

Foundations of Artificial Intelligence

4. Introduction: Environments and Problem Solving Methods

Malte Helmert

University of Basel

March 8, 2021

Foundations of Artificial Intelligence

March 8, 2021 — 4. Introduction: Environments and Problem Solving Methods

4.1 Environments of Rational Agents

4.2 Problem Solving Methods

4.3 Classification of AI Topics

4.4 Summary

Introduction: Overview

Chapter overview: introduction

- ▶ 1. What is Artificial Intelligence?
- ▶ 2. AI Past and Present
- ▶ 3. Rational Agents
- ▶ 4. Environments and Problem Solving Methods

4.1 Environments of Rational Agents

AI Problems

AI Problems

AI problem: performance measure + agent model + environment

German: Performance-Mass, Agentenmodell, Umgebung

- ▶ agent model:
 - ▶ Which actions are at the agent's disposal?
 - ▶ Which observations can it make?
- ▶ environment:
 - ▶ Which aspects of the world are relevant for the agent?
 - ▶ How does the world react to the agent's actions?
 - ▶ Which observations does it send to the agent?

Example Problem: Autonomous Taxi

Example (Autonomous Taxi)

environment:

- ▶ streets, vehicles, pedestrians, weather, ...

performance measure:

- ▶ punctuality, safety, profit, legality, comfort, ...

agent model:

- ▶ actions: steering, accelerating, braking, changing gears, honking, ...
- ▶ observations: cameras, acceleration sensors, GPS, touchpad, ...

Example Problem: Web Shopping Bot

Example (Web Shopping Bot)

environment:

- ▶ web pages, products, sellers, ...

performance measure:

- ▶ cost and quality of bought products, shipping time, ...

agent model:

- ▶ actions: querying the user, following links, filling in forms, ...
- ▶ observations: HTML pages (text, images, scripts, metadata), user input, ...

Classification of Environments

- ▶ properties of environment determine character of an AI problem
- ▶ classify according to criteria such as:
 - ▶ static vs. dynamic
 - ▶ deterministic vs. non-deterministic vs. stochastic
 - ▶ fully vs. partially vs. not observable
 - ▶ discrete vs. continuous
 - ▶ single-agent vs. multi-agent

Properties of Environments

Example (Properties of Environments)

| | Rubik's Cube | backgammon | shopping bot | taxi |
|---------------|--------------|------------|--------------|------|
| static | yes | (yes) | (yes) | no |
| deterministic | | | | |
| observability | | | | |
| discrete | | | | |
| agents | | | | |

static vs. dynamic

Does the state of the environment remain the same while the agent is contemplating its next action?

German: statisch, dynamisch

Properties of Environments

Example (Properties of Environments)

| | Rubik's Cube | backgammon | shopping bot | taxi |
|---------------|--------------|------------|--------------|------|
| static | yes | (yes) | (yes) | no |
| deterministic | yes | stochastic | (yes) | no |
| observability | | | | |
| discrete | | | | |
| agents | | | | |

deterministic vs. non-deterministic vs. stochastic

Is the next state of the environment fully determined by the current state and the agent's next action?

If not: is the next state affected by randomness?

German: deterministisch, nichtdeterministisch, stochastisch

Properties of Environments

Example (Properties of Environments)

| | Rubik's Cube | backgammon | shopping bot | taxi |
|---------------|--------------|------------|--------------|-----------|
| static | yes | (yes) | (yes) | no |
| deterministic | yes | stochastic | (yes) | no |
| observability | fully | fully | partially | partially |
| discrete | | | | |
| agents | | | | |

fully vs. partially vs. not observable

Do the agent's observations completely determine the state of the environment?

If not: can the agent at least determine some aspects of the state of the environment?

German: vollständig/teilweise/nicht beobachtbar

Properties of Environments

Example (Properties of Environments)

| | Rubik's Cube | backgammon | shopping bot | taxi |
|---------------|--------------|------------|--------------|-----------|
| static | yes | (yes) | (yes) | no |
| deterministic | yes | stochastic | (yes) | no |
| observability | fully | fully | partially | partially |
| discrete | yes | yes | yes | no |
| agents | | | | |

discrete vs. continuous

Is the environment's state given by discrete or by continuous parameters?

also applies to: actions of the agent, observations, elapsing time

German: diskret, stetig

Properties of Environments

Example (Properties of Environments)

| | Rubik's Cube | backgammon | shopping bot | taxi |
|---------------|--------------|-----------------|--------------|-----------|
| static | yes | (yes) | (yes) | no |
| deterministic | yes | stochastic | (yes) | no |
| observability | fully | fully | partially | partially |
| discrete | yes | yes | yes | no |
| agents | 1 | 2 (adversaries) | (1) | many |

single-agent vs. multi-agent

Must other agents be considered?

If yes: do the agents behave cooperatively, selfishly, or are they adversaries?

German: ein/mehrere Agenten; Gegenspieler

Properties of Environments

Example (Properties of Environments)

| | Rubik's Cube | backgammon | shopping bot | taxi |
|---------------|--------------|-----------------|--------------|-----------|
| static | yes | (yes) | (yes) | no |
| deterministic | yes | stochastic | (yes) | no |
| observability | fully | fully | partially | partially |
| discrete | yes | yes | yes | no |
| agents | 1 | 2 (adversaries) | (1) | many |

suitable problem solving algorithms

Environments of different kinds (according to these criteria) usually require different algorithms.

The "real world" combines all unpleasant (in the sense of: difficult to handle) properties.

4.2 Problem Solving Methods

Three Approaches to Problem Solving

We can solve a **concrete AI problem** (e.g., backgammon) in several ways:

Three Problem Solving Methods

- 1 **problem-specific**: implement algorithm "by hand"
- 2 **general**: create problem description + use general algorithm (**solver**)
- 3 **learning**: **learn** (aspects of) algorithm from experience

German: problemspezifisch, allgemein, lernend

- ▶ all three approaches have strengths and weaknesses (**which?**)
- ▶ combinations are possible
- ▶ we will mostly focus on **general** algorithms, but also consider other approaches

General Problem Solvers

General problem solving:

problem instance \implies language \implies solver \implies solution

- ① **models** to classify, define and understand problems
 - ▶ What is a problem **instance**?
 - ▶ What is a **solution**?
 - ▶ What is a **good/optimal** solution?
- ② **languages** to represent problem instances
- ③ **algorithms** to find solutions

German: Problem Instanz, Sprache, Solver/Löser, Lösung, Modelle

4.3 Classification of AI Topics

Classification of AI Topics

Many areas of AI are essentially characterized by

- ▶ the **properties of environments** they consider and
- ▶ which of the three **problem solving approaches** they use.

We conclude the introduction by giving some examples

- ▶ within this course and
- ▶ beyond the course (“advanced topics”).

Examples: Classification of AI Topics

Course Topic: Informed Search Algorithms

environment:

- ▶ **static** vs. dynamic
- ▶ **deterministic** vs. non-deterministic vs. stochastic
- ▶ **fully** vs. partially vs. not **observable**
- ▶ **discrete** vs. continuous
- ▶ **single-agent** vs. multi-agent

problem solving method:

- ▶ **problem-specific** vs. general vs. learning

Examples: Classification of AI Topics

Course Topic: Constraint Satisfaction Problems

environment:

- ▶ **static** vs. dynamic
- ▶ **deterministic** vs. non-deterministic vs. stochastic
- ▶ **fully** vs. partially vs. not **observable**
- ▶ **discrete** vs. continuous
- ▶ **single-agent** vs. multi-agent

problem solving method:

- ▶ problem-specific vs. **general** vs. learning

Examples: Classification of AI Topics

Course Topic: Board Games

environment:

- ▶ **static** vs. dynamic
- ▶ **deterministic** vs. non-deterministic vs. stochastic
- ▶ **fully** vs. partially vs. not **observable**
- ▶ **discrete** vs. continuous
- ▶ single-agent vs. **multi-agent** (**adversarial**)

problem solving method:

- ▶ **problem-specific** vs. general vs. learning

Examples: Classification of AI Topics

Advanced Topic: General Game Playing

environment:

- ▶ **static** vs. dynamic
- ▶ **deterministic** vs. non-deterministic vs. (stochastic)
- ▶ **fully** vs. partially vs. not **observable**
- ▶ **discrete** vs. continuous
- ▶ single-agent vs. **multi-agent** (**adversarial**)

problem solving method:

- ▶ problem-specific vs. **general** vs. learning

Examples: Classification of AI Topics

Course Topic: Classical Planning

environment:

- ▶ **static** vs. dynamic
- ▶ **deterministic** vs. non-deterministic vs. stochastic
- ▶ **fully** vs. partially vs. not **observable**
- ▶ **discrete** vs. continuous
- ▶ **single-agent** vs. multi-agent

problem solving method:

- ▶ problem-specific vs. **general** vs. learning

Examples: Classification of AI Topics

Course Topic: Acting under Uncertainty

environment:

- ▶ **static** vs. dynamic
- ▶ deterministic vs. non-deterministic vs. **stochastic**
- ▶ **fully** vs. partially vs. not **observable**
- ▶ **discrete** vs. continuous
- ▶ **single-agent** vs. multi-agent

problem solving method:

- ▶ **problem-specific** vs. general vs. learning

Examples: Classification of AI Topics

Advanced Topic: Reinforcement Learning

environment:

- ▶ **static** vs. dynamic
- ▶ deterministic vs. non-deterministic vs. **stochastic**
- ▶ **fully** vs. partially vs. not **observable**
- ▶ **discrete** vs. continuous
- ▶ **single-agent** vs. multi-agent

problem solving method:

- ▶ **problem-specific** vs. general vs. **learning**

4.4 Summary

Summary (1)

AI problem: performance measure + agent model + environment

Properties of **environment** critical for choice of suitable algorithm:

- ▶ **static** vs. **dynamic**
- ▶ **deterministic** vs. **non-deterministic** vs. **stochastic**
- ▶ **fully** vs. **partially** vs. **not observable**
- ▶ **discrete** vs. **continuous**
- ▶ **single-agent** vs. **multi-agent**

Summary (2)

Three **problem solving methods**:

- ▶ **problem-specific**
- ▶ **general**
- ▶ **learning**

general problem solvers:

- ▶ **models** characterize problem instances mathematically
- ▶ **languages** describe models compactly
- ▶ algorithms use languages as **problem description** and to **exploit problem structure**