Machine Learning HW6

Hsueh Lin Huang

hsuehlih

Problem 1.

(a)

1. $S \perp R$ (F)
2. $S \perp PU \mid T$ (F)
3. $T \perp C \mid S$ (T)
4. $S \perp PU \mid T, R$ (F)
5. $T \perp C \mid PU$ (F)
6. $S \perp PU \mid R$ (T)
7. $S \perp PU \mid T, C$ (T)
8. $S \perp PU \mid T, C, R$ (T)
9. $S \perp PU \mid C$ (F)
10. $T \perp R \mid S$ (T)

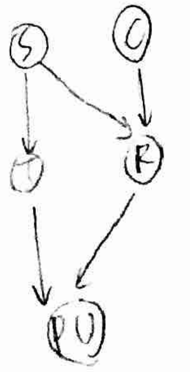
(b)

1. No variables are d-separated from R
2. T is d-separated from R given S
3. Need to condition on C or S, there are two ways to achive this d-separation

(c)

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

(d)



(e)

1. 0.7
2. $\sum_c \sum_t P(PU=Yes \mid T=t, C=c, R=Yes)\,P(T=t \mid S=summer)\,P(C=c \mid S=summer)$

$= 0.1946$

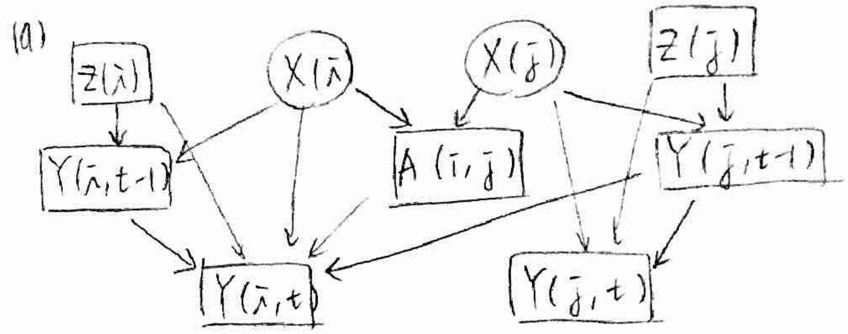
3. $P(PU = Yes \mid C = Yes)$

$$= \sum_r \sum_t \sum_s P(PU = Yes \mid T = t, C = Yes, R = r)\, P(T = t \mid S = s)\, P(R = r \mid C = Yes)\, P(S = s)$$

$= 0.719225$

Problem 2.

(a)

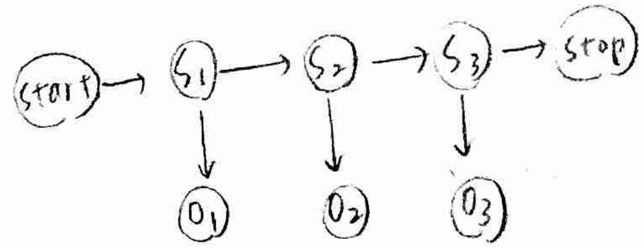


(b) Given $Z(i)$, $X(i)$, $Y(i, t-1)$, then we can know if $Y(i, t)$ is influenced by $Y(j, t-1)$

(c) No, because the change in $Y(i, t)$ might caused by $Y(i, t-1)$, $Z(i)$, or $X(i)$.

Problem 3.

(a)



(b) Through observation, we try to estimate the relationship of the states.

(c) Given S_i, O_i and all the other states and observations are independent.

Given S_i, the previous states are independent of the states after S_i.

. The markov blanket of a node B given, then it would break all the node's active path and it will be d-separated from the rest of the network.

(e)

$$\ell(\theta, \theta_{start}, \theta_{stop}, \gamma) = \log \left(P_{S_{0:T+1}, O_{1:T}}(start, S_{1:T}, stop, O_{1:T}; \theta, \theta_{start}, \theta_{stop}, \gamma)\right)$$

$$= \log \left[\prod_{\bar{i},\bar{j}} \theta_{\bar{i}\bar{j}}^{C_{trans}(\bar{i},\bar{j})}\right]\left[\prod_{\bar{i}} \theta_{\bar{i},stop}^{C_{trans}(\bar{i},stop)}\right]\left[\prod_{\bar{j}} \theta_{start,\bar{j}}^{C_{trans}(start,\bar{j})}\right]\left[\prod_{\bar{i},\bar{j}} \gamma_{\bar{i}\bar{j}}^{C_{obs}(\bar{i},\bar{j})}\right]$$

$$= \sum_{\bar{i},\bar{j}} C_{trans}(\bar{i},\bar{j}) \log \theta_{\bar{i}\bar{j}} + \sum_{\bar{i}} C_{trans}(\bar{i},stop) \log \theta_{\bar{i},stop} + \sum_{\bar{j}} C_{trans}(start,\bar{j}) \log \theta_{start,\bar{j}}$$

$$+ \sum_{\bar{i},\bar{j}} C_{obs}(\bar{i},\bar{j}) \log \gamma_{\bar{i}\bar{j}}$$

(f)

$$\ell(\theta_{start}) = \sum_{\bar{j}} C_{trans}(start,\bar{j}) \log \theta_{start,\bar{j}}$$

$$\sum_{\bar{i}} \ell(\gamma_{\bar{i},:}) = \sum_{\bar{i},\bar{j}} C_{obs}(\bar{i},\bar{j})$$

$$\sum_{\bar{i}} \ell(\theta_{\bar{i},:}) = \sum_{\bar{i},\bar{j}} C_{trans}(\bar{i},\bar{j}) \log \theta_{\bar{i}\bar{j}}$$

$$\sum_{\bar{i}} \ell(\theta_{stop}) = \sum_{\bar{i}} C_{trans}(\bar{i},stop) \log \theta_{\bar{i},stop}$$

The statistic we need are the $C_{trans}(start,\bar{j})$, $C_{trans}(\bar{i},\bar{j})$, $C_{trans}(\bar{i},stop)$ and $C_{obs}(\bar{i},\bar{j})$

(g)

$$\operatorname{argmax} P(start, S_{1:T}, stop, O_{1:T})$$

$$= \operatorname*{argmax}_{S_{1:T}} P_{S_{0:T+1}|O_{1:T}}(start, S_{1:T}, stop \mid O_{1:T}) P_{O_{1:T}}(O_{1:T})$$

$$= \operatorname*{argmax}_{S_{1:T}} P_{S_{0:T+1}|O_{1:T}}(start, S_{1:T}, stop \mid O_{1:T}) = \hat{S}_t$$

(h) $O(m^T)$

(I)
Input: observations of length T, state-graph of length N
Output: best-path

Create a path probability matrix viterbi[N+2, T]
Create a path backpointer matrix backpointer[N+2, T]

for each state s from 1 to N
    forward[s,1] ← $\theta_{0,s} \times \gamma_s(O_1)$
    backpointer[s,1] ← 0
end

for each time step t from 2 to T
    for each state s from 1 to N
        viterbi[s,t] ← $\max_{s'=1}^{N}$ viterbi[s', t-1] × $\theta_{s',s} \times \gamma_s(O_t)$
        backpointer[s,t] ← $\operatorname{argmax}_{s'=1}^{N}$ viterbi[s', t-1] × $\theta_{s',s}$
    end
end
viterbi[q_F, T] ← $\max_{s=1}^{N}$ viterbi[s,T] × θ_{s,q_F}
backpointer[q_F, T] ← $\operatorname{argmax}_{s=1}^{N}$ viterbi[s,T] × θ_{s,q_F}
return the backtrace path by following backpointers to states back in time from backpointer[q_F, T]

(J) $O(N^2T)$

(O) bl_acc_train = 0.8522
bl_acc_test = 0.8106
hmm_acc_train = 0.9618
hmm_acc_test = 0.9387

The base line counts only the frequency of a state cooccurs with a word throughout the data set, while the Viberti counts the transition probability.

The resul

The result of alpha_obs = 0.1 performs better.

bl_acc_train = 0.8522

bl_acc_test = 0.8106

hmm_acc_train = 0.9641

hmm_acc_test = 0.9246

It adds a small constant to the words that never seen in the training data. So the probability of it will not always be 0 when we observe new words.