Omsairam

CS 156-02

Prof. L Wesley

Book ch: 2 and 3 and 5.1, 5.2, 5.3

Plus slides

Plus homeworks

Plus in class exercise

Idea:

Cant we use IR with AI to get information related to required topic ? like combine chatGPT (AI) with google search (IR)

Syllabus:

1. What goes under AI system to make it smart
2. Python review and history and state of the art in AI. Intelligent Agents
3. Solving problems by search
4. Knowledge, reasining, and planning
5. Knowledge representation
6. Uncertain knowledge and reasoning
7. Machine learning
8. Applications: how to integrate and apply the above trchniques to real-world applications

Wednesday due assignments

Definition:

Search , representation and reasoning schemes-> important

ML not intelligent on own , only if above 3 in ML then only AI

Textbook-> by norvik

Artificial intelligence A Modern Approach fourth edition by stuart russel and peter Norvig,

Python 3.9 or above : pycharm

Group of 3 or 4 project

No final but quizzes

Lec-1

Intelligent agents/system:

1. Representation
2. Search
3. Reasoning

Search ->

# of planets in the universe ~ 10^20

# of atoms in universe ~ 10^50

# of possible chess moves ~10^110 (legal moves) ~10^123 (illegal moves)

Intelligent agent can be able to search solutions that are correct or workable

Reasoning ->

Forming an analogy between a current problem to a situation to a situation you probably have had, predict, reasoning by analogy

Reasoning by inference -> logical deduction

Ex; a vehicle, washing machine doing it how …

Representation ->

Sequence of transporting example of animals across the bank.

Can use representation to do that: Like a 2D matrix,

To capture State of env or of possible solution over time

Intelligent Agent doesn’t need ML as such for everything

Omsairam lec 2

* What is the assignment about ?
* What is the format to submit the assignment
* File and everything
* What about the feedback ? same file to be uploaded with answer ?
* Where is the Programming assignment guidelines in canvas?
* Where is the extra-credit section on assignment submission page ?

Files themselves directly upload rather than the link to the file.

Min-max tree search problem -> practice with classes and manipulation of classes

Assignment guidelines:

Add these (check cs156\_pr1.py)

1. # coding: utf-8 # <- This is an encoding declaration REQUIRED
2. Name of module
3. Description of module
4. Modification history and info (optional)
5. Naming convention follow standards and in description show what is what
6. Mention the execution and imported module at the end

If execute then module then \_\_name\_\_ == ‘\_\_main\_\_’:

To install pkgs : pip3 install <pkg> in terminal

Pass in python: for forward reference,

Class B:

Pass

Class A:

Var = B()

Class B:

Var= A()

So we don’t have B defined in before so when it comes to A it says it doesn’t know B to avoid that we pass B and then later define B so, the latest B will be used in python

Init function in python: constructor in java

Class return string object to be printed

Class A:

Def \_\_init\_\_(self,…):

….

Def \_\_str\_\_(self,..):

…

Var = A()

Print(var.\_str\_\_())

We must use self in functions as one of the arguments while defining the function.

While calling the function no need to use the object in function parameter

Min-max tree:

Start state = root

Branches are possible moves

Next node is the new state after the move made

Alternate level is urs and ur opponents

Ex: Checker or chess games

Access how good to some depth with some sequence of moves

Doubt: how can u init with none as name cannot be none . will show error if MM\_node() is called right ?

1. When setting left and right child should parent get updated too?
2. When setting parent should left or right child of parent node be updated too?
3. When should the error msgs be displayed ? if true or false or always? If in true displayed will the updation happen?
4. If new parent not in tree should the new node be inserted ?
5. To update the new parent of the new node, how to update its left or child child meaning , should the new node be updated in left\_child or right\_child ? and similarly we need to update the left or right child’s parent to none as its position is updated with the new node
6. What things needed in the file as a must have like any file name or functions or anything else ?

omsairam

Lec-03

Lisp, prolog designed to build AI systems

These are functional programming language

Intelligent Agent:

Agent has special knowledge and skills ,

Agent function maps historic data to actions

Agent program runs on physical architecture to produce f

Agent = architecture + program

To design an agent use PAGE process -> perception, action, goals, environment

For example :

In chess board can see whole environment

In car, agent cannot see whole environment

Or a dynamic env -> real life driving , video game driving

Is percepts same like a representation scheme that shows the state of the agent?

Current, and future too states

Internet personal financial planning agent:

Percepts (PERCETS OF ENV: eg: env is tough competition then u use the percepts of the competitors, ): current market performance, current portfolio / financial situation of customer, interested stocks of customer, political situation, customer information: ( age of candidate, risk taking value of customer, company employee or not, family, income)

Actions: pursue different investment opportunity, create portfolio of customers, predict cashflow/budget, advise on tax returns

Goals: minimize risk, maximize profit, not invest illegal stocks, specific customer configurable needs/goals, don’t go broke

Environment: network or web or mobile app, financial situation (interest rate, supply, demand) or market rate, financial constraints and laws, competitors

Omsairam

Lec-04

Textbook -> agent on toy world games, vaccum-Cleaners .. referred to those slides.

Why is search important for an agent?

Search states and eliminate subsets and still get to a goal within limited time and resources. To get possible list of solutions to choose from.

Eg: x number of trading options , need to prune out the subsets and get list of possible solutions

//actions taking also might need to check if at this state the action is appropriate

Eg: chess

Percept -> moves

Formulate prob: what can be dobe and what are the states that can result into

Eg: financial

State: what if I use this opportunity, ..

Deterministic -> sure of future state

Non-deterministic -> don’t know what will be future state

Non-observable -> don’t know any state

Unknown -> periodically consider possible states, dynamically

Vacuum example: Perception, updating states, formulating action, to get goal state.

Why cost ? -> get optimal sol.

Single state prob.

S(a) = {a->b, b} ie (action, state)

Sol: seq of actions . set of states to reach goal state

State space = all possible states , given no start state. Everything possible

See vacuum ex in slides

Tree -> no loops, graphs have loops

Connecting cities can be a tree search algo

Frontier -> expanded nodes list.

Omsairam

Lec-05

Ex: Performance measure for financial planning agent :

Ans: if any profit obtained

Need to understand for environment which are changing, what is the rational behavior in certain environments.

Performance metrics : how you are going to reach the goal

Goal: end goals

Time slices in environment : if dynamic env, have small size period , else can be bigger time period. To determine the next action to take

Semi episodic : beginning to turn right,

At one episodic period : nothing was there except speed

At 2nd episodic period: someone came on the road

So sometime episodic period depends partially on previous episodic period

Investment ex: semi episodic

Model based agent vs reflex agent :

More information like state, what action what reaction as extra info

Utility v goal agent

Goal -> is a state u want to be.

Utility -> how valuable the goal is.

omsairam

Lec-06

Search algos:

Table search (percept, action) : reflex agent

Depth limited search: pick a level u want to go to. Even if there are more levels, u can reverse from the level you chose. Adv : somewhat time saving if solution state is somewhere within that chosen level

Iterative deepening search: if don’t find within a certain level, then can increase the chosen level and search within that new space.

B = branching factor

D = depth

BFS -> can have a clever algo to remove the space related problem. Is complete

DFS-> incomplete if loop or infinite space depth

Graph vs tree search -> graph need to see if node already visited or not, to avoid loops

A\* search->heuristic, acceptable sol in reasonable amount of time and space. Limits going to expensive routes.

Example: City travel. A -> n -> d

A -> n travelled already , total cost = g(n)

N -> d to travel, total estimated cost = h(n)

How to get h(n) or what’s the heuristic? estimated distance can be Euclidian distance between current node to dest node (provided you have the map of node’s location)

Do that for frontier nodes and chose the least cost f(n).

A to d, estimated cost = f(n) = g(n) + h(n)

If h(n) is admissible, then A\* using tree-search is optimal

Omsairam

Take min f(n) from all frontier nodes

**Optimal** means if path exists, it will find the answer

h is aways less than actual cost

Q. as h(n) is an estimate cost, how to make sure which path to take if there is a tie between f(n)s ?

Q. Is it possible to take h(n) for any solution? How to guarantee that its always the optimal sol?

Ans : Relaxation of problem. Eg: car always on the road is a constraint. So if relax this constraint then a direct distance h(n) will be optimal for sure.

Suboptimal : g2 not as good as G

H\*(n) from n to G2

**Graph** search -> h(n) is consistent, then A\* is optimal => f(n’) >= f(n)

**Tree** search -> h(n) is admissible, then A\* is optimal

How know graph or tree ? depends on the structure of the state space of the problem

Game -> tree

Robot moving -> graph

Manhattan distance -> least distance from cur to goal position

Dominant h2>h1 and consistent prove should .

Q. Prove that is a heuristic is consistent, then it must be admissible

Use proof by induction

Construct an admissible heuristic that is not consistent

1. Prove that consistent heuristic is admissible

If F(n) is consistent

F’(n) >= f(n) means it increases

For being admissible

H(n) <= h\*(n)

F(n’) > = f(n)

H(n) + g(n’) > = h(n) + g(n)

if n is start node and we have 2 goals G and G2 then

g(n) = g(n’)

from above

h(n’) + g(n’) > = h(n) + g(n)

h(n’) + g(n) > = h(n) + g(n)

h(n’) > = h(n)

which proves that heuristic h(n’) is more than the h(n), hence proving consistent heuristic is admissible too.

b) Now to prove that admissible heuristic is not consistent

h(n) < = h(n’) where n’ is true cost to reach goal. This is an admissible definition

f(n) = g(n) + h(n)

f(n’) = g(n’) + h(n’)

we don’t have a certainty that f(n’) > = f(n)

g(n) >= g(n’) ? not sure .

so we cannot say that admissible heuristic is a consistent

omsairam

quiz : open book? Yes

game playing : what will be the next move is the process to expand, called evaluation function

using a heuristic approach : we get to fnal state best search, here not possible

why slide 13 of min max-:

2 versions of min-max: a) picking the route to minimize your risk of losing, backed up value will determine the route so use the same value as root 2-> 2 -> 2. = root value, you want to keep the value

b) Take the path -> that will min your opponent and maximize you A31

root node is your turn move

states can vary per the opponent’s decision. If brand new state, then use that as root node and make a tree on top of that.

2 opponents -> equally good, same evaluation function, what will the result be? Neutral state. Tie.

What would make one of the players a winner if same intellect (exclude human error)? Not making optimal decision: how ? a) better evaluation function. B) Depth of the tree, better decision making

omsairam

Project

1. Pick agent
2. PEAS process
3. Game playing -> based on opponent skills and level of difficulty.
4. Build a map internally in the same env, like save the map like in learning
5. Web-crawling agent for stock investing --- may be? We have data from various website, need to integrate with middle tier and show in tableau or some graph structure.

1. Can dive deeper, apply better evaluation function at the frontier <- what can be done in the extra resources we get after pruning.
2. Alpha-beta pruning -> Use depth first. Go to depth level, apply evaluation function, go back to its parent , check if parent can be evaluated, if not go to next child, apply evaluation function, g back to parent, is it ok to evaluate that now , if yes, evaluate parent , go to grand parent . repeat. If an evolution function is such that it has reached its limit, so we have enough information to get the result to parent, then we prune the other child. If can get the percentage reduction in the pruning then can use that to other things.
3. If at maxnode and u know u don’t need to do any calculation below it, alpha cutoff
4. Same as above for minnode is called beta cutoff

Omsairam

Wumpus world: game playing world

Episodic -> partial definition like may only partially depend on what happened previously. Strict definition : current doesn’t depend on previous . for wumpus world its not episodic

Propositional logic is either true or false

A-> B = not A or B

A<-> B = (not A or B) and (not B or A)

KB or knowledge base= set of propositional sentences

Entails means KB has something and that something is true where KB is true

Sementics Meaning of a propositional sentence comes from whether its true or false

Ways to carry out reasoning:

No new info: Entailment: for some propositional sentence, does current KB entails that’s sentence

New info: Inference , is asserting a new sentence from the statement already known

1. Model checking : create all models that have a sentence is true in KB. Create all models that needs to assert a sentence a1, check if KB entails a1
2. Inference : derive using inference procedures

Inference rule is sound if inference rule actually asserts it, what follows is that KB entails that

Inference rue cannot derive a false inference if its sound

Completeness : if KB entails a, then all true sentences can be derived

Here some false statements might be derived if complete

Desirable if sound and complete

1. Truth table creation also finds all states that are true

A^(A=>B)

=A^-AvB

=B

Omsairam

1. Ask about the assignment, the sol. Wrong data and what to submit for programming , format , the details or only the updated code? And the entire flder ? pseudo codes needed what file name ?

Omsairam

1. Different inference rules

Equivalence

Modus pones

And elimination

1. Resolution : to prove or disprove some propositional statement

Convert to CNF

Put KB ^ -a

if get empty after resolution, then proved that a belongs to KB

KB = (B 1,1 ⇔ (P1,2∨ P2,1)) ∧¬ B1,1 α = ¬P1,2

From previous we have :

Conversion to CNF  
B1,1 ⇔ (P1,2 ∨ P2,1)  
1. Eliminate ⇔, replacing α ⇔ β with (α ⇒ β)∧(β ⇒ α).  
(B 1,1 ⇒ (P1,2 ∨ P2,1)) ∧ ((P1,2 ∨ P2,1) ⇒B 1,1)  
2. Eliminate ⇒, replacing α ⇒ β with ¬α∨ β.  
(¬B 1,1 ∨ P1,2 ∨ P2,1) ∧ (¬(P1,2 ∨ P2,1) ∨B 1,1)  
3. Move ¬ inwards using de Morgan's rules and double-negation:  
(¬B 1,1 ∨ P1,2 ∨ P2,1) ∧ ((¬P1,2 ∨ ¬P2,1) ∨B 1,1)  
4. Apply distributivity law (∧ over ∨) and flatten:  
(¬B 1,1 ∨ P1,2 ∨ P2,1) ∧ (¬P1,2 ∨ B 1,1) ∧ (¬P2,1 ∨ B 1,1)

(¬B 1,1 ∨ P1,2 ∨ P2,1) ∧ (¬P1,2 ∨ B 1,1) ∧ (¬P2,1 ∨ B 1,1) ∧¬ B1,1^P1,2

Can cancel only 1 at a time

Use pairs to cancel

All clauses found, when we have one clause from above and do nota with the single literal

P1,2 ^ ¬ P1,2 = empty then we proved that ¬ P1,2 is true

1. Can do conjunction for all clauses in KB. Horn clause may or may not have disjunctions within each clause
2. (A,a=>b)/b … given a , and we can imply b is true
3. Forward chaining if antecedents trie then consequents true
4. Sound whatever it asserts its true

* only required derived

1. Completeness – all true statements can be derived

* Issue KB size increases

Create horn clause and forward chaining to assert all statements that can be derived from it

R1 : ~P11

R2 : B11 ⬄ (P12 v P21)

(B11 => (P12 v P21)) ^ ((P12 v P21) => B11)

= (~B11 v (P12 v P21)) ^ (~(P12 v P21) v B11)

= (~B11 v P12 v P21) ^ ((~P12 ^ ~P21) v B11)

= (~B11 v P12 v P21) ^ (~P12 v B11) ^ (~P21 v B11)

R3: B21 ⬄ (P11 v P22 V P31)

(B21 => (P11 v P22 V P31)) ^ ((P11 v P22 V P31)=>B21)

= (~B21 v (P11 v P22 V P31)) ^ (~(P11 v P22 V P31)v B21)

= (~B21 v P11 v P22 v P31) ^ ((~P11 ^ ~P22 ^ ~P31)v B21)

= (~B21 v P11 v P22 v P31) ^ (~P11 v B21) ^ (~P22 v B21) ^ (~P31 v B21)

R4: ~B11

R5: B21

So horn clauses so formed are :

1. B11 => (P12 v P21)
2. P12 v P21=> B11
3. B21 => (P11 v P22 V P31)
4. (P11 v P22 V P31)=>B21

Then using forward chaining we have

1. Using B11 => (P12 v P21) and ~B11 we have ~(P12 v P21) which implies ~P12 and ~P21 means no pit in P12 and no pit in 21 adding extra 2 statements to KB :

R6: ~P12

R7: ~P21

1. Using B21 => (P11 v P22 V P31) and B21 we have P11 v P22 v P31 . we have ~P11 in R1, using that we have

P22 v P31 .. hence adding R8 to KB

R8: P22 v P31

Omsairam

Rule based representation: not in book

Rules : R1:if .. then…

R2 : if… then..

If percept makes the antecedent true then consequent action is performed.

If conditions are like cond1 ^ cond2 v cond3 …

Disadv: have to bring down to propositional logical statements way.

Lots of computation

Conflicting rules need another set of rules that resolves the conflicts

Omsairam

Frame: City

Slot: (population IF-NEEDED<…..> ) then automatically calls a procedure to determine the value to save in that attribute

Look for keywords in slots and execute. Function and instantiate another frame to do some task.

Slot: (Cur\_Temperature IF-ADDED<…>) if value is saved in other or this slot, as soon as something is added this procedure is fired. Not necessarily to store but also do something to this slot value.

Frame same as class. Is-A for class type

Instantiate frame to create objects of the frame/class instance-of shows what class its from

Slot = same as fields of classes

If-added is inherited procedure. When some slot/field is added then do something

If-needed is inherited procedure. – same like procedure but does when needed. To fill some values.

Trip17,lodgingStay is an object of class Trip

Rule based system can sometimes be called expert system ES. As experts defines the rules.

Omsairam

Traditional prob. Reasoning

//need to know elements prob to derive others

M: [0,1]

Theta

Sch that

Sum (m(p))=1

Where p belongs to theta

H= a set

Of mutually disjoint exclusive and exhaustive propositions.

Evidential reasoning -> extension of traditional reasoning.

// not forced to apply prob. To a=some elements , can get intervals of prob. Rather than a precision number.

M=2^H -> [0,1] // set of all subsets of theta

Such that

Sum(m(p))=1

Where p belongs to theta

Its possible that some elements present in power set that have no probability number.

Frame based is like oops

Can be converted to first order

Can be converted to semantic network, where objects are nodes and relations are edges. Relations with more than 2 nodes in a relation can be converted to nodes called reification.

[bird]------is a ----- animal

| /

|is /is-a

[alive] [mammals]

\

\ is a member of

[4-legged animal]

Omsairam

Ans = 2.0 \* 2.3 – 3 = 1.6 (input to transfer fucntion)

Neuron utput

Hard limit =1 as > 0

Linear output = 1.6

Sigmoid function : 1/(1+e^(-1.6))

Omsairam

A=4.0343

B=0.0042

X=1200

Using formula A-bX

Ln(p/(1-p))=4.0343 - 0.0042x1200= -1.0057

p/(1-p) = e^ (-1.0057) = 0.3658

p= 0.3658 /(1+ 0.3658) = 0.2678

omsairam

1. SVM : SVC and SVR

SVC - support vector classifier: is a binary classifier but can be used to classify more classes, more classes used to classify then accuracy decreases.

Decision boundary : Boundary should maximize the distance between the 2 sets of data.

Margin : how string is the analysis

LSVC – linear support vector classifier

(Datapoint, class)

D= #features

N = #samples

X1 = x11w11 + x12w12 + … x1dw1d +b = y1

X2 = x21w21 + x22w22 + … x2dw2d +b = y2

Xn = xn1wn1 + xn2wn2 + … xndwnd +b = yn

Solve for w and b such that decision boundary holds and from sample or test dataset, we get either -1 or 1.

The final eq will be of this format: so that we only use the data set values and get the final decision {-1, 1} for y

Y = a1x1+a2x2 + … adxd+ b, all the ai are the LaGrange multiplier/ coefficients , all that are 0 then remove those from the eq. so we have only

Y= a5x5 + a11x11 + a16x16 + b, so only few feature points needs to calc y now. Benefit to use LaGrange multiplier. Only the data points on the margin will be considered.

Anything on decision boundary should be wx +b =0, if anrgin then wx+b =-1 , if plus boundary the wx+b=1