B551 Final Practice Problems: Fall 2011

The final exam will be similar in breadth, length, and difficulty to these problems. All topics covered in class will be fair game for the final exam.

I. Traveling on a roadmap

You are given a roadmap of some country in the form of a connected non-directed graph in which nodes represent cities and edges represent roads between cities. (A connected graph is one in which every two nodes are connected by a path made of one or several consecutive edges.) Each edge (i,j) is labeled by the length $l(i,j)$ of the road between cities i and j.

Two friends live in two different cities, a and b, of the map. They want to meet in a city of the map (any one). To do this, they move in successive turns. On every turn, the two friends start moving at the same time. Each friend moves to a neighboring city on the map; he/she cannot stay in the same city. The amount of time needed to move from city i to neighboring city j is equal to the length $l(i,j)$ of the road between cities i and j. So, the two friends may not reach their respective new cities at the same time. The friend that arrives first to his/her new city must wait until the other arrives to his/her new city (each one calls the other on his/her cell phone when he/she arrives to a new city) before the next turn can begin. The two friends want to meet as quickly as possible. Note that the goal for the two friends is to meet in a city, not anywhere on a road.

1. Formulate this problem as a state-space search problem:
   a) What is the state space?
   b) What are the initial and the goal states?
   c) What is the successor function?
   d) What is the step cost function?

2. Let $D(i,j)$ be the straight-line distance between any two cities i and j in the map. Which, if any, of the following heuristic functions are admissible? Why?
   a) $D(i,j)$
   b) $D(i,j)-2$
   c) $D(i,j)/2$

3. Is the following statement true or false: "There are connected maps for which no solution exists"? If you answer 'true', give an example of such a map. If you answer is 'false', prove it.

1.
   a) Pairs of cities (i,j)
   b) Initial state: (a,b). Goal state: (i,i) for any i
   c) (i,j) -> (m,n) for any m adjacent to i, n adjacent to j
   d) max{D(i,m), D(j,n)}

2. Only c is admissible.
3. True. (1) ---- (2) is such a map, and with initial state (1,2) the two friends will keep swapping places.

II. Approximately optimal search

The two objectives of finding a solution as quickly as possible and finding an optimal solution are often conflicting. In some problems, one may design two heuristic functions h_A and h_N, such that h_A is admissible and h_N is not admissible, with h_N resulting in much faster search most of the time. Then, one may try to take advantage of both functions.

1. A best-first search algorithm called A_ε^* uses the evaluation function $f(N) = g(N) + h_A(N)$. At each iteration, A_ε^* expands a node N' such that $f(N') \leq (1+\varepsilon)\times\min_{N\in FRINGE}\{f(N)\}$, where ε is any strictly positive number. What can you say about the cost of the solution returned by A_ε^*?

2. Explain briefly how A_ε^* can use the second heuristic function h_N to reduce the time of the search. What tradeoff is being made in choosing ε?

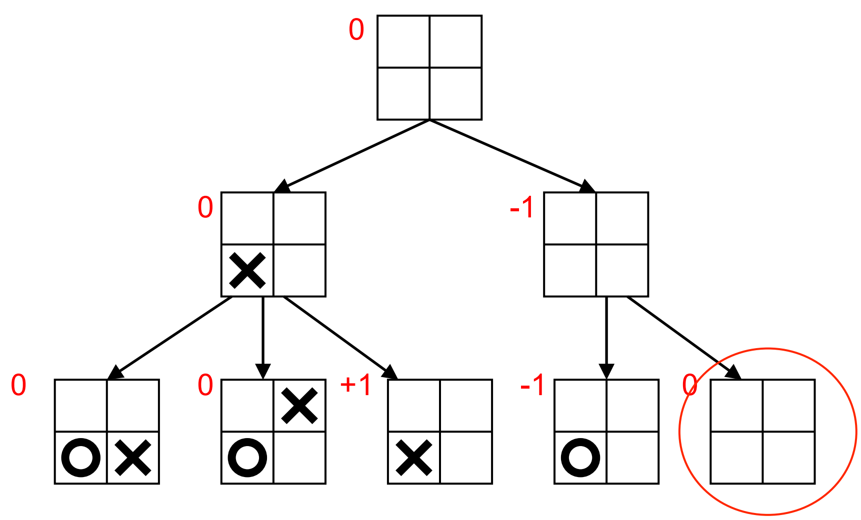
   1. The cost of the solution returned by A_ε^* is no more than (1+ε) times the cost of the optimal path, which can be proven as follows. If N' is the goal node expanded by A_ε^*, and there is another node N on an optimal path to the goal, then by admissibility, f(N) ≤ g*, where g* is the optimal cost to the goal. So, $f(N') = g(N') \leq (1+\varepsilon)\times\min_{N\in FRINGE}\{f(N)\} \leq (1+\varepsilon)\, g^*$ as desired.
   2. Of all the nodes N' that satisfy $f(N') \leq (1+\varepsilon)\times\min_{N\in FRINGE}\{f(N)\}$, A_ε^* can pick the one with the lowest value of h_N. This strategy will use the information encoded in h_N to speed up the search. The tradeoff is that if ε is low, then the search cannot choose from very many nodes and it will not be able to take much advantage of the information in h_N. If ε is high, then the search may return a higher cost path to the goal.

III. Modified Tic-Tac-Toe

Consider the game of 2×2 tic-tac-toe where each player has the additional option of passing, i.e., of marking no square on the 2×2 board. The two players, MAX and MIN, take turns, with MAX going first.

1. Draw the full game tree down to depth 2 (recall that the root of the tree is at depth 0). Do not show the nodes that are rotations or reflections of siblings already shown (your tree should have five leaves). [Draw your tree nicely since you will have to use it again for your answers to questions 2 and 3.]
2. Let the evaluation function of MAX be the number of MAX's marks on the board minus the number of MIN's marks. Give the values of the evaluation function for all leaves of the tree constructed in Question 1 and the values backed-up by the Minimax algorithm for all internal nodes. [Show these values on the tree drawn in Question 1.]
3. Circle all nodes [in the tree that you drew in Question 1] that would not be evaluated by the Alpha-Beta algorithm during a left-to-right depth-first exploration of your tree.
4. Suppose we wanted to solve the game to find the optimal move of MAX (i.e., by constructing a game tree with no depth limit). Explain why Alpha-Beta pruning with an appropriate node ordering can do it, while Minimax can't.

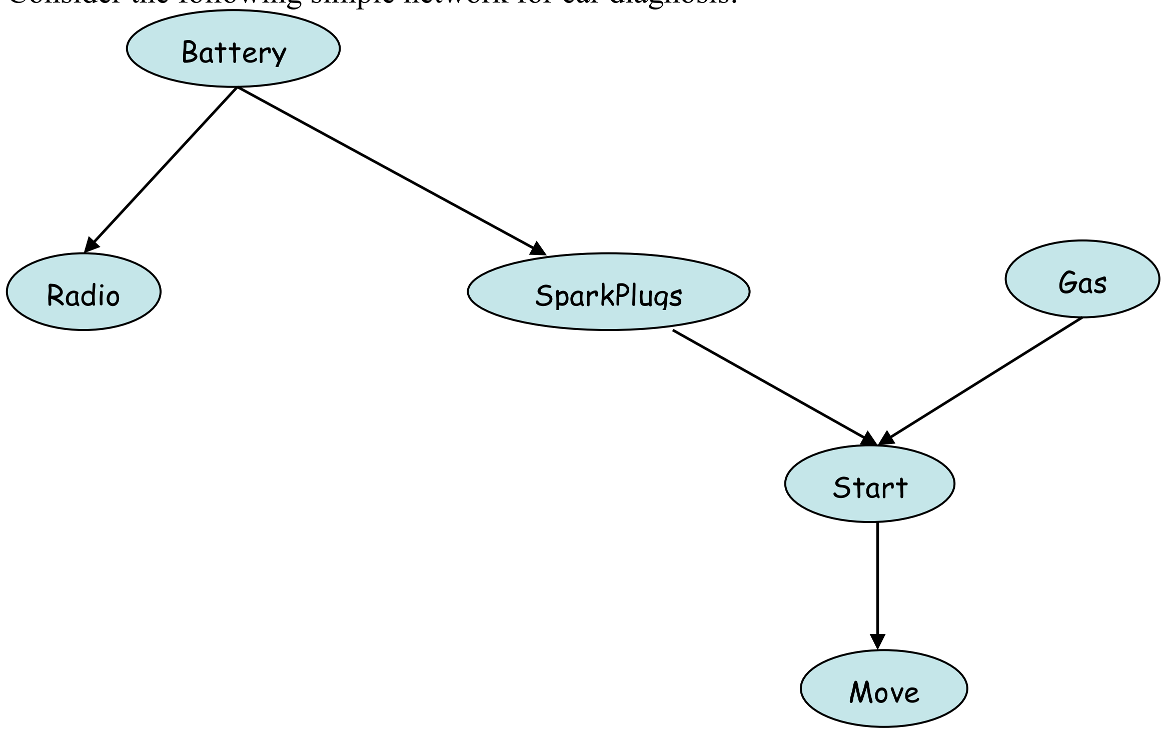
1-3.



4. Because there are loops, Minimax will expand the game tree forever. If Alpha-Beta pruning finds the optimal policy which always makes a play, then it will prune out all self loops. For example, one node ordering that would accomplish this would search the nodes that make a move before the nodes that do nothing,

IV. Automotive Diagnosis

Consider the following simple network for car diagnosis:



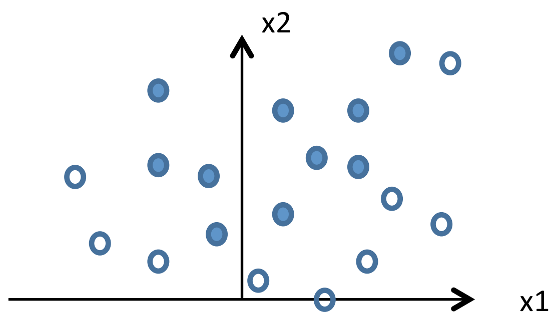
Each variable is Boolean, and a value of True indicates that the aspect of the car is working properly.

1. How many independent probability values would be listed in the joint probability table for these six variables, if no independence assumptions were made?
2. How many independent probability values are listed in the conditional probability tables of this BN?
3. Given what other variables can you say that Battery is independent of Moves? Given what other variables can you say that Battery is NOT independent of Moves?
4. Given what other variables can you say that Gas is independent of Radio? Given what other variables can you say that Gas is NOT independent of Radio?
5. List at least two algorithms that can perform inference on this Bayesian network.

1. $2^6 = 64$ (or 63, eliminating a redundant entry)
2. 1+1+2+2+4+2 = 12
3. Independent given Start and or SparkPlugs. NOT independent given nothing, Gas, or Radio.
4. Independent given nothing, Battery and SparkPlugs. NOT independent given Start, or Moves
5. Variable elimination, stochastic sampling, or belief propagation

V. Support Vector Machines

1. Construct by hand a 2-dimensional linear classifier that is consistent with the positive examples (0,3), (1,4), (2,5) (2,3) and the negative examples (3,2), (1,1),(2,2), (4,3). Give the equation defining this classifier.
2. For the above example, consider the linear classifier -2*x + y + 2. What is the geometric margin of each of the data points? (If an example is misclassified, then its margin is negative)
3. Consider the following 2D dataset (filled circles indicate positive examples, empty circles indicate negative ones).



How might you transform the data into a higher-dimensional feature space in order to get a good linear classifier in that space? Hint: consider that the positive examples seem to be above a parabola.

1. One possible equation is -0.5*x + y - 1.5.
2. The geometric margin is gotten by taking the value of (-2*x + y + 2) for each example and dividing it by the length of (-2,1), which is √5. Then, the value is flipped if the example is negative. So the margins are 5/√5, 4/√5, 3/√5, 1/√5, for the positive examples, and 2/√5, -1/√5, 0, 0 for the negative examples.
3. Consider that $y= x_2-x_1^2$ seems like it would be > 0 for positive examples and < 0 for negative examples, and hence a good linear classifier would be y. You could add a third feature $x_3=x_1^2$ with the linear classifier x_2-x_3 achieving a good fit.

VI. Iterated Rock-Paper-Scissors

Consider playing the game of rock-paper-scissors (RPS) against an opponent over many rounds. One round of RPS consists of both players simultaneously choosing either rock (R), paper (P), or scissors (S). Each player does not know which move the other will make. The winner of the round is determined as follows: rock beats scissors, scissors beat paper, and paper beats rock. You receive a reward of +1 for each win, and -1 for each loss. When both players choose the same move, the round is a tie (resulting in a reward of 0). This problem will consider modeling the opponent using probabilistic models.

1. Assume you are playing an opponent that chooses R, P, or S independently at random at each round. Describe a probabilistic model for this opponent (Hint: it contains two parameters). In terms of those parameters, what is the probability that the opponent plays R? What is the probability that the opponent plays R three times in a row? Given you won the past round by playing P against the opponent's R, what is the probability that the opponent plays R in the next three rounds?
2. Suppose you have observed that the opponent has played rock N_R times, paper N_P times, and scissors N_S times. What are the maximum likelihood estimates of the parameters of your model? What is your expected reward for choosing R? What are two potential disadvantages of maximum likelihood estimation?
3. Now you switch to playing a smarter opponent, where the opponent tries to modify its behavior depending on how well it has done in the past. Assume now that this opponent always chooses the move that has given it the highest average reward in the past (if multiple moves have equal average reward, the move is chosen at random). Can you do better, the same, or worse against this opponent than one that picks moves independently at random?

1. Let θ_R be the probability that the opponent chooses R, θ_P the probability that the opponent chooses P, and let $1-\theta_R-\theta_P$ be the probability that the opponent chooses S.
   The probability that the opponent chooses R 3 times in a row is θ_R^3. The same is true given that you won the past round because the opponent's choices are independent.
2. By extension from the coin-flip model, the ML estimates are $\theta_R = N_R/(N_R+N_P+N_S)$ and $\theta_P = N_P/(N_R+N_P+N_S)$.
   The expected reward for choosing R is sum over all possible outcomes of the reward times the probability of the outcome. Let R(X) denote the reward observed when the opponent plays X, and let P(X) be the estimated probability of the opponent playing X. Then the expected reward is R(R)P(R)+ R(P)P(P)+ R(S)P(S). Entering in the values R(R)=0, R(P)=-1, and R(S)=1, and using the ML estimates of P(X), we have that the expected reward is $(N_S-N_P)/(N_R+N_P+N_S)$.
   Maximum likelihood has the problems of being unstable and inaccurate for small N.

3. Yes, you can “game” this opponent using by predicting its move and playing the appropriate counter move, leading to a better outcome than if the player was random.