CSCE 580: Introduction to AI
CSCE 581: Trusted AI

Lecture 23 & 24: Making Decisions – Simple and Complex

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

14TH NOV & 16TH NOV, 2023

Carolinian Creed: "I will practice personal and academic integrity."

Organization of Lectures 23 & 24

- Introduction Segment
  - Recap of Lectures 21 and 22
- Main Segment
  - Making Decisions
  - Making simple decisions - Maximum Expected Utility (MEU)
  - Making complex decisions - Markov Decision Processes (MDPs)
- Concluding Segment
  - Course Project Discussion
  - Quiz 4
  - About Next Lecture – Lecture 25
  - Ask me anything

Introduction Section

Recap of Lecture 21 and 22

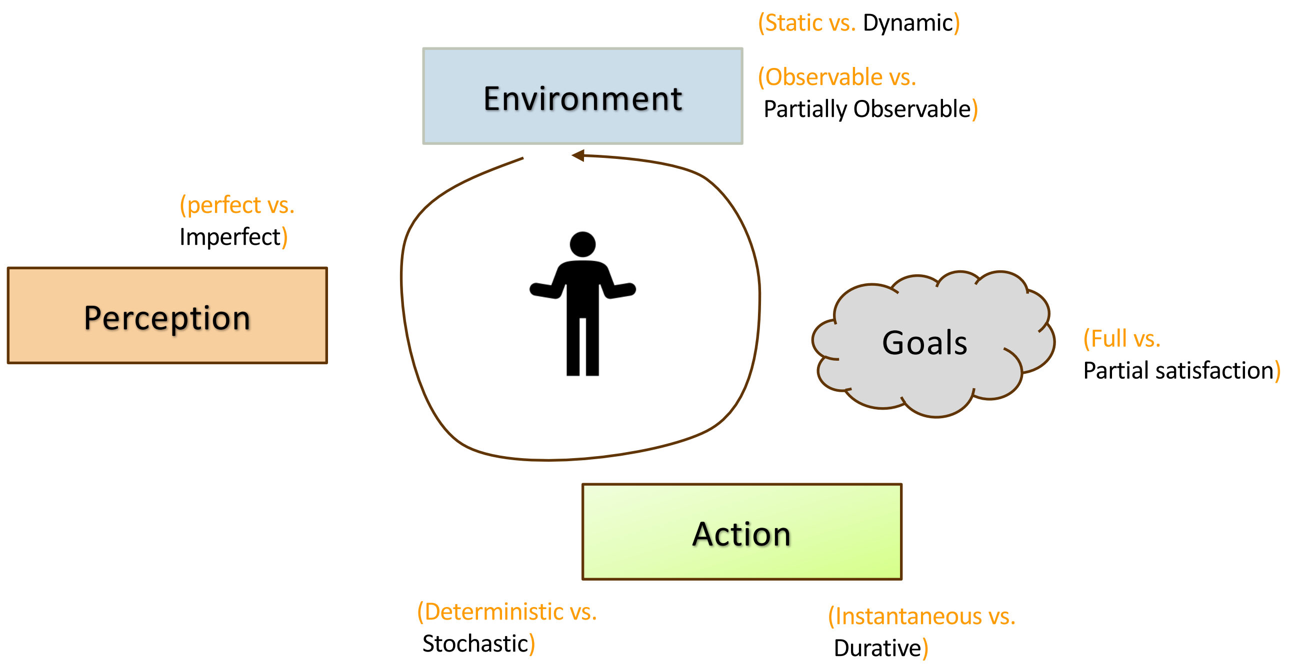
- Topic discussed
  - Text Processing
  - Language Models (LMs)
  - Learning for LMs with NN
  - Large LMs

Graduate Paper Presentation

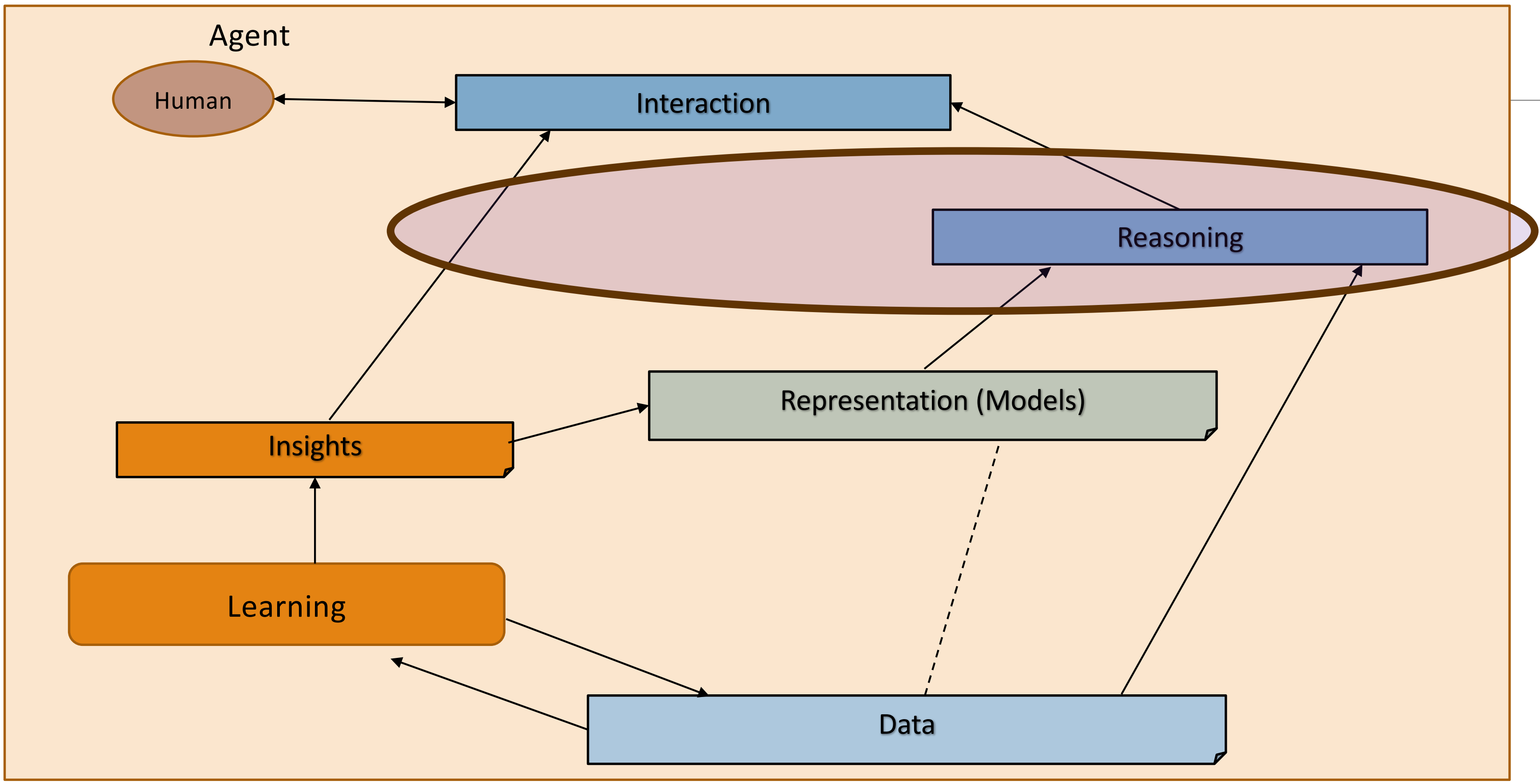
- Papers between 2021-2023 (last 3 years)
- At top AI venues: AAAI, Neurips, IJCAI, ICML, ICLR, or discuss with instructor

- Guideline on presentation
  - Summary of the paper
  - Critique (+ves/ -ves)
  - Relevance to your and anyone else's project in the class

Intelligent Agent Model



Relationship Between Main AI Topics



Where We Are in the Course

CSCE 580/ 581 – In This Course

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 4-5: Search, Heuristics - Decision Making
- Week 6: Constraints, Optimization – Decision Making
- Week 7: Classical Machine Learning – Decision Making, Explanation
- Week 8: Machine Learning - Classification
- Week 9: Machine Learning - Classification – Trust Issues and Mitigation Methods
- Topic 10: Learning neural network, deep learning, Adversarial attacks
- Week 11: Large Language Models – Representation, Issues
- Topic 12: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 13: Planning, Reinforcement Learning – Sequential decision making
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

Main Section

Credit: Retrieved from internet

Making Decisions

Real World Decisions

Decision situation: going to airport from home

- Actions:
  - Take own car
  - Take a cab/ limo
  - Take a ride-share
  - Take a bus
  - Hitch-hike
  - Walk

Students at a College Campus

An ideal solution should be:

- free of any errors (Ex: grammatical, calculation, etc.)
- utilize all the information given by the user completely and give a reasonable, practical, and optimal solution.

Example Query:
I am making a purchase of $1000 using my credit card. I have a due of $2000 on my account. My total credit line is $2,800. Would you recommend I make the purchase now or later in the future?

Ideal Solution:
Based on the information you have provided, it is not advisable to make the purchase now as you already have a due of $2000 on your account, which is close to your total credit line of $2,800. This means you are utilizing a significant portion of your available credit, and adding another $1000 to your balance would further increase your credit utilization ratio (CUR), which can negatively impact your credit score.

The Quality of Everyday Decisions

Source: https://www.umassd.edu/fycm/decision-making/process/

Major variability due to:
- Emotions
- Biases
- Increasing data volume
- Cognitive ability to process
  - Decreases under stress and constraints
  - Decreases with age*

* **Source**: A Review of Decision-Making Processes: Weighing the Risks and Benefits of Aging, Mara Mather, https://www.ncbi.nlm.nih.gov/books/NBK83778/

Evidence #1:
Poor Medical Adherence

Taking medicines

- 20 -30 % of medication prescriptions are never filled
- ~50 % of medications for chronic disease are not taken as prescribed

Impact

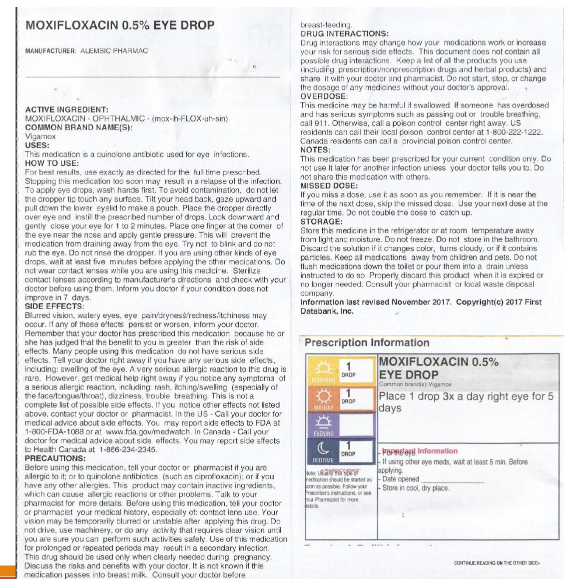
- causes 125,000 deaths, at least 10 percent of hospitalizations
- [Costs the American health care system](#) between $100 billion and $289 billion a year.

Finding relevant guidance is hard, one reason for non-adherence and high costs in health

Sources:

- Medication Nonadherence, A Diagnosable and Treatable Medical Condition, Zachary A. Marcum, Mary Ann Sevick, Steven M. Handler, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3976600/, 2013.
- https://www.nytimes.com/2017/04/17/well/the-cost-of-not-taking-your-medicine.html

Example:
Hard to understand medicine's information



MOXIFLOXACIN 0.5% EYE DROP

MANUFACTURER: ALEMBIC PHARMAC

ACTIVE INGREDIENT:
MOXIFLOXACIN - OPHTHALMIC - (mox-ih-FLOX-uh-sin)
COMMON BRAND NAME(S):
Vigamox
USES:
This medication is a quinolone antibiotic used for eye infections.
HOW TO USE:
For best results, use exactly as directed for the full time prescribed. Stopping this medication too soon may result in a relapse of the infection. To apply eye drops, wash hands first. To avoid contamination, do not let the dropper tip touch any surface. Tilt your head back, gaze upward and pull down the lower eyelid to make a pouch. Place the dropper directly over eye and instill the prescribed number of drops. Look downward and gently close your eye for 1 to 2 minutes. Place one finger at the corner of the eye near the nose and apply gentle pressure. This will prevent the medication from draining away from the eye. Try not to blink and do not rub the eye. Do not rinse the dropper. If you are using other kinds of eye drops, wait at least five minutes before applying the other medications. Do not wear contact lenses while you are using this medicine. Sterilize contact lenses according to manufacturer's directions and check with your doctor before using them. Inform you doctor if your condition does not improve in 7 days.
SIDE EFFECTS:
Blurred vision, watery eyes, eye pain/dryness/redness/itchiness may occur. If any of these effects persist or worsen, inform your doctor. Remember that your doctor has prescribed this medication because he or she has judged that the benefit to you is greater than the risk of side effects. Many people using this medication do not have serious side effects. Tell your doctor right away if you have any serious side effects, including: swelling of the eye. A very serious allergic reaction to this drug is rare. However, get medical help right away if you notice any symptoms of a serious allergic reaction, including: rash, itching/swelling (especially of the face/tongue/throat), dizziness, trouble breathing. This is not a complete list of possible side effects. If you notice other effects not listed above, contact your doctor or pharmacist. In the US - Call your doctor for medical advice about side effects. You may report side effects to FDA at 1-800-FDA-1088 or at www.fda.gov/medwatch. In Canada - Call your doctor for medical advice about side effects. You may report side effects to Health Canada at 1-866-234-2345.
PRECAUTIONS:
Before using this medication, tell your doctor or pharmacist if you are allergic to it; or to quinolone antibiotics (such as ciprofloxacin); or if you have any other allergies. This product may contain inactive ingredients, which can cause allergic reactions or other problems. Talk to your pharmacist for more details. Before using this medication, tell your doctor or pharmacist your medical history, especially of: contact lens use. Your vision may be temporarily blurred or unstable after applying this drug. Do not drive, use machinery, or do any activity that requires clear vision until you are sure you can perform such activities safely. Use of this medication for prolonged or repeated periods may result in a secondary infection. This drug should be used only when clearly needed during pregnancy. Discuss the risks and benefits with your doctor. It is not known if this medication passes into breast milk. Consult your doctor before breast-feeding.
DRUG INTERACTIONS:
Drug interactions may change how your medications work or increase your risk for serious side effects. This document does not contain all possible drug interactions. Keep a list of all the products you use (including prescription/nonprescription drugs and herbal products) and share it with your doctor and pharmacist. Do not start, stop, or change the dosage of any medicines without your doctor's approval.
OVERDOSE:
This medicine may be harmful if swallowed. If someone has overdosed and has serious symptoms such as passing out or trouble breathing, call 911. Otherwise, call a poison control center right away. US residents can call their local poison control center at 1-800-222-1222. Canada residents can call a provincial poison control center.
NOTES:
This medication has been prescribed for your current condition only. Do not use it later for another infection unless your doctor tells you to. Do not share this medication with others.
MISSED DOSE:
If you miss a dose, use it as soon as you remember. If it is near the time of the next dose, skip the missed dose. Use your next dose at the regular time. Do not double the dose to catch up.
STORAGE:
Store this medicine in the refrigerator or at room temperature away from light and moisture. Do not freeze. Do not store in the bathroom. Discard the solution if it changes color, turns cloudy, or if it contains particles. Keep all medications away from children and pets. Do not flush medications down the toilet or pour them into a drain unless instructed to do so. Properly discard this product when it is expired or no longer needed. Consult your pharmacist or local waste disposal company.
Information last revised November 2017. Copyright(c) 2017 First Databank, Inc.

Prescription Information

MORNING 1 DROP
MIDDAY 1 DROP
EVENING
BEDTIME 1 DROP

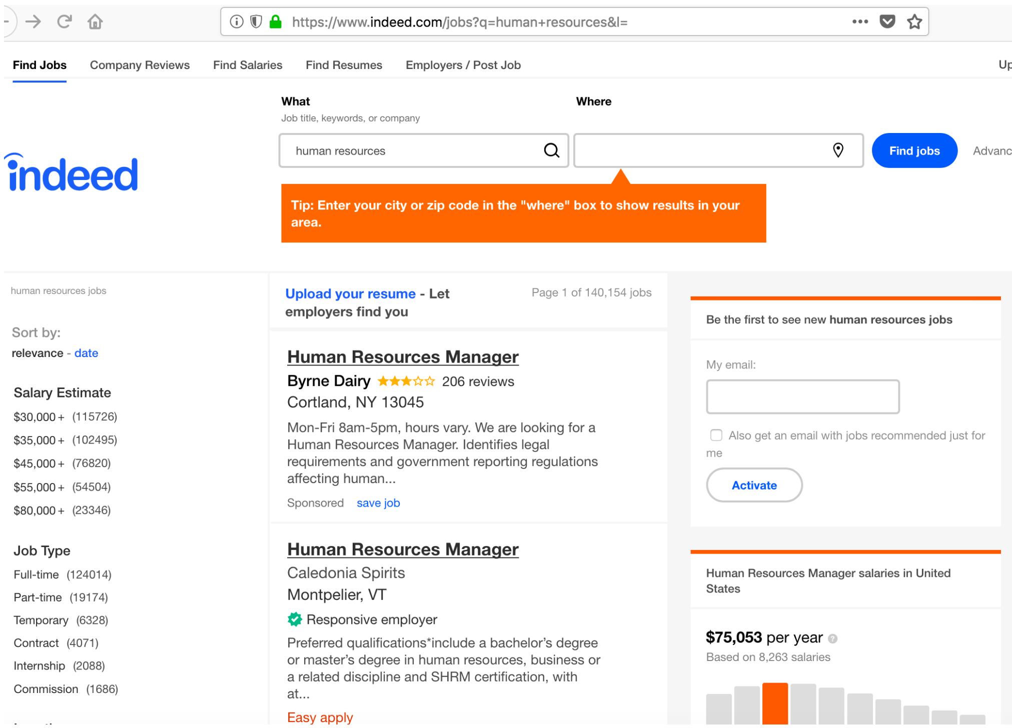
MOXIFLOXACIN 0.5% EYE DROP
Common brand(s): Vigamox
Place 1 drop 3x a day right eye for 5 days

PHARMACY ADVICE
Note: Usually this type of medication should be started as soon as possible. Follow your Prescriber's instructions, or see your Pharmacist for more details.

Important Information
- For the eye.
- If using other eye meds, wait at least 5 min. Before applying.
- Date opened ______________
- Store in cool, dry place.

CONTINUE READING ON THE OTHER SIDE>

Evidence #2: Matching Demand to Supply of Jobs is Inadequate Demand-Supply Gap in Jobs Market [1] and Yet, Low Work Satisfaction/ Engagement [2]



Job search at a portal

- Finding jobs was generally hard around the world (Dec 2019), except for in tight labor markets like US (3.5% unemployment)
- Workforce satisfaction/ engagement was generally low around the world – people did not find jobs they were match for [1,2]
- COVID-19 impact [3]:
  - *Nearly half of global workforce at risk of losing livelihoods in informal sector*
  - *9-12% job loss in the formal sector around the world*
  - *14.7% unemployment in US by end of April 2020 [4]*

1. **Source**: Global Skills Trends, Training Needs and Lifelong Learning Strategies for the Future of Work, ILO & OECD Report 2018, http://www.g20.utoronto.ca/2018/g20_global_skills_trends_and_lll_oecd-ilo.pdf
2. **Source**: For 2016, job satisfaction: US – 32%, Global – 13%, https://www.gallup.com/workplace/236495/worldwide-employee-engagement-crisis.aspx
3. **https://www.ilo.org/global/about-the-ilo/newsroom/news/WCMS_743036/lang--en/index.htm**
4. **https://www.bls.gov/news.release/empsit.nr0.htm**

Decision Imperative: Corona Virus Pandemic

Emerging Scenario Around the World*

- Millions of cases, hundreds of thousands of deaths
- Businesses disrupted, millions going out of business
- Millions loosing jobs

* Numbers changing continuously; see reference for details

Decisions Need to be Made

- About disease
  - Understand disease
  - Tackle disease
- Understand impact to society: economy, supply chain
- Advise on actions to take
  - Individual
  - Group
  - Societal policy

Resource: https://github.com/biplav-s/covid19-info/wiki/Important-Information-About-COVID19

Before and After: (AI) Decision Support

Today's tools: Static, non-interactive, non-contextual, lack explanations

Future tools: Dynamic to data, interactive, contextual, explaining with data, anywhere, multi-modal, social (group dependency), societally relevant, …

Future has potential to improve people's lives, promote well-being and reduce waste

Simple Decisions

Setting for a Decision

- An agent has a set of actions available, $A = \{a_i\}$ and is in a state s
- There may be an uncertainty about current state. So, the agent assigns a probability to current state P(s) to each possible current state.
- When an action is taken, there may be uncertainty about outcome. So, resulting state is:
  $P(s' \mid s, a)$
- The probability of reaching state s' after executing a in the current state is:
  $P(RESULT(a) = s') = \Sigma_s P(s) P(s' \mid s, a)$

Note: P(RESULT(a) = s') requires perception, learning, knowledge representation and inference

Adapted from:
Russell & Norvig, AI: A Modern Approach

Making a Simple Decision

- Choose best action based on the desirability of immediate outcome
- Have a utility function U(s) expressing desirability of a state (s)
- Expected utility of an action given the evidence, EU(a), is the average utility value of the outcome, weighted by the probability of that outcome.

$$EU(a) = \Sigma_{s'} P(RESULT(a) = s')\, U(s')$$

- Principle of maximum expected utility (MEU): rational agent chooses an action which maximizes its maximum expected utility

$$action = \text{argmax}_a\, EU(a)$$

Decision situation: going to airport from home

- Actions:
  - Take own car
  - Take a cab/ limo
  - Take a ride-share
  - Take a bus
  - Hitch-hike
  - Walk

Adapted from:
Russell & Norvig, AI: A Modern Approach

Utility Functions: Modeling Preferences

- Notations
  - A > B: agent (decision maker) prefers A over B
  - A ~ B: agent (decision maker) is indifferent between A and B
  - A ≳ B: agent (decision maker) prefers A over B oris indifferent between A and B
- Convention
  - Outcome of an action is a lottery: $L = [p_1,S_1;\ p_2,S_2;\ \ldots;\ p_n,S_n]$
- Utility function U
  - U(A) > U(B), if and only if, A > B
  - U(A) = U(B), if and only if, A ~ B

Example: Choosing a Winning

- Won a game and have to choose
  - Choice 1: Take $1M
  - Choice 2: Toss coin; Heads => $2.5 M, Tails => 0
- What will you choose?

Example: Choosing a Winning

- Won a game and have to choose
  - Choice 1: Take $1M
  - Choice 2: Toss coin; Heads => $2.5 M, Tails => 0
- Expected Monetary Value (EMV)
  - Choice 1: $1M
  - Choice 2: ½ . $2.5M + ½ . 0 = $1.25M
- Expected Utility depends on current money

Example: Choosing a Winning

- Won a game show and have to choose
  - Choice 1: Take \$1M
  - Choice 2: Toss coin; Heads => \$2.5 M, Tails => 0
- Expected Utility depends on current money(k)
  - EU(Accept) = ½ $U(S_k)$ + ½ $U(S_{k+\$2.5M})$
  - EU(Decline) = $U(S_{k+\$1M})$

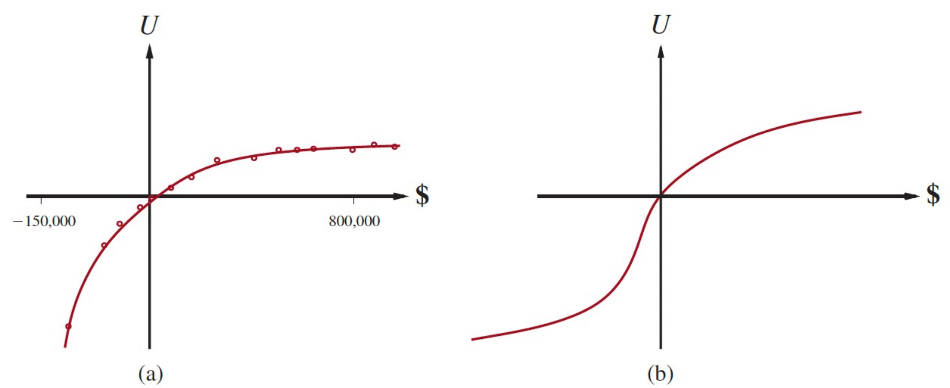


Figure 16.2 The utility of money. (a) Empirical data for Mr. Beard over a limited range. (b) A typical curve for the full range.

Adapted from/ image credit:
Russell & Norvig, AI: A Modern Approach

Example: S-Curve, Risk

- S-Curve: Fig 16.2(b)
- utility of a lottery is less than a sure thing
  - U(Lottery) < U($SureThing_{EMV(L)}$)
  - **Risk averse agents**: prefer sure payoff than expected monetary value of a gamble
  - **Risk seeking agents**: (people already in debt)
  - **Certainty equivalent** of lottery: agent will accept in lieu of a lottery
- According to studies, people will accept $400 (approx.) in lieu of a gamble than gives $1,000 half the time and $0 other
- **Insurance premium**: difference between EMV of a lottery and its certainty equivalent
  - Risk aversion / positive insurance premium is the basis of insurance industry

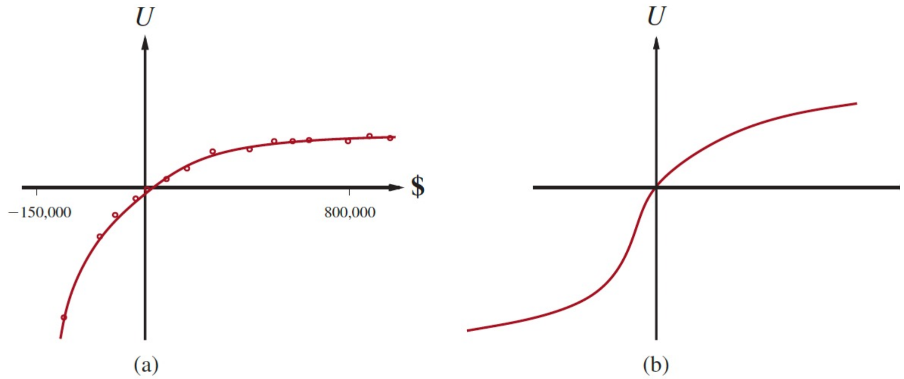


Figure 16.2 The utility of money. (a) Empirical data for Mr. Beard over a limited range. (b) A typical curve for the full range.

Adapted from/ image credit:
Russell & Norvig, AI: A Modern Approach

Humans <u>STILL</u> Do Now Always Follow Utility Theory

- Subjects in this experiment are given a choice between lotteries A and B:
  - Comparison scenario 1
    - A : 80% chance of $4000
    - B : 100% chance of $3000
  - Comparison scenario 2
    - C : 20% chance of $4000
    - D : 25% chance of $3000

Tversky and Kahneman (1982) experiment

- The majority of survey respondents choose B over A and C over D.
  - Comparison scenario 1:
    - A: 0.8 * 4000 + 0.2 * 0 = **3200**
    - B: 3000
  - Comparison scenario 2:
    - C: 0. 2* 4000 + 0.8 * 0 = **800**
    - D: 0.25 * 3000 + 0.75 * 0 = 750

Consistent utility: A over B and C over D.

Source: Russell & Norvig, AI: A Modern Approach

Multi-Attribute/ Objective Optimization

Decision situation: going to airport from home

- Actions:
  - Take own car
  - Take a cab/ limo
  - Take a ride-share
  - Take a bus
  - Hitch-hike
  - Walk

Attributes: cost, time, comfort, certainty of arrival time, …

Choosing By Decision Dominance

Two attribute case shown

- Choose by dominance

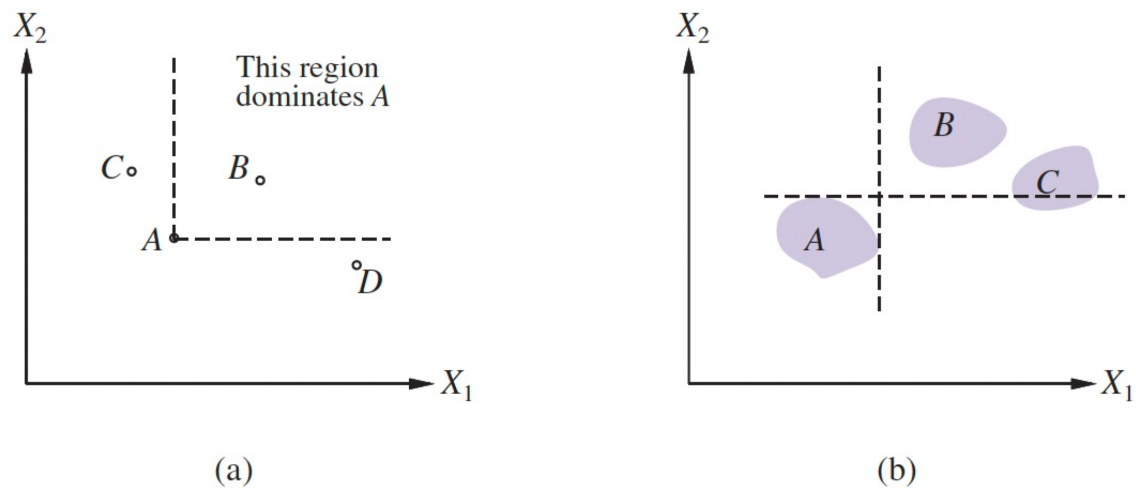


Figure 16.4 Strict dominance. (a) Deterministic: Option A is strictly dominated by B but not by C or D. (b) Uncertain: A is strictly dominated by B but not by C.

Adapted from/ image credit:
Russell & Norvig, AI: A Modern Approach

Choosing by Formal Verification of Correctness

Table 1: Different product interaction categories considered, query identifiers, queries posed under each category, variables used in each query with their corresponding chosen values and constraints to consider while answering the user queries.

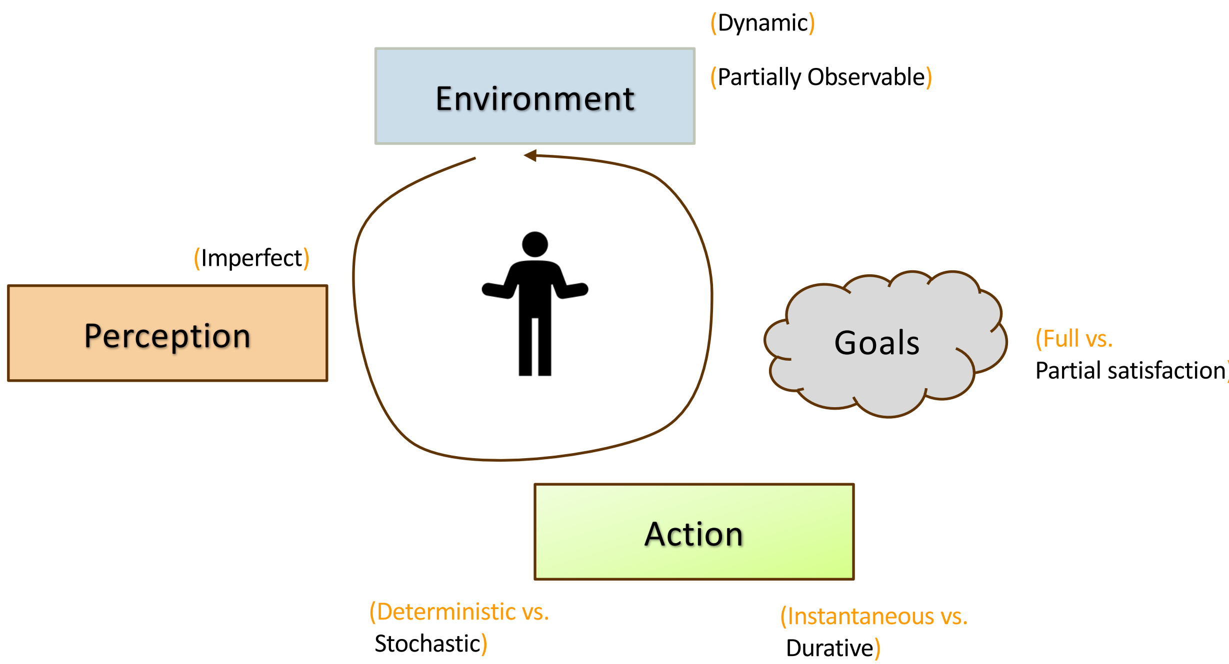
Product Interactions	Query Identifier	Queries	Variables with their values	Constraints
CC	Q1	I am making a **purchase of $1000** using my credit card. My **billing cycle is from March 25th to April 24th**. Today is March 31st, and I have a **due of $2000** on my account. My total **credit line is $2,800**. Would you recommend I make the purchase now or later in the future?	x_{PA} = 1000, x_{BC} = (March 25th - April 24th), x_{DA} = 2000, x_{CL} = 2800	$x_{DA} + x_{PA} < x_{CL}$
	Q2	I am making a **purchase of $1000** using my credit card. My **billing cycle is from March 25th to April 24th**. Today is March 31st, and I have a **due of $2000** on my account. My total **credit line is $3,800**. Would you recommend I make the purchase now or later in the future?	x_{PA} = 1000, x_{BC} = (March 25th - April 24th), x_{DA} = 2000, x_{CL} = 3800	
	Q3	I get **5% cashback** if I buy furniture using my credit card. I am **buying furniture worth $1000** using my credit card. My **billing cycle is from March 25th to April 24th**. Today is March 31st, and I have a **due of $2000** on my account. My total **credit line is $2,800**. Would you recommend I make the purchase now or later in the future?	x_{CP} = 5%, x_{PA} = 1000, x_{BC} = (March 25th - April 24th), x_{DA} = 2000, x_{CL} = 2800	
	Q4	I get **5% cashback** if I buy furniture using my credit card. I am **buying furniture worth $1000** using my credit card. My **billing cycle is from March 25th to April 24th**. Today is March 31st, and I have a **due of $2000** on my account. My total **credit line is $3,800**. Would you recommend I make the purchase now or later in the future?	x_{CP} = 5%, x_{PA} = 1000, x_{BC} = (March 25th - April 24th), x_{DA} = 2000, x_{CL} = 3800	
CC (AAVE)	Q5	I be makin' a **purchase of $1000** usin' i's credit card. I's **billin' cycle be from march 25th to april 24th**. Today be march 31ts, and i done a **due of $2000** on i's account. I's total **credit line be $2,800**. Would you recommend i make de purchase now o lateh in de future?	x_{PA} = 1000, x_{BC} = (March 25th - April 24th), x_{DA} = 2000, x_{CL} = 2800	

Source: Can LLMs be Good Financial Advisors?: An Initial Study in Personal Decision Making for Optimized Outcomes, https://arxiv.org/abs/2307.07422

Complex Decisions

Complex Decisions

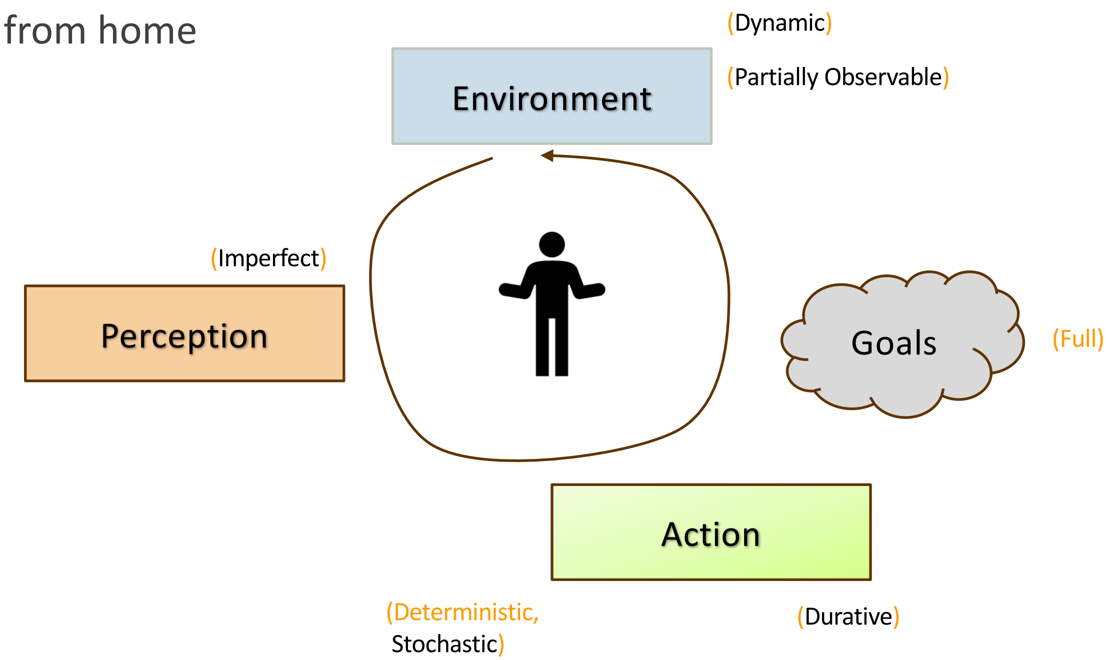
- Making a sequence of decisions
- Making a single decision but with
  - Environment changing
  - Actions not being deterministic
  - Perception not being perfect
  - …



Making a Sequence of Decisions

Decision situation: driving to airport from home

- Actions:
  - Take a LEFT at first intersection
  - ENTER a highway
  - GETOUT a highway at EXIT-X
  - Turn RIGHT at intersection
  - PARK in Premium lot
  - ..



Optimal Decision

- What is it? There is no absolute answer. In AI, there is the concept of a **rational** agent.
- Acting rationally: acting such that one can achieve one's goals given one's beliefs (and information)
  - But what are one's goals
  - Are the goals always of achievement?
- Some options
  - Perfect rationality: maximize expected utility at every time instant
    - Given the available information; can be computationally expensive
    - "Doing the right thing"
  - Bounded optimality: do as well as possible given computational resources
    - Expected utility as high as any other agent with similar resources
  - Calculative rationality: *eventually* returns what would have been the rational choice

What Is It?

- As a working principle
  - Bounded or Calculative Rationality
- In observable and deterministic scenarios
  - Maximize utility: (benefit – cost)
- In scenarios with uncertainty and/ or unobservable
  - Maximize ***expected*** utility: (benefit – cost)

Example Situation – Course Selection

- A person wants to pass an academic program in two majors: A and B
- There are three subjects: A, B and C, each with three levels (*1, *2, *3). There are thus 9 courses: A1, A2, A3, B1, B2, B3, C1, C2, C3
- To graduate, at least one course at beginner (*1) level is needed in major(s) of choice(s), and two courses at intermediate levels (*2) are needed
- **Optimality considerations** in the problem
  - Least courses, fastest time to graduate, class size, friends attending together, …
- Answer questions
  - Q1: How many minimum courses does the person have to take ?
  - Q2: Can a person graduate in 2 majors studying 3 courses only?
  - …

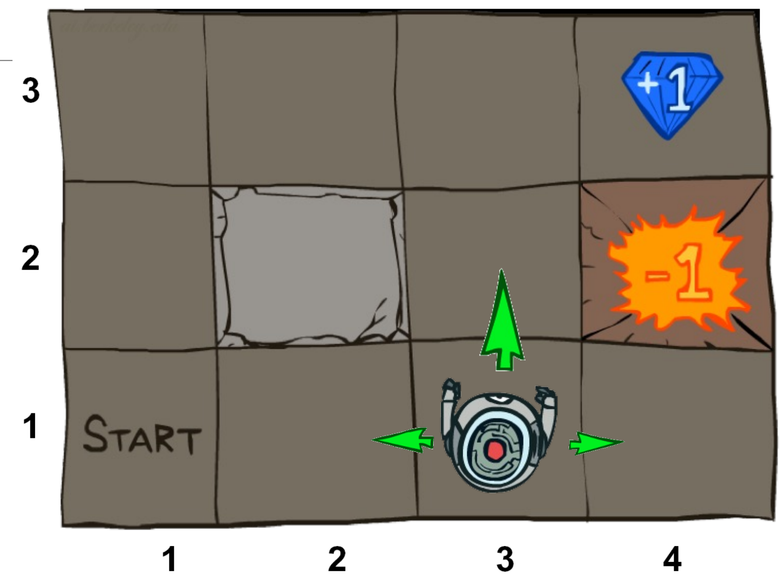
Algorithms for Optimality

- Problem specific methods
  - Path finding
  - Linear programming
  - Constraint satisfaction and optimization

- General Purposed - methods for optimality in search

Synthetic Example: Grid World

- A maze-like problem
  - The agent lives in a grid
  - Walls block the agent's path
- Noisy movement: actions do not always go as planned
  - 80% of the time, the action North takes the agent North (if there is no wall there)
  - 10% of the time, North takes the agent West; 10% East
  - If there is a wall in the direction the agent would have been taken, the agent stays put
- The agent receives rewards each time step
  - Small "living" reward each step (can be negative)
  - Big rewards come at the end (good or bad)

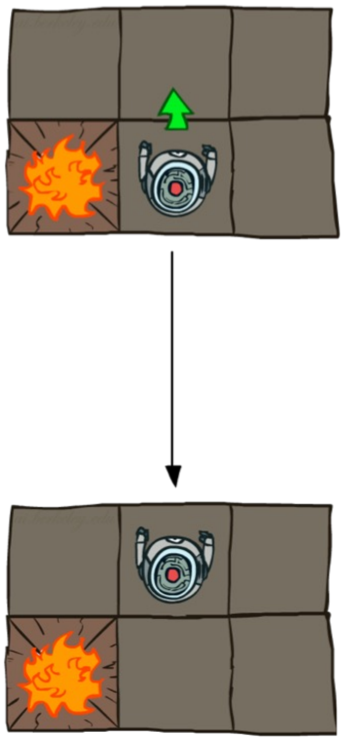


Slide adapted from: Dan Klein and Pieter Abbeel's AI lecture
Original example in Russell & Norvig's AI book

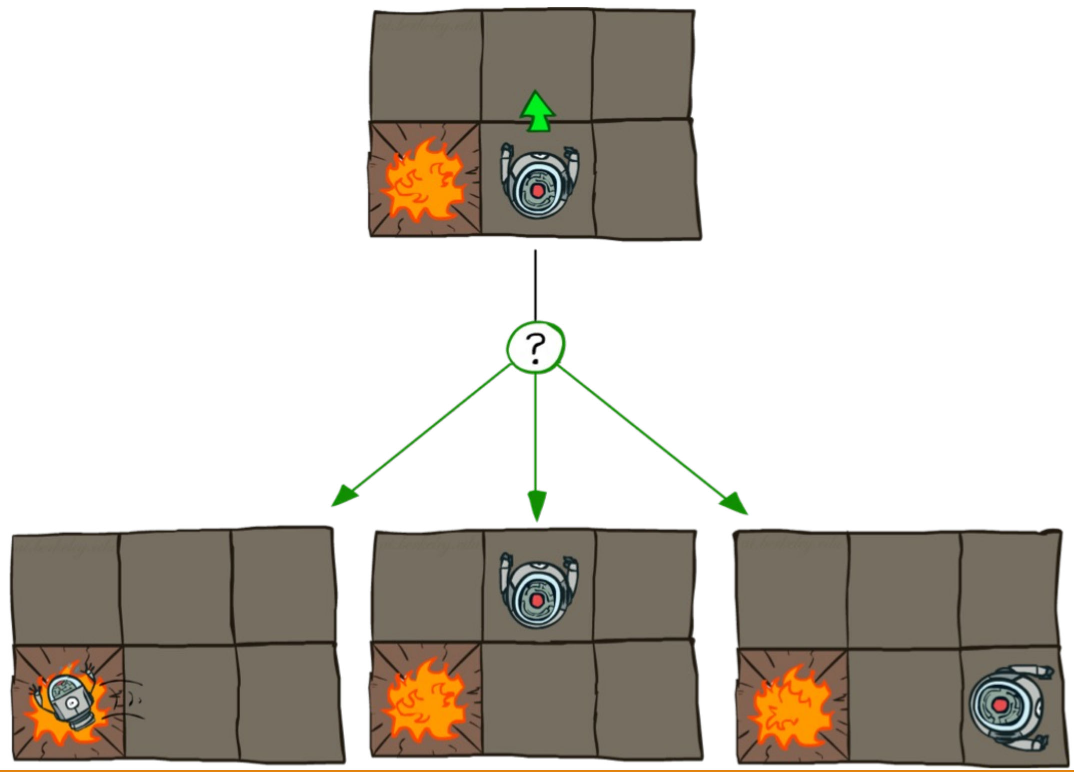
Slide adapted from: Dan Klein and Pieter Abbeel's AI lecture
Original example in Russell & Norvig's AI book

Grid World Actions

Deterministic Grid World



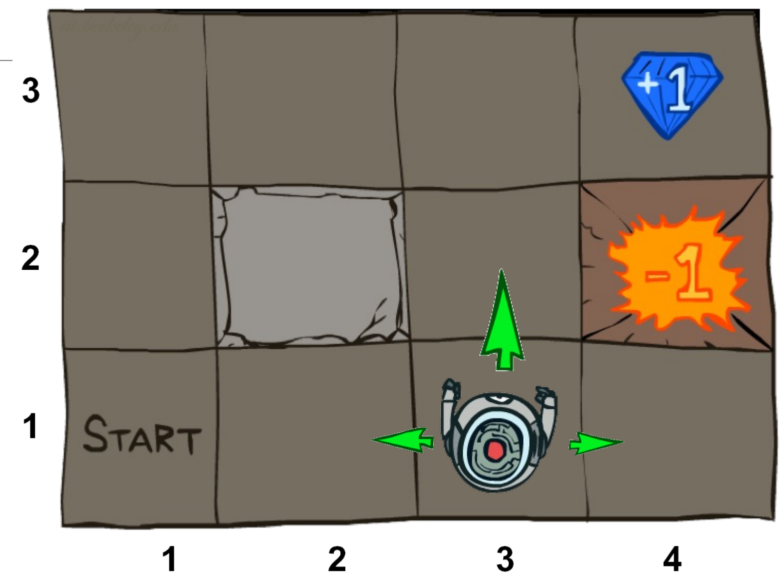
Stochastic Grid World



Markov Decision Processes

An MDP is defined by:

- A set of states $s \in S$
- A set of actions $a \in A$
- A transition function T(s, a, s')
  - Probability that a from s leads to s', i.e., P(s'| s, a)
  - Also called the model or the dynamics
- A reward function R(s, a, s')
  - Sometimes just R(s) or R(s')
- A start state
- Maybe a terminal state



MDPs are non-deterministic search problems

Slide adapted from: Dan Klein and Pieter Abbeel's AI lecture
Original example in Russell & Norvig's AI book

[Demo – gridworld manual intro (L8D1)]

Markovian Assumption

"Markov" generally means that given the present state, the future and the past are independent

For Markov decision processes, "Markov" means action outcomes depend only on the current state

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1}, \ldots S_0 = s_0)$$

$$=$$

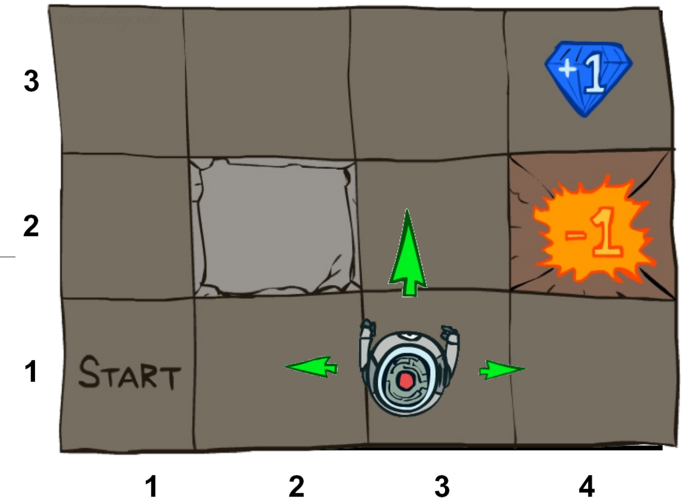
$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t)$$

Andrey Markov
(1856-1922)

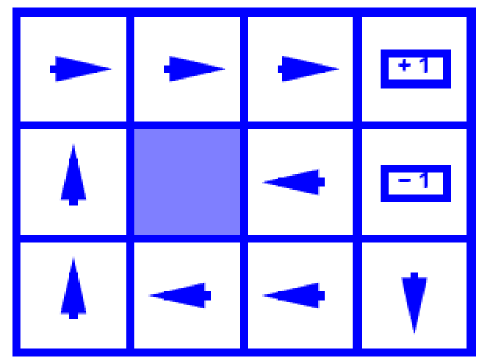
Slide adapted from: Dan Klein and Pieter Abbeel's AI lecture

Output: Policies

- In deterministic single-agent search problems, we have a plan, or sequence of actions, from start to a goal



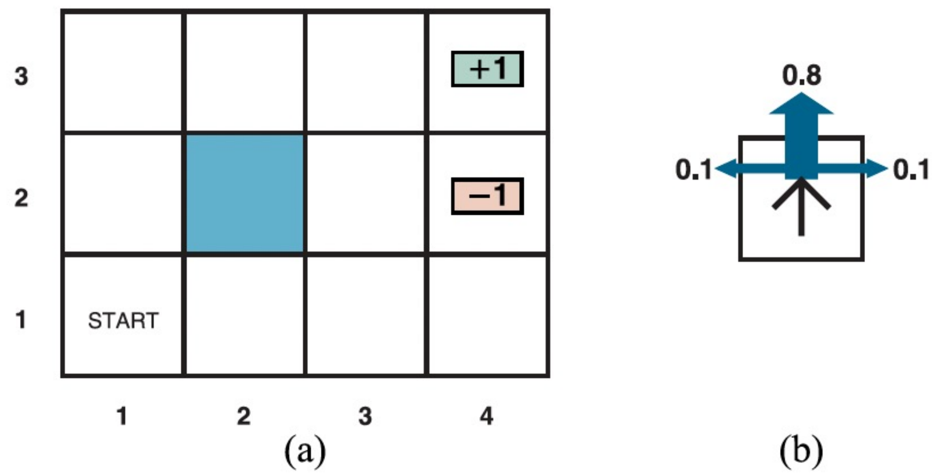
- For MDPs, we want an optimal policy $\pi^*: S \rightarrow A$
  - A policy π gives an action for each state
  - An optimal policy is one that maximizes expected utility if followed



R(s) = -0.01

Slide adapted from: Dan Klein and Pieter Abbeel's AI lecture

Example 2:



r = -0.04 for
non-terminal states

Figure 17.1 (a) A simple, stochastic 4×3 environment that presents the agent with a sequential decision problem. (b) Illustration of the transition model of the environment: the "intended" outcome occurs with probability 0.8, but with probability 0.2 the agent moves at right angles to the intended direction. A collision with a wall results in no movement. Transitions into the two terminal states have reward +1 and –1, respectively, and all other transitions have a reward of –0.04.

Adapted from/ image credit:
Russell & Norvig, AI: A Modern Approach

Example 2: Optimal Policies Under Different Situations

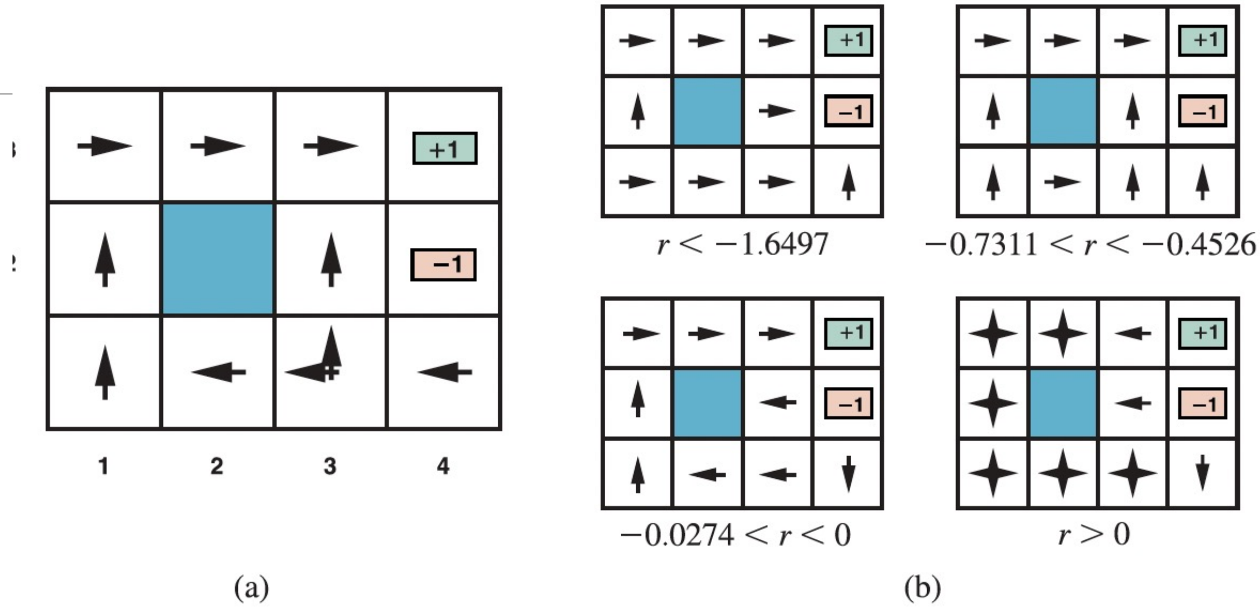


Figure 17.2 (a) The optimal policies for the stochastic environment with $r = -0.04$ for transitions between nonterminal states. There are two policies because in state (3,1) both *Left* and *Up* are optimal. (b) Optimal policies for four different ranges of r.

Adapted from/ image credit:
Russell & Norvig, AI: A Modern Approach

Example 2:

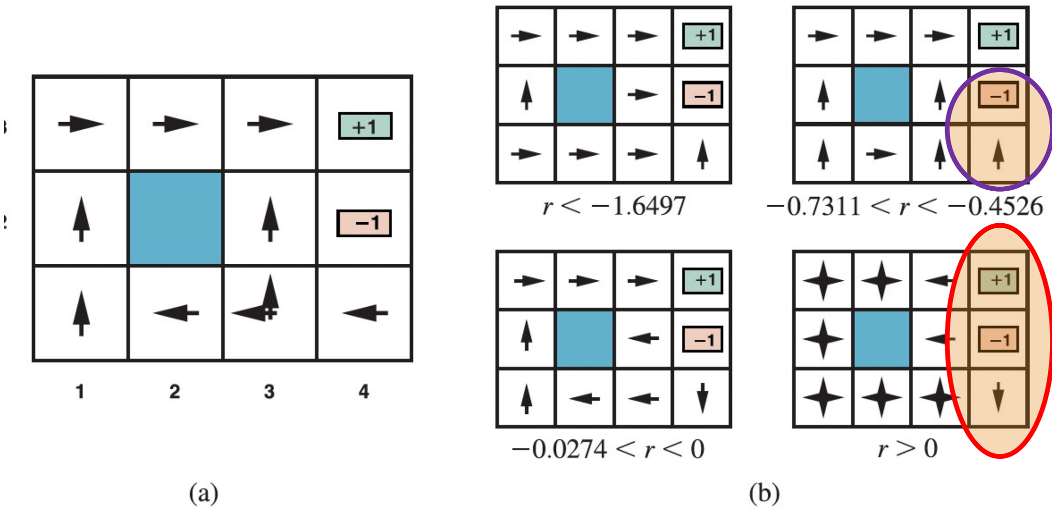


Figure 17.2 (a) The optimal policies for the stochastic environment with $r = -0.04$ for transitions between nonterminal states. There are two policies because in state (3,1) both *Left* and *Up* are optimal. (b) Optimal policies for four different ranges of r.

Agent decides to directly go to -1 state rather than high cost of trying to go to +1

Agent decides NOT to go to any terminal state

Adapted from/ image credit:
Russell & Norvig, AI: A Modern Approach

On Finding Solution to MDP

- Dynamic programming: simplifying a problem by recursively breaking it into smaller pieces, solving it and assembling full solution from optimal solutions to sub-problems
- Optimal policy: a policy that yields the highest expected utility
- Setting: how much time we have – finite v/s infinite horizon
  - For **finite horizon**, solution may depend on time left. Policy is called **nonstationary**.
  - For in**finite horizon**, solution will not depend on time left. Policy is called **stationary**.
- Utility of a state sequence – by **additive discounted rewards**
  - U_h ($[s_0,a_0; s_i,a_1; \ldots]) = R(s_o, a_o, s_1) + \gamma R(s_1, a_1, s_2) + \gamma^2 R(s_2, a_2, s_3) + \ldots$

On Finding Solution to MDP

- **Key Idea**: in an optimal policy, one would have chosen the action that maximizes the reward for the next step plus the expected discounted utility of the subsequent state
  - $\pi^*(s) = \text{argmax}_a \ \Sigma_{s'} \ P(s' \mid s, a) \ [\ R(s, a, s) \ + \gamma \ U(s')]$

- **Key Idea**: The utility of a state is the expected reward for the next transition plus the discounted utility of the next state, assuming the agent chooses the optimal action
  - $U(s) = \max_a \ \Sigma_{s'} P(s' \mid s, a) \ [R(s, a, s) \ + \gamma \ U(s')]$
  - Bellman equation

Finding Policy

- Value Iteration – iterate over value of states; offline; optimal
- Policy Iteration – iterate over policies ; offline; optimal
- Linear programming - offline; optimal
- Monte carlo planning – online; approximate

Exercise and Code

- MDP Solution Methods
  - From Book: AI – A Modern Approach, https://github.com/aimacode/aima-python/blob/master/mdp.ipynb
  - More applications https://github.com/aimacode/aima-python/blob/master/mdp_apps.ipynb

Source: Russell & Norvig, AI: A Modern Approach

Two Party Decisions - Games

- Games
  - Cooperative games
  - Non-cooperative games
    - Adversarial games

- What is value of cooperation ?
  - Prisoner's dilemma

Two Party Decisions - Games

Prisoner's dilemma

- Two prisoners are caught for a robbery. They can testify against each other (-5 years to other; 0 themselves), stay silent (-10 year if other testifies, but -1 if they do not).
- For A: testifying (defecting) is a better choice (- 0 - 5 * ½) = -2.5 over remaining silent (cooperating) (-1 -10 * ½) = -6.5 // Assuming B will decided with probability 0.5
- For B: similarly, testifying is better
- For both, cooperating is better: -1 each, but the authorities would try to prevent it

Prisoner B / **Prisoner A**	**Prisoner B stays silent (*cooperates*)**	**Prisoner B testifies (*defects*)**
Prisoner A stays silent (*cooperates*)	Each serves 1 year	Prisoner A: 10 years Prisoner B: goes free
Prisoner A testifies (*defects*)	Prisoner A: goes free Prisoner B: 10 years	Each serves 5 years

Application of Decision Theory

- Help with individual decisions:
  - driving,
  - buying/ auctions, …
- Help with group decisions:
  - hiring/ interviewing,
  - merger/ acquisition, …
- Help with adversarial situations
  - Price discovery
  - Avoiding collusion
- Help with autonomous systems
  - Space crafts, drones, underwater navigation, …

Course Project

Project Discussion: What Problem Fascinates You ?

- Data
  - Water
  - Finance
  - …
- Analytics
  - Search, Optimization, Learning, Planning, …
- Application
  - Building chatbot
- Users
  - Diverse demographics
  - Diverse abilities
  - Multiple human languages

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

Project Discussion: Dates and Deliverables

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
- Nov 30, 2023
  - Project report due
- Dec 5 / 7, 2023
  - In-class presentation

Skeleton: A Basic Chatbot

- Run in an infinite loop until the user wants to quit
- Handle any user response
  - User can quit by typing “Quit” or “quit” or just “q”
  - User can enter any other text and the program has to handle it. The program should write back what the user entered and say – “I do not know this information”.
- Handle known user query types // Depends on your project
  - “Tell me about N-queens”, “What is N ?”
  - “Solve for N=4?”
  - “Why is this a solution? ”
- Handle chitchat // Support at least 5, extensible from a file
  - “Hi” => **“Hello”**
  - …
- ***Store session details in a file***

Illustrative Project

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Discussion: Illustration

1. Create a private Github repository called “CSCE58x-Fall2023-<studentname>-Repo”. Share with Instructor (biplav-s) and TA (kausik-l)
2. Create Google folder called “CSCE58x-Fall2023-<studentname>-SharedInfo”. Share with Instructor (prof.biplav@gmail.com) and TA (lakkarajukausik90@gmail.com)
3. Create a Google doc in your Google repo called “Project Plan” and have the following by next class (Sep 5, 2023)

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Illustration: N-Queens

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
    - Method 1: Random solution
    - Method 2: Search – BFS
    - Method 3: Search - …
  - Human interaction: Develop a basic chatbot (no AI) as outlined
  - Deliverable
    - Code structure in Github
      - ./data
      - ./code
      - ./docs
      - ./test
    - Presentation: Make sprint presentation on Oct 12, 2023

Reference: Project Rubric - NEW

- **Project report** – 60%
  - Project description: problem, related work, approach, evaluation – 40%
  - Working system demo/ video – 10%
    - Well organized Github with code (./data, ./code, ./docs, ./test) – 10%
- **Project presentation** – 40%
  - Evaluation by peers, instructor and TA
- **Bonus**
  - Instructor discretion – 10%
- **Penalty**
  - Lack of timeliness as per announced policy (right) - up to 30%

Milestones and **Penalties**

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
  - **Penalty: presentation not ready by Oct 10, 2023 [-10%]**
- Nov 30, 2023
  - Project report due
  - **Project report not ready by date [-10%]**
- Dec 5 / 7, 2023
  - In-class presentation
  - **Project presentations not ready by Dec 4, 2023 [-10%]**

Evaluation of Presentation

1. An online form will be available during presentation
2. During a presentation, three students will be assigned to review along with instructor and TA
3. They will enter following survey questions:
   1. Their name
   2. Presentation number
   3. How useful is the system – will you use it? [1-5 scale]
   4. How well have you understood the project from the presentation? [1-5 scale]
4. Top and bottom scores will be removed. Average of remaining three will be used for final presentation marks

Lecture 23 & 24: Summary

- We talked about
  - Making Decisions
  - Simple Decisions
  - Complex Decisions

Concluding Section

Quiz 4

- November 14-21, 2023
  - Data Science
  - NN/ Keras with non-text media

About Next Lecture – Lecture 25, 26

Lecture 25-26: Sequential Decisions

- Planning
- Reinforcement Learning