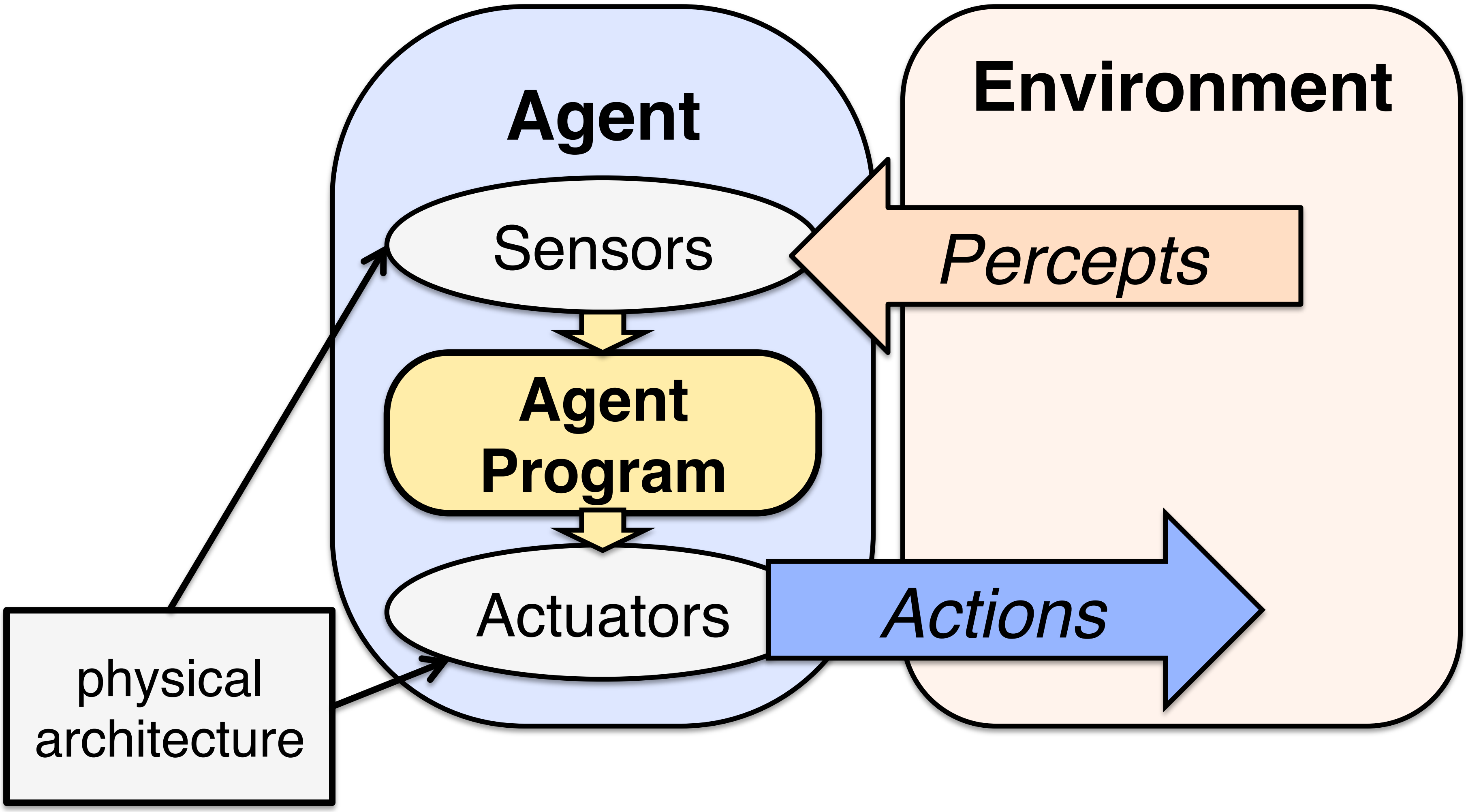
1. e4 e5
2. Qh5 Nc6
3. Bc4 Nf6
4. Qxf7# 1–0



The environment: a chess game
The agent: a game
The task: play chess, win match

Agents

Agents operate in an environment



Definitions….

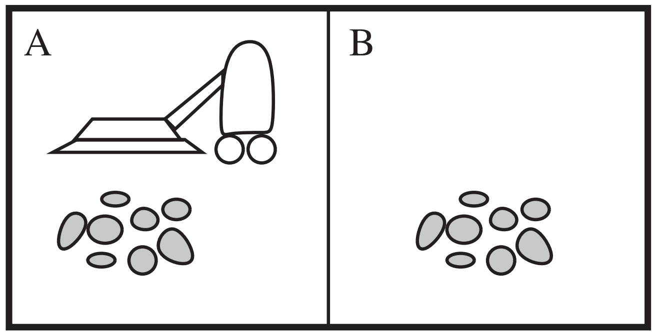
Sensor: eyes, ears, nose (human); camera, microphone (robot); stdin (NLP system),

Percept: the perceptual input at any instant.

Percept sequence: the complete history of what the agent has perceived

Actuator: arms, legs (human, robot), hose (vacuum), stdout (NLP system),

Toy example: the vacuum world



Sensor: a camera
Percepts: current location, is clean or dirty
Actions: move left, move right, suck

The agent program (the 'brain')

The agent program decides what action to take in each situation.

– *Situation* = the current percept sequence

– It implements a mapping from percept sequences to actions *(=the agent function)*

Percept sequence	**Action**
[A, Clean]	Right
[A, Dirty]	Suck
[A, Clean], [A, Dirty]	Suck

Agent program vs. agent function

Agent function:

A (complete) mapping from sequences of percepts to actions:

$$\text{AgentFunction}(\langle p^{(1)} p^{(2)} \ldots p^{(t)} \rangle) = a^{(t)}$$

Agent program:

What is *actually* implemented in the agent. Typically an approximation of the agent function.

Simple reflex agents

Action depends *only* on current percept.
Agent has no memory.

Last percept	Action
[Clean]	Right
[*cat*]	RUN!

May choose actions stochastically
to escape infinite loops.

Last percept	Action
[Clean]	Right (p=0.8) Left(p=0.2)

Model-based reflex agents

Agent has an **internal model** of the current state of the world.

Examples: the agent's previous location; current locations of all objects it has seen;

Last percept	Last location	Action
[Clean]	Left of current	Right
[Clean]	Right of current	Left

Model-based reflex agents

Agent may also have (generic) domain knowledge of the world.

Examples: effects of agent's actions, behavior of entities in the environment

Action	Effect
agent sucks cat	cat bites agent
agent moves down stairs	agent breaks

Goal-based agents

Agent has a goal, which may require a **sequence of actions**.
(This requires *searching* or *planning*)

Goal	Required actions
a clean house	clean every room

Utility-based agents

Agent can choose between multiple actions to achieve its goal. Which is best?

Agent may have conflicting goals.
Which one to pursue first?

Agents may have a **utility function**, which (ideally) approximates the external performance measure.

Learning-based agents

We cannot foresee every eventuality.
Agents need to change their behavior
to adapt to their environment.

This requires an (external) **teacher** or
reward which tell the agent how well
it is currently doing.

Evaluating agents

How well does the agent perform?

***Objective* evaluation:**
What are the consequences of its actions on the environment?

Performance measure:
Do the agent's actions reach the desired *state of the environment* ?

- We don't care about the state of the agent
- We usually don't care how the agent behaves to reach its goal

Rationality

A rational agent should always choose the action that maximizes its expected performance,
given the current situation

How rational an agent can be depends on
a) the performance measure
b) its prior knowledge of the environment
c) what actions it can perform
d) its current percept sequence

NB: rationality ≠ omniscience

Strategies for maximizing expected performance

Information gathering (first look, then act):
Choose an action that yields a more informative percepts for the following action

Learning (act according to prior experience)
Augment or modify knowledge of the task or the environment according to experience.
NB: learning requires autonomy

Comparing agents and tasks

The **task environment** specifies the problem that the agent has to solve.

It is defined by:
1. the objective **P**erformance measure
2. the external **E**nvironment
3. the agent's **A**ctuators
4. the agent's **S**ensors

PEAS descriptions for….

… medical diagnosis system?
… assembly line robot?
… chess computer?
… autonomous car?

Performance measure
Environment
Actuators
Sensors

Comparing task environments

1. What information do the *sensors* provide?

Fully observable:
The percepts contain **all relevant properties** of the environment

Partially observable:
The percepts contain only ***some*** **relevant properties** of the environment

2. What is the agent's *knowledge* about the environment?

Known:
Agent knows **all the rules** that hold in the environment. Can **predict outcomes** from complete observations.

Unknown:
Agent **doesn't know the rules** of the environment. Can't predict outcomes.

3. What effect do the *actions* have?

Deterministic:
Outcome of actions (next state of the environment) is fully determined by the current state.

Nondeterministic:
Each action has a set of possible outcomes.

Stochastic:
There is a probability distribution over possible outcomes.

4. How do the *percepts* change over time?

Episodic:
Agent receives a single percept per episode.
Its action in this episode does not affect future percepts (e.g. classification)

Sequential:
Agent receives a sequence of percepts.
The current action affects future percepts.
(e.g. navigation, game playing)

5. How does the *environment* change over time?

Static: Environment doesn't change unless agent performs an action. (e.g. crossword puzzles)

Dynamic: Environment changes even when the agent doesn't do anything. (e.g. traffic)

Semi-dynamic: Environment is static, but agent's performance score changes over time. (e.g. chess against a clock)

6. Is the world discrete or continuous?

Continuous:

Time, percepts and actions are continuous.
Example: driving a car.

Discrete:

Time, percepts and actions are discrete.
Example: playing a board game.

7. Does the performance depend on *other agents*?

Single-agent environment:
Performance of agent A does not depend on any other agents.

Competitive multi-agent environment:
Maximizing performance of agent A minimizes performance of agent B. (competitive games)

Cooperative multi-agent environment:
Maximizing performance of agent A maximizes performance of agent B.
(avoiding traffic accidents)

To conclude…

Today's key concepts

Agents:

– Different kinds of agents
– The structure and components of agents

Describing and evaluating agents:

– Performance measures
– Task environments

Rationality:

– What makes an agent intelligent?

Your tasks

Reading:
Chapter 2

Compass quiz:
Online after 2pm

Assignments:
Read up on (basic) Java if you don't know Java yet.