HW 4

Emmanuel Adebayo

November 2023

1 Problem 1

For each Node in each sequence of the trellis diagram will be denoted with $a_i, b_i, c_i, d_i,$, where i ranges from 1 to 3. N=1, M=2, V=3.

1. (a) $a_1 = p(joe|n) * p(n) = 0.3 * 0.8 = 0.24$,
$a_2 = p(joe|m) * p(m) = 0 * 0.1 = 0$,
$a_3 = p(joe|V) * p(v) = 0 * 0.1 = 0$
$b_1 = P(can|n)*P(n|n)a_1+p(can|n)*p(n|m)*a_2+p(can|n)*p(n|v)*a_3$,
= 0.1*0.1 * 0.24 + 0.1 * 0.3 * 0 + 0.1 * 0.8 * 0 = 0.0024
$b_2 = P(can|m) * P(m|n)a_1 + p(can|m) * p(m|m) * a_2 + p(can|m) * p(m|v) * a_3$,
= 0.5*0.4*0.24 + 0.5*0.1*0 + 0.5*0.1*0 = 0.048
$b_3 = P(can|v)*P(v|n)a_1+p(can|v)*p(v|m)*a_2+p(can|v)*p(v|v)*a_3$,
= 0.2*0.5*0.24 + 0.2*0.6*0 + 0.2*0.1*0 = 0.024
$c_1 = P(see|n)*P(n|n)b_1+p(see|n)*p(n|m)*b_2+p(see|n)*p(n|v)*b_3$,
= 0.1*0.1*0.0024 + 0.1*0.3*0.048 + 0.1*0.8*0.024 = 0.003384
$c_2 = P(see|m) * P(m|n)b_1 + p(see|m) * p(m|m) * b_2 + p(see|m) * p(m|v) * b_3$,
= 0*0.4*0.0024 + 0*0.1*0.048 + 0*0.1*0.024 = 0
$c_3 = P(see|v)*P(v|n)b_1+p(see|v)*p(v|m)*b_2+p(see|v)*p(v|v)*b_3$,
= 0.4*0.5*0.0024 + 0.4*0.6*0.048 + 0.4*0.1*0.024 = 0.01296
$d_1 = P(tom|n)*P(n|n)c_1+p(tom|n)*p(n|m)*c_2+p(tom|n)*p(n|v)*c_3$,
= 0.2*0.1*0.003384 + 0.2*0.3*0 + 0.2+0.8*0.01296 = 0.00214128
$d_2 = P(tom|m) * P(m|n)c_1 + p(tom|m) * p(m|m) * c_2 + p(tom|m) * p(m|v) * c_3$,
= 0*0.4*0.003384 + 0*0.1*0 + 0*0.1*0.01296 = 0
$d_3 = P(tom|v)*P(v|n)c_1+p(tom|v)*p(v|m)*c_2+p(tom|v)*p(v|v)*c_3$,
= 0*0.5*0.003384 + 0*0.6*0 + 0*0.1*0.01296 = 0
Using the forward algorithm, The sentence (Joe can see Tom) is likely to occur in a sequence with the probability of $d_1 + d_2 + d_3$,
=0.00214128 + 0 + 0 =0

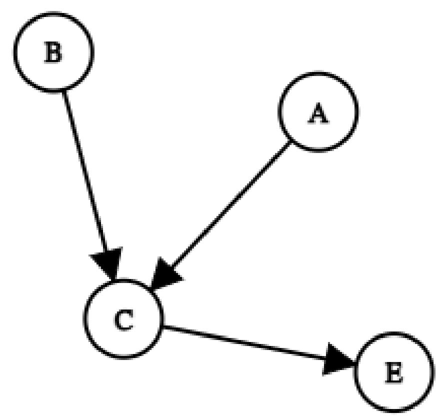
(b) $a_1 = p(will|n) * p(n) = 0.1 * 0.8 = 0.08$,
$a_2 = p(will|m) * p(m) = 0.5 * 0.1 = 0.05$,
$a_3 = p(will|V) * p(v) = 0.1 * 0.1 = 0.01$
$b_1 = max(P(joe|n)*P(n|n)a_1, p(joe|n)*p(n|m)*a_2, p(joe|n)*p(n|v)*a_3)$,
= max(0.3*0.1*0.08, 0.3*0.3*0.05, 0.3*0.8*0.01) = 0.024
$b_2 = max(P(joe|m) * P(m|n)a_1, p(joe|m) * p(m|m) * a_2, p(joe|m) * p(m|v) * a_3)$,
= max(0*0.4*0.08, 0*0.1*0.05, 0*0.1*0.01) = 0
$b_3 = max(P(joe|v)*P(v|n)a_1, p(joe|v)*p(v|m)*a_2, p(joe|v)*p(v|v)*a_3)$,
= max(0*0.5*0.08, 0*0.6*0.05, 0*0.1*0.01) = 0
$c_1 = max(P(spot|n) * P(n|n)b_1, p(spot|n) * p(n|m) * b_2, p(spot|n) * p(n|v) * b_3)$,
= max(0.2*0.1*0.024, 0.2*0.3*0, 0.2*0.8*0) = 0.00048
$c_2 = max(P(spot|m)*P(m|n)b_1, p(spot|m)*p(m|m)*b_2, p(spot|m)*p(m|v) * b_3)$,
= max(0*0.4*0.024, 0*0.1*0, 0*0.1*0) = 0
$c_3 = max(P(spot|v) * P(v|n)b_1, p(spot|v) * p(v|m) * b_2, p(spot|v) * p(v|v) * b_3)$,
= max(0.3*0.5*0.024, 0.3*0.6*0, 0.3*0.1*0) = 0.0036
$d_1 = max(P(tom|n) * P(n|n)c_1, p(tom|n) * p(n|m) * c_2, p(tom|n) * p(n|v) * c_3)$,
= max(0.2*0.1*0.00048, 0.2*0.3*0, 0.2*0.8*0.0036) = 0.000576
$d_2 = max(P(tom|m)*P(m|n)c_1, p(tom|m)*p(m|m)*c_2, p(tom|m)*p(m|v) * c_3)$,
= max(0*0.4*0.0048, 0*0.1*0, 0*0.1*0.0036) = 0
$d_3 = max(P(tom|v) * P(v|n)c_1, p(tom|v) * p(v|m) * c_2, p(tom|v) * p(v|v) * c_3)$,
= max(0*0.5*0.00048, 0*0.6*0, 0*0*0.0036) = 0
Using Viterbi algorithm, the most likely tag sequence of "will Joe spot Tom" is N, N, V, N

2 problem 2

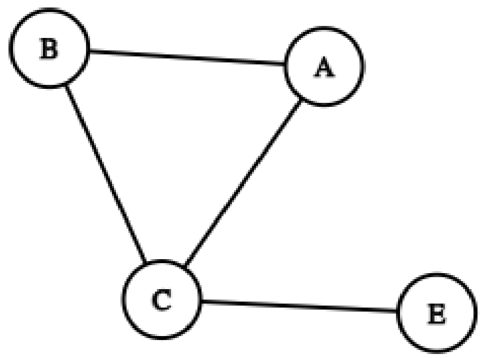
1. (a) This is a proof.
   The following must be true for conditional independence, $P(x, z|y) = P(x, z, y)/p(y)$
   we start by solving,
   $p(x, z|y) = p(x, z, y)/p(y)$
   According to the graph p(x,z,y) = $p(x|y)p(y|z)p(z)$. so,
   $p(x, z|y) = p(x|y)p(y|z)p(z)/p(y)$
   if we rearrange the numerator, then,
   $p(x, z|y) = p(x|y)p(z|y)p(y)/p(y)$
   then reduce the numerator by the denominator, then,
   $p(x, z|y) = p(x|y)p(z|y)$
   Q.E.D

   (b) the same technique applies to b
   $P(x, z|y) = P(x, z, y)/p(y)$
   we start by solving,
   $p(x, z|y) = p(x, z, y)/p(y)$
   According to the diagram
   $p(x, z|y) = p(x, z, y)/p(y)$
   According to the graph p(x,z,y) = $p(x|y)p(z|y)p(y)$.
   $p(x, z|y) = p(x|y)p(z|y)p(y)/p(y)$
   then reduce the numerator by the denominator, then,
   $p(x, z|y) = p(x|y)p(z|y)$
   Q.E.D

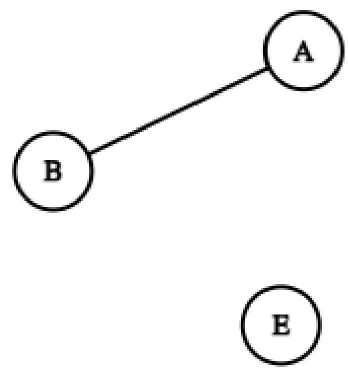
(c) i. we make an ancestral graph where E, A and C and their ancestors are kept



we then proceed to make a moral graph where the parents of each node have an undirected edge between them and convert the all of the edges to an undirected edge.

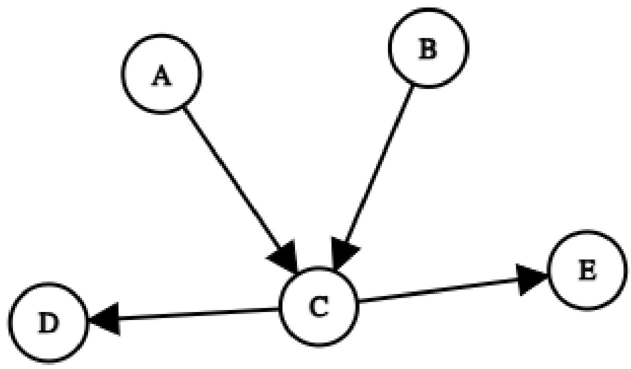


finally, we remove the node c and check if there is a path between E and A

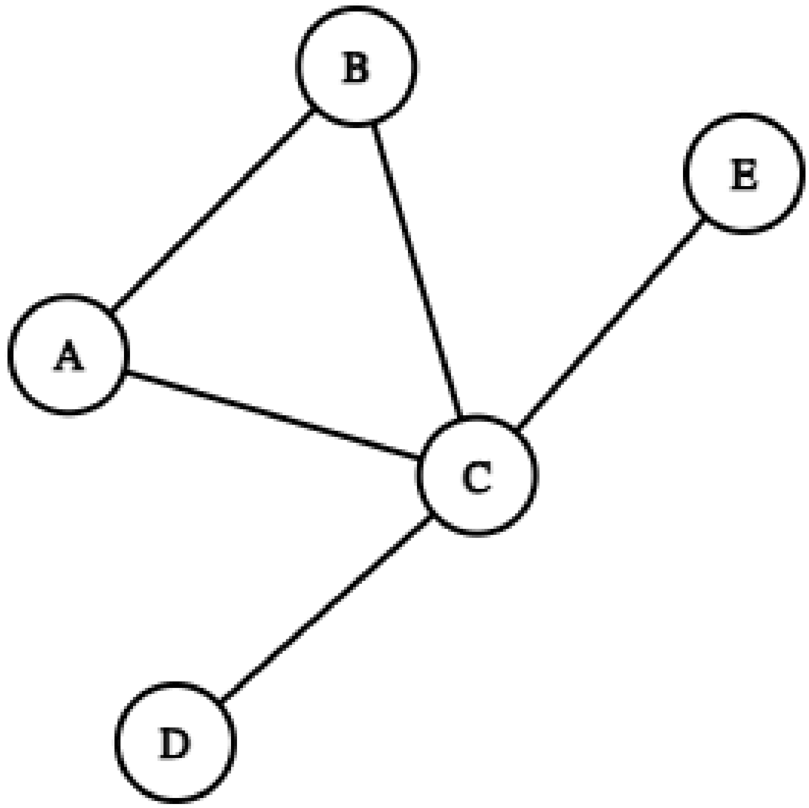


Evidently, there is no path between E and A which indicates D separation. which means E and A are conditionally independent given C

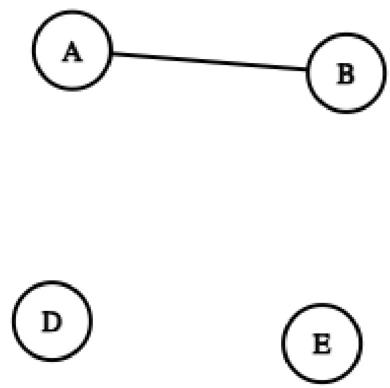
ii. Are E and D conditionally independent, given C? we make an ancestral graph where E, A and C and their ancestors are kept



we then proceed to make a moral graph where the parents of each node have an undirected edge between them and convert the all of the edges to an undirected edge.



finally, we remove the node c and check if there is a path between E and A



Evidently, there is no path between E and D which indicates D separation. which means E and D are conditionally independent given C

Hence, P (E | $ACD) = P(E \mid C)$ is true

3 problem 3

The table represents the distance of each exampple to each cluster. $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ we use the euclidean distance to calculate distance first iteration

Data	Red (6.2, 3.2)	Green (6.6, 3.7)	Blue (6.5, 3.0)
(5.9, 3.2)	0.3	0.86	0.63
(4.6, 2.9)	1.6	2.15	1.9
(6.2, 2.8)	0.4	0.98	0.3
(4.7, 3.2)	1.5	1.96	1.8
(5.5, 4.2)	1.22	1.21	1.56
(5.0, 3.0)	1.22	1.75	1.5
(4.9, 3.1)	1.30	1.80	1.60
(6.7, 3.1)	0.51	0.61	0.22
(5.1,3.8)	1.25	1.50	1.61
(6.0,3.0)	0.28	0.92	0.5

Table 1: iteration: a table calculating the euclidean distances of each instances

second iteration

Data	Red (5.170, 3.170)	Green (5.500, 4.200)	Blue (6.450, 2.950)
(5.9, 3.2)	0.73	1.08	0.60
(4.6, 2.9)	0.63	1.58	1.85
(6.2, 2.8)	1.09	1.57	0.29
(4.7, 3.2)	0.47	1.28	1.77
(5.5, 4.2)	1.08	0	1.57
(5.0, 3.0)	0.24	1.3	1.45
(4.9, 3.1)	0.28	1.25	1.56
(6.7, 3.1)	1.53	1.63	0.29
(5.1,3.8)	0.63	0.57	1.60
(6.0,3.0)	0.85	1.3	0.45

Table 2: iteration: a table calculating the euclidean distances of each instances

third iteration

1. (a) Red's center: (5.9,3.2) + (4.6,2.9) + (4.7, 3.2) + (5.0,3.0) +(4.9,3.1) + (5.1,3.8)+ (6.0,3.0) / 7 = [5.170, 3.170]
   green's center (5.5,4.2) / 1 = [5.500, 4.200]
   blue's center = (6.2,2.8) + (6.7,3.1) / 2 = [6.450,2.950]

Data	Red (4.800, 3.050)	Green (5.300, 4.000)	Blue (6.200, 3.025)
(5.9, 3.2)	1.11	1	0.35
(4.6, 2.9)	0.25	1.30	1.60
(6.2, 2.8)	1.42	1.5	0.23
(4.7, 3.2)	0.18	1	1.5
(5.5, 4.2)	1.35	0.28	1.37
(5.0, 3.0)	0.21	1.04	1.2
(4.9, 3.1)	0.11	0.98	1.3
(6.7, 3.1)	1.9	1.66	0.51
(5.1,3.8)	0.81	0.28	1.35
(6.0,3.0)	1.20	1.22	0.2

Table 3: iteration: a table calculating the euclidean distances of each instances

(b) Red's center = (4.6,2.9)+(4.7,3.2)+(5.0,3.0)+(4.9,3.1) / 4 = [4.800, 3.050]
green's center = (5.5,4.2)+(5.1,3.8) / 2 = [5.300,4.000]
blue's center = (5.9,3.2)+(6.2,2.8)+(6.7,3.1)+(6.0,3.0) / 4 = [6.200,3.025]

(c) when it converges. that is after three iterations Red's center = (4.6,2.9)+(4.7,3.2)+(5.0,3.0)+(4.9,3.1) / 4 = [4.800, 3.050]
green's center = (5.5,4.2)+(5.1,3.8) / 2 = [5.300,4.000]
blue's center = (5.9,3.2)+(6.2,2.8)+(6.7,3.1)+(6.0,3.0) / 4 = [6.200,3.025]

(d) The cluster stops changing after three iterations. so three iterations is required for the clusters to converge