Introduction to Artificial Intelligence

Wen-Chih Peng (彭文志), CS, NYCU

AI – 人工智慧

- 什麼是AI？
  - AI 就是人工智慧
  - 教電腦或機器，學習人類做的事
    - 下棋
    - 開車
    - 對話
  - 產生智慧，可以做人類可以做的事並有可能做得比人類更好

Computer could see, hear, talk and understand human being
Bill Gates, 1991

看得懂(電腦視覺)

聽得懂(自然語言處理)

言之有物(語音處理)

AI isn't as smart as you think -- but it could be

AI Today

(Neural networks + computational power)

Next Generation AI Systems

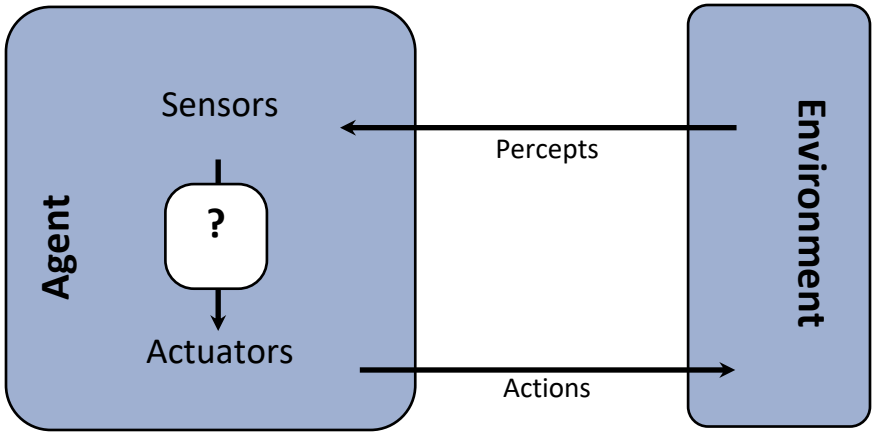
(How to make our technical approaches better)

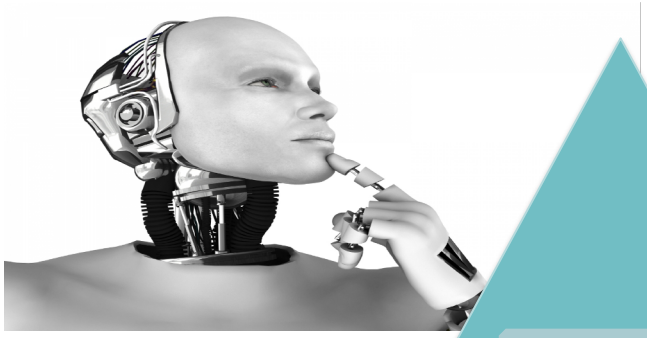
Goals of Artificial Intelligence

- "Artificial" means "man-made". So, what kind of artificial intelligence do we want to achieve?
- Two dimensions:
  - To be **human-like**, or to be **rational**? (After all, human beings are not always rational.)
  - To focus on the process of **thinking**, or to focus on the **behaviors**?
- Therefore, four different possible goals of AI:
  - Thinking humanly
  - Acting humanly
  - Thinking rationally
  - Acting rationally

Designing Rational Agents

- 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





Behaving/
Acting
Reasoning
Learning
Sensing and Collecting Behavior and
environmental data

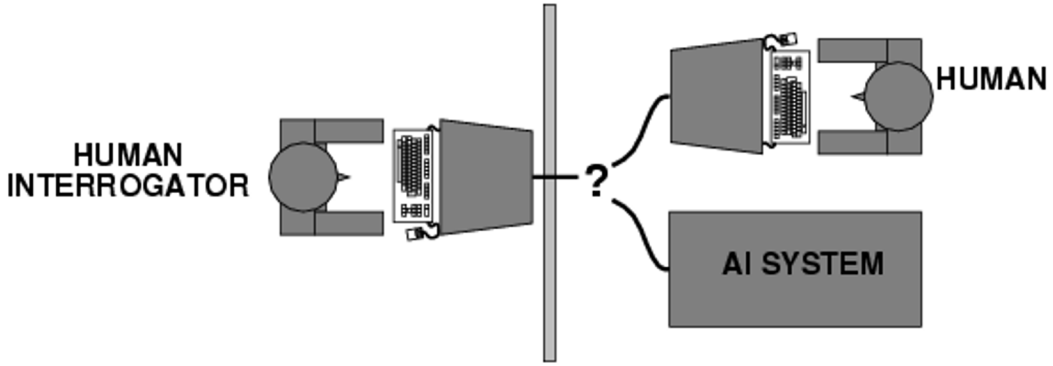
General vs. Narrow AI

- **General AI**:
  - Human-like intelligence: Thinking, reasoning, learning, creativity, etc., that are not limited to specific fields.
  - A popular topic of science fictions.
  - Generalized learning is the key.
  - Still out of reach.
- **Narrow AI**:
  - Designed for specific tasks.
  - Human experts control "what" and "how" to learn.
  - The focus of "practical AI".

Acting Humanly: Turing Test

Turing's (1950) "*Computing machinery and intelligence*":

- "Can machines think?" □ "Can machines behave intelligently?"
- Operational test for intelligent behavior: the Imitation Game



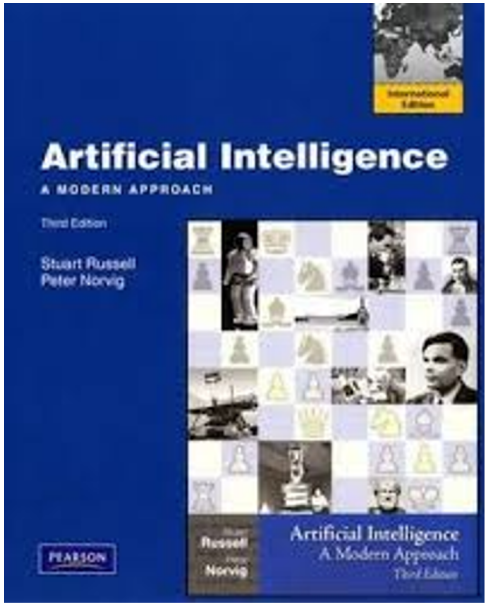
- Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes
- Anticipated all major arguments against AI in the following 50 years
- Suggested major components of AI: knowledge, reasoning, language understanding, learning.

Q: How much do you think today's technology can pass the Turing Test?

A Brief History of AI

- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's “Computing Machinery and Intelligence”
- 1950s Early AI programs: checkers, Logic Theorist, etc.
- 1956 **Dartmouth meeting: The term “Artificial Intelligence” adopted**
- 1965 Robinson's complete algorithm for logical reasoning (a theorem prover)
- 1966-74 AI discovers computational complexity; neural network research almost disappears; the start of "AI winter"
- 1980-93 **Expert systems industry booms and busts**
- 1985 Neural networks return to popularity
- 1995 The start of "data science"
- 1997 Deep Blue
- 2009 **The "Big Bang" of deep learning with GPU computing**
- 2016+ AlphaGo, etc.

About This Course



- Instructor information:
  - Contact: wcpengcs@nycu.edu.tw
  - Office hour: by appointment
- Textbook:
  - *Artificial Intelligence: A Modern Approach* by Russell & Norvig.
  - A comprehensive book for good foundations, but a little too old for current developments.
- CS 221 AI @Standford (https://reurl.cc/AK3NoK)

Assignment Overview (Tentative)

1. 1 warmup + 4 assignments (60%)
   a. Code and report
   b. Define -> Solve -> Analyze
   c. Can you outperform the SOTA method? (if you can, let's publish your results)
2. 1 in-class exam (20%)
3. 1 final project (20%)

HW0 - Python 101

The purpose of this assignment it to help you set up your environment and get started with Python.

- Set up your environment.
- Read files and do some simple operations.

Ref: Python 100 Days , Python numpy tutorial

HW1 - Computer Vision

Crowd detection

- Implement Viola-Jones object detection for crowd detection
- Call existing package or model to observe SOTA performance (e.g. YOLOv5)
- Can you make it better ?

HW2 - Natural Language Processing

Language models and sentiment classification

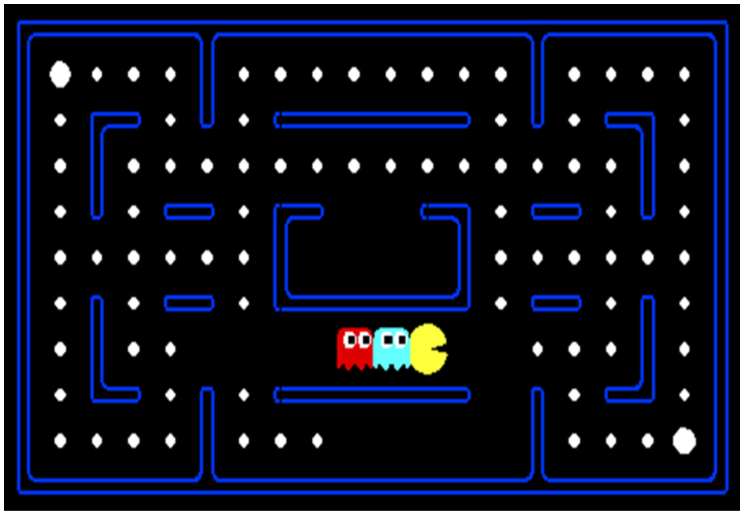
- Implement n-gram as the language model
- Implement different smoothing techniques to improve the performance
- Apply in the downstream task
- Call existing package to observe SOTA performance

HW3 - Games

Design agents for the classic version of Pac-man.

- Design agents by implement minimax search, alpha-bata pruning and expectimax search.
- Get a better performance by **traditional reinforcement learning** and **deep reinforcement learning**, and try to explain the differences

Ref: The Pac-Man Projects



HW4 - Explainable AI

Use the sentiment classifier in HW2 to do some explanations.

- Traditional explainable techniques
- Explain Bert-based model by existing package, e.g. ExBERT

Final Project

Topic: AI for Social Impact

- E.g. Agriculture, Education, Financial, Mobility/Transportation, Ethical Issues, Security and Privacy, Social Welfare, Justice, Fairness and Equality…
- 2 stages: Project Proposal and Project Presentation
- 3-4 members in a group

TA Information

- 王威堯 sf1638.cs05@nctu.edu.tw
- 杜葳葳 wwdu.cs10@nycu.edu.tw
- 王廣達 gdwang.cs10@nycu.edu.tw
- Public questions: Microsoft Teams
  - 線上討論參與程度將列為期末加分考量之一
- Private questions: New E3 mail（**Please send to all TAs**）

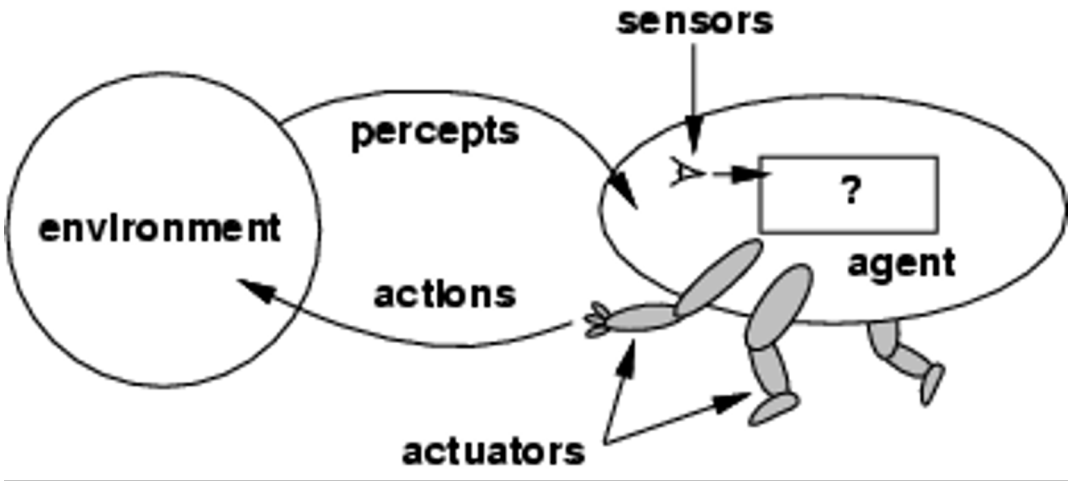
The Concept of "Agents"

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

- Human agents:
  - Sensors: eyes, ears, etc.
  - Actuators: hands, legs, mouth, etc.

- Robotic (nonhuman) agents:
  - Sensors: cameras, range finders, etc.
  - Actuators: motors, speakers, etc.

The Concept of "Agents"



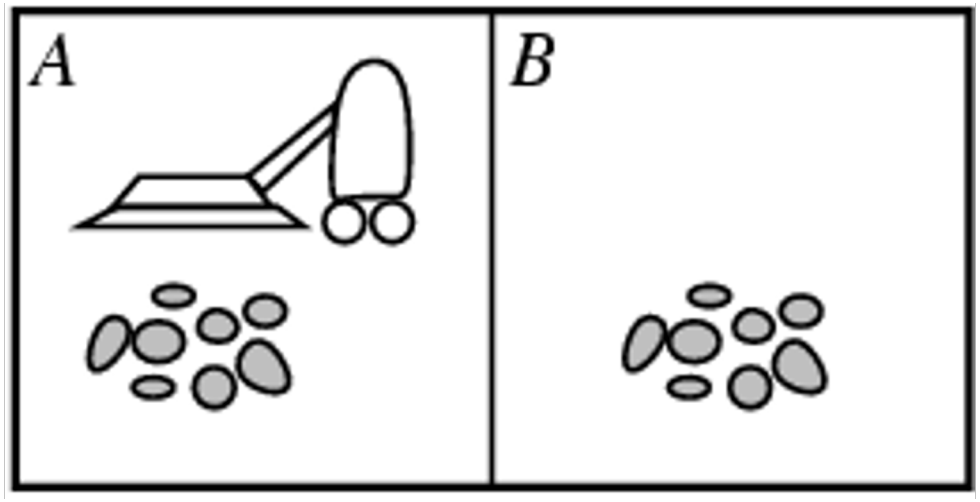
- Abstractly, an agent is a function from percept histories to actions:

$$f : P^* \rightarrow A$$

- This course is about designing rational agents: For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance.

Example: Vacuum-Cleaner World

- Percepts: location and contents, e.g., [*A*, *Dirty*]
- Actions: *Left*, *Right*, *Suck*, *NoOp*.



Example: A Vacuum-Cleaner Agent

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
⋮	⋮

function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

if *status* = *Dirty* **then return** *Suck*
else if *location* = *A* **then return** *Right*
else if *location* = *B* **then return** *Left*

PEAS

To design a rational agent, we must specify the following.

Example: An automated taxi

- **P**erformance measure: safety, destination, profits, legality, comfort, …
- **E**nvironment: roads, traffic, pedestrians, weather, …
- **A**ctuators: steering, accelerator, brake, horn, speaker, display, …
- **S**ensors: video, accelerometers, gauges, engine sensors, GPS, …

PEAS Example

Additional examples: Game-playing agent

- Performance measure:
- Environment:
- Actuators:
- Sensors:

PEAS Example

Additional examples: Interactive route planning

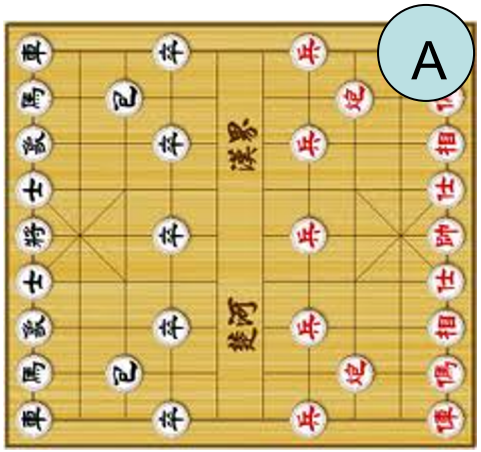
- Performance measure:
- Environment:
- Actuators:
- Sensors:

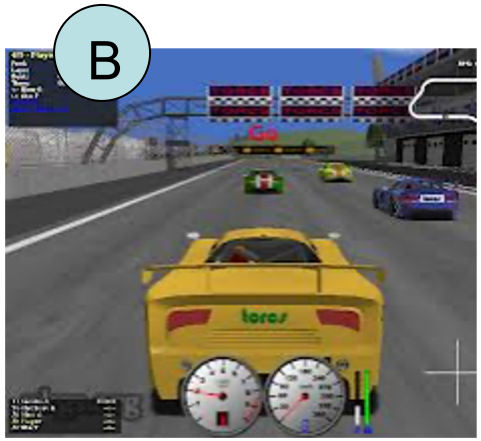
Environment Types

- Fully observable or partially observable:
- Single agent or multi-agent:
- Deterministic or stochastic: (Is the current state of the environment completely determined by its past states and the agents' actions?)
- Episodic or sequential: (Are future decisions affected by past ones?)
- Static or dynamic: (Does the environment change between actions by the agent?)
- Discrete or continuous:
- Known or unknown: (rules or "laws of physics" known to the agent)

Example Environment Types

Try to characterize the environment types of these games:





	A	B
Observable?		
Single-agent?		
Deterministic?		
Episodic?		
Static?		
Discrete?		
Known?		

Agent Types

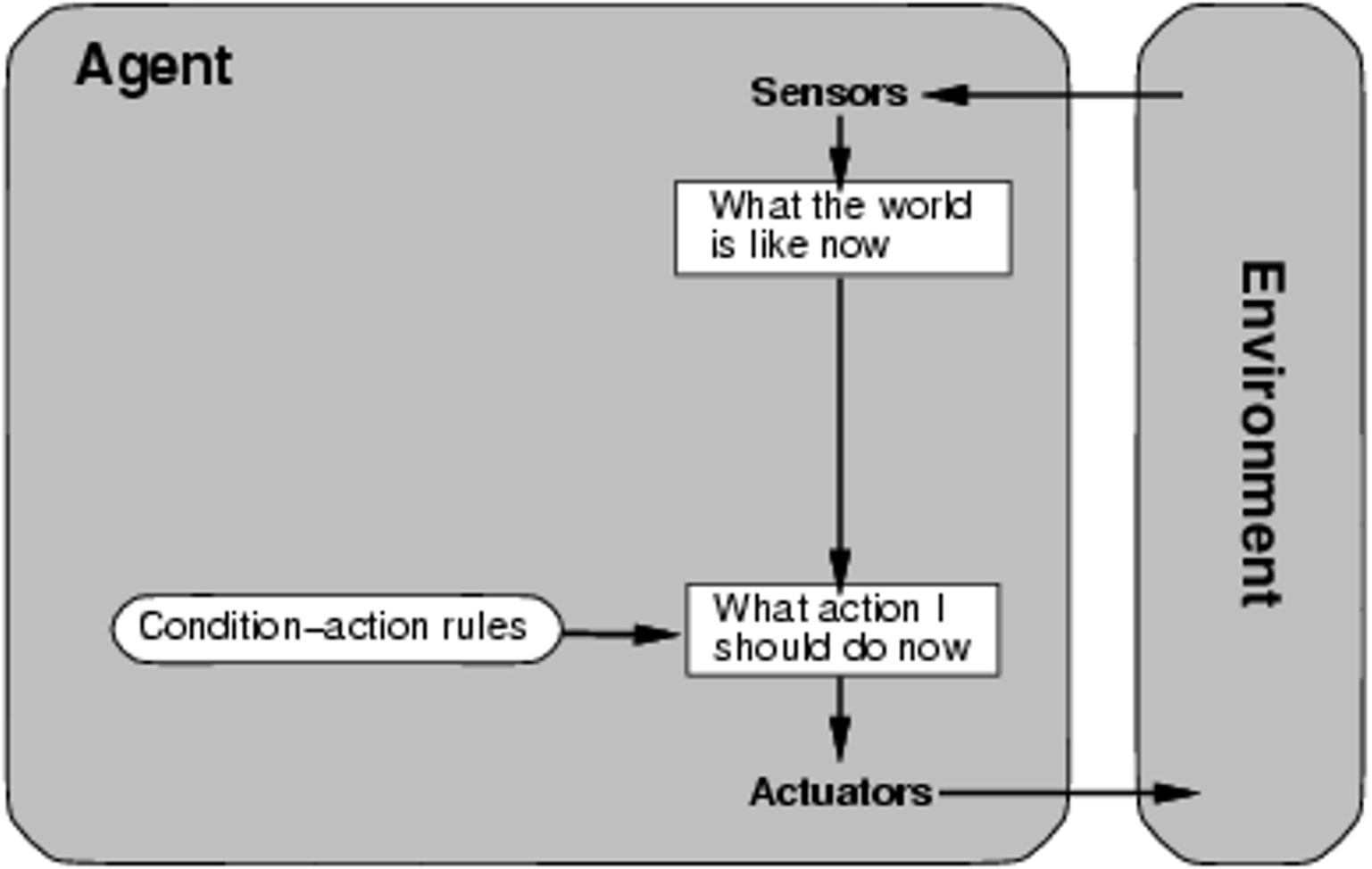
Four basic types, in order of increasing generality:

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents

All these can be turned into learning agents.

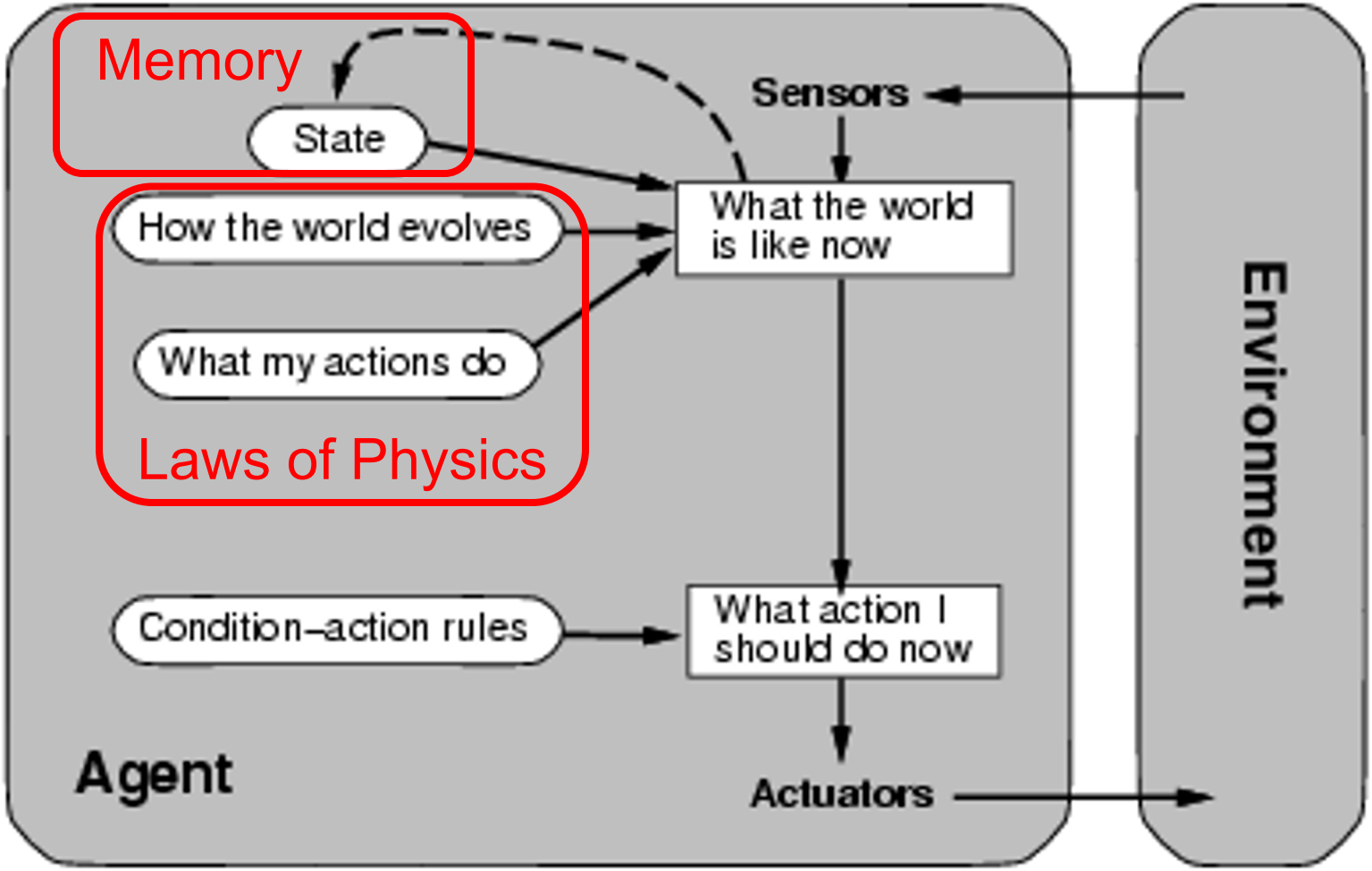
Simple Reflex Agent

Actions are only based on rules.



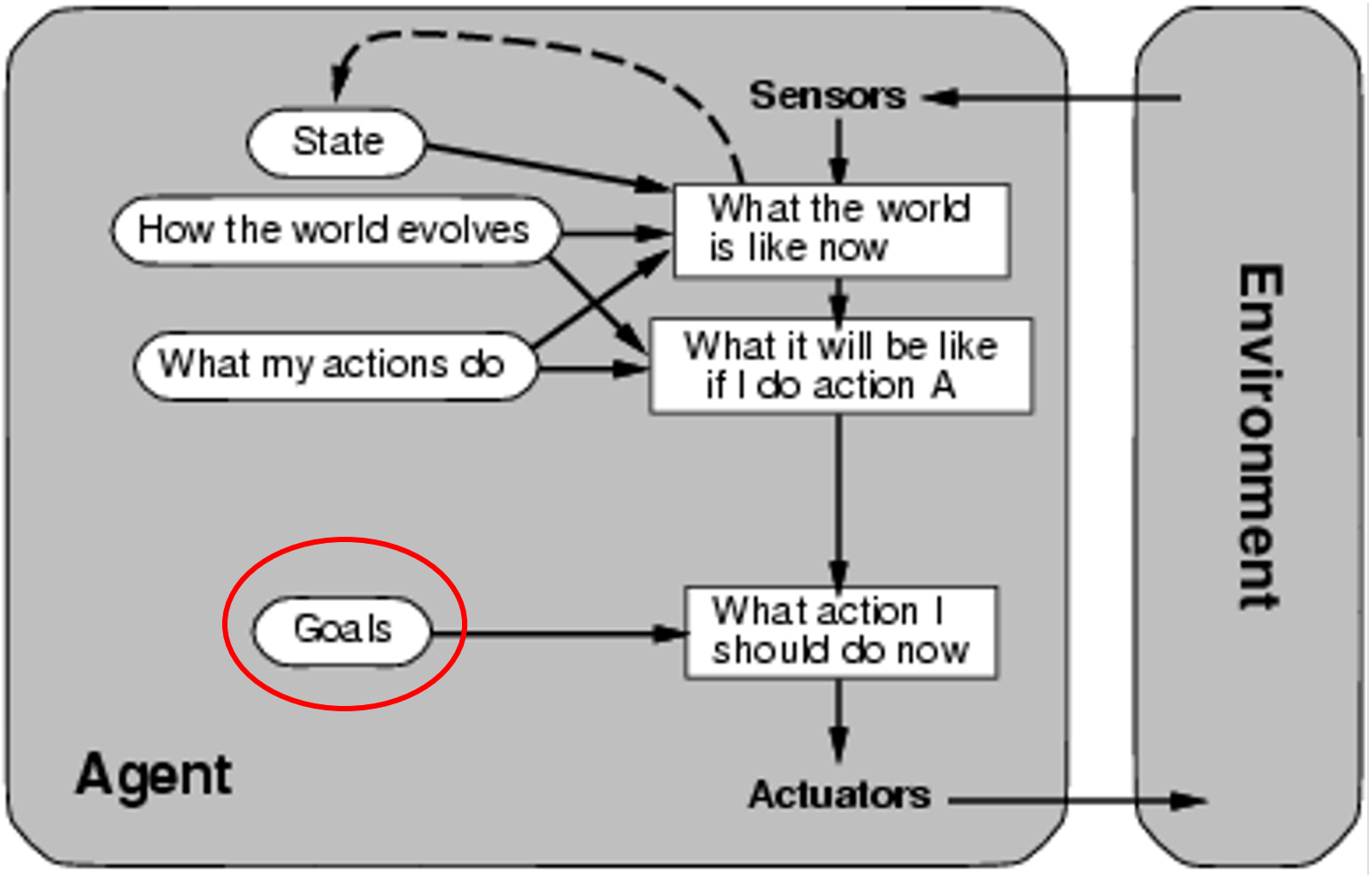
Model-Based Reflex Agent

The agent keeps track of the "**state**" of itself and the world. It knows how the state is changed.



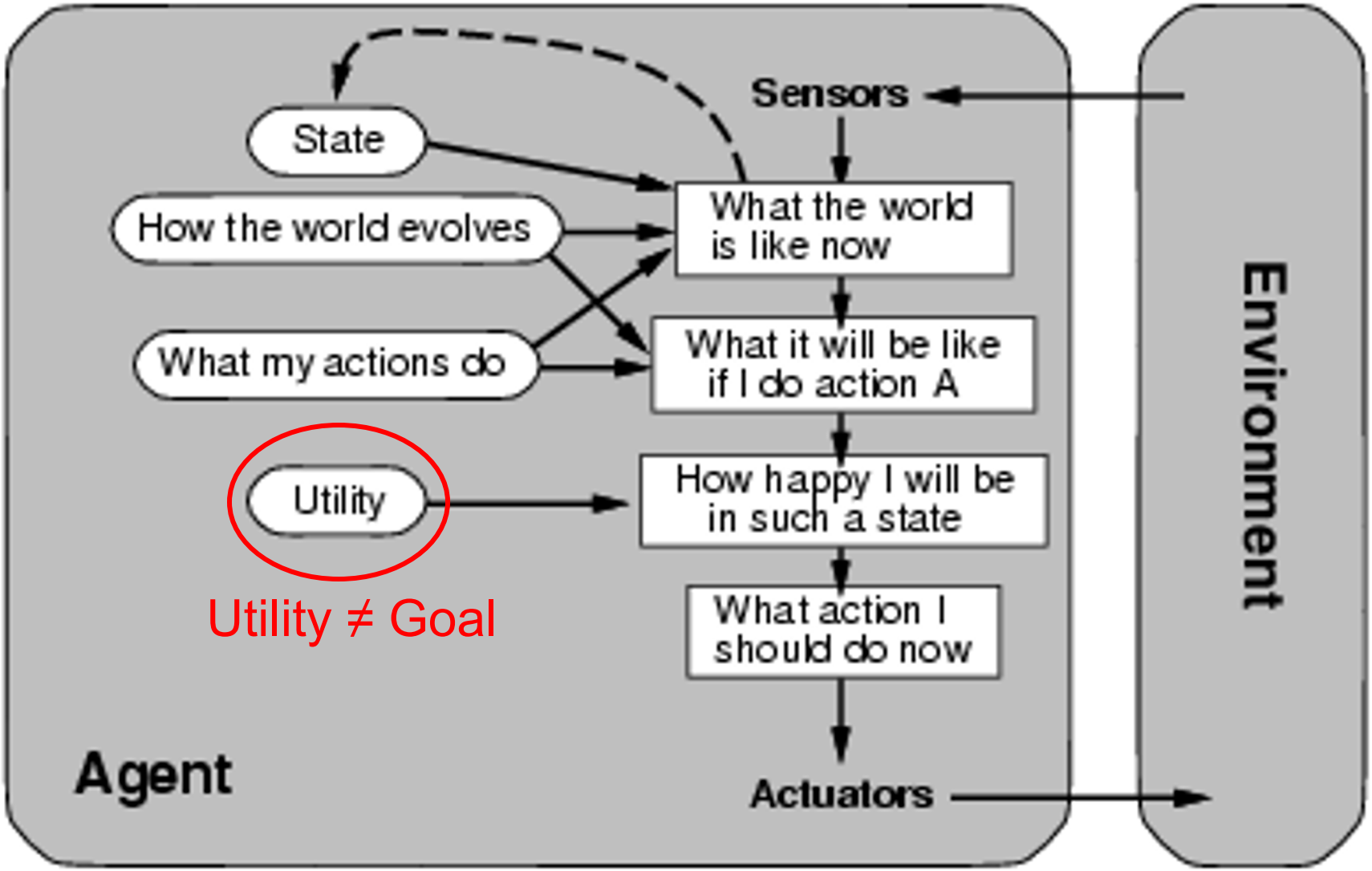
Goal-Based Agent

The agent aims to reach a "goal" state.
This affects how it selects the action.

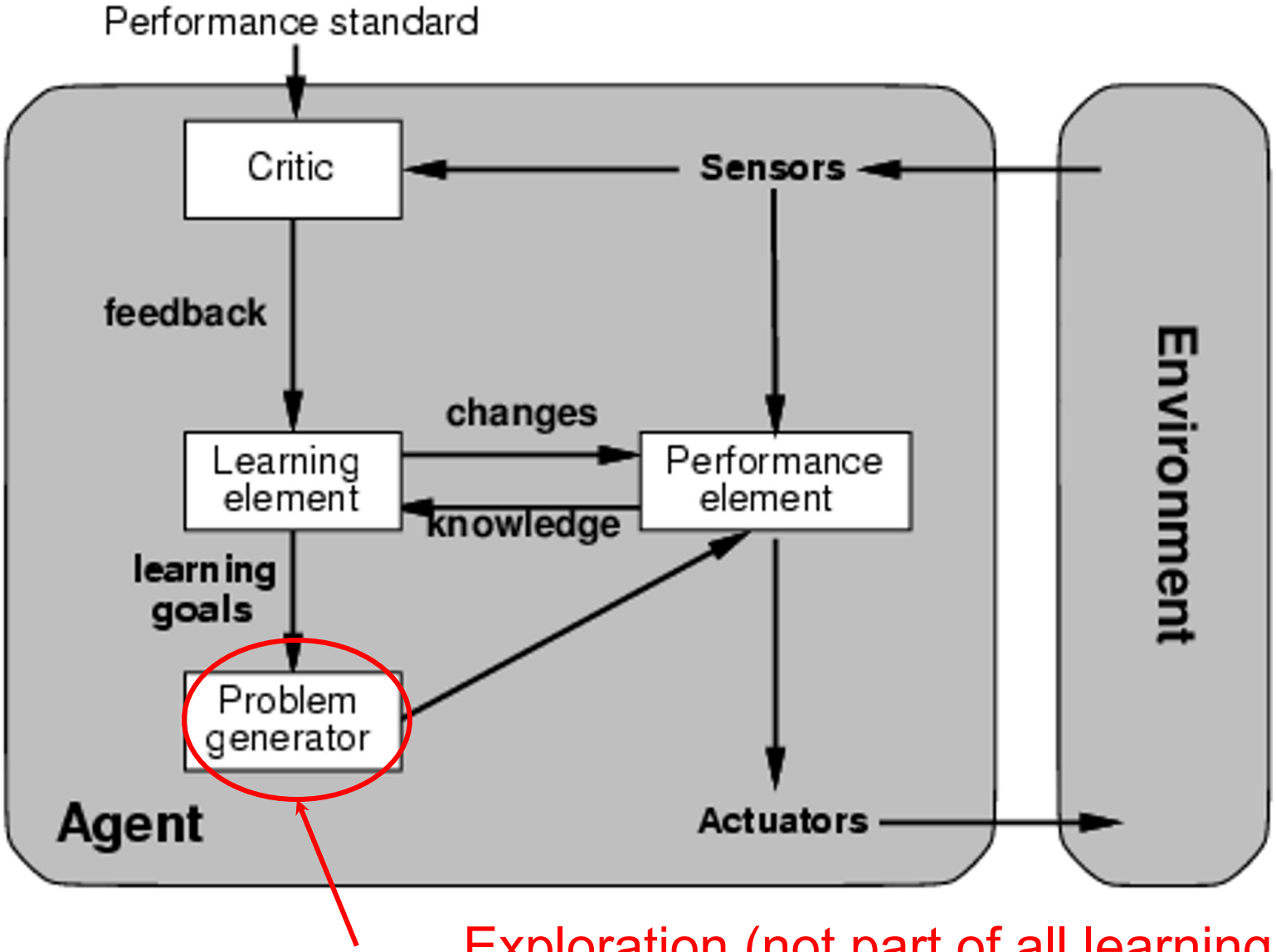


Utility-Based Agent

Utility is more general than goals.



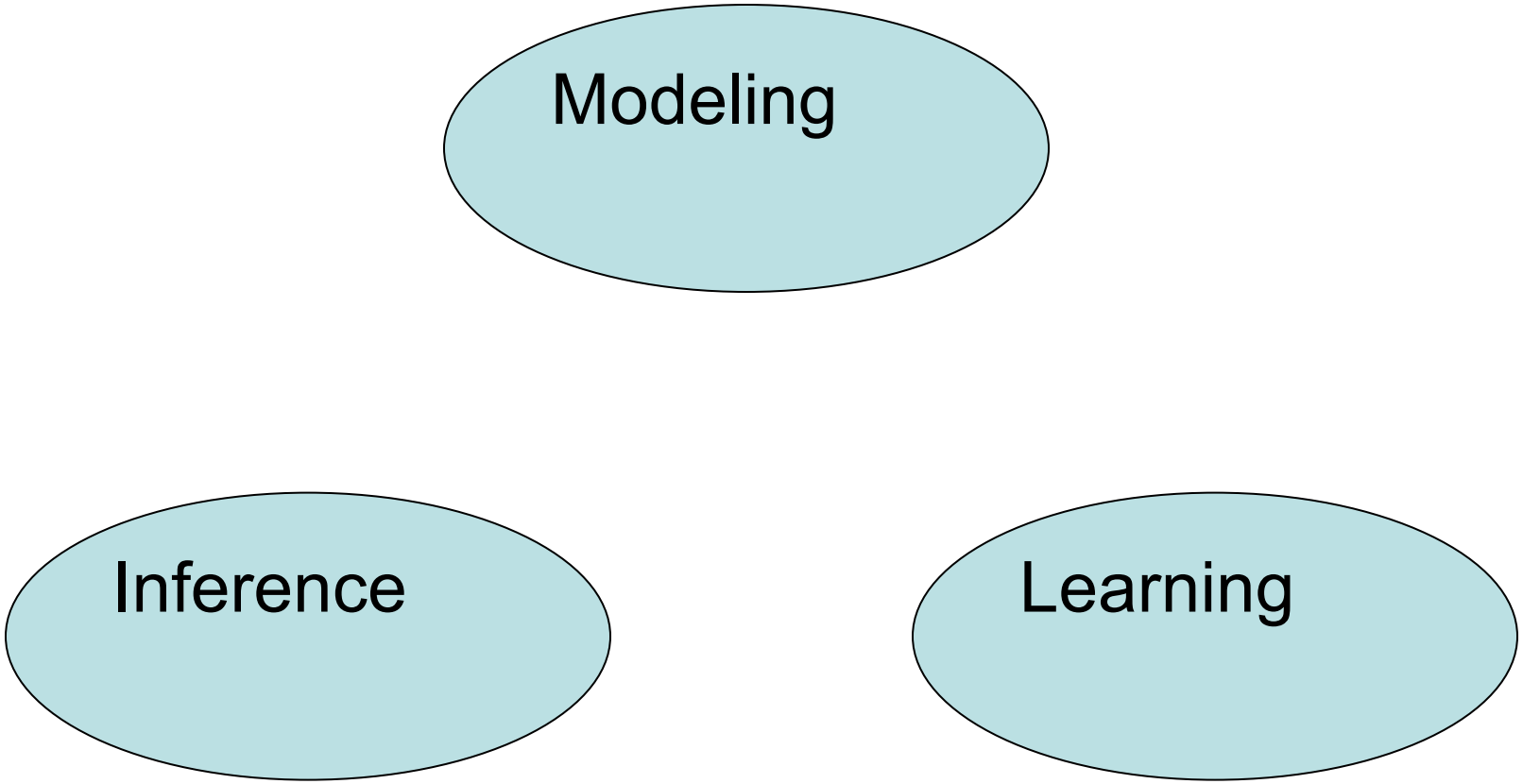
Learning Agent



Exploration (not part of all learning agents)

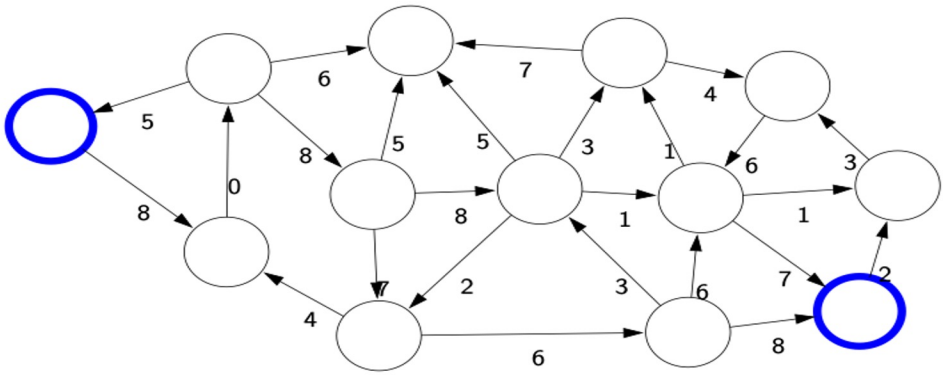
Key issues

from CS221

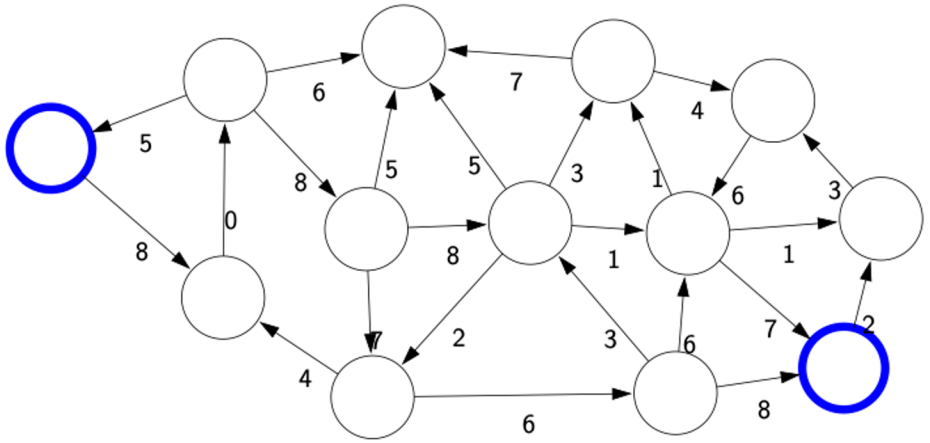


Modeling

Model

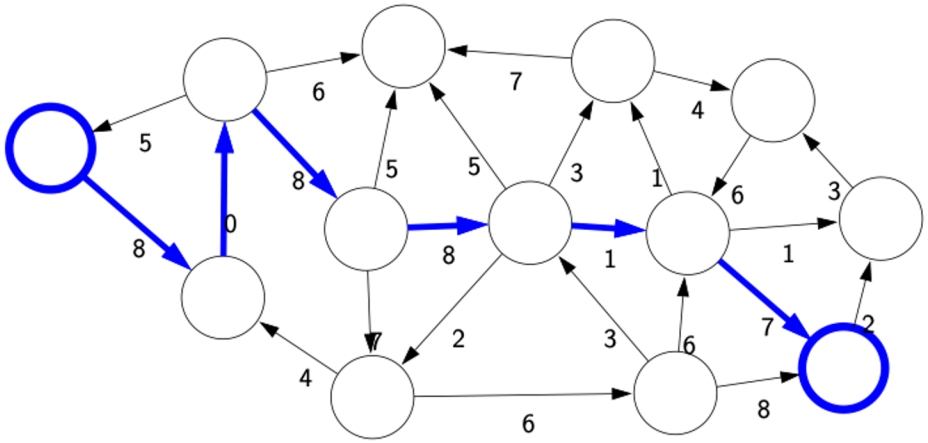


Model

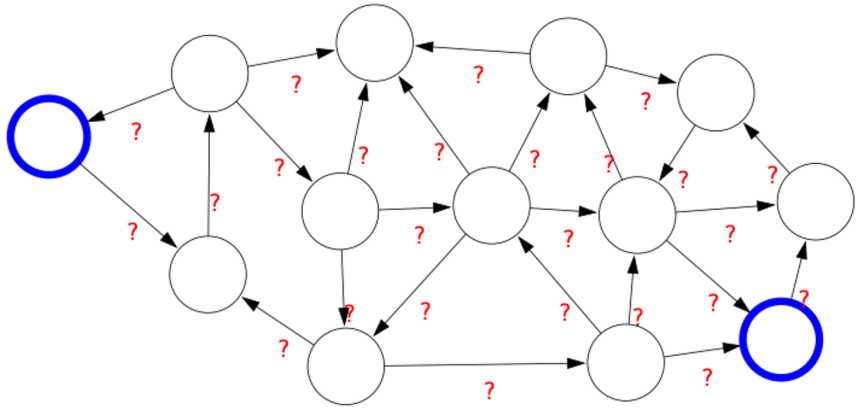


Inference

Predictions



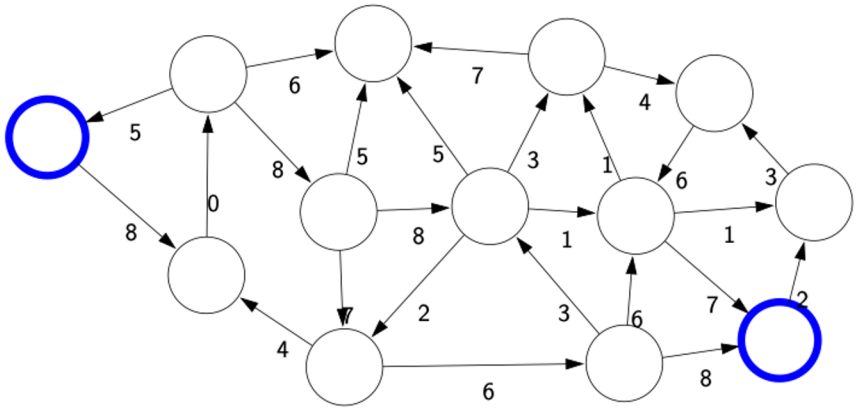
Model without parameters



+data

Learning

Model with parameters



Topics covered in this course