CSCI 5512: Artificial Intelligence II (Spring'13)

Homework 2 (Due Mar 13, at 11:59pm)

1. (25 points) [Programming Assignment] Consider the rain network in Figure 1. Assume that $Sprinkler = true$ and $WetGrass = true$. For simplicity, we denote these two events by s and w respectively. Further, let r denote the event $Rain = true$.

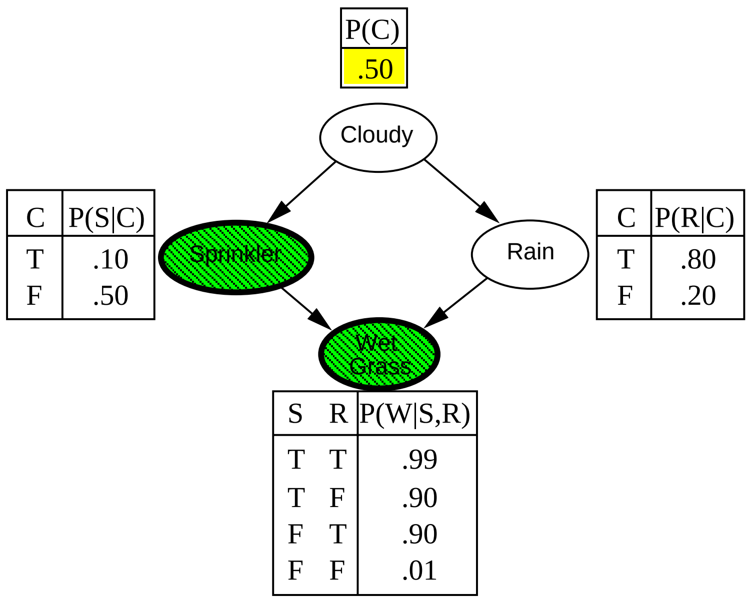

Figure 1: The Rain Network

(a) (10 points) A Gibbs sampler for the problem will need the following conditional probabilities: $P(c|r, w, s)$, $P(c|\neg r, w, s)$, $P(r|c, w, s)$, and $P(r|\neg c, w, s)$, as well as their complements: $P(\neg c|r, w, s)$, $P(\neg c|\neg r, w, s)$, $P(\neg r|c, w, s)$, and $P(\neg r|\neg c, w, s)$. Using the numeric values given in Figure 1 and using the formula for conditional probability of a variable given its Markov blanket, compute the numeric values of the above conditional probabilities.

(b) (15 points) Using the above conditional probabilities, estimate $P(r|s, w)$ using Gibbs sampling for 100 and 10,000 steps.

In addition to the numeric estimates, you have to submit code for GibbsRain implementing Gibbs sampling for the rain network. The code should take one input argument: numSteps, the number of steps, and output an estimate of $P(r|s, w)$. The output should be clearly displayed on screen after running the program.

For language specific and general coding instructions, please see detailed instructions at the end of the homework. Please follow these instructions carefully. Code submitted without adhering to these instructions will not receive any credit.

2. (25 points) Consider the distribution

$$p(a,b,c,d,e,f,g,h,i) = p(a)p(b|a)p(c|a)p(d|a)p(e|b)p(f|c)p(g|d)p(h|e,f)p(i|f,g) \ .$$

   (a) (3 points) Draw the belief network for this distribution.

   (b) (3 points) Draw the moralized graph.

   (c) (3 points) Draw the triangulated graph. Your triangulated graph should contain cliques of the smallest size possible.

   (d) (3 points) Draw a junction tree for the above graph and verify that it satisfies the running intersection property.

   (e) (3 points) Describe a suitable initialization of the clique potentials.

   (f) (10 points) Describe an appropriate message passing schedule giving equations for the absorption at every step of the schedule.

3. (20 points) Consider the joint distribution

$$p(x_1,\ldots,x_n,y) = p(y|x_1,\ldots,x_T)p(x_1)\prod_{t=2}^{T} p(x_t|x_{t-1}) \ ,$$

where all variables are binary.

   (a) (10 points) Draw the junction tree for this distribution and discuss the computational complexity of computing the marginal $p(x_T)$, as suggested by the junction tree algorithm (JTA).

   (b) (10 points) By using an approach different from the plain JTA above, explain how $p(x_T)$ can be computed in time that scales linearly with T.

4. (30 points) [Programming Assignment] Consider the Hidden Markov Model in Figure 2. Assume each of the hidden variables $X_i, i = 0, 1, 2, 3, \ldots$ and the evidence variables $E_i, i =$

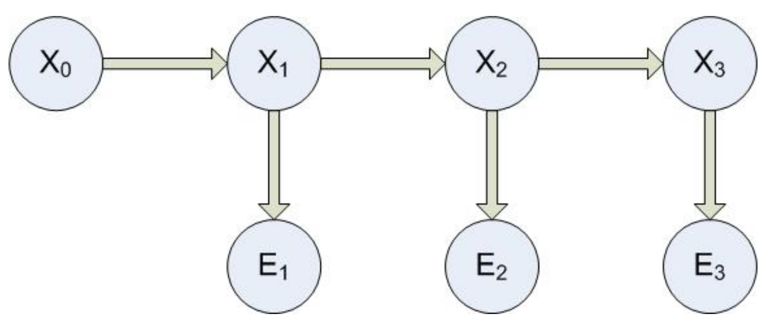

Figure 2: Hidden Markov Model

$1, 2, 3, \ldots$ to be boolean, and can take two values T and F. Let $P(X_0 = T) = P(X_0 = F) = 0.5$. Let the transition matrix $P(X_{t+1}|X_t)$ and sensor matrix $P(E_t|X_t)$ be given by

$$T = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \qquad E = \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} ,$$

where, in the T matrix,

$$T_{11} = P(X_{t+1} = T|X_t = T) , \qquad T_{12} = P(X_{t+1} = F|X_t = T) ,$$
$$T_{21} = P(X_{t+1} = T|X_t = F) , \qquad T_{22} = P(X_{t+1} = F|X_t = F) ,$$

and in the E matrix,

$$E_{11} = P(E_t = T|X_t = T) , \qquad E_{12} = P(E_t = F|X_t = T) ,$$
$$E_{21} = P(E_t = T|X_t = F) , \qquad E_{22} = P(E_t = F|X_t = F) .$$

Consider two sequences of evidence $\mathbf{e}_{1:10}$ over 10 time steps:

Evidence sequence 1: $\mathbf{e}_{1:10} = \langle F, F, F, T, T, T, T, F, F, F \rangle$

Evidence sequence 2: $\mathbf{e}_{1:10} = \langle F, T, F, T, F, T, F, T, F, T \rangle$

For each of the above two sequences of evidence:

(a) (15 points) Compute the smoothed estimates of $X_t, t = 1, \ldots, 10$ given evidence $\mathbf{e}_{1:10}$.

(b) (15 points) Find the most likely sequence of states $X_{1:10}$ given evidence $\mathbf{e}_{1:10}$.

In addition to the smoothed estimates and sequence of states, you have to submit code for SmoothHMM implementing computation of smoothed estimate and MaxSeq implementing computation of the most likely sequence. The code for both algorithms should take two input arguments:

(i) n, the length of the evidence ($n = 10$ for the two examples above)

(ii) Evidence, which is an array of length n containing a '1' for T and '0' for F. Thus, for the first example $\mathbf{e}_{1:10} = \langle F, F, F, T, T, T, T, F, F, F \rangle$, implying Evidence = [0 0 0 1 1 1 1 0 0 0].

The output for SmoothHMM should be an array of length n with smoothed estimates of $P(X_t = T), t = 1, \ldots, n$. The output should be clearly displayed on screen after running the program.

The output for MaxSeq should be a binary array of length n containing the most likely sequence of states with 1 corresponding to T and 0 corresponding to F. The output should be clearly displayed on screen after running the program.

For language specific and general coding instructions, please see detailed instructions at the end of the homework. Please follow these instructions carefully. Code submitted without adhering to these instructions will not receive any credit.

Instructions

Follow the rules strictly. If we cannot run your programs, you get 0 points. Also be sure to cite any and all sources used.

Programming Assignments

For each programming assignment (PA), you have to submit the code as required by the problem and a README file (described below). For any PA, the algorithm must be implemented using a main file as named in the problem (with proper extensions). For example, for 1(b), your file should be GibbsRain.m if you are using Matlab. There may be additional source files (implementing subroutines) associated with the main file.

You can submit code written in Matlab, C, Java, or Python. Code in any other language will not be graded and will not receive any credit. If you implement the algorithms in C, you also have to submit a makefile that will compile your code and produce the proper binary executables.

Note that your program should run/build on the CSE Labs workstations. Assignments that fail to build and run there will be considered as incorrect.

For each algorithm, you must include a README file which describes your program. The README file for a particular algorithm algo should be named README_algo. For example, for 1(b), the file should be named README_GibbsRain.

The README file for each algorithm must contain the following:

1. How to compile your program from the shell/command line. It is important that you give instructions on how to compile *from the command line*. Other options, e.g., compile by opening Eclipse, is not acceptable and will be considered incorrect.
2. How to use the program from the shell/command line.
3. What exactly your program does (briefly).

At the top of the README files and the main source file(s), please include your name, student ID, algorithm, and email.

Things to submit

1. hw2.pdf: Report that contains detailed solutions to all problems.
2. GibbsRain.* and README_GibbsRain: Code and README for Problem 1.
3. SmoothHMM.*, MaxSeq.*, README_SmoothHMM, and README_MaxSeq: Code and READMEs for Problem 4.
4. Any other files which are necessary for your programs.