ADITYA GAUTAM

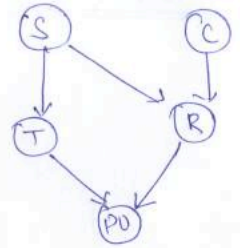
ASSIGNMENT#6 (MACHINE LEARNING)

Problem 1

a) 1) F
2) F
3) T
4) F
5) F
6) F
7) T
8) T
9) F
10) T

b) 1) No variables are d-separated from R given nothing.
2) ~~From~~ T is d-separated from R given S.
3) We need to provide value of PU and one of the value of variable S or C.
There ~~are~~ are two ways of achieving d-separation.
We can provide any variables from S or C to make T and R d-separated.

d) $P(S,T,C,R,PU) = P(S)\,P(C)\,P(PU|T,R)\,P(R|S,C)\,P(T|S)$



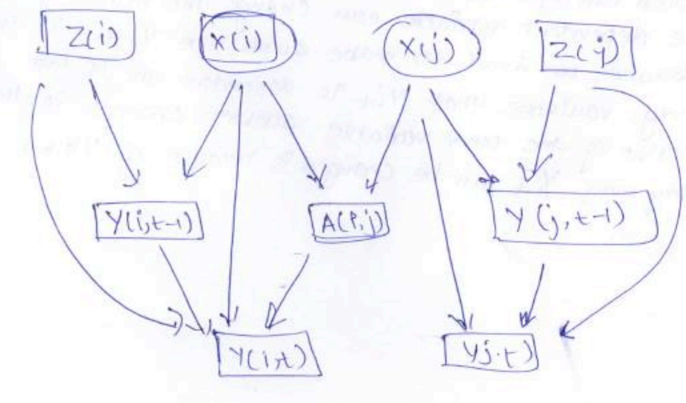
Probability queries

e)

1) $P\left(\text{Sky}=\text{clear} \,/\, \text{Season}=\text{summer}\right) = 0.7$

Problem 2:

a)



b) If we see a change in $Y_{i,t}$, then this change can be due to variable variable which $Y_{i,t}$ will depends upon.

We can write the following thing in the D-separation terms like this.

"Is there any variable, which is not d-separate from $Y_{i,t}$ other than $Y_{j,t-1}$"

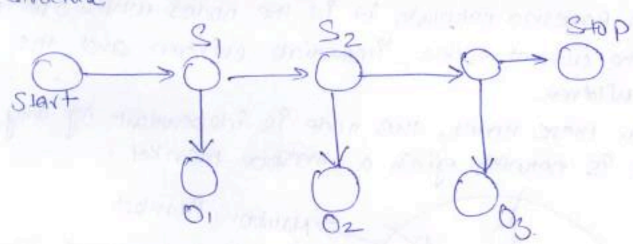
If the answer is yes, that means other variables ~~which~~ can influence the value of $Y_{i,t}$. thus change in value of $Y_{j,t-1}$ doesn't necessarily mean/guarantee that it is because of social link between them.

c) To guarantee that the change in $Y_{i,t}$ is due to $Y_{j,t-1}$, we need to make sure that there is no other variable which can influence the value of the $Y_{i,t}$. Because all the dependent variable can change the value of $Y_{i,t}$. Likewise, we need to make sure the $Y_{j,t-1}$ is the only variable that Y_{it} is dependent on or the value of the other variable doesn't change. So, the only way $Y_{i,t}$ can be changed is because of $Y_{j,t-1}$.

c) To guarantee that the change in $y_{i,t}$ is due to $y_{j,t-1}$, we need to make sure that there is no other variable which can influence the value of the $y_{i,t}$. Because all the dependent variable can change the value of $y_{i,t}$. Likewise, we need to make sure the $y_{j,t-1}$ is the only variable that y_{it} is dependent on or the value of the other variable doesn't change. So, the only way $y_{i,t}$ can be changed is because of $y_{j,t-1}$.

Problem 3)

a) Structure



b) A Generative HMM Model needs 3 things:

① Initial probabilities of states.
This is the prior and would be needed for the initial state probability estimation.

② Transition probabilities of states.
Probabilities tables of transition from one state to every possible state. The dimension would be size(state) x size(state)

③ Emission probabilities:
given the observed variables, we need to know the probability of occuring the observed variable given the state.

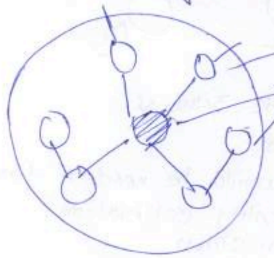
These three thing defines HMM model.
We can predict the most probable sequence of words or do inferences of the model.

c) Given the previous state and current observed variable, the current state is independent of any other variable in HMM.
Observed variable are independent of previous states and other observed variable. It depends only on the current state.

d) Markov Blanket is a set of nodes which would make the given variable independent of all other nodes in a network.

In a Bayesian network, it is the nodes which are parents of the current nodes, immediate children and the parents of children.

Given these nodes, the node is independent of any other node is network given a markov blanket.



→ Markov Blanket.

→ main node is a separated from all the node except the one inside the circle.

Estimation

e) likelihood (MLE) $= \prod_{t=1}^{t+1} P_{s|s'}(s_t|s_{t-1};\theta,\theta_{start},\theta_{stop})$

$\prod_{t=1}^{T} P_{o|s}(o_t|s_t;\gamma)$

log likelihood L

$l = \log(MLE)$

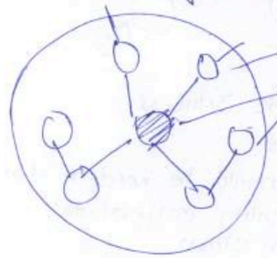
$l = \sum \log P_{s|s'}(s_t|s_t;\theta,\theta_{start},\theta_{stop}) +$

$\sum \log P_{o|s}(o_t|s_t;\gamma)$

d) Markov Blanket is a set of nodes which would make the given variable independent of all other nodes in a network.

In a Bayesian network, it is the nodes which are parents of the current nodes, immediate children and the parents of children.

Given these nodes, the node is independent of any other node in network given a markov blanket.



Estimation

e) likelihood (MLE) $= \prod_{t=1}^{t+1} P_{s|s'}(s_t|s_{t-1};\theta,\theta_{start},\theta_{stop}) \prod_{t=1}^{T} P_{o|s}(o_t|s_t;\gamma)$

log likelihood L

$l = \log(MLE)$

$l = \sum \log P_{s|s'}(s_t|s_t;\theta,\theta_{start},\theta_{stop}) + \sum \log P_{o|s}(o_t|s_t;\gamma)$

Analysis

(O) Alpha_obs = 0.1
Alpha_trans = 0

Baseline accuracy:
Training = 85.22%.
Testing = 93.87%.

Viterbi decoding:
Trang = 96.18%.
Testing = 93.87%.

The Baseline accuracy doesn't takes into account the dependence of one state into another. It consider all the states as independent of each other, which is not always correct in the word corpus. Thus Baseline accuracy is not good as compared to Viterbi decoding which takes state transition and conditional dependence into account.

(P) Alpha-obs = 0.0 , Alpha-trans = 0.0

Baseline accuracy:
Training = 85.22%
Testing = 81.06%.

Viterbi decoding:
Training : ~~92.69~~% 96.41%.
Testing : 92.70%.

Keeping Alpha_obs to 0 will not do smoothing will keep the model consistence with the training data but may provide more error on the testing which is what is happening in the Baseline case. Viterbi results are better.

9)

h) Naïve way of Viterbi decoding.

Try all the possible combination of states in sequence of length T.
Check the HMM probability of that model and the value which gives the maximum probability is the one most probable.
So, for each place is a sequence of length T, the number of possible states are n.
Time complexity $\boxed{O(n^T)}$

i) Viterbi Algorithm tries to get the most probable sequence of states given the HMM model parameters i.e. transition probabilities, initial priors and the emission probabilities.
In Viterbi Algorithm, for each sequence, we calculate the maximum probability of score keeping the present state constant and varying all the possible previous state.

for iteration layers i
    for each ~~of~~ state
        go through all states in previous iter
        get max_val.

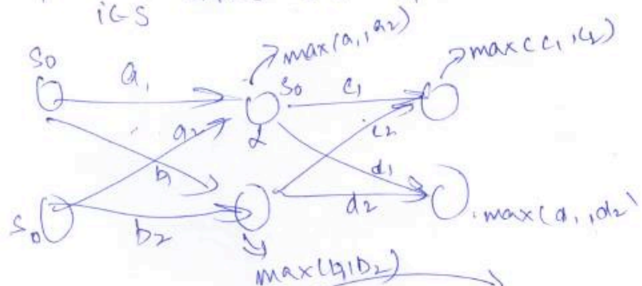
i) , & Remaining part

$$\alpha_1(j) = P_{S|S'}(j|\text{start})\, P_{O|S}(O_1|j)$$

We need to get the maximum score for all the previous layer score combined with the

$$\alpha_{t+1}(j) = \max_{i \in S} \alpha_t(i)\, P_{S|S'}(j|i)\, P_{O|S}(O_{t+1}|j)$$

$$\alpha_{T+1} = \max_{i \in S} \alpha_T(i)\, P_{S|S'}(\text{stop}|i)$$



$\alpha_t \leftarrow a_1 = P(PS_0/S_0)\, P(S_0)\, P(O^t/S_0)$ α_{t-1}

$\alpha_t \leftarrow a_2 = P(S_0/S_p)\, P(S_1) P(O^t/S_0)$ α_{t-1}

Score = max (a_1, a_2) ⟶ for α node.

j) Time complexity = $O(N^2 T)$
$N \to$ no. of states, $T \to$ sequence length

as we are going through all the possible state calculating the score and getting the max at time T. We do the same thing for all n states at time T. Otherwise, time complexity $= O(N^2 T)$