**Csci 543 Advanced AI,**

**Assignment 2**

**Wei Chen**

**Q1. [20 + Optional] Temporal Reasoning in BN**

We examine what happens to the probabilities in the umbrella world in the limit of long time sequences. Use the Bayesian Network structure and conditional distributions in Fig.15.2.

1. [10] Suppose we observe an unending sequence of days on which the umbrella appears.

Show that, as the days go by, the probability of rain on the current day increases monotonically towards a fixed point (i.e. it converges to a fixed value). Calculate this fixed point.

1. [10 + Optional] Now consider forecasting further and further into the future, given just the first two umbrella (U1, U2) observations.
2. [10, Optional] Compute the probability *P(R2+k|U1, U2)* for *k*=1, …, 20, and see that the probability converges towards a fixed point. You can write a program to compute it.
3. [10] Calculate the exact value of this fixed point.

Note: In general, when any *P(Rk)* converges to a fixed point *c*,

*. ∀n∈N,*

**Answers:**

**1.** We want to retrieve the probability that the current day is a rainy day, that is to say *Rt*, knowing that we saw *u*1:*t*. To compute this probability we can use the filtering Formula:   
 *P* (*Rt*|*u*1:*t*) = *αP* (*ut*+1|*Rt*+1)   
  
Furthermore we want to compute the fixed point. This condition gives us the relation :  
 *P* (*Rt*|*u*1:*t*) = *P* (*Rt-*1|*u*1:*t-*1)  
Replacing in the previous equation we get the relation :  
 *P* (*Rt*|*u*1:*t*) = *αP* (*ut*+1|*Rt*+1)   
  
As there is only 2 states for the weather : there is rain or there is not... We can replace *P* (*Rt-*1|*u*1:*t-*1) by *p* when there is rain and by 1 - p when there is no rain. That leads us to a system of 2 equations:   
 *p* = *α*0*.*9 *\** 0*.*7*p* + 0*.*3 *\** (1 *- p*)  
 1 *- p* = *α*0*.*2 *\* 0.3p* + 0*.*7 *\** (1 *- p*)  *Solving this system; we find that p ≈* 0*.*8933

**2. (A).** To compute all those probabilities, the easier is to find a recursive relationship between

*P* (*R*2+*k*|*U*1, *U*2) and *P* (*R*2+*k-*1|*U*1, *U*2) Using **Bayes rules** we know that:   
 *P* (*R*2+*k*|*U*1, *U*2) =   
Hence we have:

*P* (*R*2+*k*|*U*1, *U*2) = 0*.*7*P* (*r*2+*k-*1| *U*1, *U*2) + 0*.*3(1 *- P* (*r*2+*k-*1| *U*1, *U*2))  
 *P* (*R*2+*k*| *U*1, *U*2) = 0.4*P* (*R*2+*k-*1| *U*1, *U*2) + 0*.*3  
 When this relation converges, we have:

*P* (*R*2+*k*| *U*1, *U*2) = *P* (*R*2+*k-*1| *U*1, *U*2),  
 hence we have to solve :  
 *P* (*R*2+*k*| *U*1, *U*2) = 0.4*P* (*R*2+*k*| *U*1, *U*2) + 0*.*3

**(B).** In general, when any *P(Rk)* converges to a fixed point *c*,

*. ∀n∈N,*

The solution is trivial: =0.5 Also, knowing the **convergence point** we can now subtract it to each terms.

and we get :  
*P* (*R*2+*k*| *U*1, *U*2)*-*0*.*5 = 0*.*4*P* (*R*2+*k-*1| *U*1, *U*2)*-*0*.*2 = 2*/*5[*P* (*R*2+*k-*1| *U*1, *U*2)*-*0*.*5]  
Rewriting *W* (*R*2+*k*| *U*1, *U*2) = *P* (*R*2+*k*| *U*1, *U*2) *–* 0*.*5 we have :  
*W* (*R*2+*k*| *U*1, *U*2) = 2/5*W* (*R*2+*k-*1| *U*1, *U*2)  
This is a geometric serie so :  
*W* (*R*2+*k*| *U*1, *U*2) = (2*/*5)*kW* (*R*2| *U*1, *U*2)  
Replacing *W* by *P* we finally get :  
*P* (*R*2+*k*| *U*1, *U*2) = (2*/*5)*k*(*P* (*R*2+*k-*1| *U*1, *U*2) *–* 0*.*5) + 0*.*5

**Q2. [10] Variable Elimination in DBN**

Consider applying the variable elimination algorithm to the umbrella DBN unrolled for 3 slices, where the query is **P**(*R3* | *U1, U2, U3* ). Show that the complexity of the algorithm – the size of the largest factor – is the same, regardless of whether the rain variables are eliminated in forward or backward order. For the probability of transition model and that of sensor model, refer to Fig. 15.1.

**Answers:**

We view the world as a series of snapshots, or time slices, each of which contains a set of random variables, some observable and some not. For example, I want to know whether it’s raining today, but I only access to the outside world occurs each morning when I see the director coming in with, or without, an umbrella.

Applying the variable elimination algorithm to the umbrella DBN is exactly a very complex work, which is appeared in 3 slices: Filtering and prediction, Smoothing, and Finding the most likely sequence.

Variable Elimination

P(R3|U1, U2, U3) = P(R3|R2)·(R2|R1) · P(U3|R3) · P(U2|R2) ·P(U1|R1)

= P(R3|R2) · P(U3|R3) ·(R2|R1) · P(U2|R2) ·P(U1|R1)

= fR3(R3) ·fU3(R3) ·R2(R3) ·fU2(R2) · fU1(R1)

= fR3(R2) · fU3(R3) ·R2(R1, R2) · fU2U1(R1R2)

= fR3(R2) · fU3(R3) ·R2U2U1(R1R2)

= fR3(R2) · fU3(R3) ·R2U2U1(R1)

= fR3(R2) · fR2U1U2U3(R1R3)

=fR3R2U1U2U3(R1R3)

=fR2R3U1U2U3(R1)

**Q3. [60] DBN and HMM models in Temporal Reasoning**

A professor wants to know if students are getting enough sleep.

Each day, the professor observes whether the students sleep in class, and whether they have red eyes.

The professor has the following domain theory:

* The prior probability of getting enough sleep, with no observations, is 0.7.
* The probability of getting enough sleep on night t is 0.7 given that the student got enough sleep the previous night; otherwise, 0.3.
* The probability of having red eyes is 0.2 if the student got enough sleep; otherwise, 0.7.
* The probability of sleeping in class is 0.1 if the student got enough sleep; otherwise, 0.3.

1. [10] Formulate this information as a **dynamic Bayesian Network (DBN)** that the professor could use to filter or predict from a sequence of observations.

NOTE: Drawing a DBN of your model is a part of the formulation.

1. [30] For the evidence values below, perform the following inference.

**e1** = not red eyes, not sleeping in class

**e2** = red eyes, not sleeping in class

**e3** = red eyes, sleeping in class

1. [10] State estimation: Compute P(*EnpoughSleep*t | **e1:t**) for each of *t*=1, 2, 3.
2. [10] Smoothing: Compute P(*EnpoughSleep*t | **e1:3**) for each of *t*=1, 2, 3.
3. [10] Compare the filtered and smoothed probabilities in A and B for *t*=1 and *t*=2.
4. [10] Suppose that a particular student show up with red eyes and sleeps in class every day.

Given the model above, explain why the probability that the student had enough sleep the previous night converges to a fixed point rather than continuing to go down as we gather more days of evidence. What is the fixed point? Answer it both numerically and analytically.

1. [10] Formulate the problem as a **Hidden Markov Model** that has only a single observation variable.

You should draw a HMM and give the complete (conditional) probability distribution for the model in its Transition matrix and in its Sensor matrix.

**Answers:**

From the questions we assume:

**St**be the random variable of the student having enough sleep,

**Rt** be the random variable for the student having red eyes,

and **Ct** be the random variable of the student sleeping in class on day t.

**1).** This information could be formulated as the following DBN diagram

……

We’ll accept the answer that either takes S0 or S1 as prior.

Then, the professor could use to filter or predict from a sequence of observations by the following probability tables if S0 is assumed for prior distribution.

|  |  |  |
| --- | --- | --- |
| Ct | St | P(Ct|St) |
| c  c | s  s | 0.1  0.9  0.3  0.7 |

|  |  |  |
| --- | --- | --- |
| St+1 | St | P(St+1|St) |
| st1  st1  st1  st1 | st  st  st  st | 0.7  0.3  0.3  0.7 |

|  |  |  |
| --- | --- | --- |
| Rt | St | P(Rt|St) |
| r  r | s  s | 0.2  0.8  0.7  0.3 |

|  |  |
| --- | --- |
| S0 | P(S0) |
| s | 0.7  0.3 |

**2). We know that:**

**e1** = not red eyes, not sleeping in class = *¬r1, ¬c1*

**e2** = red eyes, not sleeping in class =*r2, ¬c2*

**e3** = red eyes, sleeping in class =*r3, c3*

**(A).** **When t=1**, if *S0* is assumed as prior: We will first compute *P*(*S1*) to get *P*(*S1|¬r1, ¬c1*). This will correspond to predict and update step of our forward algorithm.

Predict: P(s1) = (s1|s0) P(s0) = 0.7\*0.7+0.3\*0.3= .58

P(*¬*s1)=1-P(s1) = .42

Update: P(*s1|¬r1, ¬c1*) = αP(*¬r1, ¬c1|s1*) P(*s1*) =α(0.8\*0.9\*.58) = α.4176

P(*¬s1|¬r1, ¬c1*) = αP(*¬r1, ¬c1|¬s1*) P(*¬s1*) =α(0.3\*0.7\*.42) = α.0882

From these two we get, α= =.5058

***So, P(S1|¬r1, ¬c1) = = .8256 P(¬S1|¬r1, ¬c1) = = .1744***

If *S1* is assumed as prior, then *P(S1)* = 0.7 and *P(¬S1)* = 0.3. The observation at time 1 won’t matter.

**When t=2**, *S0* as prior, again using forward algorithm.

Predict: P(s2*|¬r1, ¬c1*) = (s2|s1) P(s1*|¬r1, ¬c1*) = 0.7\*.8256+0.3\*.1744= .6302

P(*¬*s2*|¬r1, ¬c1*) = 1- P(s2*|¬r1, ¬c1*) = .3698

Update: P(s2*|r1:2, c1:2*) = αP(*r2, ¬c2|s2*) P(s2*|¬r1, ¬c1*) = α(.2\*.9 \*.6302) = α \*.1134

P(*¬*s2*|r1:2, c1:2*) = αP(*r2, ¬c2|¬s2*) P(*¬*s2*|¬r1, ¬c1*) = α(.7\*.7 \*.3698) = α \*.1812

***So, P(s2|r1:2, c1:2) = = .38493***

***P(¬s2|r1:2, c1:2) = = .61507***

**When t = 3,** *S0* as prior,

Predict: P(s3*|r1:2, c1:2*) = (s3|s2) P(s2*| r1:2, c1:2*) = .7\*.3849 +.3\*.6151 = .4540

P(*¬*s3*|r1:2, c1:2*) = 1- P(s3*|r1:2, c1:2*) = .5460

Update: P(s3*|r1:3, c1:3*) = αP(*r3, c3|s3*) P(s3*|r1:2, c1:2*) = α(.2\*.1 \*.4540) = α \*.00908

P(*¬*s3*|r1:3, c1:3*) = αP(*r3, c3|¬s3*) P(*¬*s3*| r1:2, c1:2*) = α(.7\*.3 \*.5460) = α \*.11466

***So, P(s3|r1:3, c1:3) = = .0734***

***P(¬s3|r1:3, c1:3) = = .9266***

**(B).** Smoothing:

**When t=1**, in order to compute P(*S1| r1:3, c1:3*), we should first compute backward message P(*r3, c3|S1*).

P(*r3, c3|S1*) = (*r3, c3|S2*)P(*s2|s1*)

P(*r3, c3|s1*) = .02\*.7+.21\*.3 = .0770, P(*r3, c3|¬s1*) = 0.279

P(*S1| r1:3, c1:3*) = αP(*S*1*| r1:2, c1:2*) P(*r3,c3|s1*)

= α\*.8256\*.770 = α\*.6357

P(*¬s1| r1:3, c1:3*) = α\*.1744\*.279 = α\*.0487

***So, P(S1| r1:3, c1:3) = .6885; P(¬s1| r1:3, c1:3) = .3115***

**When t = 2**, in order to compute P(*S2| r1:3, c1:3*), we should first compute backward message P(*r3, c3|S2*).

P(*r3, c3|S2*) = (*r3, c3|S3*)P(*s3|s2*)

P(*r3, c3|s2*) = .02\*.7+.21\*.3 = .0770, P(*r3, c3|¬s2*) = 0.279

P(*S2| r1:3, c1:3*) = αP(*S*2*| r1:2, c1:2*) P(*r3,c3|s2*)

= α\*.3849\*.770 = α\*.2964

P(*¬s2| r1:3, c1:3*) = α\*.6151\*.279 = α\*.1716

***So, P(S2| r1:3, c1:3) = .7604; P(¬s2| r1:3, c1:3) = .2396***

**When t = 3**, same as part (a, t=3)

***P(S3| r1:3, c1:3) = .0734; P(¬s3|r1:3, c1:3) = .9266***

**C).** From A and B we can observe:

On day 0 Prior = <0.7000, 0.3000>

On day 1(t =1) P(State Elimination)= <0.8256, 0.1744>

P(Smoothing) = <0.6885, 0.3115>

On day 2(t =2) P(State Elimination)= <0.38493, 0.61507>

P(Smoothing) = <0.7604, 0.2396>

On day 2(t =3) P(State Elimination)= <0.734, 0.9266>

P(Smoothing) = <0.734, 0.9266>

So, we can conclude:

There is a downtrend in of Filtered probabilities (A) and uptrend in Smoothing probabilities in (B) from t =1 to t = 2. We can observe a tuple (rt, ct) telling us whether the student had red eyes and whether they were sleeping in class. When t=3, the filtered in A is same as the smoothed probabilities in B. That means reach a fixed point.

**3). <0.0423, 0.9577>; There is a long calculation**

**4)** First to draw a HMM diagram, and again we will accept the diagram with *S0* or *S1*

……

Then, give the complete (conditional) probability distribution.

Instead of two observation variables in DBN, we will shrink two observations to one. This new

observation variables *Ot* can take four values. *S0* can be *S1*

|  |  |  |
| --- | --- | --- |
| Ot | St | P(Ot|St) |
| r, c  r, c  r, c  r, c  r, c  c | s  s  s  s | 0.02  0.18  0.08  0.72  0.21  0.49  0.09  0.21 |

|  |  |  |
| --- | --- | --- |
| St+1 | St | P(St+1|St) |
| st1  st1  st1  st1 | st  st  st  st | 0.7  0.3  0.3  0.7 |

|  |  |
| --- | --- |
| S0 | P(S0) |
| s | 0.7  0.3 |

**Q4.** [30] Robot Sensing Problem

Robot is in the world with 3 locations A, B, C.

The probability that the Robot is in one of location is: P(A) = P(B) = P(C ) = 1/3.

The color of location A is Red while the colors of location B and C are Green.

Robot has a sensor to detect a color, but the sensor is unreliable. The probability that it senses the color of location correct is only 0.9 for each location.

P(Red | A) = P(Green | B) = P(Green | C) = 0.9

Robot is seeing Red currently.

1. [10] What is the probability that the Robot is in location A?
2. [10] What is the probability that it is in location B?
3. [10] What is the probability that the Robot is in C?

**Answers:**

Condition: Robot is Red.

P (A) = P (B) = P(C) = 1/3.

P (Red | A) = P (Green | B) = P (Green | C) = 0.9

So,

1. The probability that the Robot is in location A:

P(A | Red) = P(Red | A) \* P(A) / P(Red) (Bayes’ rule)

Because the probability of being bed in each location is not equal (P(Red) = 1/3 is **wrong**), so we should calculate the probability of being red in each location, and then add together.

P (Red) = P(Red | A) \* P(A) + P(Red | B) \* P(B) + P(Red | C) \* P(C)

= 0.9 \* (1/3) + (1-P(Green | B)) \* P(B) + (1-P(Green | C)) \* P(C)

= 0.9 \* (1/3) + (1-0.9) \* (1/3) + (1-0.9) \* (1/3) = 11/30

So, P(A | Red) = P(Red | A) \* P(A) / P(Red)

= 0.9 \* (1/3) / (11/30)

= 9/11

1. because P(Red) has already calculated in question (1) equal to (11/30), So,

The probability that the Robot is in location B:

P(B | Red) = P(Red | B) \* P(B) / P(Red)(Bayes’ rule)

= (1-P(Green | B)) \* P(B) / P(Red)

= (1-0.9) \* (1/3) / (11/30)

= 1/11

1. because P(Red) has already calculated in question (1) equal to (11/30), So,

The probability that the Robot is in location C:

P(C | Red) = P(Red | C) \* P(C) / P(Red)(Bayes’ rule)

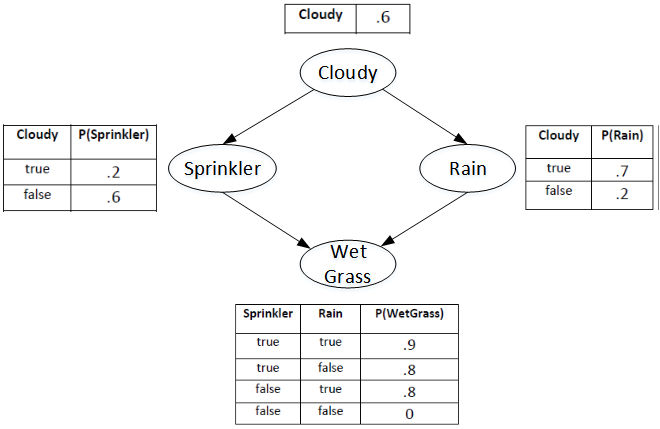
= (1-P(Green | C)) \* P(C) / P(Red)

= (1-0.9) \* (1/3) / (11/30)

= 1/11

**Q5. [**30**]**MCMC

Consider the query *P(Rain|Sprinkler = false, WetGrass=true)*in Figure and how MCMC can answer it.



1. [5] How many states does the Markov chain have?
2. [25] Calculate the transition matrix **Q** containing ***q(y → y’)***for all *y, y’*.

First of all, compute the sampling distribution for each variable, conditioned on its Markov blanket.

* 1. [2] *P(C|r, s)*
  2. [2] *P(C|¬r, s)*
  3. [2] *P(R|c, s,w)*
  4. [2] *P(R|¬c, s,w)*
  5. [17] Using the above probabilities, compute Q below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Q*** | *(c, r)* | *(c, ¬r)* | *(¬c, r)* | *(¬c, ¬r)* |
| *(c, r)* |  |  |  |  |
| *(c,¬r)* |  |  |  |  |
| *(¬c, r)* |  |  |  |  |
| *(¬c, ¬r)* |  |  |  |  |

**Answers:**

1. There are two uninstantiated Boolean variables (Cloudy and Rain) and therefore four possible states.
2. First, I compute the sampling distribution for each variable, conditioned on its Markov blanket.

(a). P (*C|r, s*) = *α*P(*C*)P(*s|C*)P(*r|C*) = α (0.5, 0.5) (0.1, 0.5) (0.8, 0.2) = *α* (0.04, 0.05)

= (4/9, 5/9)

(b). P (*C|¬r, s*)= *α*P(*C*)P(*s|C*)P(¬r*|*C) = α(0.001,0.20) = (1/21, 20/21)

(c). P (*R|c, s,w*) = *α*P(R*|*c)P(w*|*s,R) = *α*(0.8, 0.2) (0.99, 0.90) = *α*(0.792, 0.180) = (22/27, 5/27)

(d). P (*R|¬c, s,w*) = *α*P(R*|¬*c)P(w*|*s,R) = *α*(0.2, 0.8) (0.99, 0.90) = *α*(0.198, 0.720) = (11/51, 40/51) (e). Strictly speaking, the transition matrix is only well-defined for the variant of MCMC in which the variable to be sampled is chosen randomly. (In the variant where the variables are chosen in a fixed order, the transition probabilities depend on where we are in the ordering.) Now consider the transition matrix

·Entries on the diagonal correspond to self-loops. Such transitions can occur by sampling either variable.

For example,

q((c, r) (c, r)) = 0.5 *P* (*C|r, s*) + 0.5 *P* (*r |c, s,w*) = 17/27

·Entries where one variable is changed must sample that variable. For example,

q((c, r) (c, *¬*r)) = 0.5 *P* (*¬r|c, s,w*) = 5/54

·Entries where both variables change cannot occur. For example,

q((c, r) (*¬*c, *¬*r)) = 0

·This gives us the following transition matrix, where the transition is from the state given by the row label to the state given by the column label:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Q*** | *(c, r)* | *(c, ¬r)* | *(¬c, r)* | *(¬c, ¬r)* |
| *(c, r)* | 17/27 | 5/54 | 5/18 | 0 |
| *(c,¬r)* | 11/27 | 22/189 | 0 | 10/21 |
| *(¬c, r)* | 2/9 | 0 | 59/153 | 20/51 |
| *(¬c, ¬r)* | 0 | 1/42 | 11/102 | 310/357 |