CSCI 5512: Artificial Intelligence II (Spring'10)

Homework 2 (Due Mar 17 by11:59 PM)

1. (20 points) Consider the Rain network in Figure 1. Assume that WetGrass = true. For simplicity, we denote the events by c, s, r and w for Cloudy=true, Sprinkler=true, Rain=true and WetGrass=true respectively. Each of the 4 variables in the network is boolean, i.e., can take two possible values. Hence, in order to determine the marginal distributions of each variable, it is sufficient to determine the probability of one of the possible outcomes for each variable. Using the sum-product algorithm, compute the marginal probabilities P(c),P(s),P(r),P(w).

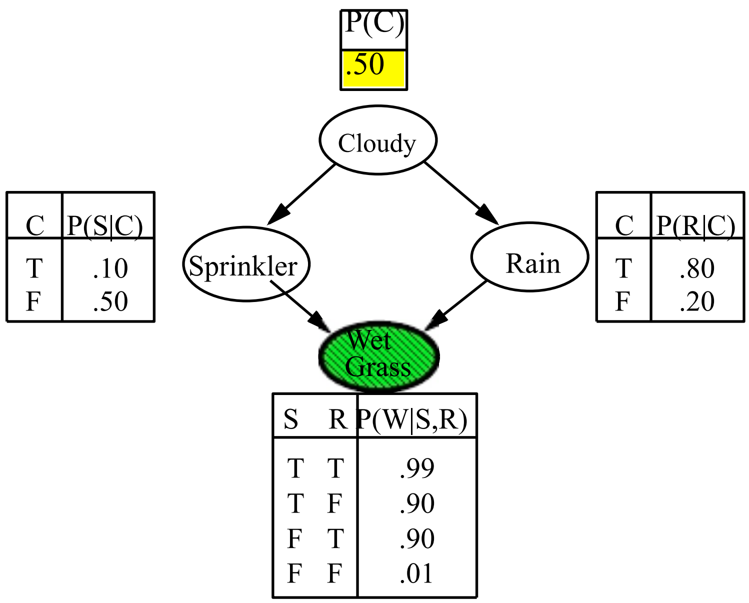

Figure 1: The Rain Network

2. (30 points) [Programming Assignment] Consider the rain network in Figure 1.
   (a) (10 points) A Gibbs sampler for the problem will need the following conditional probabilities: P(c|r,w,s), P(c|¬r,w,s), P(r|c,w,s), and P(r|¬c,w,s), as well as their complements: P(¬c|r,w,s), P(¬c|¬r,w,s), P(¬r|c,w,s), and P(¬r|¬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. Condition on both possible values of Sprinkler.

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).

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.

3. (20 points) Show that any second-order Markov process can be rewritten as a first-order Markov process with an augmented set of state variables. Can this always be done parsimoniously, i.e., without increasing the number of parameters needed to specify the transition model?

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

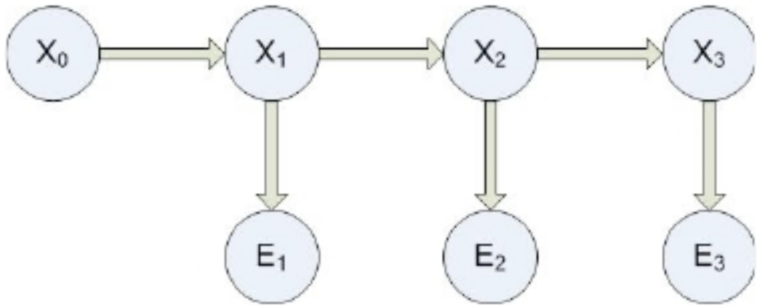

Figure 2: Hidden Markov Model

1,2,3,... 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 give
n by

$$T = \begin{matrix} 0.66 & 0.34 \\ 0.25 & 0.75 \end{matrix} \qquad E = \begin{matrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{matrix} ,$$

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 $e_{1:10}$ over 10 time steps:

Ev1dence sequence 1: $e_{1:10} =$ F,F,F,T,T,T,T,F,F,F
Ev2dence sequence 2: $e_{1:10} =$ F,T,F,T,F,T,F,T,F,T

For each of the above two sequences of evidence:

(a) (15 points) Compute the smoothed estimates of $X_{1:10}$ given evidence $e_{1:10}$.
(b) (15 points) Find the most likely sequence of states $X_{1:10}$ given evidence e.

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, which is the length of the evidence (n = 10 for the two examples above), and (ii) Evidence, which is an array of length numEvidence containing a 1 for T and 0 for F. Thus, for the first example $e_{1:10} =$ F,F,F,T,T,T,T,F,F,F , Evidence = [0 0 0 1 1 1 1 0 0 0]. The output for SmoothHMM should be an array of length (n + 1) with smoothed estimates of $P(X_t = T), t = 0,...,n$. The output for MaxSeq should be an binary array of length n containing the most likely sequence of states with 1 corresponding to T and 0 corresponding to F.

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 for 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(a), your file should be rsRain.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, or Java. 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 IT 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(a), the file should be named README rsRain.

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).

The README files should not be too long, as long as it properly describes the above points. Proper in this case means that a first-time user will be able to answer the above questions without any confusion after reading the README. Within your code you should write detailed comments, although you do not need to comment every line of your code.

At the top of the README files and the main source file(s), please include the following comments:

```
/*CSci5512  Spring'11  Homework  2
*login:  login_used_to_submit
*date:  mm/dd/yy
*name:  full_name
*id:  umn_student_id
*algorithm:  algorithm_name*/
```