General Instructions for Labs

Students enrolled in the course are expected to work on programming problems assigned for laboratories in groups. Groups once formed would remain fixed for the entire duration of the course. Evaluation of lab assignments should primarily be taken as a two-way feedback, both for the students and the instructor.

A group can have a maximum of 4 students from the same section.

Attending laboratory sessions is a must and absence during laboratory sessions without prior intimation to the course instructor (only genuine cases) will attract 0 credit for that particular session.

Feel free to contact me or the TAs in case you have any query or confusion regarding submissions.

Course Instructor:

Pratik Shah
pratik@iiitvadodara.ac.in

Course TA(s):

TBA

Evaluation Policy

Component 1: 35%

A. Attendance and participation - 15%
B. Peer Assessment - 20%
    a. Four student groups will be identified per week to solve, discuss and explain the respective laboratory problem for the coming week to the respective sections.
    b. Each group will have 45 minutes of pre-assigned slots during the laboratory hours. Each group will explain their solution with the codes within the first 30 minutes followed by 15 minutes of discussion/ question answer on the solution/ codes with the section.
    c. The peers/ students present during the lab will assess the performance of each group by assigning the score on the scale of 0-5 for the following
        i. Quality of explanation and clarity of exposition and discussion
        ii. Quality of the code base
        iii. Resolution of queries/ questions raised by peers during the session.

Component 2: 50%

Students are required to work in groups on laboratory problems. Each group is expected to submit a total of two reports containing the discussion on the solution to five-five problems each. One such report will be due just before the mid-semester examination and the other just before the end-semester examination.

Report/ Term Paper Template : IEEE Transactions on Systems Man and Cybernetics: Systems in LaTeX format. https://drive.google.com/file/d/1sdEyE-9DxDNFbuR_n3xJ0u0Og6aw9zKU/view?usp=sharing

It is expected that you will write the report as if you are writing a tutorial to explain the problems and the corresponding solutions.

Component 3: 15%

A project/ lab viva voce will be conducted immediately before/ after the mid/ end-semester examination.

Moral Code of Conduct

Zero tolerance to plagiarism. If the report submitted by any group is found to have copied extracts from any of the other submitted reports the group members will receive no credit for the entire report. In a report if some material is found to have been reproduced from some source without proper citation or reference, the report will attract zero credits.

Week 0 :

Lab Assignment - 0

Learning Objective: To be able to model a given problem in terms of state space search problem and solve the same using BFS/ DFS

Reference:

[1] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition)

Chapter 1, 2, 3

Problem Statement:

A. In the rabbit leap problem, three east-bound rabbits stand in a line blocked by three west-bound rabbits. They are crossing a stream with stones placed in the east west direction in a line. There is one empty stone between them. The rabbits can only move forward one or two steps. They can jump over one rabbit if the need arises, but not more than that. Are they smart enough to cross each other without having to step into the water?
B. The missionaries and cannibals problem is usually stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place. This problem is famous in AI because it was the subject of the first paper that approached problem-formulation from an analytical viewpoint.

For the above two problems,

1. Model the problem as a state space search problem. How large is the search space?
2. Solve the problem using BFS. The optimal solution is the one with the fewest number of steps. Is the solution that you have acquired an optimal one? The program should print out the solution by listing a sequence of steps needed to reach the goal state from the initial state.
3. Solve the problem using DFS. The program should print out the solution by listing a sequence of steps needed to reach the goal state from the initial state
4. Compare solutions found from BFS and DFS. Comment on solutions. Also compare the time and space complexities of both.

Week 1 :

Lab Assignment - 1

Learning Objective: To design a graph search agent and understand the use of a hash table, queue in state space search.

Reference:
[1] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition)
Chapter 2 and 3
[2] A first course in Artificial Intelligence, Deepak Khemani
Chapter 2

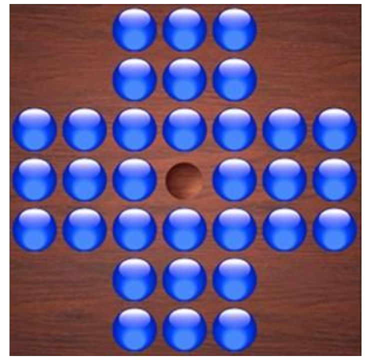
Problem Statement:

A. Write a pseudocode for a graph search agent. Represent the agent in the form of a flow chart. Clearly mention all the implementation details with reasons.
B. Write a collection of functions imitating the environment for Puzzle-8.
C. Describe what is Iterative Deepening Search.
D. Considering the cost associated with every move to be the same (uniform cost), write a function which can backtrack and produce the path taken to reach the goal state from the source/ initial state.
E. Generate Puzzle-8 instances with the goal state at depth “d”.
F. Prepare a table indicating the memory and time requirements to solve Puzzle-8 instances (depth “d”) using your graph search agent.

Week 2 :

Lab Assignment - 2

Learning Objective: To understand the use of Heuristic function for reducing the size of the search space. Explore non-classical search algorithms for large problems.



Reference:

[1] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition)
Chapter 3 and 4

[2] A first course in Artificial Intelligence, Deepak Khemani
Chapter 3, 4

Problem Statement:

A. Read about the game of marble solitaire. Figure shows the initial board configuration. The goal is to reach the board configuration where only one marble is left at the centre. To solve marble solitaire, (1) Implement priority queue based search considering path cost, (2) suggest two different heuristic functions with justification, (3) Implement best first search algorithm, (4) Implement A*, (5) Compare the results of various search algorithms.
B. Write a program to randomly generate k-SAT problems. The program must accept values for k, m the number of clauses in the formula, and n the number of variables. Each clause of length k must contain distinct variables or their negation. Instances generated by this algorithm belong to fixed clause length models of SAT and are known as uniform random k-SAT problems.
C. Write programs to solve a set of uniform random 3-SAT problems for different combinations of m and n, and compare their performance. Try the Hill-Climbing, Beam-Search with beam widths 3 and 4, Variable-Neighborhood-Descent with 3 neighborhood functions. Use two different heuristic functions and compare them with respect to *penetrance*.

Week 3 :
Lab Assignment 3

Learning Objective:
Non-deterministic Search | Simulated Annealing
For problems with large search spaces, randomized search becomes a meaningful option given partial/ full-information about the domain.

Reference:
[1] A first course in Artificial Intelligence, Deepak Khemani (Chapter 4)

Problem:
Travelling Salesman Problem (TSP) is a hard problem, and is simple to state. Given a graph in which the nodes are locations of cities, and edges are labelled with the cost of travelling between cities, find a cycle containing each city exactly once, such that the total cost of the tour is as low as possible.

For the state of Rajasthan, find out at least twenty important tourist locations. Suppose your relatives are about to visit you next week. Use Simulated Annealing to plan a cost effective tour of Rajasthan. It is reasonable to assume that the cost of travelling between two locations is proportional to the distance between them.

An interesting problem domain with TSP instances:
VLSI: http://www.math.uwaterloo.ca/tsp/vlsi/index.html#XQF131
(Attempt at least five problems from the above list and compare your results.)

Week 4 :

Week 1-3 : Challenge Problem

Learning Objective:
Non-deterministic Search | Simulated Annealing | Genetic Algorithm
For problems with large search spaces, randomized search becomes a meaningful option given partial/ full-information about the domain.

Reference:
[1] Simulated annealing from basics to applications, Daniel Delahaye, Supatcha Chaimatanan, Marcel Mongeau
[2] A first course in Artificial Intelligence, Deepak Khemani (Chapter 4)
[3] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition)
Chapter 3 and 4

Problem:

Ever played jigsaw puzzles in childhood? Time to make an agent learn to solve the puzzle.

Some rough, confusing code snippets are posted on the codes folder of the course along with the scrambled_lena.mat

Formulate the problem as a state space search problem. Try to solve the same using simulated annealing.

Have fun!

This is a bonus challenge! If you solve this, you are up for some extra credits.

Week 5 :
Lab Assignment 4

Learning Objective:
Game Playing Agent | Minimax | Alpha-Beta Pruning

Systematic adversarial search can lead to savings in terms of pruning of sub-trees resulting in lesser node evaluations

References:
[1] A first course in Artificial Intelligence, Deepak Khemani (Chapter 8)
[2] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition) (Chapter 5)

Problem:

1. What is the size of the game tree for Noughts and Crosses? Sketch the game tree.
2. Read about the game of Nim (a player left with no move losing the game). For the initial configuration of the game with three piles of objects as shown in Figure, show that regardless of the strategy of player-1, player-2 will always win. Try to explain the reason with the MINIMAX value backup argument on the game tree.

3. Implement MINIMAX and alpha-beta pruning agents. Report on number of evaluated nodes for Noughts and Crosses game tree.
4. Use recurrence to show that under perfect ordering of leaf nodes, the alpha-beta pruning time complexity is $O(b^{m/2})$, where b is the effective branching factor and m is the depth of the tree.

Week 6 :

Lab Assignment 5

Learning Objective:

Understand the graphical models for inference under uncertainty, build Bayesian Network in R, Learn the structure and CPTs from Data, naive Bayes classification with dependency between features.

Reference:

1. https://www.bnlearn.com/
2. http://gauss.inf.um.es/umur/xjurponencias/talleres/J3.pdf

Problem Statement:

A table containing grades earned by students in respective courses is made available to you in (codes folder) 2020_bn_nb_data.txt.

1. Consider grades earned in each of the courses as random variables and learn the dependencies between courses.
2. Using the data, learn the CPTs for each course node.
3. What grade will a student get in PH100 if he earns DD in EC100, CC in IT101 and CD in MA101.
4. The last column in the data file indicates whether a student qualifies for an internship program or not. From the given data, take 70 percent data for training and build a naive Bayes classifier (considering that the grades earned in different courses are independent of each other) which takes in the student's performance and returns the qualification status with a probability. Test your classifier on the remaining 30 percent data. Repeat this experiment for 20 random selection of training and testing data. Report results about the accuracy of your classifier.
5. Repeat 4, considering that the grades earned in different courses may be dependent.

Week 7 :

Lab Assignment 6

Learning Objective:

To implement Expectation Maximization routine for learning parameters of a Hidden Markov Model, to be able to use the EM framework for deriving algorithms for problems with hidden or partial information.

Reference:

1. A Revealing Introduction to Hidden Markov Models, Mark Stamp, 2018
2. What is the expectation maximization algorithm? Chuong B Do and Serafim Batzoglou, Nature Biotechnology, Vol 26, Num 8, August 2008

Problem Statement:

A. Read through the reference carefully. Implement routines for learning the parameters of HMM given in section 7. In section 8, “A not-so-simple example”, an interesting exercise is carried out. Perform a similar experiment on “War and Peace” by Leo Tolstoy.
B. Ten bent (biased) coins are placed in a box with unknown bias values. A coin is randomly picked from the box and tossed 100 times. A file containing results of five hundred such instances is presented in tabular form with 1 indicating head and 0 indicating tail. Find out the unknown bias values. (2020_ten_bent_coins.csv) To help you, a sample code for two bent coin problem along with data is made available in the work folder: two_bent_coins.csv and embentcoinsol.m
C. A point set with real values is given in 2020_em_clustering.csv. Considering that there are two clusters, use EM to group together points belonging to the same cluster. Try and argue that k-means is an EM algorithm.

Week 8 :

Lab Assignment 7

Learning Objective:

To model the low level image processing tasks in the framework of Markov Random Field and Conditional Random Field. To understand the working of Hopfield network and use it for solving some interesting combinatorial problems

Reference:

1. http://www.cs.utoronto.ca/~strider/Denoise/Benchmark/
2. Interesting way to denoise an image using random walk: http://www.cs.toronto.edu/~fleet/research/Papers/BMVC_denoise.pdf
3. MRF Image Denoising: https://web.cs.hacettepe.edu.tr/~erkut/bil717.s12/w11a-mrf.pdf
4. Single Neuron and Hopfield Network: Chapter 40, 41, 42
   Information Theory, Inference and Learning Algorithms, David MacKay
   http://www.inference.phy.cam.ac.uk/mackay/itila/

Problem Statement:

A. Many low level vision and image processing problems are posed as minimization of energy function defined over a rectangular grid of pixels. We have seen one such problem, image segmentation, in class. The objective of image denoising is to recover an original image from a given noisy image, sometimes with missing pixels also. MRF models denoising as a probabilistic inference task. Since we are conditioning the original pixel intensities with respect to the observed noisy pixel intensities, it usually is referred to as a conditional Markov random field. Refer to (3) above. It describes the energy function based on data and prior (smoothness). Use quadratic potentials for both singleton and pairwise potentials. Assume that there are no missing pixels. Cameraman is a standard test image for benchmarking denoising algorithms. Add varying amounts of Gaussian noise to the image for testing the MRF based denoising approach. Since the energy function is quadratic, it is possible to find the minima by simple gradient descent. If the image size is small (100x100) you may use any iterative method for solving the system of linear equations that you arrive at by equating the gradient to zero. Extra Credit Challenge: Implement and compare MRF denoising with Stochastic denoising (reference 2).
B. For the sample code hopfield.m supplied in the lab-work folder, find out the amount of error (in bits) tolerable for each of the stored patterns.
C. Solve a TSP (traveling salesman problem) of 10 cities with a Hopfield network. How many weights do you need for the network?

Week 9 :

Lab Assignment for FUN

Objective: (unsupervised learning)

Smoothing filter as a clustering mechanism, Hierarchical Agglomerative Clustering

References:

1. Diffusion and Confusions in Signal and Image Processing
   http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=C857BB546B94BFA72996815EE6DAB6C4?doi=10.1.1.228.3667&rep=rep1&type=pdf
2. Hierarchical Agglomerative Clustering
   Chapter 16 and 17, Introduction to Information Retrieval, Manning, Raghvan and Schutze

Problems:

1. Given the noisy image test_noisy.jpg (available in the codes folder). Use adaptive smoothing (Ref. Diffusions and Confusions in Signal and Image Processing) process to remove the noise. Explain the selected values of the parameters in the algorithm.
2. Use HAC with Euclidean/ Manhattan distance as a measure (Single link, complete link, Ward's distance, Group average, Centroid, Clusteroid) cluster the states of India based on the feature vector comprising of the following parameters (for one of the financial year values available in the data-set)
   a. Percentage of schools with electricity
   b. Percentage of schools with girls toilet
   c. Percentage of schools with drinking water
   d. Percentage of schools with boys toilet

For second problem you should get the data from the following:
https://data.gov.in/

As an example, a simple visualization based on percentage of schools with electricity is available at https://data.gov.in/major-indicator/percentage-schools-electricity

This lab is more fun than serious derivations and implementations. I hope you will enjoy working on the problems.

Week 10 :

Lab Assignment 8

Learning objective:

Basics of data structure needed for state-space search tasks and use of random numbers required for MDP and RL

Title: Matchbox Educable Naughts and Crosses Engine

Reference: http://people.csail.mit.edu/brooks/idocs/matchbox.pdf

Problem Statement:

Read the reference on MENACE by Michie and check for its implementations. Pick the one that you like the most and go through the code carefully. Highlight the parts that you feel are crucial. If possible, try to code the MENACE in any programming language of your liking.

Week 11:

Lab Assignment 9

Learning objective:

Understanding Exploitation - Exploration in simple n-arm bandit reinforcement learning task, epsilon-greedy algorithm

Title: n-armed bandit

Reference: Reinforcement Learning: an introduction by R Sutton and A Barto (Second Edition) (Chapter 1-2)

Problem Statement:

(1) Consider a binary bandit with two rewards {1-success, 0-failure}. The bandit returns 1 or 0 for the action that you select, i.e. 1 or 2. The rewards are stochastic (but stationary). Use an epsilon-greedy algorithm discussed in class and decide upon the action to take for maximizing the expected reward. There are two binary bandits named binaryBanditA.m and binaryBanditB.m are waiting for you.

(2) Develop a 10-armed bandit in which all ten mean-rewards start out equal and then take independent random walks (by adding a normally distributed increment with mean zero and standard deviation 0.01 to all mean-rewards on each time step).
{function [value] = bandit_nonstat(action)}

(3) The 10-armed bandit that you developed (bandit_nonstat) is difficult to crack with a standard epsilon-greedy algorithm since the rewards are non-stationary. We did discuss how to track non-stationary rewards in class. Write a modified epsilon-greedy agent and show whether it is able to latch onto correct actions or not. (Try at least 10000 time steps before commenting on results)

Credits: Be as descriptive as possible while coding. Provide substantial evidence in the form of graphs, analysis etc. supporting your claim. Prepare a small report on the assignment (preferably in tex).

Week 12:
Lab Assignment 10

Learning Objective: Understand the process of sequential decision making (stochastic environment) and the connection with reinforcement learning

Title: Markov Decision Process and Dynamic Programming
Reference:
[1] Artificial Intelligence a Modern Approach, Russell and Norvig (third edition)
Chapter 16, 17, 21
[2] Reinforcement Learning, Sutton and Barto (second edition)
Chapter 3, 4

Problem Statement:

(1) Suppose that an agent is situated in the 4x3 environment as shown in Figure 1. Beginning in the start state, it must choose an action at each time step. The interaction with the environment terminates when the agent reaches one of the goal states, marked +1 or -1. We assume that the environment is fully observable, so that the agent always knows where it is. You may decide to take the following four actions in every state: Up, Down, Left and Right. However, the environment is stochastic, that means the action that you take may not lead you to the desired state. Each action achieves the intended effect with probability 0.8, but the rest of the time, the action moves the agent at right angles to the intended direction with equal probabilities. Furthermore, if the agent bumps into a wall, it stays in the same square. The immediate reward for moving to any state (s) except for the terminal states S^+ is r(s)= -0.04. And the reward for moving to terminal states is +1 and -1 respectively. Find the value function corresponding to the optimal policy using value iteration.

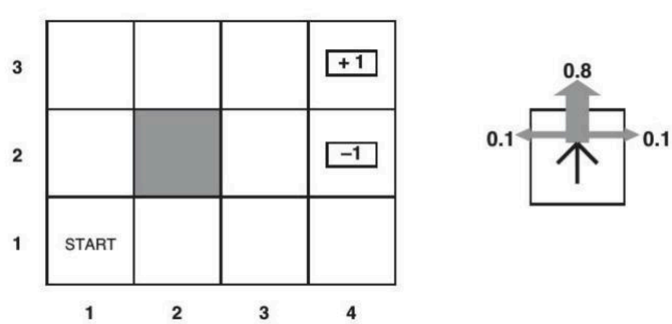


Figure 1: 4×3 grid world with uncertain state change.

Find the value functions corresponding optimal policy for the following:

(a) r(s)=-2
(b) r(s)=0.1
(c) r(s)=0.02
(d) r(s)=1

(2) [Gbike bicycle rental] You are managing two locations for Gbike. Each day, some number of customers arrive at each location to rent bicycles. If you have a bike available, you rent it out and earn INR 10 from Gbike. If you are out of bikes at that location, then the business is lost. Bikes become available for renting the day after they are returned. To help ensure that bicycles are available where they are needed, you can move them between the two locations overnight, at a cost of INR 2 per bike moved.

Assumptions: Assume that the number of bikes requested and returned at each location are Poisson random variables. Expected numbers of rental requests are 3 and 4 and returns are 3 and 2 at the first and second locations respectively. No more than 20 bikes can be parked at either of the locations. You may move a maximum of 5 bikes from one location to the other in one night. Consider the discount rate to be 0.9.

Formulate the continuing finite MDP, where time steps are days, the state is the number of bikes at each location at the end of the day, and the actions are the net number of bikes moved between the two locations overnight.

Download and extract files from gbike.zip. Try to compare your formulation with the code. Before proceeding further, ensure that you understand the policy iteration clearly.

(3) Write a program for policy iteration and resolve the Gbike bicycle rental problem with the following changes. One of your employees at the first location rides a bus home each night and lives near the second location. She is happy to shuttle one bike to the second location for free. Each additional bike still costs INR 2, as do all bikes moved in the other direction. In addition, you have limited parking space at each location. If more than 10 bikes are kept overnight at a location (after any moving of cars), then an additional cost of INR 4 must be incurred to use a second parking lot (independent of how many cars are kept there).

Note: Problem (2) and (3) are directly taken from the book by Sutton and Barto. The problem if attempted independently (without referring to the solutions available) is a good exercise in understanding the implementation issues and finite MDP. Changing the Jack's car rental to Gbike bicycle rental is just to put things in context. The instructor has no intention of earning credit for posing the problem. Credit goes to Sutton and Barto.

Week 13:
Lab Assignment 11

Objective: to understand the design of type-1 (Mamdani) Fuzzy expert system
Problem:

We all have seen, at least in advertisements, washing machines work. Some of us have used them also. Don't feel surprised if I tell you that it was one of the initial devices to have used fuzzy logic.

In this laboratory, we will try to identify the appropriate time needed to wash the load of clothes, given the dirtiness and the volume of the load. This is a simplified design, in practice a lot more variables are involved including: water level, amount of detergent to be dispensed, and temperature of water (recent development) and variables that you and I may imagine in future.

A simple rule-base for the washing machine time problem is given below for reference.

Load Dirtiness/ Load Volume	**vd**	**md**	**ld**	**nd**
fl	vlot	vlot	lot	lit
ml	vlot	mt	mt	lit
ll	lot	lot	lit	lit

vd: very dirty, md: medium dirty, ld: lightly dirty, nd: not dirty
fl: full load, ml: medium load, ll: low load
vlot : very long time, lot: long time, mt: medium time, lit: little time

The rule table consists of 12 rules. For example, the entry in the second row and the third column of the table specifies the rule:

If	load volume	is	medium load
And	load dirtiness	is	lightly dirty
Then	washing time	is	medium time