COMP 424 Artificial Intelligence - Winter 2019

Wenzong Xia

January 17, 2019

Contents

Part I

Introduction

Biological Intelligence

What is biological intelligence? A mix of general-purpose and special-purpose algorithms. **General purpose:** Memory formation, updating, retrieval. Learning new tasks. **Special-purpose:** Recognizing visual patterns. Recognizing sounds. Learning language. All are integrated seamlessly.

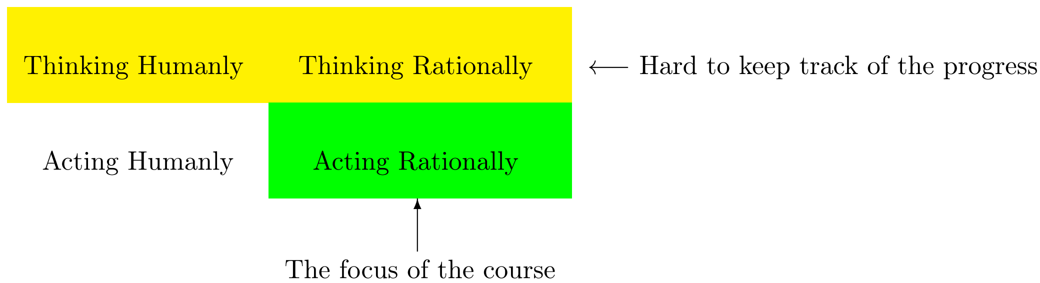
Definition of AI

Working definition of AI: Developing models and algorithms that can produce rational behaviors in response to incoming stimulus and information.

Human intelligence:		Artificial Intelligence:
• Sensory processing:		
• Visual cortex	➔	Computer vision
• Auditory cortex	➔	Signal/speech processing
• Somatosensory cortex	➔	Haptics
• Motor cortex	➔	Robotics
• Cognitive functions		
• Memory	➔	Knowledge representation
• Reasoning	➔	Search, inference
• Executive control	➔	Planning, decision-making
• Learning	➔	Model learning
• Language	➔	Language understanding

Goals of AI

The general goals of AI:



Acting Rationally

- Rational behavior = doing the 'right' thing.
- Doing what is expected to **maximize goal achievement**, given the **available information** and **available resources**.
  - Does not necessarily require thinking (e.g. blinking reflex).
  - But in many cases, thinking serves rational behavior.
  - This is the flavor of AI we will focus on.

Rational Agents

- This course is about designing **rational agents**.
  - An agent is an entity that perceives and acts.
  - Goal: Learn a function mapping percept histories to actions: $f : p^h \rightarrow A$
- A rational agent implements this function such as to maximize performance.
- **Caveat**: resources constraints (time, space, energy, bandwidth,...) which make perfect rationality unachievable.

- **Objective**: Find best function for given information and resources.

Part II

Basic Tools

1 Search

1.1 Uninformed Search

Search is at the heart of all AI systems. Typical setup of an AI task includes:

- **Knowledge**:
  - Formal representation of the problem to be solved.
  - Includes definitions of state, goal, available actions.
  - Other factual or probabilistic knowledge about the problem.
- **Search**:
  - Process of finding a solution (or possibly, the **best** solution).

1.1.1 Defining a Search Problem

Before we solve a problem, we need to formulate the problem into a search problem:

- **State space** S: all possible configurations of the domain.
- **Initial state** $s_0 \in S$: the start state.
- **Goal states** $G \subset S$: the set of end states.
- **Operators** A: the actions available (often defined in terms of mapping from state to successor state).
- **Path**: a sequence of states and operators.
- **Path cost** C: a number associated with any path.
- **Solution**: a path from s_0 to $s_g \in G$.
- **Optimal solution**: a path with minimum cost.

1.1.2 Basic Assumptions

For next few lectures, simplifying assumptions for now. Later, we will consider search in other settings.

- **Static:** The environment will not change in the course of the search. vs **dynamic** environment.
- **Observable:** The agent always know the current state. vs **unobservable** environment.
- **Discrete:** At any given states, there are only finite many actions to choose from. vs **continuous** states.
- **Deterministic:** Each action has exactly one outcome. vs **stochastic** environment.

Under these assumptions, the solution to any problem is a fixed sequence of actions.

Eight Puzzle Example

1.1.3 Search Graph vs Search Tree

The process of looking for a sequence of actions that reach the goal is called **search**. A search algorithm takes a problem as input and returns a solution in the form of an action sequence. Once a solution is found, the actions it recommended can be carried out and thus in a execution phase.

Search Graph

Search graph can visualize the state space search in terms of a graph. Graph defined by a set of vertices and a set of edges connecting the vertices. Nodes correspond to states in S. Edges correspond to operators.

Search Tree

Search tree represents the exploration paths in a search procedure. Nodes represent partial solutions, including a state in S. Edges correspond to operators.

1.1.4 Data Structure for Search Tree

Defining a search tree node:

- Each node contains a state id (from states in the graph).
- Nodes also contain additional information:
  - The parent state and the operator used to generate it.
  - Cost of the path so far.
  - Depth of the node.

Expanding a search tree node:

- Applying all legal operators to the state.
- Generating nodes for all the corresponding successor states.

1.1.5 Generic Search Algorithm

Algorithm 1.1.1: Generic Search Algorithm

```
Result: A solution/Failure
Initialize the search tree using the initial state of the problem;
Repeat
if No candidate nodes can be expanded then
  | return Failure
Choose a node for expansion, according to some search strategy;
if The node contains a goal state then
  | return the corresponding path
else
  | Expand the node by applying each applicable operator, generating the successor state and adding
  |   the resulting node to the tree
end
```

Part III

Logical Representations

2 | Game Playing

3 | Logical Reasoning

4 | Classical Planning

Part IV

Probabilistic Representations

5 | Probabilistic Reasoning

6 | Learning Probabilistic Models

Part V

Utility Theory

7 | Reasoning With Utilities

8 | Sequential Reasoning And Decision-making

9 | Learning Complex Sequential Decisions

10 Applications