Artificial Intelligence

Christos Dimitrakakis

May 16, 2024

Contents

1 Introduction

1.1 About the course

1.1.1 Aims

This course will focus on algorithms and models for Artificial Intelligence. We will concentrate mainly on the decision making, rather than the learning, side of artificial intelligence. Learning is already addressed in statistics courses, as well as the machine learning course in the final year.

1.1.2 Philosophy

The philosophy of this course is as follows:

- We give example problems.
- We use theory to explain and generalise from those examples to general problems.
- We describe algorithms to *solve* general problems.
- We implement algorithms to solve the specific examples.

In general, the course will start from the simplest problems and slowly progress to the more complex ones.

1.1.3 How to use this notebook

- This notebook contains a summary of all the foundational material in the course. It is not sufficient for studying for the exam, or as a reference material.
- For details, use the links to external resources.
- There are some more detailed slides available for each lecture.

1.1.4 Books and schedule

Artificial Intelligence: Foundations of Computational Agents, 3rd Edition Artificial Intelligence: a Modern Approach, 4th Edition

Module	Topics	AI:FoCA	AI: aMA
1	- Preferences - Utility - States - Actions - Beliefs - Fairness	1. AI and Agents 1.2. Complexity 1.3. Application domains 1.4. Knowledge representation 2. Architecture 2.1. Control 2.2. Hierarchical control 2.3. Moral machines	2.1. Agents and environments 2.2 Rationality 2.3. Environments 2.4. Agents 2.4.1 Programs 2.4.2-3 Reflex agents 2.4.4. Goals 2.4.5. Utility 2.4.6. Learning
2	Depth-First Search Breadth-First Search	3. Search 3.1. Search in graphs	3.1 Problem-solving 3.2. Examples. 3.3. Best-First search 3.4.1. Breadth-first 3.4.3. Depth-first search
3	Heuristic Search A* Search	3.2. Uninformed search 3.3. Heuristic search	3.5.2. A* 3.6. Heuristic Functions
4	Dynamic Programming Branch and bound	 3.4. Dynamic programming 3.5. Branch and bound	3.4.2 Dijkstra
5	Constraint programming Logical reasoning Deterministic planning	4. Reasoning with constraints 4.1. Variables and Constraints 4.2. CSPs and Search 4.6. Local Search 4.8. Optimization	6. CSP 7. Logical Agents
6	Uncertainty Aleatory/Epistemic Probability Theory Bayes Theorem Probabilistic inference	9. Reasoning with Uncertainty 9.1. Probability 9.2. Independence	12.1. Acting under uncertaint 12.2. Basic probability notation 12.3. Inference 12.4. Independence 12.5. Bayes's theorem
7	Expected Utility Theory	12.1 Preferences and Utility 12.2 One-off decisions	16.1. Beliefs and Desires 16.2. utility theory 16.3. Utility functions 16.5. Decision networks
8	Markov Decision Processes Dynamic Programming	12.3 Sequential Decisions 12.4 The value of information 12.5 Decision processes	17.1. Sequential decision proble 17.2. Algorithms for MDPs
9	Alternating Zero-Sum Games Stochastic Zero-Sum Games Linear programming	14.1. Multi-agent framework 14.2. Representations of games 14.3. Perfect information games	5.1. Game Theory 5.2. Zero-Sum Games 5.3. Alpha-Beta Search 5.5. Stochastic Games
10	Belief networks	9.3. Belief Networks 9.4. Probabilistic Inference	13.1. Representing knowledge 13.2. Bayesian Networks 13.3. Exact inference in BNs
11	Supervised learning Learning as inference Learning as optimisation Stochastic gradient descent		
12	Reinforcement learning Bandit problems Q-learning Stochastic approximation		

1.1.5 Notation

- $\mathbb{R}, \mathbb{R}^d$: the real line and \$d\$-dimension Euclidean space
- $\gvertneqq^d$ the \$d\$-dimensional simplex
- $\gvertneqq(A)$ the set of distributions over A.
- $\mathbb{I}\{x\}$: indicator function (1 if x is true, 0 otherwise)
- $\mathbb{P}$: probability
- $\mathbb{E}$: expectation
- $\pi \in \mathit{\Pi}$: policies, or algorithms.
- $\mu \in \mathcal{M}$: models
- $\theta \in \mathit{\Theta}$: parameters (i.e. models parameterised by vectors in $\mathbb{R}^d$)
- u: utility
- c: cost / constraints
- $s \in S$: state
- $a \in A$: action
- $r \in \mathbb{R}$: reward

1.2 Project

1.2.1 Application project.

Application projects proposals need to contain the following:

- Domain description and goals: What is the problem, in general terms, and which aspect would you try and solve in an AI framework? Make sure to cite relevant literature and course material.
- Methodology: How would you formalise the problem mathematically? Which algorithms and/or models do you intend to apply at different stages of the project? Feel free to read widely about both the problem and algorithms and do cite relevant literature. Make sure to employ techniques taught in the course, e.g. logic + search or probabilities and MDPs etc.
- Experiment design: How would you know that the method is working? How would you compare with existing solutions? In what context would you expect an improvement? How would you measure it? How will you test the robustness of your solution over variations in the problem instance?
- Expected results: What results do you expect to obtain, and what do you think might go wrong? In what way do you expect an improvement?

1.2.2 Algorithmic project.

- Algorithmic/theory problem and goals: What is the deficiency, in general terms, of current theory and algorithms that your method would try to improve? As an example, the goal could be reducing computational complexity, increasing data efficiency, improving robustness or applicability of a specific family of algorithms; or introducing a slightly different setting to existing ones. In other words, which is the open problem you will be addressing? Make sure to cite relevant literature to better identify the problem.
- Methodology: What kind of existing algorithms, theory or technical result would you rely on? Would you be combining various existing results? What would be the most significant novelty of your methodology? Do cite relevant literature.
- Experiment design (if applicable): How would you know that the method is working? How would you compare with existing solutions? In what context would you expect an improvement? How would you measure it?
- Expected results: What results do you expect to obtain, and what do you think might go wrong? In what way do you expect an improvement?

1.2.3 Grading for projets:

Grades will be adjusted based on group size with on letter grade up/down for double/half the mean group size.

- Environments: A. Complex, well described environment that captures all of the elements of the application or algorithmic problem. B. The environment is simple or lacks description. C. An adequate environment that captures the basic setting. D. Insufficient environment or description. E. Insuffcient environment and description.
- Algorithms: A. Significantly novel algorithms that are well described. B. Some novelty in the algorithms, with good descriptions. C. Some novelty in the algorithms, but descriptions are lacking. D. Insufficient novelty or descriptions. E. Insufficient novelty and descriptions.
- Experiments: A. Thorough experiments with ablation tests and comparisons over algorithms and environments, that are well-described. B. Somewhat incomplete experiments or descriptions. C. Sufficient experiments and descriptions. D. Insufficient experiments or descriptions. E. Insufficient experiments and descriptions.

Criteria for full marks in each part of the project are the following.

1. Documenting of the work in a way that enables reproduction.

2. Technical correctness of their analysis.

3. Demonstrating that they have understood the assumptions underlying their analysis.

4. Addressing issues of reproducibility in research.

5. Addressing scientific and ethical questions where applicable, and if not, clearly explain why they are not.

6. Consulting additional resources beyond the source material with proper citations.

The follow marking guidelines are what one would expect from students attaining each grade.

Detailed grading

1.2.4 A (6)

1. Submission of a detailed report from which one can definitely reconstruct their work without referring to their code. There should be no ambiguities in the described methodology. Well-documented code where design decisions are explained.

2. Extensive analysis and discussion. Technical correctness of their analysis. Nearly error-free implementation.

3. The report should detail what models are used and what the assumptions are behind them. The conclusions of the should include appropriate caveats. When the problem includes simple decision making, the optimality metric should be well-defined and justified. Simiarly, when well-defined optimality criteria should given for the experiment design, when necessary. The design should be (to some degree of approximation, depending on problem complexity) optimal according to this criteria.

4. Appropriate methods to measure reproducibility. Use of cross-validation or hold-out sets to measure performance. Use of an unbiased methodology for algorithm, model or parameter selection. Appropriate reporting of a confidence level (e.g. using bootstrapping) in their analytical results. Relevant assumptions are mentioned when required.

5. A clear definition of a scientific question. When dealing with data relating to humans, ethical concerns, such as privacy and/or fairness should be addressed.

6. The report contains some independent thinking, or includes additional resources beyond the source material with proper citations. The students go beyond their way to research material and implement methods not discussed in the course.

1.2.5 B (5.5)

1. Submission of a report from which one can plausibly reconstruct their work without referring to their code. There should be no major ambiguities in the described methodology.

2. Technical correctness of their analysis, with a good discussion. Possibly minor errors in the implementation.

3. The report should detail what models are used, as well as the optimality criteria, including for the experiment design. The conclusions of the report must contain appropriate caveats.

4. Use of cross-validation or hold-out sets to measure performance. Use of an unbiased methodology for algorithm, model or parameter selection.

5. When dealing with data relating to humans, ethical concerns such as privacy and/or fairness should be addressed. While an analysis of this issue may not be performed, there is a substantial discussion of the issue that clearly shows understanding by the student.

6. The report contains some independent thinking, or the students mention other methods beyond the source material, with proper citations, but do not further investigate them.

1.2.6 C (5)

1. Submission of a report from which one can partially reconstruct most of their work without referring to their code. There might be some ambiguities in parts of the described methodology.

2. Technical correctness of their analysis, with an adequate discussion. Some errors in a part of the implementation.

3. The report should detail what models are used, as well as the optimality criteria and the choice of experiment design. Analysis caveats are not included.

4. Either use of cross-validation or hold-out sets to measure performance, or use of an unbiased methodology for algorithm, model or parameter selection - but in a possibly inconsistent manner.

5. When dealing with data relating to humans, ethical issues are addressed superficially.

6. There is little mention of methods beyond the source material or independent thinking.

1.2.7 D (4.5)

1. Submission of a report from which one can partially reconstruct most of their work without referring to their code. There might be serious ambiguities in parts of the described methodology.
2. Technical correctness of their analysis with limited discussion. Possibly major errors in a part of the implementation.
3. The report should detail what models are used, as well as the optimality criteria. Analysis caveats are not included.
4. Either use of cross-validation or hold-out sets to measure performance, or use of an unbiased methodology for algorithm, model or parameter selection - but in a possibly inconsistent manner.
5. When dealing with data relating to humans, ethical issues are addressed superficially or not at all.
6. There is little mention of methods beyond the source material or independent thinking.

1.2.8 E (4)

1. Submission of a report from which one can obtain a high-level idea of their work without referring to their code. There might be serious ambiguities in all of the described methodology.
2. Technical correctness of their analysis with very little discussion. Possibly major errors in only a part of the implementation.
3. The report might mention what models are used or the optimality criteria, but not in sufficient detail and caveats are not mentioned.
4. Use of cross-validation or hold-out sets to simultaneously measure performance and optimise hyperparameters, but possibly in a way that introduces some bias.
5. When dealing with data relating to humans, ethical issues are not discussed.
6. There is no mention of methods beyond the source material or independent thinking.

1.2.9 F (<3)

1. The report does not adequately explain their work.
2. There is very little discussion and major parts of the analysis are technically incorrect, or there are errors in the implementation.

3. The models used might be mentioned, but not any other details.
4. There is no effort to ensure reproducibility or robustness.
5. When applicable: Ethical issues are not mentioned.
6. There is no mention of methods beyond the source material or independent thinking.

1.3 Schedule

2.22	1-2 Introduction
2.29	3.1-3.5 Search, State Spaces, Graphs, Uniformed Search.
3.07	3.6. Informed Search, Heuristics, A*
3.14	4.1-4.2, 4.8 Constrained Search
3.21	Exercises and Project Day
3.28	9.1 Probability, Independence, Belief Networks
4.11	12.1. Preferences and Utility. 12.2 Probability. Decision making.
4.18	Markov decision processes
4.25	Markov games
5.02	Simultaneous games
5.16	Gradient methods
5.23	Fairness and privacy
5.30	Project presentations

2 single agent problems with certainty

2.1 Uninformed search

Graph definitions	10
Tree example	5
Shortcut example	5
Depth-first search	10
The shortest path problem	10
Goals and DFS	10
Shortest-path DFS	10
Breadth-first search	10
Iterative deepening	10
Uniform cost search	10
	95

2.2 Informed search

2.2.1 Heuristics

2.2.2 A^*-search

2.2.3 Dynamic programming

2.2.4 Branch and Bound

2.3 Constraints

Local search	20
Constraint Satisfaction	20
Graph Colouring	20
Meeting Scheduling	20
onstraint Optimisation	20
Travelling Salesman	20
Maximum Flow	20
Logical constraints	20
Towers of Hanoi	20
	160

2.4 Infinite choices

2.4.1 Lipschitz search

If we know the function f is Lipschitz-smooth, i.e.

$$\exists L > 0 : |f(x) - f(y)| \leq L|x - y|,$$

then we also know that for any point z:

$$f(z) < f(x) + L|x - z|, \qquad f(z) < f(y) + L|y - z|$$

1. Schubert's Algorithm (Schubert, 1972)

```
**Input:** $L > 0$, $X$, $x_0 \in X$.
**for** $t = 1, \ldots, T$ **do**
  $x_t = \arg\max_{x \in X} \min \{f(x_k) + L|x_k - x| \mid k = 0, \ldots, t - 1\}$
**end for**
```

2. Discussion

   - This is guaranteed to **converge** to the optimal solution.
   - If L is **unknown**, DIRECT (Jones et al. 1993) can be used.
   - If f is noisy, the problem becomes a **continuum bandit** problem.

2.4.2 First-order gradient methods

- Gradient descent
- Stochastic gradient descent

1. Properties
   - Incremental algorithms
   - Can converge to a **local** optimum

2.4.3 Single-variable gradient descent

1. Setting
   - Input: $f : \mathbb{R} \to \mathbb{R}$
   - Problem: $\max_x f(x)$
   - Derivative: $\frac{d}{dx} f(x) \triangleq \lim_{\Delta \to 0} \frac{f(x+\Delta)-f(x)}{\Delta}$.
2. Algorithm
   (a) Input: $x^{(0)}$, f
   (b) For $t = 1, \ldots$:
   (c) Calculate direction $g_t = \frac{d}{dx} f(x_{t-1})$
   (d) Select step-size α_t
   (e) Update $x^{(t)} = x^{(t-1)} + \alpha_t g_t$.

2.4.4 Multiple-variable gradient descent

1. Setting
   - Input: $f : \mathbb{R}^d \to \mathbb{R}$, $x = (x_1, \ldots, x_d)$
   - Problem: $\max_x f(x)$
   - Partial Derivative: $\frac{\partial}{\partial x_i} f(x) \triangleq \lim_{\Delta \to 0} \frac{f(x_1, \ldots, x_i + \Delta, \ldots, x_d) - f(x)}{\Delta}$.
   - Gradient $\nabla_x f(x) = \left[\frac{\partial}{\partial x_1} f(x), \ldots, \frac{\partial}{\partial x_i} f(x), \ldots, \frac{\partial}{\partial x_d} f(x)\right]^\top$.
2. Algorithm
   (a) Input: x_0, f
   (b) For $t = 1, \ldots$:
   (c) Calculate direction $g_t = \nabla_x f(x_{t-1})$
   (d) Select step-size α_t
   (e) Update $x_t = x_{t-1} + \alpha_t g_t$.

2.4.5 Stochastic gradient descent

1. As gradient descent with errors
   - Calculate direction $g_t = \nabla_x f(x_{t-1}) + \epsilon_t$
   - ϵ_t is typically zero-mean noise.
2. In learning from data The gradient can be broken up into a sum of gradients:
   $$f(x) = \sum_t v(x, z_t), \qquad \nabla_x f(x) = \sum_t \nabla_x v(x, z_t),$$
   $x_t = x_{t-1} + \alpha_t \nabla_x v(x, z_t)$.
3. In Bayesian quadrature The function is an expectation:
   $$f(x) = \int_Z v(x, z) p(z) dz. \qquad \nabla_x f(x) \approx \sum_t \nabla_x v(x, z_t),$$
   where $z_t \sim p(z)$ are samples from p.

3 single agent problems with uncertainty

3.1 Probability

3.1.1 Probability fundamentals

1. Probability measure P
   - Defined on a universe Ω
   - $P : \Sigma \to [0, 1]$ is a function of subsets of Ω.
   - A subset $A \subset \Omega$ is an **event** and P measures its likelihood.
2. Axioms of probability
   - $P(\Omega) = 1$
   - For $A, B \subset \Omega$, if $A \cap B = \emptyset$ then $P(A \cup B) = P(A) + P(B)$.
3. Marginalisation If $A_1, \ldots, A_n \subset \Omega$ are a partition of Ω
   $$P(B) = \sum_{i=1}^{n} P(B \cap A_i).$$

3.2 Conditional probability and independence

3.2.1 Conditional probability

1. Conditional probability The conditional probability of an event A given an event B is defined as

$$P(A|B) \triangleq \frac{P(A \cap B)}{P(B)}$$

   The above definition requires $P(B)$ to exist and be positive.

2. Conditional probabilities as a collection of probabilities More generally, we can define conditional probabilities as simply a collection of probability distributions:

$$\{P_\theta(A) \mid \theta \in \varTheta\},$$

   where $\varTheta$ is an arbitrary set.

3.2.2 The theorem of Bayes

1. Bayes's theorem

$$P(A|B) = \frac{P(B|A)}{P(B)}$$

2. The general case If $A_1, \ldots, A_n$ are a partition of $\varOmega$, meaning that they are mutually exclusive events (i.e. $A_i \cap A_j = \emptyset$ for $i \neq j$) such that one of them must be true (i.e. $\bigcup_{i=1}^n A_i = \varOmega$), then

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

   and

$$P(A_j|B) = \frac{P(B|A_j)}{\sum_{i=1}^n P(B|A_i)P(A_i)}$$

3.2.3 Independence

1. Independent events A, B are independent iff $P(A \cap B) = P(A)P(B)$.

2. Conditional independence A, B are conditionally independent given C iff $P(A \cap B|C) = P(A|C)P(B|C)$.

3.3 Random variables and expectation

3.3.1 Random variables

A random variable $f : \varOmega \to \mathbb{R}$ is a real-value function measurable with respect to the underlying probability measure P, and we write $f \sim P$.

1. The distribution of f The probability that f lies in some subset $A \subset \mathbb{R}$ is

$$P_f(A) \triangleq P(\{\omega \in \Omega : f(\omega) \in A\}).$$

2. Independence Two RVs f, g are independent in the same way that events are independent:

$$P(f \in A \wedge g \in B) = P(f \in A) P(g \in B) = P_f(A) P_g(B).$$

   In that sense, $f \sim P_f$ and $g \sim P_g$.

3.3.2 Expectation

For any real-valued random variable $f : \Omega \to \mathbb{R}$, the expectation with respect to a probability measure P is

$$\mathbb{E}_P(f) = \sum_{\omega \in \Omega} f(\omega) P(\omega).$$

1. Linearity of expectations For any RVs x, y:

$$\mathbb{E}_P(x + y) = \mathbb{E}_P(x) + \mathbb{E}_P(y)$$

2. Independence If x, y are independent RVs then $\mathbb{E}_P(xy) = \mathbb{E}(x)\,\mathbb{E}(y)$.
3. Correlation If x, y are **not** correlated then $\mathbb{E}_P(xy) = \mathbb{E}(x)\,\mathbb{E}(y)$.
4. IID (Independent and Identically Distributed) random variables A sequence x_t of r.v.s is IID if $x_t \sim P$ $(x_1, \ldots, x_t, \ldots, x_T) \sim P^T$.

3.3.3 Conditional expectation

The conditional expectation of a random variable $f : \Omega \to \mathbb{R}$, with respect to a probability measure P conditioned on some event B is simply

$$\mathbb{E}_P(f|B) = \sum_{\omega \in \Omega} f(\omega) P(\omega|B).$$

3.4 Statistical Decision Theory

3.4.1 Expected utility

1. Actions, outcomes and utility In this setting, we obtain random outcomes that depend on our actions.
   - Actions $a \in A$
   - Outcomes $\omega \in \Omega$.
   - Probability of outcomes $P(\omega \mid a)$
   - Utility $U : \Omega \to \mathbb{R}$

2. Expected utility The expected utility of an action is:

$$\mathbb{E}_P[U \mid a] = \sum_{\omega \in \Omega} U(\omega) P(\omega \mid a).$$

3. The expected utility hypothesis We prefer a to a' if and only if

$$\mathbb{E}_P[U \mid a] \geq \mathbb{E}_P[U \mid a']$$

3.5 Supervised learning

3.5.1 Supervised learning

3.6 Markov decision processes

3.6.1 Markov decision process

- Action space A.
- State space S.
- Transition kernel $s_{t+1} = j \mid s_t = s, a_t = a \sim P_\mu(j \mid s, a)$.
- Reward $r_t = \rho(s_t, a_t)$ (can also be random).
- Utility

$$U_t = \sum_{k=t}^{T} r_t.$$

3.6.2 Value functions

1. The state value function For any given MDP μ and policy π we define

$$V^\pi_{\mu,t}(s) \triangleq \mathbb{E}^\pi_{\mu,t} \left[U_t \mid s_t = s \right]$$

2. The state-action value function

$$Q^\pi_{\mu,t}(s, a) \triangleq \mathbb{E}^\pi_{\mu,t} \left[U_t \mid s_t = s, a_t = a \right]$$

3. The optimal value functions For an optimal policy π^*

$$V^*_{\mu,t}(s) \triangleq V^{\pi^*}_{\mu,t}(s) \geq V^\pi_{\mu,t}(s), \qquad Q^*_{\mu,t}(s, a) \triangleq Q^{\pi^*}_{\mu,t}(s, a) \geq V^\pi_{\mu,t}(s, a)$$

3.6.3 The Bellman equations

1. State value function

$$\begin{aligned}
V^{\pi}_{\mu,t}(s) &\triangleq \mathbb{E}^{\pi}_{\mu}[U_t \mid s_t = s] \\
&= \mathbb{E}^{\pi}_{\mu}[r_t + U_{t+1} \mid s_t = s] \\
&= \mathbb{E}^{\pi}_{\mu}[r_t \mid s_t = s] + \mathbb{E}^{\pi}_{\mu}[U_{t+1} \mid s_t = s] \\
&= \mathbb{E}^{\pi}_{\mu}[r_t \mid s_t = s] + \sum_{j \in S} \mathbb{E}^{\pi}_{\mu}[U_{t+1} \mid s_{t+1} = j] \, \mathbb{P}^{\pi}_{\mu}(s_{t+1} = j \mid s_t = s) \\
&= \mathbb{E}^{\pi}_{\mu}[r_t \mid s_t = s] + \sum_{j \in S} V^{\pi}_{\mu,t+1}(j) \, \mathbb{P}^{\pi}_{\mu}(s_{t+1} = j \mid s_t = s) \\
&= \mathbb{E}^{\pi}_{\mu}[r_t \mid s_t = s] + \sum_{j \in S} V^{\pi}_{\mu,t+1}(j) \sum_{a \in A} P_{\mu}(j \mid s, a) \pi(a_t \mid s_t).
\end{aligned}$$

2. State-action value function

$$Q^{\pi}_{\mu,t}(s) = \rho(s,a) + \sum_{j \in S} V^{\pi}_{\mu,t+1}(j) P_{\mu}(j \mid s, a)$$

3.6.4 Optimal policies

1. Bellman optimality condition The value function of the optimal policy satisfies this:

$$V^{*}_{\mu,t}(s) = \max_{a} [\rho(s,a) + \sum_{j \in S} V^{*}_{\mu,t+1}(j) P_{\mu}(j \mid s, a)]$$

2. Dynamic programming To find V^*, Q^*, first initialise $V^{*}_{\mu,T}(s) = \max_a \rho(s,a)$. Then for $t = T-1, T-2, \dots, 1$:

$$\begin{aligned}
Q^{*}_{\mu,t}(s,a) &= \rho(s,a) + \sum_{j \in S} V^{*}_{\mu,t+1}(j) P_{\mu}(j \mid s, a). \\
V^{*}_{\mu,t}(s) &= \max_{a} Q^{*}_{\mu,t}(s,a).
\end{aligned}$$

3. The optimal policy The optimal policy is deterministic with:

$$a_t = \arg\max_{a} Q^{*}(s_t, a)$$

4 Multi-player Games

4.1 Introduction

4.1.1 Multi-agent decision making

- **Two** versus n-player games

- **Co-operative** games
- **Zero-sum** games
- General-sum games
- **Stochastic** games
- Partial information games

4.1.2 Prisoner's Dilemma

4.1.3 Prisoner's Choice

4.2 Two-Player zero-sum Games

4.2.1 Extensive-form alternating-move games

- At time t:
- Player chooses action a_t, which is revealed.
- Player chooses action b_t.
- Player a receives $\rho(a_t, b_t)$ and b receives $-\rho(a_t, b_t)$.

The utility for each player is $U = \sum_t \rho(a_t, b_t)$.

4.2.2 Backwards induction for ZSG

```
for $t = T, T-1, \dots, 1$ do
  x
end for
```

4.2.3 Normal-form simultaneous-move games

- Player a chooses action a in secret.
- Player b chooses action b in secret.
- Players observe both actions
- Player a receives $U(a, b)$, and b receives $-U(a, b)$.

4.2.4 Linear Programming

4.2.5 The linear programming problem

Linear programming is a constrained minimisation problem where the objective and the constraints are both linear.

$$\min_x \ \theta^\top x \\ \text{s.t. } c^\top x \geq 0.$$

We can have

5 Project plan

1. Define a deterministic, fully observable Wumpus world
2. Find optimal policies for any instance of this world.
3. Consider a randomised version of the Wumpus world, where

3.1. Your moves are not deterministic 3.2. The monster moves in a way relative to your position

1. Advanced versions

Only one of then can be done in practice 4.1. Partially-observable wumpus (where the positions are unknown) (logic-based) 4.2. Same, but with randomised observations. (probability-based) 4.3. Adversarial Wumpus (where the monster also solves its own optimisation problem)

5.1 Wumpus world project

5.1.1 Wumpus world

- State: $s_t = (x_t, y_t, d_t, w_t)$, the x-y location of the agent, the direction, and the amount of arrows left.
- Actions: $a \in \{U, D, L, R, S\}$ for up, down, left, right, and maybe shoot
- Rewards are given for each step, for killing the Wumpus, dying, or finding the treasure

1. Deterministic/Stochastic Wumpus
    - An action/observation is always the same/is random
2. Observable/Unobservable Wumpus
    - We know where the holes, the treasure and the Wumpus is/they are unknown
3. Static/Dynamic/Strategic/ Wumpus
    - The Wumpus is stationary/moves according to a fixed policy/has goals to achieve

5.1.2 Deterministic, Observable Wumpus

This is the simplest setting. It is a deterministic planning problem. For this, you can

1. Define a way to describe the Wumpus world

2. Find a policy for solving the Wumpus world as given. This policy is going to be deterministic and Markov.

Of course, the optimal policy for each **instance** of the Wumpus problem is going to be different.

I recommend summarising the Wumpus problem in two parts: (a) A matrix G where $G[x, y]$ is a number indicating what is contained in this location, (b) x_t, y_t, d_t, w_t being the agent-relevant variables.

You can either use any logical planning algorithm, or an MDP algorithm with deterministic transitions for this problem.

5.1.3 Stochastic, Observable Wumpus

To make the environment stochastic, we can add the following extensions

(a) The Wumpus moves according to some stochastic policy. For example, the Wumpus could randomly move in a direction, so that on average it moves away from us. (b) Our actions do not always work (e.g. we may turn in the wrong direction) (c) We do not always die when we encounter a hole or the Wumpus.

For this, you can

1. Define a way to describe the Wumpus world

2. Find a policy for solving the Wumpus world as given. This policy is going to be deterministic and Markov.

Of course, the optimal policy for each **instance** of the Wumpus problem is going to be different.

I recommend summarising the Wumpus problem in two parts: (a) A matrix G where $G[x, y]$ is a number indicating what is contained in this location, (b) x_t, y_t, d_t, w_t being the agent-relevant variables.

You can either use any logical planning algorithm, or an MDP algorithm with deterministic transitions for this problem.

5.1.4 Deterministic, Unobservable Wumpus

This setting is significantly harder to work with. Now we have observations whenever we are near a hole or the Wumpus.

You can either: (a) Use a logical description of the world, and a SAT algorithm. (b) Use a probabilistic description with all probabilities being 0 or 1, and an MDP algorithm.

In either case, a simple idea is to summarise the knowledge of the Wumpus problem as a matrix G where $G[x, y]$ indicates one of:

- Empty.

- Hole.

- Wumpus.

- Treasure.
- Breeze Observed.
- Stink Smelled.
- Unknown.

For simplicity, you can always start with the setting where you know you are dealing with one of a **small number** of possible worlds. Then the problem can be split in two parts:

1. Inferring the set of possible worlds
2. Choosing the optimal policy given what you know.

1. Inferring the world

   For each observation, you can simply eliminate all worlds incompatible with it. Then you are left with a set of possible worlds.

2. Choosing a policy

   You can evaluate any policy in every remaining possible world. Suppose that all of them agree that some action a is better than some action a', then it is obvious that a is better than a'. It is not trivial to find the optimal policy under uncertainty, but you can try.

5.1.5 Static, Stochastic-Observation, Unobservable Wumpus

Here we assume the Wumpus does not move, and observations are stochastic: sometimes we feel a breeze, sometimes not. We assume we know the probability of a breeze.

The first problem is to summarise what we know about the Wumpus problem. Now we can have an entry $G[x, y]$ in the matrix which is a **vector of probabilities** for the possible contents of the co-ordinate: (Empty, Hole, Wumpus, Treasure)

For simplicity, you can always start with the setting where you know you are dealing with one of a **small number** of possible worlds. Then you only need to deal with the probability of each world being the right one.

1. Inferring the probabilities of possible worlds
2. Choosing the optimal policy given what you know.

1. Inferring the world

   For each observation, you calculate the posterior probability of all worlds, $P_t(\mu)$

2. Choosing a policy

   You can evaluate any policy in every remaining possible world. In fact, you can find the optimal policy for each one of them.