CS2109s - Tutorial 1

Eric Han (TG03, TG04)

Aug 30, 2023

Introduction to AI and Machine Learning

Your Tutor Eric Han, PhD pending Defence

- [Pioneer JC 2009-2010] Took 'A' levels and fell in love with Computing
  - H2 Computing, Interested in research and in AI.
- [B.Com. NUS 2013-2018] Not so long ago I was in your seat
  - A*STAR Scholarship, Turing Programme
  - [University of Southern California, 2016] Student Exchange
- [PhD. NUS 2018-2023] Submitted my thesis, pending Defence
  - My research is in AI/Machine Learning regarding scaling and robustness.
  - Some of the courses I taught: CS3217(1), CS3243(2), CS3203(5), CS2030(1)
  - Teaching this course is coming full circle for me, to teach the next generation.

You are welcome to check my profile & research: https://eric-han.com.

Expectations / Commitment

Expectations of you

1. Fill seats from the front.
2. Good students are always prepared:
   1. Attempt your Tutorial
   2. Review lecture content
   3. Be on time
3. Refrain from taking pictures of the slides.
   1. Learn to take good notes.
   2. Slides/notes will be distributed, but after lesson: https://www.eric-han.com/teaching

Commitment from me

1. Be avaliable for your learning as much as possible.
2. Strive to make the lessons interesting and fun.
3. Pass on a good foundation in AI/ML (not just the `A+`).

Any comments or suggestions for the lessons welcome!

Administrative

- Plagiarism - Passing work or ideas as your own w/o full ack/consent.
  - Problem Sets are **individual** work.
  - Tutorial are **individual** work.
  - About ChatGPT.
- In case if you cannot make it for tutorial (for any valid reason); makeup:
  - Attend my other tutorial interchangeably (I teach both, but don't abuse)
  - Attend other tutors TG (Inform me; Let me know which)
- If you still cannot make it (Send me an email with valid reason with proof)
- Consultations are avaliable in 1hr slots (By Appt only)
  - Wednesday 1-6pm

- Keep slides/notes within this class

AMA - chat about AI/ML, R&D, sch etc:

- Telegram `@Eric_Vader`, Email `eric_han@nus.edu.sg`

Transparent Grading

Tutorial EXP Guidelines (Adapted from Module Policy)

EXP	Category	Comments
400	Valid MC	Valid MC must be submitted.
0	Absent	Absent w/o valid excuse
250	Silent	Attended but tutorial not attempted.
300	Silent, Attempted Tutorial	Incomplete Tutorial
350	Completed Tutorial	Completed Tutorial
400	Active	Completed Tutorial w Active Discussion
450	Active	Completed Tutorial w Active Participation
500	Exceptional	Completed Tutorial w Active Participation w Bonus

My goal is to be as objective as possible, and to help u get EXP.

Transparent Grading

Tutorial EXP Rubrics

EXP	Item
+250	Attendance
+50	Attempted Tutorial (Not all Qns)
+50	Completed Tutorial (All Qns - attempt is good enough)
+50	Active Discussion (Contribute to Group/Buddy Discussion)
+50	Active Participation (Contribute to Class Discussion)
+50	Bonus (Completed Bonus Qn, Awarded by Eric)

Buddy System is from past feedback Your buddy will give you EXP

Transparent Grading

Problem Set EXP Notes

- ChatGPT: The use of LLMs are *not encouraged* but you must declare it:
  - OK as a search engine: What is np.reduce?
  - Not OK: Code a function that (1) Do not use numpy…
  - Not OK: Help me debug…
- You can consult your buddy for help:
  - OK: Hey do u know what the qn means by dont use iterative?
  - Not OK: Yo bro/sis can send me your code?
- You can ask on our Tutorial chat
  - OK: Hey anyone got problem with PS1.1?
  - Not OK: Can someone help me debug…
- You can PM me
  - OK: Hey Eric, I am having problems with np.reduce, what…
  - Not OK: Hey, this is my code…

Buddy System

- Everyone will be paired with a buddy (in pairs)

  - He/She will be your buddy for this class/sem
  - If there are odd numbers then we will have one group of 3
- Buddy will be responsible for taking your attendance/rating
  - I will penalise collusion.
  - Check on your buddy
  - Help your buddy
- Learning is social and I hope that we are able to build friendships in this class.
- The buddy must declare cheating and plagrism when there is suspect of that
  - Please email me at `eric_han@nus.edu.sg` with proof
  - I place a heavy penalty on Plagiarism/Cheating
  - I will recommend max penalty to the Prof

Annoucements

Important admin:

1. Join our Telegram Group: https://t.me/+Uvk5h0tAgjRmYjNl
2. Welcome Survey (also attendance): https://forms.gle/x5hxtfpRhmib6Att8
3. Take Attendance for your buddy: will be flashed at the end.

Figure 1: Our Telegram Group

Figure 2: Survey

Question 1

Determine the environment properties of a Sudoku game.

Recap

- **Fully / Partially Observable**: Is the complete state of the environment accessible to the agent's sensors?
- **Single / Multi-Agent**: Are there more than one actor in the environment? (competitive vs cooperative)
- **Deterministic / Stochastic**: Is the next state determined by the current state and action by the agent?
- **Episodic / Sequential**: Is the next episode dependent on the action taken previously?
- **Static / Dynamic**: Can the environment change while the agent is deliberating?
  - **Semi-Dynamic**: Env doesn't change but agent's metric does.
- **Discrete / Continuous**: Is the state of the environment discretized or varying continuously?
- [**Known / Unknown**] Are the rules of the game known to the agent?

Easiest: Fully, Single, Deterministic, Episodic, Static

Answer

Environment Characteristic	Sudoku Puzzle Generation
Fully / Partially Observable	Fully Observable
Single / Multi-Agent	Single agent
Deterministic / Stochastic	Deterministic
Episodic / Sequential	Episodic / Sequential
Static / Dynamic	Static
Discrete / Continuous	Discrete

Explaination on Episodic / Sequential: Depending on the **environment** formulation, the next episode is dependent on the previous action.

Question 2

Determine the PEAS for the SIRI function of iPhones.

Recap

- Explain PEAS and why is it used?

. . .

PEAS:

- **Performance measure**: How good or bad?
- **Environment**: External context
- **Actuators**: Output/Actions
- **Sensors**: Input

Answer

PEAS	Description
Performance measure	Correctly processes users' voice into text based on different language with possible accents. Make correct action with acceptable reaction speed.
Environment	iOS device with a functional audio input system. Internet connection and supported language voice.
Actuators	Speaker or/ and visual output to show answers to user.
Sensors	Speech Listener/ Microphone. Touchscreen Interactions.

Question 3 [G]

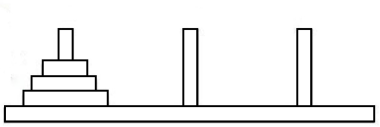

Figure 3: Tower of Hanoi with $n = 3$ disks

Tower of Hanoi is a game with 3 (L,M,R) pegs and a set of n disks with ascending (different) sizes. The aim is to move all disks from the L to the R peg with the minimum moves, considering:

- Move one disk at a time, from one peg to the other.
- Larger disks cannot be placed ontop of smaller disks.

Discuss its environment formulation.

Recap: What are the 5 parts of the environment formulation?

Answer

Sample (no right answer) Environment Formulation:

- **State Space**:
  - Each disk has an index from 1 to n corresponding to its size,
  - 3 tuples for the pegs $[L, M, R]$
  - **Invariant**: each tuple must be in ascending order
- **Initial State**: $[(1, 2, \cdots, n), (), ()]$
- **Final State**: $[(), (), (1, 2, \cdots, n)]$
- **Action**: 6 actions - L>M, L>R, M>L, M>R, R>L, R>M
- **Transition Model**: (Incomplete; Need to describe the rest of the 5)
  - Action a_1: L peg to M peg
    - $T\Big([(l_1, ...), (m_1, ...), (...)], a_1\Big) \rightarrow [(...), (l_1, m_1, ...), (...)], l_1 < m_1$
    - $T\Big([(l_1, ...), (), (...)\Big), a_1] \rightarrow [(...), (l_1), (...)]$

Some representations are better than others - Compute complexity, No of states, etc...

Quality of Representation

Surjection must be formed between the set of all possible valid state representations:

- For any possible real world config, there is a corresponding state representation.

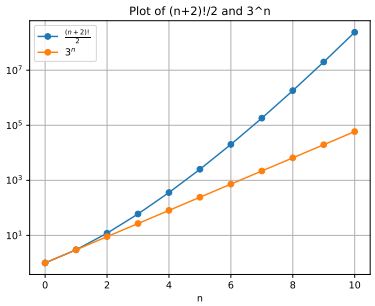

Figure 4: With/Without invariant

Without invariant: $\frac{(n+2)!}{2}$: Shuffle then split the items into partitions

With invariant: 3^n: Consider the state tree of placing largest to smallest disks

Question 4 [G]

Let S be the initial state and G be the goal state.

a. Trace uniform-cost search using *tree*
b. Trace uniform-cost search using *graph*
c. Why doesn't the algorithm halt and return the search result since the goal has been found?
d. What's the difference between uniform-cost search (using graph search) and Dijkstra's algorithm?

Recap

- What is the difference between Tree Search and Graph Search?
- AIMA Chapter 3, on graph and tree-like search.

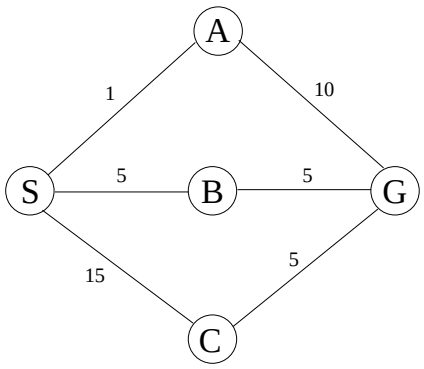

Figure 5: Graph with cost.

Answer 4a (check full trace in Notes, Slides can't fit)

```
S(0)
A(1) B(5) C(15)
S(2) B(5) G(11) C(15)
A(3) B(5) B(7) G(11) C(15) C(17)
S(4) B(5) B(7) G(11) G(13) C(15) C(17)
A(5) B(5) B(7) B(9) G(11) G(13) C(15) C(17) C(19)
B(5) S(6) B(7) B(9) G(11) G(13) C(15) G(15) C(17) C(19)
S(6) B(7) B(9) G(10) S(10) G(11) G(13) C(15) G(15) C(17) C(19)
A(7) B(7) B(9) G(10) S(10) B(11) G(11) G(13) C(15) G(15) C(17) C(19) C(21)
B(7) S(8) B(9) G(10) S(10) B(11) G(11) G(13) C(15) G(15) C(17) G(17) C(19) C(21)
S(8) B(9) G(10) S(10) B(11) G(11) G(12) S(12) G(13) C(15) G(15) C(17) G(17) C(19) C(21)
A(9) B(9) G(10) S(10) B(11) G(11) G(12) S(12) B(13) G(13) C(15) G(15) C(17) G(17) C(19)
    C(21) C(23)
B(9) G(10) S(10) S(10) B(11) G(11) G(12) S(12) B(13) G(13) C(15) G(15) C(17) G(17) C(19)
    G(19) C(21) C(23)
G(10) S(10) S(10) B(11) G(11) G(12) S(12) B(13) G(13) G(14) S(14) C(15) G(15) C(17) G(17)
    C(19) G(19) C(21) C(23)
```

Answer 4b

frontier	explored
S(0-)	
A(1-S) B(5-S) C(15-S)	S
B(5-S) G(11-SA) C(15-S)	S A
G(10-SB) C(15-S)	S A B

Answer 4c

- Goal check is done on `frontier.pop()`, this means that only when the goal is the cheapest item then the algorithm terminates.
- If the algorithm terminates on discovery, we may miss the node with smaller cost.
- The algorithm works by always selecting the node with the least path cost for expansion and it ensures that the first goal node selected for expansion is an optimal (i.e., shortest path) solution.

Answer 4d

The algorithms are the same, with minor differences, ie.:

- Dijkstra finds the shortest path to every node from a single source
- UCS concerns itself with the shortest path to the goal states.

Question 5

Describe a state space in which iterative deepening search performs much worse than depth-first search.

Bonus Question

To help you further your understanding, not compulsory; Work for Snack/EXP!

Tasks

1. Fork the repository https://github.com/eric-vader/CS2109s-2324s1-bonus
2. We will be first solving Question 4 using code, `uniform_cost_search(graph, inital_node, goal_test, is_tree, is_update)` that returns the path found:
    1. Able to solve 4a via `is_tree=True,is_update=False`
    2. Able to solve 4b via `is_tree=False,is_update=True`
3. Some code have been implemented for you, including the priority queue.
4. You should print the frontier and explored at the beginning of the loop.

To claim your snack & EXP, show me your forked repository and your code's output.

Useful Links

- AIMA Python Implementation - https://github.com/aimacode/aima-python.

Buddy Attendance Taking

Important admin:

1. Join our Telegram Group: https://t.me/+Uvk5h0tAgjRmYjNl
2. Welcome Survey: https://forms.gle/x5hxtfpRhmib6Att8
3. Take Attendance for your buddy: https://forms.gle/Ckkq639TNwWEx3NT6

Figure 6: Telegram Group

Figure 7: Survey

Figure 8: Buddy Attendance