CMPT 310
Artificial Intelligence Survey

Simon Fraser University
Summer 2011

Instructor: Oliver Schulte

Assignment 2: Chapters 4, 5, 6. Revised: June 20.

Total Marks: 63 + 5 bonus.
Due Date: June 22, 11 pm.

Instructions: Check the instructions in the syllabus. The university policy on academic dishonesty and plagiarism (cheating) will be taken very seriously in this course. *Everything submitted should be your own writing or coding.* You must not let other students copy your work. Discussion of the assignment is okay, for example to understand the concepts involved. If you work in a group, put down the name of all members of your group. On your assignment, put down your **name**, the number of the assignment and the number of the course. Spelling and grammar count.

Handing in the Assignment. Please post your assignment on our course management server https://courses.cs.sfu.ca/1114-cmpt-310-d100/ . You should post a Word, rtf or Open Office document, so we can add comments on your file directly (*not* pdf). For diagrams please use a drawing program (recommended) or scan a paper and pencil drawing and add it to your Word/Office/rtf file. If the system doesn't take .docx files, try saving your file as .doc instead. *The time when you upload your assignment is the official time stamp.*

We also need a printout. Please hand in the printout to the assignment box in CSIL (Computing Science Instructional Lab). *You need an access card for CSIL*. You should put the printout in the assignment box on day after the due date, but it doesn't have to be by 11 pm.

Chapter 4. Local Search in Continuous Spaces.

1. (26) + 5 bonus. A common problem in machine learning is to find hypotheses that explain the data as well as possible. Let's consider a simple but important instance: modelling coin flips. Suppose you flip a coin 10 times and you observe 7 heads and 3 tails. You assume that the coin flips are mutually independent, and that the chance of getting heads on any given toss is some probability *p* between 0 and 1 (inclusive). Which value of *p* best explains the data?

For a fixed *p*, the probability of seeing 7 heads and 3 tails is given by

$f(p) = p^7 * (1-p)^3.$

Since our goal is to maximize this function, and the logarithm is monotone increasing, we can also consider the natural logarithm $ln(f)$, that is, the function

$l(p) = 7ln(p) + 3ln(1-p).$

I. Use gradient ascent to try and find a maximizing value of *p*. You may do this using either the *f* or the *l* function.
   a. (3) Write down the gradient (derivative) of the function you chose. (Hint: this is probably easier for *l*.)
   b. (10) Try to get close to the maximum in 5 gradient ascent steps. Use as your initial guess *p=1/2* (the coin is fair), and for the first 3 step sizes, use 0.04, 0.02, 0.01. The last 2 step sizes you can choose for yourself. For your answer fill in the table below.

step	p	Step size
0	1/2	0.04
1	Fill in	0.02
2	Fill in	0.01
3	Fill in	Fill in
4	Fill in	Fill in
5	Fill in	

II. Use the Newton-Raphson Method to try and find a maximizing value of *p*. The update formula is given in the lecture notes. You may do this using either the *f* or the *l* function.
   a. (3) Write down the second-order gradient (derivative) of the function you chose. (Hint: this is probably easier for *l*.)
   b. (10) Show the results of 5 Newton-Raphson steps. Use as your initial guess *p=1/2* (the coin is fair).

step	p

0	1/2
1	Fill in
2	Fill in
3	Fill in
4	Fill in
5	Fill in

III. **Bonus Question (5 marks).** Usually local search methods have problems with getting stuck in local maxima. Is there an issue with local maxima in this problem? Why or why not?

Chapter 5. Adversarial Search.

2. (27) Consider the following simple game. There are five (5) pennies on the table. You and your opponent take turns picking up 1, 2, or 3 coins until none is left. You get to keep each penny you pick up. But, if you pick up the last coin, you have to pay 2 cents to your opponent. The object of the game for each player is to finish with as much money as they can.

I. (3) Consider building a game tree to solve the above problem. What would each state in the game tree represent? What are the operator(s)?
II. (10) Show the (complete) resulting state tree for this problem. Each node should have an entire state description. This will be easiest to do with a drawing tool so that you can copy similar parts of the state tree.
III. (2) What utility function/performance measure (i.e., static evaluation function), do you use to evaluate terminal nodes?
IV. (10) Show the (complete) minimax state tree for this problem; be sure to back up the values, etc.
V. (2) Your opponent courteously offers to let you go first. If you accept, what is your first move? How much will you earn, assuming optimal play? You don't need to redraw the entire state tree, just the answer with a brief justification suffices.

Chapter 6. Constraint Satisfaction.

3. (10) The following questions are concerned with the notion of arc consistency.

I. (2) Explain what it means for an arc to be consistent.
II. (2) Explain what it means for a network to be arc consistent.
III. (2) How can you enforce the consistency of an arc that connects variables X and Y?
IV. (4) Suppose that the arc consistency algorithm terminates and for some variables, their domains have multiple values. Is there guaranteed to be a solution to the CSP?
One suggestion if you find this difficult: Think about the CSP problem with three binary variables A, B, and C, and the three inequality constraints stating that none of the three variables may have the same value. You could draw the corresponding constraint network, and apply the arc consistency algorithm.
You don't have to approach the question in this way; the right yes/no answer with a brief justification suffices, no matter how you get it.