14.21 Three soccer teams A, B, and C, play each other once. Each match is between two teams, and can be won, drawn, or lost. Each team has a fixed, unknown degree of quality an integer ranging from 0 to 3 and the outcome of a match depends probabilistically on the difference in quality between the two teams.
a. Construct a relational probability model to describe this domain, and suggest numerical values for all the necessary probability distributions.
b. Construct the equivalent Bayesian network for the three matches.
c. Suppose that in the first two matches A beats B and draws with C. Using an exact inference algorithm of your choice, compute the posterior distribution for the outcome of the third match.
d. Suppose there are n teams in the league and we have the results for all but the last match. How does the complexity of predicting the last game vary with n?
e. Investigate the application of MCMC to this problem. How quickly does it converge in practice and how well does it scale?

a. Construct a relational probability model to describe this domain, and suggest numerical values for all the necessary probability distributions.

We can define the class Team with two attributes: name and quality. There will be three teams which names will be A, B and C. On every team instantiation we will assign to it a random integer value in the range 0 to 3 to the quality attribute. We will define a procedure called match(team_1, team_2) that will generate three possible values: WIN, TIE or LOSE (making reference to the team_1). If there is no quality difference between teams, then we can define that the probability to win for team_1 is 1/3, the probability to lose is 1/3 and the probability to make a tie is 1/3. This is the same for team_2. Then we can use the quality difference between the teams in the match to increase or decrease the probabilities of the match result.

Team: {A, B, C}
Quality: Team -> {0,1,2,3}
Match: Team1, Team2 -> {Win, Tie, Lose}
Quality(team): < ¼, ¼, ¼, ¼ >
Match(Team1, Team2): CPT presents the probabilities of the match between Team1 vs Team2 given its quality difference: Quality(Team1) - Quality(Team2).
All the teams present the same Quality uniform distribution and all the matches present the same pseudo CPT (below). The real CPT for matches will present two columns for Team1 and Team2 qualities and not the Quality Difference column, but I did this instead to avoid drawing 16 rows.

Quality Difference	P(WIN | QDiff)	P(TIE | QDiff)	P(LOSE | QDiff)	Sum
-3	**1/3 - 0.15**	**1/3 - 0.15**	**1/3 + 0.30**	1
-2	**1/3 - 0.10**	**1/3 - 0.10**	**1/3 + 0.20**	1
-1	**1/3 - 0.05**	**1/3 - 0.05**	**1/3 + 0.10**	1
0	**1/3**	**1/3**	**1/3**	1
+1	**1/3 + 0.10**	**1/3 - 0.5**	**1/3 - 0.05**	1
+2	**1/3 + 0.20**	**1/3 - 0.10**	**1/3 - 0.10**	1
+3	**1/3 + 0.30**	**1/3 - 0.15**	**1/3 - 0.15**	1

In bold are the match probabilities distribution.

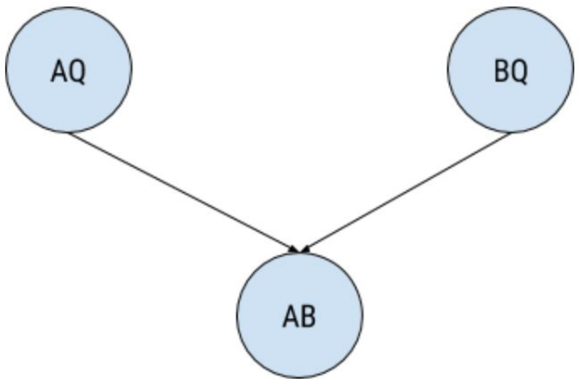
As an example I use a step value of 0.10 per positive point of quality difference to increase the probability of winning and decrease 0.05 the probabilities of losing or tying, but there are a lot of options to use in this model.

We could define a class tournament to test our model that will process all the matches. In this case just three matches: A vs B, A vs C and B vs C.

b. Construct the equivalent Bayesian network for the three matches.

First I build a bayesian network with just two teams, the variables AQ and BQ represent the quality of every team with four possible states each one. And then the variable AB that represents the possible output of the match between A and B based on theirs qualities. Here we show the CPTs where you can see that the quality probability is uniform distributed for each team. Here I show a possible CPT of the variable AB. AB=Win means that A wins B or that B lose against A. The number of combination between AQ and BQ possible states is 16. But the probabilities looks all similar because here is the difference between AQ and BQ what increase or decrease probabilities. And there are only 7 possible differences (-3, -2, -1, 0, 1, 2, 3) like we showed above.

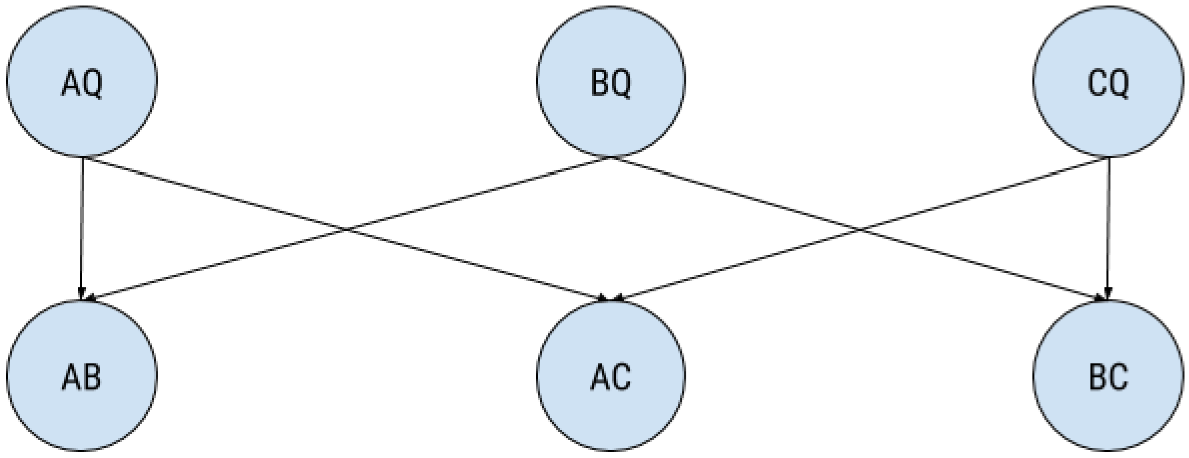
AQ	P(AQ)
0	0.25
1	0.25
2	0.25
3	0.25



BQ	P(BQ)
0	0.25
1	0.25
2	0.25
3	0.25

AQ	BQ	P(AB=W | AQ, BQ)	P(AB=T | AQ, BQ)	P(AB=L | AQ, BQ)
0	0	1/3	1/3	1/3
0	1	1/3 - 0.05	1/3 - 0.05	1/3 + 0.10
0	2	1/3 - 0.10	1/3 - 0.10	1/3 + 0.20
0	3	1/3 - 0.15	1/3 - 0.15	1/3 + 0.30
1	0	1/3 + 0.10	1/3 - 0.05	1/3 - 0.05
1	1	1/3	1/3	1/3
1	2	1/3 - 0.05	1/3 - 0.05	1/3 + 0.10
1	3	1/3 - 0.10	1/3 - 0.10	1/3 + 0.20
2	0	1/3 + 0.20	1/3 - 0.10	1/3 - 0.10
2	1	1/3 + 0.10	1/3 - 0.05	1/3 - 0.05
2	2	1/3	1/3	1/3
2	3	1/3 - 0.05	1/3 - 0.05	1/3 + 0.10
3	0	1/3 + 0.30	1/3 - 0.15	1/3 - 0.15
3	1	1/3 + 0.20	1/3 - 0.10	1/3 - 0.10
3	2	1/3 + 0.10	1/3 - 0.05	1/3 - 0.05
3	3	1/3	1/3	1/3

And the full bayesian network variables (no CPTs here) will look like:



Where the variables in the Bayesian network come with the distribution probabilities based on our model:

AQ: Quality(A)
BQ: Quality(B)
CQ: Quality(C)
AB: Match(A, B)
AC: Match(A, C)
BC: Match(B, C)

c. Suppose that in the first two matches A beats B and draws with C. Using an exact inference algorithm of your choice, compute the posterior distribution for the outcome of the third match.

We will need to calculate **P**(BC | AB = Win, AC = Tie) but we can say something before:

A wins B then AQ is likely to be greater than BQ
A tie with C then AQ and CQ are more likely to be similar values.
Then C is more likely to beat B.

Looking at the AB CPT we can see that **P**(AQ | AB = Win) = <62/350, 77/350, 95/350, 116/350> representing the quality probability distribution (AQ=0, AQ=1, AQ=2, AQ=3) given AB=win.
This means that AQ is more likely to be closer to 3 than to 0.

P(BQ | AB = Win) = <116/350,, 95/350, 77/350, 62/350 >
This means that BQ is more likely to be closer to 0 than to be 3.

Looking at the match CPT we see that the greater probability that a match ends in a TIE happens when both teams qualities are the same. This means that AQ and CQ are more likely to present the same quality score. Then it is more likely that CQ presents a score closer to 3 than to 0.

Using Exact Inference (by Enumeration):

P(BC | AB = Win, AC = Tie) = alpha x **P**(BC, AB=Win, AC=Tie)

$$P(BC \mid AB = Win,\ AC = Tie) = \alpha \times P(BC,\ AB = Win,\ AC = Tie)$$

$$P(BC \mid AB = Win,\ AC = Tie) = \alpha \times \sum_{y \in (AQ, BQ, CQ)} P(BC,\ AB = Win,\ AC = Tie,\ y)$$

Where variable *y* represent all the possible joint combination of all other variables.
y belongs to (AQ, BQ, CQ) and generates 4*4*4 combinations .

We can rewrite the same as:

$$P(BC \mid AB = Win,\ AC = Tie) = \alpha \times \sum_{a \in AQ} \sum_{b \in BQ} \sum_{c \in CQ} P(a,\ b,\ c,\ AB = Win,\ BC,\ AC = Tie)$$

$$= \alpha \times \sum_{a \in AQ} \sum_{b \in BQ} \sum_{c \in CQ} P(a) \times P(b) \times P(c) \times P(AB = Win \mid a,\ b) \times P(BC \mid b, c) \times P(AC = Tie \mid a,\ c)$$

$$= \alpha \times \sum_{a \in AQ} P(a) \times \sum_{b \in BQ} P(b) \times \sum_{c \in CQ} P(c) \times P(AB = Win \mid a,\ b) \times P(BC \mid b, c) \times P(AC = Tie \mid a,\ c)$$

I am going to do less effort because I defined the same uniform probability for all teams quality:

$$= \alpha \times P(a) \times P(b) \times P(c) \times \sum_{a \in AQ} \sum_{b \in BQ} \sum_{c \in CQ} P(AB = Win \mid a,\ b) \times P(BC \mid b, c) \times P(AC = Tie \mid a,\ c)$$

$$= \alpha \times 0.25^3 \times \sum_{a \in AQ} \sum_{b \in BQ} \sum_{c \in CQ} P(AB = Win \mid a,\ b) \times P(BC \mid b, c) \times P(AC = Tie \mid a,\ c)$$

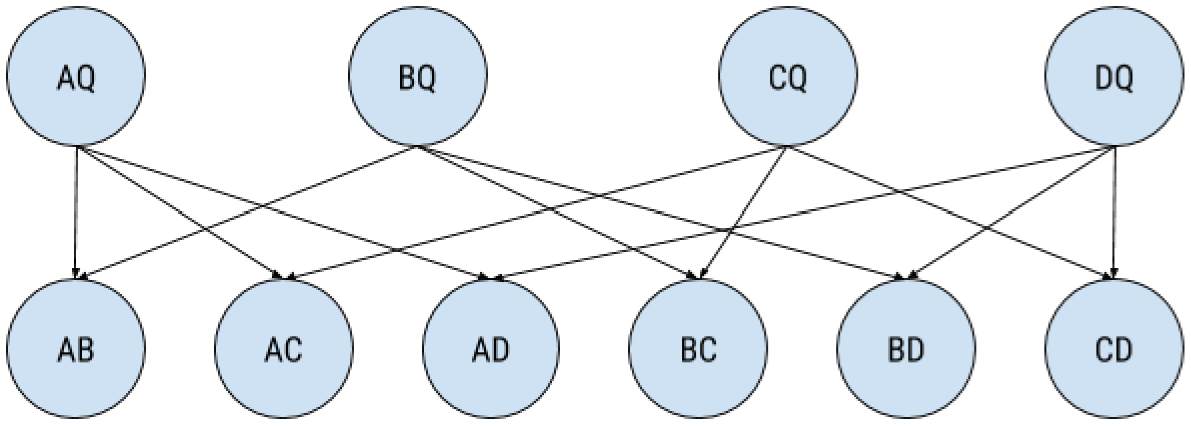
$$= \alpha' \times \sum_{a \in AQ} \sum_{b \in BQ} \sum_{c \in CQ} P(AB = Win \mid a,\ b) \times P(BC \mid b, c) \times P(AC = Tie \mid a,\ c)$$

$$= \alpha' \times \sum_{a \in AQ} \sum_{b \in BQ} P(AB = Win \mid a,\ b) \times \sum_{c \in CQ} P(BC \mid b, c) \times P(AC = Tie \mid a,\ c)$$

P(BC | AB=Win, AC=Tie) is approx: <0.342, 0.271, **0.387**>
It is more likely that C wins.

d. Suppose there are n teams in the league and we have the results for all but the last match. How does the complexity of predicting the last game vary with n?

When n=4 the Bayesian network is:



Number of nodes m = n + n*(n-1)/2 (10 in this case)
Number of nodes in the network is quadratic based on the number of teams ($m = n^2$ with $n \to \infty$)

The CPT for team quality presents 4 rows and the CPT for every match presents 16 rows.
Space required: 4*n + 16*n*(n-1)/2 = 4n + 8nn - 8n = 8nn - 4n = 4n(2n - 1)

Here the query is **P**(CD | ab, ac, ad, bc, bd)

$$P(CD \mid ab, ac, ad, bc, bd) = \alpha \times \sum_a \sum_b \sum_c \sum_d P(a, b, c, d, ab, ac, ad, bc, bd, CD)$$

$$= \alpha \times \sum_a \sum_b \sum_c \sum_d P(a)P(b)P(c)P(d)P(ab|a, b)P(ac|a, c)P(ad|a, d)P(bc|b, c)P(bd|b, d)P(CD|c, d)$$

P(a), P(b), P(c) and P(d) are the same for all elements.

$$= \alpha \times P(a)P(b)P(c)P(d) \sum_a \sum_b P(ab \mid a, b) \sum_c P(ac \mid a, c)P(bc \mid b, c) \sum_d P(bd \mid b, d)P(CD \mid c, d)$$

$$= \alpha' \times \sum_a \sum_b P(ab \mid a, b) \times \sum_c P(ac \mid a, c) \times P(bc \mid b, c) \times \sum_d P(ad \mid a, d) \times P(bd \mid b, d) \times P(CD \mid c, d)$$

Number of iterations: $4^4 = 256$ (summations)

And for any n > 1 the number of matches : *n * (n-1) / 2*
Then we have information on *(n*(n-1)/2)-1* matches results.

P(Tn-1Tn | T1T2, T1T3, ,,,, T1Tn, T2T3, …, T2Tn, … Tn,Tn-2Tn)

An exact inference:

$$P(X_{n-1}X_n \mid x_1x_2,\ x_1x_n,\ \ldots,\ x_{n-2}x_n) = \alpha \times \sum_{x1\ \in X_1}\ \sum_{x2\ \in X_2} \cdots \sum_{x_n\ \in X_n} P(x_1,\ x_2, \ldots,\ x_n,\ x_1x_2,\ x_1x_n,\ \ldots,\ x_{n-2}x_n, X_{n-1}X_n)$$

$$= \alpha \times \sum_{x1\ \in X_1}\ \sum_{x2\ \in X_2} \cdots \sum_{x_n\ \in X_n} P(x_1)P(x_2)....P(x_n).P(x_1x_2 \mid x_1, x_2)....P(x_1x_n|x_1, x_n).\ ...P(x_{n-2}x_n \mid x_{n-2}, x_n)P(X_{n-1}X_n|X_{n-1}, X_n)$$

$$= \alpha \times P(x_1)P(x_2)....P(x_n) \sum_{x1\ \in X_1}\ \sum_{x2\ \in X_2} \cdots \sum_{x_n\ \in X_n} P(x_1x_2 \mid x_1, x_2)....P(x_1x_n|x_1, x_n).\ ...P(x_{n-2}x_n \mid x_{n-2}, x_n)P(X_{n-1}X_n|X_{n-1}, X_n)$$

$$= \alpha \times 0.25^n \times \sum_{x1\ \in X_1}\ \sum_{x2\ \in X_2} \cdots \sum_{x_n\ \in X_n} P(x_1x_2 \mid x_1, x_2)....P(x_1x_n|x_1, x_n).\ ...P(x_{n-2}x_n \mid x_{n-2}, x_n)P(X_{n-1}X_n|X_{n-1}, X_n)$$

$$= \alpha' \times \sum_{x1\ \in X_1}\ \sum_{x2\ \in X_2} \cdots \sum_{x_n\ \in X_n} P(x_1x_2 \mid x_1, x_2)....P(x_1x_n|x_1, x_n).\ ...P(x_{n-2}x_n \mid x_{n-2}, x_n)P(X_{n-1}X_n|X_{n-1}, X_n)$$

$$= \alpha' \times \sum_{x1}\sum_{x2} P(x_1x_2 \mid x_1, x_2) \times \sum_{x3} P(x_1x_3 \mid x_1, x_3) \times P(x_2x_3 \mid x_2, x_3) \times \sum_{x4} \ldots \times \sum_{x_n} P(x_1x_n|x_1, x_n) \times \ldots \times P(x_{n-2}x_n \mid x_{n-2}, x_n) \times P(X_{n-1}X_n|X_{n-1}, X_n)$$

Number of iterations: 4^n
Number of factors: n*(n-1) / 2
(n-1) generated 4^n
(n-2) generated 4^{n-1}
(n-3) generated 4^{n-2}
...
(2) generated 4^3
(1) generated 4^2
(0) generated 4^1
Complexity: $O(4^n)$

e. Investigate the application of MCMC to this problem. How quickly does it converge in practice and how well does it scale?

I adapted the pseudo code of GIBBS in the book for the problem. The algorithm converge very quickly but it does not scale very well. The scale problem comes from the Z variable. The algorithm requires to sample values for all variables that do not belong to the evidence plus the query variable. All the teams qualities are in Z. So, we need to sample for each team from the distribution **P**(t | mb(t)). The calculation of this distribution is heavy: presents *U-1* nodes (being U the number of matches). And this is done N times (for all teams). After the teams we sample from the last match distribution **P**(last_match | mb(last_match)) and this is complex too, but at least we can say that it is required just once per iteration. The important thing is that all of these distributions calculations grow quadratic in relation with the number of teams.

N = number of teams
U = N*(N-1)/2 = number of matches
M = N + U = number of variables in the network