CSC520 - Artificial Intelligence

Lecture 2

Dr. Scott N. Gerard

North Carolina State University

Jan 9, 2025

In The News

LinkedIn Jobs on the Rise 2025: The 25 fastest-growing jobs in the U.S.

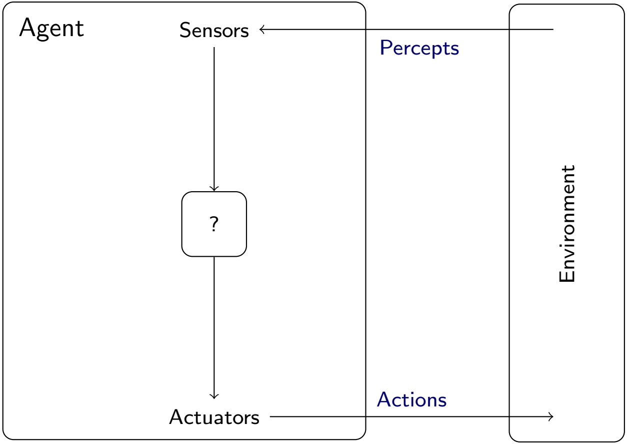
1. **Artificial Intelligence Engineer**
2. **Artificial Intelligence Consultant**
3. Physical Therapist
4. Workforce Development Manager
5. Travel Advisor
6. Event Coordinator
7. Director of Development
8. Outside Sales Representative
9. Sustainability Specialist
10. Security Guard

https://www.linkedin.com/pulse/linkedin-jobs-rise-2025-25-fastest-growing-us-linkedin-news-gryie/

Agenda

- Agents and environments
- Agent function
- Performance measures
- Rational agent
- Task environment

Agent and Environment



Definitions

- Agent perceives from its environment through sensors and acts upon it through actuators

Definitions

- Agent perceives from its environment through sensors and acts upon it through actuators
- Percept is what an agent perceives using its sensor

Definitions

- Agent perceives from its environment through sensors and acts upon it through actuators
- Percept is what an agent perceives using its sensor
- Percept sequence is a list of percepts that an agent has perceived

Definitions

- Agent perceives from its environment through sensors and acts upon it through actuators
- Percept is what an agent perceives using its sensor
- Percept sequence is a list of percepts that an agent has perceived
- *Agent function* maps a percept sequence to an action; external characterization

$$f : p_0, p_1, \ldots p_n \rightarrow a_i$$

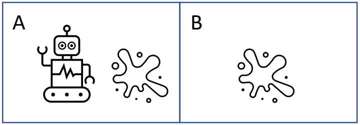
Definitions

- Agent perceives from its environment through sensors and acts upon it through actuators
- Percept is what an agent perceives using its sensor
- Percept sequence is a list of percepts that an agent has perceived
- *Agent function* maps a percept sequence to an action; external characterization

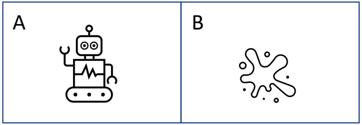
$$f : p_0, p_1, \ldots p_n \rightarrow a_i$$

- *Agent program* is a concrete implementation; internal characterization

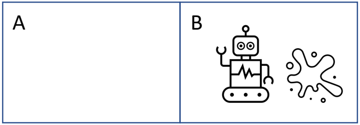
Vacuum Cleaner Example



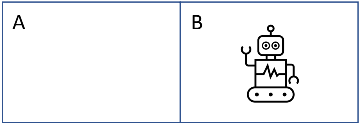
1: $[A, \mathit{dirtyA}] \rightarrow \mathit{vacA}$



2: $[A, \mathit{dirtyA}], [A, \mathit{cleanA}] \rightarrow \mathit{right}$



3: $[A, \mathit{dirtyA}], [A, \mathit{cleanA}], [B, \mathit{dirtyB}] \rightarrow \mathit{vacB}$



4: $[A, \mathit{dirtyA}], [A, \mathit{cleanA}], [B, \mathit{dirtyB}], [B, \mathit{cleanB}] \rightarrow \mathit{left}$

Performance Measure

- A rational agent is one that does the *right thing*
  - What does it mean to do the *right thing*?

Performance Measure

- A rational agent is one that does the *right thing*
  - What does it mean to do the *right thing*?
  - Performance measure is a measure of how well an agent performs

Performance Measure

- A rational agent is one that does the *right thing*
  - What does it mean to do the *right thing*?
  - Performance measure is a measure of how well an agent performs
- Performance measure evaluates a sequence of environmental states
  - Designer specifies it initially
  - Agent learns it as time goes by

Performance Measure

- A rational agent is one that does the *right thing*
  - What does it mean to do the *right thing*?
  - Performance measure is a measure of how well an agent performs
- Performance measure evaluates a sequence of environmental states
  - Designer specifies it initially
  - Agent learns it as time goes by
- What if the performance measure of a vacuum cleaner agent is the amount of dirt cleaned per hour?

Performance Measure

- A rational agent is one that does the *right thing*
  - What does it mean to do the *right thing*?
  - Performance measure is a measure of how well an agent performs
- Performance measure evaluates a sequence of environmental states
  - Designer specifies it initially
  - Agent learns it as time goes by
- What if the performance measure of a vacuum cleaner agent is the amount of dirt cleaned per hour?
  - Rational agent picks up the dirt and drop it back on the floor
    Be careful what you ask for !

Performance Measure

- A rational agent is one that does the *right thing*
  - What does it mean to do the *right thing*?
  - Performance measure is a measure of how well an agent performs
- Performance measure evaluates a sequence of environmental states
  - Designer specifies it initially
  - Agent learns it as time goes by
- What if the performance measure of a vacuum cleaner agent is the amount of dirt cleaned per hour?
  - Rational agent picks up the dirt and drop it back on the floor
    Be careful what you ask for !
- Better performance measure will be amount of clean area per time step

Rationality

- What if
  - Agent lacks perfect knowledge
  - Agent's action does not produce exactly the desired result

Rationality

- What if
    - Agent lacks perfect knowledge
    - Agent's action does not produce exactly the desired result
- Perfection is rarely possible in reality

Rationality

- What if
  - Agent lacks perfect knowledge
  - Agent's action does not produce exactly the desired result
- Perfection is rarely possible in reality
- Rationality $\neq$ perfection

Rationality

- What if
  - Agent lacks perfect knowledge
  - Agent's action does not produce exactly the desired result
- Perfection is rarely possible in reality
- Rationality $\neq$ perfection
- Rationality means maximizing *expected* performance

Rationality

Rationality

For each possible *percept sequence*, a rational agent should select an *action* that is expected to maximize its *performance measure*, given the evidence provided by the percept sequence and whatever built-in *knowledge* the agent has.

Task Environment

- **P**erformance measure, **E**nvironment, **A**ctuators and **S**ensors are collectively referred to as the *task environment*

Task Environment

- **P**erformance measure, **E**nvironment, **A**ctuators and **S**ensors are collectively referred to as the *task environment*
- PEAS description

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver				

Task Environment

- **P**erformance measure, **E**nvironment, **A**ctuators and **S**ensors are collectively referred to as the *task environment*
- PEAS description

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Maximize profits while providing safe, fast, legal trip	Roads, traffic, weather, pedestrians, customers	Steering, accelerator, brake, signal, horn	Cameras, radar, speedometer, GPS, engine sensors

Class Exercise

- Select a task of your choice and give PEAS specification for an AI agent to solve that task

Task Environment Properties

- Fully vs. Partially Observable
- Deterministic vs. Stochastic
- Episodic vs. Sequential
- Static vs. Dynamic
- Discrete vs. Continuous
- Known vs. Unknown
- Single vs. Multi-agent

Observability

- *Fully observable* if the agent's sensors can perceive complete state of the environment

Observability

- *Fully observable* if the agent's sensors can perceive complete state of the environment
- *Partially observable* if the agent's sensors can only perceive some part of the environment

Observability

- *Fully observable* if the agent's sensors can perceive complete state of the environment
- *Partially observable* if the agent's sensors can only perceive some part of the environment
  - Agent may keep an internal state with previous observations

Observability

- *Fully observable* if the agent's sensors can perceive complete state of the environment
- *Partially observable* if the agent's sensors can only perceive some part of the environment
  - Agent may keep an internal state with previous observations
- *Unobservable* task environment, the agent has no sensors

Single-agent vs. Multiagent

- *Single agent* if everything else can be treated as environment
  - E.g. an agent solving cross-word puzzle
- *Multiagent* if multiple agents exist and influence each other's performance measures
  - E.g. two agents playing chess is a *competitive multiagent* environment
  - E.g. multiple taxi drivers minimizing accidents and traffic congestion is *cooperative multiagent* environment

Deterministic vs. Stochastic

- *Deterministic* if the next state is completely determined by the current state and the action

Deterministic vs. Stochastic

- *Deterministic* if the next state is completely determined by the current state and the action
    - E.g. Vacuum cleaner

Deterministic vs. Stochastic

- *Deterministic* if the next state is completely determined by the current state and the action
    - E.g. Vacuum cleaner
- *Stochastic* if we know the probability distribution of the next state given the current state and the action
    - E.g. Taxi driver

Episodic vs. Sequential

- *Episodic* if agent's experience is divided into separate, unrelated, atomic episodes in which agent perceives and acts

Episodic vs. Sequential

- *Episodic* if agent's experience is divided into separate, unrelated, atomic episodes in which agent perceives and acts
  - Next episode does not depend on actions from previous episodes
  - E.g. Spotting defective parts on assembly line

Episodic vs. Sequential

- *Episodic* if agent's experience is divided into separate, unrelated, atomic episodes in which agent perceives and acts
  - Next episode does not depend on actions from previous episodes
  - E.g. Spotting defective parts on assembly line
- *Sequential* if agent's current decision can affect the future

Episodic vs. Sequential

- *Episodic* if agent's experience is divided into separate, unrelated, atomic episodes in which agent perceives and acts
    - Next episode does not depend on actions from previous episodes
    - E.g. Spotting defective parts on assembly line
- *Sequential* if agent's current decision can affect the future
    - E.g. Current move in chess can affect the future

Static vs. Dynamic

- *Static* if the environment stays unchanged while the agent deliberates

Static vs. Dynamic

- *Static* if the environment stays unchanged while the agent deliberates
  - E.g. In chess, the board stays unchanged
- *Dynamic* if the environment continuously changes while the agent deliberates

Static vs. Dynamic

- *Static* if the environment stays unchanged while the agent deliberates
  - E.g. In chess, the board stays unchanged
- *Dynamic* if the environment continuously changes while the agent deliberates
  - E.g. In taxi driving, the traffic changes continuously

Static vs. Dynamic

- *Static* if the environment stays unchanged while the agent deliberates
    - E.g. In chess, the board stays unchanged
- *Dynamic* if the environment continuously changes while the agent deliberates
    - E.g. In taxi driving, the traffic changes continuously
- *Semidynamic* if the environment is static but the agent's performance score changes by time

Static vs. Dynamic

- *Static* if the environment stays unchanged while the agent deliberates
    - E.g. In chess, the board stays unchanged
- *Dynamic* if the environment continuously changes while the agent deliberates
    - E.g. In taxi driving, the traffic changes continuously
- *Semidynamic* if the environment is static but the agent's performance score changes by time
    - E.g. Chess when played with a clock

Discrete vs. Continuous

- *Discrete* if the environment, percepts and actions have finite, distinct values

Discrete vs. Continuous

- *Discrete* if the environment, percepts and actions have finite, distinct values
    - E.g. Chess without clock

Discrete vs. Continuous

- *Discrete* if the environment, percepts and actions have finite, distinct values
  - E.g. Chess without clock
- *Continuous* if the environment, percepts or actions have infinite values

Discrete vs. Continuous

- *Discrete* if the environment, percepts and actions have finite, distinct values
  - E.g. Chess without clock
- *Continuous* if the environment, percepts or actions have infinite values
  - E.g. Taxi driving

Known vs. Unknown

- *Known* if the outcomes (or outcome probabilities) of the actions are known to the agent

Known vs. Unknown

- *Known* if the outcomes (or outcome probabilities) of the actions are known to the agent
- *Unknown* if the outcomes are not known and the agent needs to learn experientially

Known vs. Unknown

- *Known* if the outcomes (or outcome probabilities) of the actions are known to the agent
- *Unknown* if the outcomes are not known and the agent needs to learn experientially
  - E.g. Rover sent to a new planet

Class Exercise

- Determine the characteristics of the task environment for the task from the PEAS exercise

Examples

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle						
Medical diagnosis						
Taxi driving						

Agent Function vs. Program vs Architecture

- *Agent function* maps a percept sequence to an action; external characterization

$$f : p_0, p_1, \ldots p_n \rightarrow a_i$$

Agent Function vs. Program vs Architecture

- *Agent function* maps a percept sequence to an action; external characterization

$$f : p_0, p_1, \dots p_n \rightarrow a_i$$

- *Agent program* is a concrete implementation; internal characterization

Agent Function vs. Program vs Architecture

- *Agent function* maps a percept sequence to an action; external characterization

$$f : p_0, p_1, \ldots p_n \rightarrow a_i$$

- *Agent program* is a concrete implementation; internal characterization

- *Agent architecture* is the computing device with sensors and actuators

Agent Function vs. Program vs Architecture

- *Agent function* maps a percept sequence to an action; external characterization

$$f : p_0, p_1, \ldots p_n \rightarrow a_i$$

- *Agent program* is a concrete implementation; internal characterization

- *Agent architecture* is the computing device with sensors and actuators

- *Agent* is the combination of architecture and program
  $agent = architecture + program$

Simple Reflex Agent

- Selects action based on the current percept

Simple Reflex Agent

- Selects action based on the current percept
- Does not track history of percepts

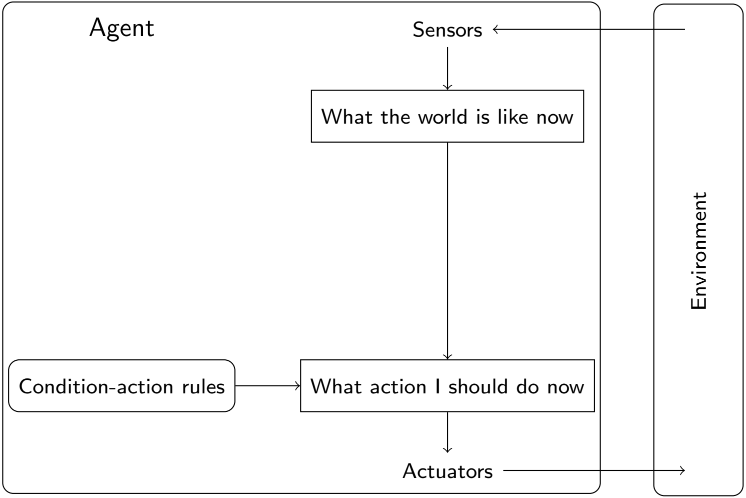
Simple Reflex Agent

- Selects action based on the current percept
- Does not track history of percepts
- Simple but can fail if the environment is not fully observable

Simple Reflex Agent

- Selects action based on the current percept
- Does not track history of percepts
- Simple but can fail if the environment is not fully observable
    - Agent cannot figure out if a car infront is braking by observing only a single image frame

Simple Reflex Agent



Simple Reflex Agent

```
function SIMPLE-REFLEX-AGENT(percept) return an action
    persistent: rules, a set of condition-action rules
    state ← INTERPRET-INPUT(percept)
    rule ← RULE-MATCH(state, rules)
    action ← rule.ACTION
    return action
```