Name:
Student ID:

CSE 473 Autumn 2012: Take-Home Final Exam

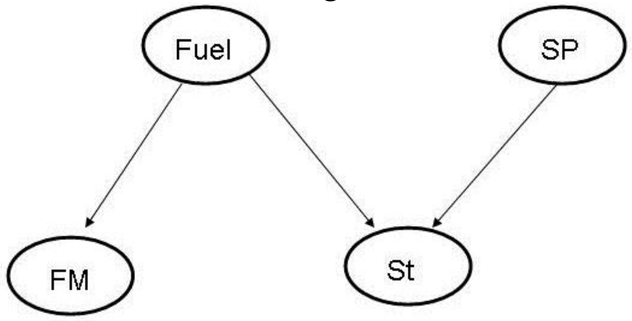
Total: 100 points, 4 questions
Open book, open notes
Due: Wednesday December 12, 2012 BEFORE 10:30AM
Submit to CSE 473 Dropbox

Instructions:

1. Type in your answers using your favorite word processor OR scan and paste your neatly handwritten solution.
2. Type your name and student ID at the top.
3. Keep your answers brief but provide enough details and explanations to let us know you understand the concepts involved.
4. If you need to draw something or write equations by hand as part of your answer, be sure to scan or photograph the page and include the image as part of your answer.

1) **(30 points: 5 points each) Important Concepts and Techniques in AI**
   Provide brief summaries (150-200 words each) of why the following concepts/techniques are important in AI:
   a) A* search
   b) Alpha-beta pruning
   c) Resolution
   d) Support vector machines
   e) Boosting
   f) Cross-validation

2) **(25 points: 4, 3, 3, 5, 10 points) Bayesian Networks**
The following Bayesian network captures some of the causal dependencies pertaining to whether your car will start in the morning:



The random variables *Start?* (St), *Fuel?* (Fuel), and *Clean-Spark-Plugs*? (SP) can each take on the values Yes or No, while *Fuel-Meter* (FM) can take on the values Full, Half, or Empty.
You know **P**(Fuel) = ⟨0.98,0.02⟩ and **P**(SP) = ⟨0.96,0.04⟩. You also know the following CPTs for **P**(FM | Fuel) and **P**(St | Fuel, SP):

Fuel	P(FM = Full)	P(FM = Half)
Yes	0.50	0.30
No	0.01	0.02

Fuel	SP	P(St = Yes)
Yes	Yes	0.99
Yes	No	0.01
No	Yes	0.001
No	No	0

a) Use Bayes rule to compute P(Fuel = Yes | FM = Half).
b) Write down an expression for the full joint distribution over the above four random variables as a product of conditional probabilities based on the above Bayesian network structure.
c) Compute the joint probability P(Fuel = Yes, SP = No, FM = Empty, St = No).
d) Use the full joint distribution in (b) and *inference by enumeration* to compute the probability P(SP = No | St = No, FM = Full). Show all the steps involved.
e) Use the *variable elimination* algorithm to compute **P**(St | FM = Half). Show all the steps involved.

3) (20 points) Decision Trees

Suppose a problem domain is described by the attributes A, B, and C, where A and B can each assume the values *Yes* or *No*, and C can assume the values *Yes*, *No*, or *Maybe*. Based on the decision tree learning algorithm discussed in class and in the textbook (best attribute at each step chosen according to information gain), construct a decision tree for this problem using the following set of training examples:

Example	**A**	**B**	**C**	**Output**
1	*Yes*	*Yes*	*Maybe*	*Yes*
2	*No*	*Yes*	*Maybe*	*No*
3	*Yes*	*No*	*No*	*No*
4	*No*	*No*	*No*	*No*
5	*Yes*	*No*	*Yes*	*Yes*

4) (25 points: 5, 5, 10, 5 points) Neural Networks

In class, we discussed neural networks (perceptrons) that have threshold and sigmoid activation functions. Consider networks whose neurons have *linear activation functions*, i.e., each neuron's output is given by $g(x) = bx+c$, where x is the weighted sum of inputs to the neuron, and b and c are two fixed real numbers.

a) Suppose you have a single neuron with a linear activation function g as above and input $\mathbf{x} = x_0, \ldots, x_n$ and weights $\mathbf{W} = W_0, \ldots, W_n$. Write down the squared error function for this input if the true output is y.

b) Write down the weight update rule for the neuron based on gradient descent on the above error function. (Hint: See the Neural Networks lecture slides)

c) Now consider a network of linear neurons with one hidden layer of m units, n input units, and *one output unit*. For a given set of weights w_{kj} in the input-hidden layer and W_j in the hidden-output layer, write down the equation for the output unit as a function of w_{kj}, W_j, and input $\mathbf{x}$. Show that there is a single-layer linear network with no hidden units that computes the same function.

d) Given your result in (c), what can you conclude about the computational power of N-hidden-layer linear networks for $N = 1, 2, 3, \ldots$?