Graphical Models: Assignment 1

Jesús Herrera Machado, Antonio Remiro Azócar, Nicholas Williams

October 31, 2016

Contributions

jesus.herrera.16@ucl.ac.uk: 2.5, 2.6, 2.7, 3.12, 3.13, 3.17.
antonio.remiro.16@ucl.ac.uk: 3.3, 3.4, 3.8, 3.9, 3.11, 3.21, 3.25.
nicholas.williams.16@ucl.ac.uk: 2.9, 2.10, 3.14, 3.17, 3.21, 3.22, 3.24

The workload for this assignment was split based on our skills and backgrounds. We created the LaTeX document together. Each of us took responsibility for writing down our part of the assignment.

- Jesús Herrera is accountable for the exercises that involved programming and the BRML Toolbox. We are all fairly new to Matlab and found easier using R for one of the exercises (2.7).
- Antonio Remiro Azócar was responsible for a big part of the exercises in Chapter 3.
- Nicholas Williams was responsible for the exercises in chapter 2 that did not involve programming and the remaining Chapter 3 exercises.

It is worth noting that there was constant collaboration throughout the process. We kept face to face meetings to the minimum. We thought that a group of 3 people would be more efficient than one of 5.

Exercise 2.5

In order to retrieve all the ancestors of a group of nodes in a DAG, lets assume we have a function that retrieves the parents of a given set of nodes.

- We can take an iterative or recursive approach to complete this task. We choose the iterative one for simplicity.
- We need two pieces of information to complete this task: the set of nodes whose ancestors we want to retrieve and the adjacency matrix that we will use to find them.
- To start the process we retrieve the parents set (first level of ascendance) for the original set of nodes that we are given to evaluate.
- At this point, there is a possibility that nodes in the original set do not have parents - in which case an empty list is returned. To make sure we capture this case (and hence, know when to finish our process) we make use of a boolean variable, *par*. The value of *par* will be set to true at the beginning of our function.
- We create a while loop which will keep on looping until the boolean variable *par* (which is true if the result of the function parents is not empty for a given set of nodes, false otherwise) becomes false.
- In each iteration we collect the set of parents for each "generation" of nodes. At the end we will have a collection with all the ancestors.
- As mentioned before, we finish our process once variable *par* is set to false. The only way this variable will be set to false is when a "generation" of nodes returns an empty set after calling the parents function. In which case we have collected all the ancestors for our original set of nodes.

Exercise 2.6

Using the property of the matrix power, we calculate A^k for matrix A, where $k = \{1, 2, 3...20\}$. Each bin in the histogram shown in Figure 1 is the count of $[A^k]_{ij} \geq 1$.

```
list = []
for var=1:20
    ret = A^var;
    list = [list sum(ret(:) ≥ 1)]
end
bar(list)
xlabel('Separation (S)')
ylabel('Number of pairs with separation S')
```

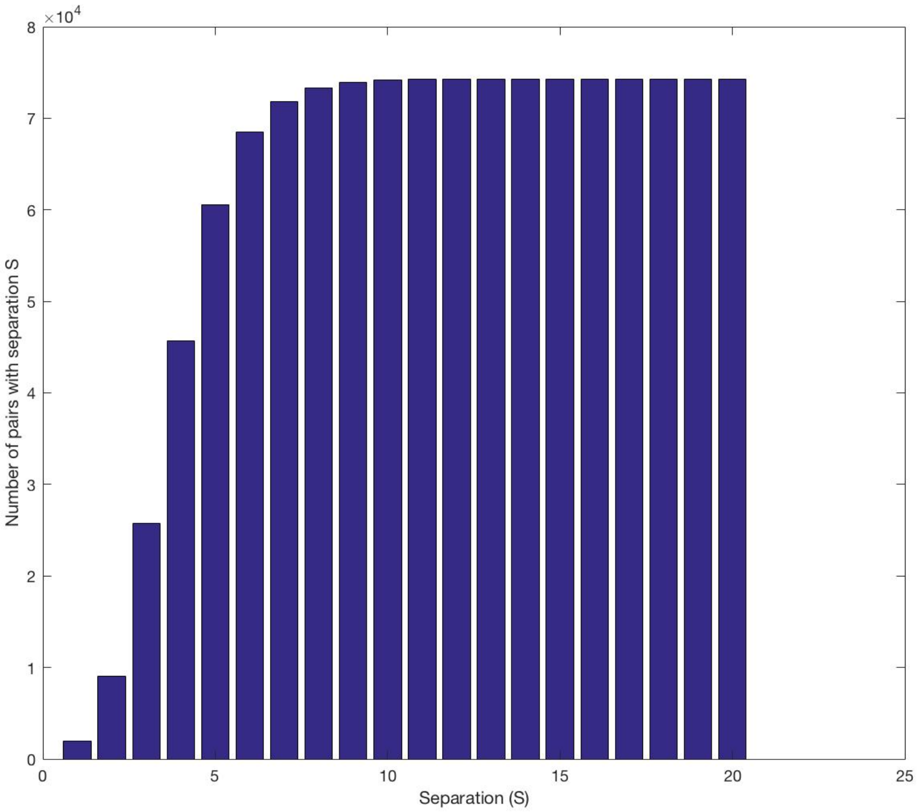


Figure 1: Separation between all users. We can observe that as we increase the separation, the number of pairs of nodes becomes constant. Suggesting that after roughly 10 steps we can reach any node in the graph

Exercise 2.7

Below is the final maximal clique list that can be found in the given set after applying the algorithm:

119, 447, 463, 474, 487, 703, 751, 755, 765, 831, 863, 886, 893, 954, 983, 1006, 1013.

- For this specific problem we are given a list of cliques and our task is to "reduce" this list to only the maximal cliques.

- The approach we take for solving this problem is one of “absorption”. The bigger cliques will “absorb” the smaller ones if all their nodes are contained in the bigger clique.
- Once we have imported our file we can begin the process.
- We create a logical array `absorbed` - initialised to FALSE. This array’s size is 100, which corresponds to the number of cliques given in the data set. The function of this data structure is to keep track of the smaller cliques that are “absorbed” by the bigger ones.
- Once the `absorbed` array is created, we need to set its values - at the end we will have an array which indicates which cliques have been absorbed and which have not (the latter are the ones that will give us the list of maximal cliques).
- To implement this we use two nested for loops. The outer loop will serve to traverse the list of 100 cliques. The inner loop will also traverse such list in order to calculate the union between every pair of cliques.
- When calculating the union between a pair of cliques, A and B, there are several possibilities:
  1. A is equal to B, in which case we do nothing.
  2. The result of the union has the same size as A, which means that A absorbs B. In this case, we set the index corresponding to B in the `absorbed` list to true - given that B belongs to a bigger clique.
  3. The result of the union has the same size as B, which means that B absorbs A. In this case, we set the index corresponding to A in the tt absorbed list to true - given that A belongs to a bigger clique.
- At the end of this process, the list `absorbed` contains information about the cliques that are part of a bigger clique (value is TRUE) and those that are not (value is FALSE). Now we are left with the required calculation for the representation.

```
absorbed = logical(length(data[[1]]))

for(i in 1:length(data_train[[1]])){
  for(j in 1:length(data_train[[1]])){
    if(i==j){
      break
    }
    else{
      if(length(union(unlist(data_train[[1]][[i]]), unlist(data[[1]][[j]]))) == ...
          length(unlist(data_train[[1]][[i]])) || ...
          length(union(unlist(data_train[[1]][[i]]), unlist(data[[1]][[j]]))) == ...
          length(unlist(data_train[[1]][[j]])))
        if(length(union(unlist(data_train[[1]][[i]]), unlist(data[[1]][[j]]))) == ...
            length(unlist(data_train[[1]][[i]]))){
          absorbed[j] = TRUE
        }
        else{
          absorbed[i] = TRUE
        }
    }
  }
}
```

- In order to start the calculation we create a matrix. This matrix will have ten columns (the number of nodes in the graph) and N rows, where N is the number of cliques that were not absorbed in the previous step (their value in the `absorbed` array is false).
- Each of the cliques contains (or not) specific nodes from the graph. To represent this in our matrix, we have set to 1 the columns that correspond to a specific node when that node is part of the clique.
- Now we should have a matrix that has the binary representation of each of the maximal cliques as required in the exercise.

```
mat <- matrix(0L, nrow = length(absorbed[absorbed == FALSE]), ncol = 10)
count <- 0

for(i in which(absorbed == FALSE)){
  count <- count + 1
  for(j in 1:length(unlist(data[[1]][[i]]))){
    mat[count,unlist(data[[1]][[i]])[j]] = 1
  }
}
```

- We are left with one more task: to represent the clique in decimal.
- We use our binary representation matrix to achieve this. To traverse the matrix we use two for loops (nested).
- We create a variable `decimal` which accumulates the calculation at each column (node) for each row (maximal clique). We do this by using the formula given.
- After calculating the decimal representation for each row, we add it to the result list, which will contain the decimal representation for each maximal clique.
- We then sort the list in ascending order.

```
dec_rep <- c()
for(i in 1:nrow(mat)){
  summ <- 0
  for(j in 1:ncol(mat)){
    summ <- summ + (2^(10-j) * mat[i,j])
  }
  dec_rep <- c(dec_rep, summ)
}
sort(dec_rep)
```

Exercise 2.9

Let $N = 3k$ where $k \in \mathbb{N}\backslash\{0\}$. Also, set:

$$G_{nodes} := \{a_i, b_i, c_i : i \in \{1, ..., k\}\}.$$

This means $|G_{nodes}| = 3k = N$. Now, let us define our subsets of nodes S_i:

$$S_i := \{a_i, b_i, c_i\} \text{ for } i \in \{1, ..., k\}.$$

Therefore, $|S_i| = 3$ and $\sum_{i=1}^{k} S_i = \sum_{i=1}^{N/3} S_i = G_{nodes}$, giving $\frac{N}{3}$ disjoint subsets as required. Now, if we define:

$$G_{edges} := \big\{(a_i, a_j), (a_i, b_j), (a_i, c_j), (b_i, b_j), (b_i, c_j), (c_i, c_j)\} : i \neq j, i, j \in \{1, ..., k\}\big\}.$$

This construction satisfies the specified set of conditions. Recall that a clique is a fully connected subset of nodes. Lets denote the set of all cliques in G with C_G. For illustration purposes, consider two examples:

1. $(N = 3,\ k = 1)$:

$$G_{nodes} = S_1 = \{a_1, b_1, c_1\},$$
$$G_{edges} = \emptyset,$$
$$C_G := \big\{\{a\}, \{b\}, \{c\}\big\} \to |C_G| = 3 = 3^{\frac{N}{3}}.$$

2. $(N = 3^2 = 9,\ k = 2)$:

$$G_{nodes} = S_1 \cup S_2 = \{a_1, b_1, c_1\} \cup \{a_2, b_2, c_2\} = \{a_1, b_1, c_1, a_2, b_2, c_2\},$$
$$G_{edges} = \big\{(a_i, a_j), (a_i, b_j), (a_i, c_j), (b_i, b_j), (b_i, c_j), (c_i, c_j)\} : i \neq j, i, j \in \{1, 2\}\big\},$$
$$C_G := \left\{ \begin{matrix} \{a_1, a_2\}, \{a_1, b_2\}, \{a_1, c_2\}, \\ \{b_1, a_2\}, \{b_1, b_2\}, \{b_1, c_2\}, \\ \{c_1, a_2\}, \{c_1, b_2\}, \{c_1, c_2\} \end{matrix} \right\} \to |C_G| = 9 = 3^{\frac{N}{3}}.$$

In general, $|C_G| = 3^k = 3^{\frac{N}{3}}$. This is true since by our construction of G_{edges}, without loss of generality, we have 3 choices for an element in S_1, then 3 choices for an element in S_2, 3 choices for an element in S_3 and finally 3 choices for an element in S_k; with the edge back to the original element in S_1 being fully determined by the definition of a clique.

Exercise 2.10

As exactly one statement is false, the claims supported by two statements must be true.

Statements 1 and 2 support each other in so far as A claims to be with B and B claims to be with A so we know for sure that A and B were in the room together. Similarly, statements 3 and 6 support each other in so far as C claims to be with F and F claims to be with C and so we know for sure C and F were in the room together.

We suspect Statement 4 with D's claim to be in the room with A and F is false. Suppose for a moment that D's claim is true. D being in the room with A and F would suggest that D is in the room with B or E as well. This is because A is in the room with E and B (Statement 1). However, D does not mention this and neither do B or E in their claims.

Therefore Statement 4, where D claims to coincide with A and F, is likely to be false.

Exercise 3.3

(a) There is a path between t and s, $t - e - d - b - s$, whose only collider is d (two incoming arrows from e and b along the path). The path is not blocked since d is in the conditioning set. Hence t and s are d-connected by d and therefore, graphically dependent conditioned on g. $t \perp\!\!\!\perp s | d$ is false.

(b) There is one possible path between l and b via s, $l - s - b$. In this path there are no colliders. Node s, a non-collider with no incoming arrows, is in the conditioning set and blocks the path. Given s, s d-separates l and b and $l \perp\!\!\!\perp b | s$ is true.

(c) There is one possible path between a and s via l, $a - t - e - l - s$. This path is blocked since l, which is in the conditioning set, is not a collider (only one incoming arrow from s). Additionally, e is a collider along this path and is not in the conditioning set. l and e block the path and $a \perp\!\!\!\perp s | l$ is true. Note that in this network there are two potential colliders, namely e (via $a - t - e - l - s$) and d (via $a - t - e - d - b - s$). If these are not in the conditioning set, a and s are d-separated and unconditionally independent.

(d) Again, there are two possible paths between a and s: $a - t - e - l - s$ and $a - t - e - d - b - s$. The path $a - t - e - l - s$ is still blocked since l, which is in the conditioning set, is not a collider (only one incoming arrow from s). Additionally, this path is also blocked by node e, which acts as a collider within it (two incoming arrows from t and l) but is not in the conditioning set. Regarding path $a - t - e - d - b - s$, collider d (two incoming arrows from b and e) is now in the conditioning set and no longer blocks the path. Note that e does not block this path either since it does not act as a collider along it (only one incoming arrow in this path). Therefore, a and s are d-connected by d and and are (graphically) dependent given d, l. $a \perp\!\!\!\perp s | l, d$ is false.

Exercise 3.4

(a) From the graphical dependencies in Figure 3.15,

$$p(d = tr) = p(d = tr|b, e)p(e|l, t)p(b|s)p(l|s)p(t|a)p(s)p(a),$$

Marginal probabilites are obtained by summation over b, e, l, t, a, s.

$$p(d = tr) = \sum_{b,e,l,t,a,s} p(d = tr|b, e)p(e|l, t)p(b|s)p(l|s)p(t|a)p(s)p(a),$$

Substituting the distribution table values into the equation we obtain,

$$p(d = tr) = 0.0000135 + 0.000000675 + 0.0002673 + 0.000013365 + 0.0002565 + 0.000012825+$$

$$0.041899275+0.001323135+0.0001215+0.000066825+0.0024057+0.001323135+0+0+0+0+0+0+ 0+0+0+0+0+0+0+0+0+0+0+0+0.002052+0.0011286+0.2117016+0.11643588+0.000007+ 0.000001225+0.0001386+0.000024255+0.000133+0.000023275+0.0137214+0.002401245+ 0.000063+0.000121275+0.0012474+0.002401245+0+0+0+0+0+0+0+0+0+0+0+0+0+0+ 0.000171+0.000329175+0.0176418+0.033960465,$$

$$p(d=tr)=0.451 \text{ (3 s.f.)}.$$

(b) From the definition of conditional probability,

$$p(d|s=tr)=\frac{p(d,s=tr)}{p(s=tr)}.$$

In the numerator, marginal probabilities are obtained from the joint probability by summation over b,e,l,t,a.

$$p(d|s=tr)=\frac{\sum_{b,e,l,t,a} p(d,s=tr,b,e,l,t,a)}{p(s=tr)}.$$

From the distribution table and the graphical dependencies in the belief network,

$$p(d|s=tr)=\frac{p(s=tr)\sum_{b,e,l,t,a} p(d|e,b)p(e|t,l)p(b|s=tr)p(l|s=tr)p(t|a)p(a)}{p(s=tr)}.$$

Cancelling out $p(s=tr)$ (note it can be taken out of the summation since it remains constant over it),

$$p(d|s=tr)=0.000027+0.0005346+0.000513+0.08379855+0.000243+0.0048114+0+0+0+0+0+ 0+0+0+0+0.004104+0.4234032+0.000014+0.0002772+0.000266+0.0274428+0.000126+0.0024948+ 0+0+0+0+0+0+0+0.000342+0.0352836,$$

$$p(d|s=tr)=0.584 \text{ (3 s.f.)}.$$

(c) From the definition of conditional probability,

$$p(d|s=fa)=\frac{p(d,s=fa)}{p(s=fa)}.$$

In the numerator, marginal probabilities are obtained from the joint probability $p(d,s=fa)$ by summation over b,e,l,t,a.

$$p(d|s=fa)=\frac{\sum_{b,e,l,t,a} p(d,s=fa,b,e,l,t,a)}{p(s=fa)}.$$

From the distribution table and the graphical dependencies in the belief network,

$$p(d|s=fa)=\frac{p(s=fa)\sum_{b,e,l,t,a} p(d|e,b)p(e|t,l)p(b|s=fa)p(l|s=fa)p(t|a)p(a)}{p(s=fa)}.$$

Cancelling out $p(s=tr)$,

$$p(d|s=fa)=0.00000135+0.00002673+0.00002565+0.00264627+0.00013365+0.00264627+0+ 0+0+0+0+0+0+0+0+0.0022572+0.23287176+0.00000245+0.00004851+0.00004655+0.00480249+ 0.00024255+0.00480249+0+0+0+0+0+0+0+0.00065835+0.06792093,$$

$$p(d|s=fa)=0.319 \text{ (3 s.f.)}.$$

Note: In this exercise, the probability associated with each instance of the system has been listed for illustrative purposes. However, one can realise that,

$$
\begin{aligned}
p(d=tr) = &\sum_{b,e,l,t,a,s} p(d=tr|b,e)p(e|l,t)p(b|s)p(l|s)p(t|a)p(s)p(a) \\
&= p(s=tr) \times \sum_{b,e,l,t,a} p(d|e,b)p(e|t,l)p(b|s=tr)p(l|s=tr)p(t|a)p(a) + \\
&\qquad p(s=fa) \times \sum_{b,e,l,t,a} p(d|e,b)p(e|t,l)p(b|s=fa)p(l|s=fa)p(t|a)p(a).
\end{aligned}
$$

The first sum is $p(d|s=tr)$ and the second sum is $p(d|s=fa)$. Since $p(s=tr)=p(s=fa)=0.5$, one can obtain the answer for (a) by summing the halves of answers (b) and (c).

Exercise 3.8

1.

(a) From the definition of conditional probability,

$$p(B=tr|W=tr) = \frac{p(B=tr, W=tr)}{p(W=tr)}.$$

In the numerator, marginal probabilities are obtained from the joint probability $p(B=tr, W=tr)$ by summation over A. In the denominator, marginal probabilites are obtained by summation over B, A.

$$p(B=tr|W=tr) = \frac{\sum_A p(B=tr, A, W=tr)}{\sum_{B,A} p(B, A, W=tr)}.$$

From the graphical dependencies in the belief network,

$$p(B=tr|W=tr) = \frac{p(B=tr)\sum_A p(W=tr|A)p(A|B=tr)}{\sum_{B,A} p(W=tr|A)p(A|B)p(B)}.$$

Substituting in the values from the table entries,

$$p(B=tr|W=tr) = \frac{0.01 \times ((0.9 \times 0.99) + (0.5 \times 0.01))}{(0.9 \times 0.99 \times 0.01) + (0.5 \times 0.01 \times 0.01) + (0.9 \times 0.05 \times 0.99) + (0.5 \times 0.95 \times 0.99)},$$

$$p(B=tr|W=tr) = 0.0171 \text{ (3 s.f.)}.$$

(b) From the definition of conditional probability,

$$p(B=tr|W=tr, G=fa) = \frac{p(B=tr, W=tr, G=fa)}{p(W=tr, G=fa)}.$$

In the numerator, marginal probabilities are obtained from the joint probability $p(B=tr, W=tr, G=fa)$ by summation over A. In the denominator, marginal probabilites are obtained by summation over B, A:

$$p(B=tr|W=tr, G=fa) = \frac{\sum_A p(B=tr, A, W=tr, G=fa)}{\sum_{B,A} p(B, A, W=tr, G=fa)}.$$

From the graphical dependencies in the belief network,

$$p(B=tr|W=tr, G=fa) = \frac{p(B=tr)\sum_A p(G=fa|A)p(W=tr|A)p(A|B=tr)}{\sum_{B,A} p(G=fa|A)p(W=tr|A)p(A|B)p(B)}.$$

Substituting in the values from the table entries,

$$p(B=tr|W=tr, G=fa) = \frac{0.01 \times ((0.3 \times 0.9 \times 0.99) + (0.8 \times 0.5 \times 0.01))}{0.002673 + 0.00004 + 0.0147 + 0.3762}$$

$$p(B=tr|W=tr, G=fa) = 0.00689 \text{ (3 s.f.)}.$$

2. The following soft evidence is introduced:

$$p(G|\tilde{G}) = \begin{cases} 0.1, & G = tr \\ 0.9, & G = fa. \end{cases}$$

$$p(W|\tilde{W}) = \begin{cases} 0.3, & W = tr \\ 0.7, & W = fa. \end{cases}$$

(a) Now we must compute:

$$p(B = tr|\tilde{W}) = p(B = tr|W = tr)p(W = tr|\tilde{W}) + p(B = tr|W = fa)p(W = fa|\tilde{W})$$

From the definition of conditional probability,

$$p(B = tr|\tilde{W}) = \frac{p(W = tr|\tilde{W})p(W = tr, B = tr)}{p(W = tr)} + \frac{p(W = fa|\tilde{W})p(W = fa, B = tr)}{p(W = fa)}.$$

From Jeffrey's rule, one can compute the model conditioned on the evidence. Firstly, marginal probabilites are obtained from joint probabilities by summation:

$$p(B = tr|\tilde{W}) = \frac{p(W = tr|\tilde{W})\sum_A p(W = tr, A, B = tr)}{\sum_{B,A} p(W = tr, A, B)} + \frac{p(W = fa|\tilde{W})\sum_A p(W = fa, A, B = tr)}{\sum_{B,A} p(W = fa, A, B)}.$$

Secondly, the equation is rearranged using the graphical dependencies in the belief network:

$$p(B = tr|\tilde{W}) = \frac{p(B = tr)p(W = tr|\tilde{W})\sum_A p(W = tr|A)p(A|B = tr)}{\sum_{B,A} p(W = tr|A)p(A|B)p(B)} + \frac{p(B = tr)p(W = fa|\tilde{W})\sum_A p(W = fa|A)p(A|B = tr)}{\sum_{B,A} p(W = fa|A)p(A|B)p(B)}$$

Substituting in the values from the table entries,

$$p(B = tr|\tilde{W}) = 0.01\times\Big(\frac{0.3 \times ((0.9 \times 0.99) + (0.5 \times 0.01))}{0.00891 + 0.00005 + 0.04455 + 0.47025} + \frac{0.7 \times ((0.1 \times 0.99) + (0.5 \times 0.01))}{0.00099 + 0.00005 + 0.00495 + 0.47025}\Big),$$

$$p(B = tr|\tilde{W}) = 0.00666 \text{ (3 s.f.)}.$$

(b) Now we must compute:

$$p(B = tr|\tilde{W}, \tilde{G}) = p(B = tr|W = tr, G = tr)p(W = tr|\tilde{W})p(G = tr|\tilde{G}) + p(B = tr|W = tr, G = fa)p(W = tr|\tilde{W})p(G = fa|\tilde{G}) + p(B = tr|W = fa, G = tr)p(W = fa|\tilde{W})p(G = tr|\tilde{G}) + p(B = tr|W = fa, G = fa)p(W = fa|\tilde{W})p(G = fa|\tilde{G})$$

From the definition of conditional probability,

$$p(B = tr|\tilde{W}, \tilde{G}) = \frac{p(W = tr|\tilde{W})p(G = tr|\tilde{G})p(W = tr, B = tr, G = tr)}{p(W = tr, G = tr)} + \frac{p(W = tr|\tilde{W})p(G = fa|\tilde{G})p(W = tr, B = tr, G = fa)}{p(W = tr, G = fa)} + \frac{p(W = fa|\tilde{W})p(G = tr|\tilde{G})p(W = fa, B = tr, G = tr)}{p(W = fa, G = tr)} + \frac{p(W = fa|\tilde{W})p(G = fa|\tilde{G})p(W = fa, B = tr, G = fa)}{p(W = fa, G = fa)}.$$

From Jeffrey's rule, one can compute the model conditioned on the evidence. Firstly, marginal probabilites are obtained from joint probabilities by summation:

$$p(B = tr|\tilde{W}, \tilde{G}) = \frac{p(W = tr|\tilde{W})p(G = tr|\tilde{G})\sum_A p(W = tr, A, B = tr, G = tr)}{\sum_{A,B} p(W = tr, A, B, G = tr)} + \frac{p(W = tr|\tilde{W})p(G = fa|\tilde{G})\sum_A p(W = tr, A, B = tr, G = fa)}{\sum_{A,B} p(W = tr, A, B, G = fa)} +$$

$$\frac{p(W=fa|\tilde{W})p(G=tr|\tilde{G})\sum_A p(W=fa,A,B=tr,G=tr)}{\sum_{A,B} p(W=fa,A,B,G=tr)}+$$
$$\frac{p(W=fa|\tilde{W})p(G=fa|\tilde{G})\sum_A p(W=fa,A,B=tr,G=fa)}{\sum_{A,B} p(W=fa,A,B,G=fa)}.$$

Secondly, the equation is rearranged using the graphical dependencies in the belief network:

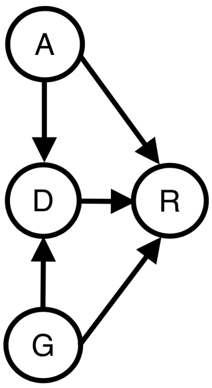
$$p(B=tr|\tilde{W},\tilde{G})=p(B=tr)\Big(\frac{p(W=tr|\tilde{W})p(G=tr|\tilde{G})\sum_A p(W=tr|A)p(G=tr|A)p(A|B=tr)}{\sum_{A,B} p(W=tr|A)p(G=tr|A)p(A|B)P(B)}+$$
$$\frac{p(W=tr|\tilde{W})p(G=fa|\tilde{G})\sum_A p(W=tr|A)p(G=fa|A)p(A|B=tr)}{\sum_{A,B} p(W=tr|A)p(G=fa|A)p(A|B)P(B)}+$$
$$\frac{p(W=fa|\tilde{W})p(G=tr|\tilde{G})\sum_A p(W=fa|A)p(G=tr|A)p(A|B=tr)}{\sum_{A,B} p(W=fa|A)p(G=tr|A)p(A|B)P(B)}+$$
$$\frac{p(W=fa|\tilde{W})p(G=fa|\tilde{G})\sum_A p(W=fa|A)p(G=fa|A)p(A|B=tr)}{\sum_{A,B} p(W=fa|A)p(G=fa|A)p(A|B)P(B)}\Big).$$

Substituting in the values from the table entries,

$$p(B=tr|\tilde{W},\tilde{G})=0.01\times\Big(\frac{0.3\times 0.1\times(0.6237+0.001)}{0.006237+0.031185+0.00001+0.09405}+\frac{0.3\times 0.9\times(0.2673+0.004)}{0.002673+0.013365+0.00004+0.3762}+$$
$$\frac{0.7\times 0.1\times(0.0693+0.001)}{0.000693+0.003465+0.00001+0.09405}+\frac{0.7\times 0.9\times(0.0297+0.004)}{0.000297+0.001485+0.00004+0.3762}\Big),$$
$$p(B=tr|\tilde{W},\tilde{G})=0.00436\text{ (3 s.f.)}.$$

Exercise 3.9

1.



2. From the graphical dependencies in the belief network, a model of the Gender, Age, Drug and Recovery data is,

$$p(G,A,D,R)=p(R|D,A,G)p(D|A,G)p(A)p(G). \tag{1}$$

From the definition of conditional probability,

$$p(R=recover|D=drug)=\frac{p(R=recover,D=drug)}{p(D=drug)}.$$

Upon summation of the absent variables,

$$p(R=recover|D=drug)=\frac{\sum_{A,G} p(R=recover,D=drug,A,G)}{p(D=drug)}.$$

The model in (1) can be substituted into the previous equation to give:

$$p(R=recover|D=drug)=\frac{\sum_{A,G} p(R=recover|D=drug,A,G)p(D=drug|A,G)p(A)P(G)}{p(D=drug)}.$$

Now $p(R = recover|D = drug)$ can be computed provided the correct procedure is followed. The doctor has to undertake a medical trial in which he not only observes the effect of the drug state on recovery, but where the drug state depends on the patient's age and gender.

3. From the definition of conditional probability,

$$p(R = recover|D = do(drug), A = young) = \frac{p(R = recover, D = do(drug), A = young)}{p(D = do(drug), A = young)}.$$

Upon summation of the absent variables,

$$p(R = recover|D = do(drug), A = young) = \frac{\sum_G p(R = recover, D = do(drug), A = young, G)}{\sum_{R,G} p(R, D = drug, A = young, G)}.$$

The model (1) is substituted into the previous equation to give,

$$p(R = recover|D = do(drug), A = young, G) = \frac{\sum_G p(R = recover|D = do(drug), A = young, G)p(D = do(drug)|A = young, G)p(A = young)p(G)}{\sum_{R,G} p(R|D = do(drug), A = young, G)p(D = do(drug)|A = young, G)p(A = young)p(G)}.$$

The *do* operator removes all parental links of variable D when setting $D = drug$. Therefore in the previous equation, $p(D = do(drug)|A = young, G) = p(D = drug)$. Additionally, with the doctor's 'intervention', the variable D puts all its probability in that single state of $D = drug$. Hence $p(R = recover|D = do(drug), A = young, G) = p(R = recover|A = young, G)$ and $p(R|D = do(drug), A = young, G) = p(R|A = young, G)$. We now have,

$$p(R = recover|D = do(drug), A = young) = \frac{\sum_G p(R = recover|A = young, G)p(A = young)p(G)}{\sum_{R,G} p(R|A = young, G)p(A = young)p(G)},$$

where $p(A = young)$ is constant over both summations and can be removed from these and then cancelled out. This gives,

$$p(R = recover|D = do(drug), A = young) = \frac{\sum_G p(R = recover|A = young, G)p(G)}{\sum_{R,G} p(R|A = young, G)p(G)},$$

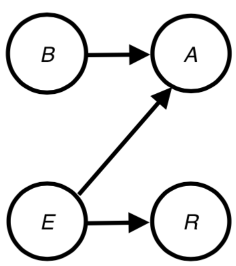
and $p(R = recover|D = do(drug), A = young)$ can be computed. Note that contrary to part (b), the doctor decides to give the drug or not independent of gender and age, with $p(D|A, G)$ playing no role in the procedure.

Exercise 3.11

Example 10 uses Bayes' rule and a number of conditional independence assumptions to present the following model:

$$p(B, E, A, R) = p(A|B, E)p(R|E)p(E)p(B). \tag{2}$$

Graphically,



(2) assumes that a burglary is surely not directly influenced by any earthquake occurring. To model the fact that in LA the probability of being burgled increases if there is an earthquake, B must be graphically dependent on E. We can repeat the procedure in Example 10 but this time $p(B|E)$ must be included in the model. Using Bayes' Rule, we can write, without loss of generality:

$$p(B, E, A, R) = p(A|B, E, R)p(B, E, R).$$

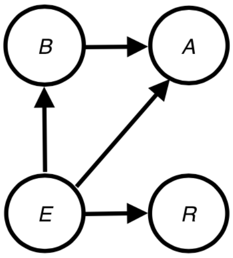
We can repeat this decomposition for $p(B, E, R)$ and $p(B, E)$:

$$p(B, E, A, R) = p(A|B, E, R)p(R|B, E)p(B|E)P(E).$$

This time we write $p(B, E) = p(B|E)P(E)$ instead of $p(B, E) = p(E|B)P(B)$ to account for the graphical dependence of B on E. Note that in Example 10, $p(E|B) = p(E)$ (an earthquake occurrence is surely not directly influenced by any burglary), so no graphical dependencies are lost. All other conditional independence assumptions are kept equal (the alarm is not influenced by any report in the radio and the report in the radio is not influenced by any burglary). The following model is obtained:

$$p(B, E, A, R) = p(A|B, E)p(R|E)p(B|E)p(E).$$

Graphically, this belief network is represented as:



Exercise 3.12

In order to determine if two DAGs, A and B, are Markov-Equivalent we must take into account the following:

1. Whether A and B contain the same vertices (have the same Skeleton - this is a requirement for Markov equivalence). This can be determined from the size of the matrix. If both A and B have the same dimensions, we assume they have the same set of vertices.

```
function adj_mat_und = Skeleton(adj_mat)
% Skeleton: this function transforms a DAG into an Undirected Graph
% The reason to do this is to being able to detemine the "Skeleton" between
% two different graphs
auxs = size(adj_mat);
adj_mat_und = adj_mat;
for i=1:auxs(1)
    for j=1:auxs(2)
        if adj_mat_und(i,j) == 1
            adj_mat_und(j,i) = 1;
        end
    end
end

adj_mat_und;

end
```

2. There is an edge between node x and node y in the graph A if and only if there is an edge between node x and node y in the graph B. We achieve this by removing the "directed" from the DAG and comparing the resulting adjacency matrices.

3. A and B have the same immoralities - this is a requirement for Markov equivalence. We have created an extra function for determining the immoralities of a given Graph. The implementation is naive. I check those nodes in the graph that have 2 or more parents. Then I iteratively check the relationship between its parents. If we have $A \rightarrow C \leftarrow B$ I assume that A and B are the parents. To determine if there is an immorality, we check the parents' respective list of neighbours. If they are not each others' neighbours, then there is an immorality. We iterate to find all the immoralities in a graph.

```
function immoral = Immoralities(adj_mat)
import brml.*

g1 = adj_mat;
s1 = size(g1);

immoral = {};

for i=1:s1(2)
    pars = parents(g1, i);
    sp = size(pars);
    if sp(2) < 2
        continue
    else
        for j=1:sp(2)
            for k=j:sp(2)
                if pars(k) == pars(j)
                    continue
                else
                    if(¬ismember(pars(j), neigh(g1, pars(k))))
                        immoral(end+1,:) = {[num2str(pars(j)) num2str(i) ...
                            num2str(pars(k))]};
                    end
                end
            end
        end
    end
end
end
```

Finally, we make use of all the conditions above to create a function that determines if two DAGs are Markov Equivalent.

```
function [markov_equivalent_yn] = MarkovEquiv(A,B)

markov_equivalent_yn = isequal(size(A),size(B)) && isequal(Skeleton(A), Skeleton(B)) ...
    && isempty(setdiff(Immoralities(A),Immoralities(B)));

end
```

Exercise 3.13

Based on the logic in Exercise 3.12 and using the code created, the result is that, indeed, the BNs A and B are Markov Equivalent.

```
import brml.*

A = [0 0 1 1 0 1 0 0 0;0 0 1 0 1 0 0 0 0;0 0 0 0 0 0 1 0 0;...
    0 0 0 0 0 0 0 1 1;0 0 1 0 0 0 1 0 0;0 0 0 1 0 0 0 1 0;...
    0 0 0 0 0 0 0 0 1;0 0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0 0];
B = [0 0 1 1 0 0 0 0 0;0 0 1 0 0 0 0 0 0;0 0 0 0 0 0 1 0 0;...
    0 0 0 0 0 0 0 1 1;0 1 1 0 0 0 1 0 0;1 0 0 1 0 0 0 1 0;...
    0 0 0 0 0 0 0 0 1;0 0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0 0];

MarkovEquiv(A, B)
```

The screen-shot below corresponds to the evaluation of adjacency matrices A and B with the `MarkovEquiv` function.

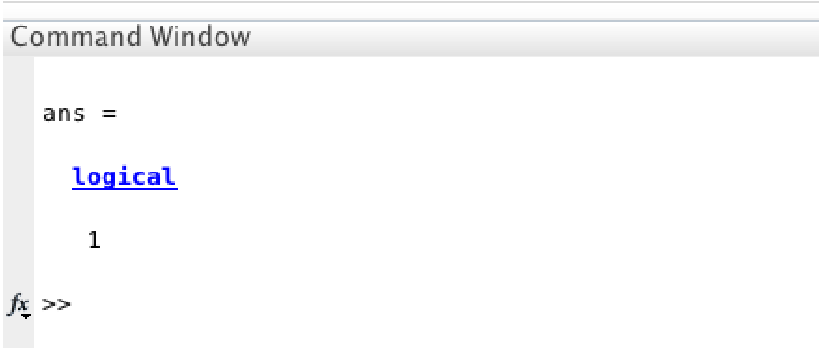
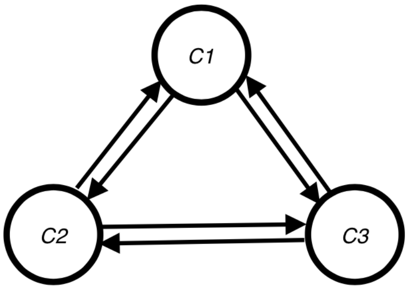


Figure 2: The Belief Networks A and B are Markov Equivalent.

Exercise 3.14

The figure below illustrates the connections in the computer network.



The communication matrix $\mathbf{C}$ can be expressed as follows:

$$\mathbf{C} = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix} = \begin{pmatrix} 1 & C_{12} & C_{13} \\ C_{21} & 1 & C_{23} \\ C_{31} & C_{32} & 1 \end{pmatrix},$$

since it is assumed $C_{ii} = 1$. Note that in $\mathbf{C}$, $C_{ij} \equiv C_{ij}(1)$. It has been assumed that a message sent directly from one computer to another can only take one timestep. We are provided with the test information $\mathcal{C} = \{C_{12}(2) = 1, C_{23}(2) = 0\}$:

$$C_{12}(2) = 1 \Leftrightarrow C_{12} = 1 \cup (C_{13} = 1 \cap C_{32} = 1). \tag{3}$$

$$C_{23}(2) = 0 \Leftrightarrow C_{23}(1) = C_{23} = 0 \text{ and } (C_{21} = 0 \cup C_{13} = 0), \tag{4}$$

Therefore, using (4) and (5), we have:

$$\begin{aligned} \mathcal{C} &= \{C_{12} = 1 \cup (C_{13} = 1 \cap C_{32} = 1), C_{23} = 0, (C_{21} = 0 \cup C_{13} = 0)\} \\ &= \{C_{12} = 1 \cup C_{13} = C_{32} = 1, C_{23} = 0, C_{21} = 0 \cup C_{13} = 0\} \end{aligned}$$

Let us look to calculate $p(\mathcal{C})$. By the independence of connections this must decompose as:

$$p(\mathcal{C}) = p\left(C_{12} = 1 \cup C_{13} = 1 \cap C_{32} = 1\right) p\left(C_{23} = 0\right) p\left(C_{21} = 0 \cup C_{13} = 0\right)$$

Since a priori, $p(C_{ij} = 1, i \neq j) = 0.1$, we can compute:

$$
\begin{aligned}
p\left(C_{12}=1 \cup C_{13}=1 \cap C_{32}=1\right) &= p\left(C_{12}=1\right)+p\left(C_{13}=1 \cap C_{32}=1\right)-p\left(C_{12}=C_{13}=C_{32}=1\right) \\
&= 0.1+0.1^2-0.1^3 \\
&= 0.109 \\
p\left(C_{23}=0\right) &= 1-p\left(C_{23}=1\right) \\
&= 1-0.1 \\
&= 0.9 \\
p\left(C_{21}=0 \cup C_{13}=0\right) &= p\left(C_{21}=0\right)+p\left(C_{13}=0\right)-p\left(C_{21}=C_{13}=0\right) \\
&= 0.9+0.9-0.9^2 \\
&= 0.99
\end{aligned}
$$

Together, this gives:

$$
p(\mathcal{C}) = 0.109 \times 0.9 \times 0.99 = 0.0972 \text{ (3 s.f.)}.
$$

Now the a posteriori probabilities can be computed[1]

$$
\begin{aligned}
p(C_{12}=1|\mathcal{C}) &= \frac{p(C_{12}=1,\mathcal{C})}{p(\mathcal{C})} \\
&= \frac{p(C_{12}=1 \cap C_{23}=0 \cap (C_{21}=0 \cup C_{13}=0))}{p(\mathcal{C})} \\
&= \frac{0.1 \times 0.9 \times 0.99}{0.097119} \\
&= \frac{100}{109} = 0.917 \text{ (3 s.f.)}.
\end{aligned}
$$

$$
\begin{aligned}
p(C_{13}=1|\mathcal{C}) &= \frac{p(C_{13}=1,\mathcal{C})}{p(\mathcal{C})} \\
&= \frac{p(C_{13}=1 \cap C_{32}=1 \cap C_{23}=0 \cap C_{21}=0)}{p(\mathcal{C})} \\
&= \frac{0.1^2 \times 0.9^2}{0.097119} \\
&= \frac{100}{1199} = 0.0834 \text{ (3 s.f.)} \\
p(C_{23}=1|\mathcal{C}) &= \frac{p(C_{23}=1,\mathcal{C})}{p(\mathcal{C})} \\
&= \frac{0}{p(\mathcal{C})}, \text{ since } p(C_{23}=1) \cap p(C_{23}=0)=0 \\
&= 0
\end{aligned}
$$

Similarly,

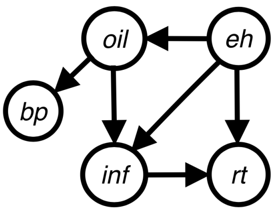
$$
\begin{aligned}
p(C_{32}=1|\mathcal{C}) &= \frac{p(C_{32}=1,\mathcal{C})}{p(\mathcal{C})} \\
&= \frac{p(C_{32}=1 \cap C_{13}=1 \cap C_{23}=0 \cap (C_{21}=0 \cup C_{13}=0))}{p(\mathcal{C})} \\
&= \frac{0.1^2 \times 0.9 \times 0.99}{0.097119} \\
&= \frac{10}{109}
\end{aligned}
$$

[1] Note how set theory is used to calculate the probabilities. For example, for $p(C_{12}=1|\mathcal{C})$, $p(C_{12}=1,\mathcal{C}) = p(C_{12}=1 \cap ((C_{12}=1 \cup C_{13}=C_{32}=1) \cap (C_{23}=0) \cap (C_{21}=0 \cup C_{13}=0))$. Since $A = A \cap (A \cup B)$, $p(C_{12}=1,\mathcal{C}) = p(C_{12}=1 \cap C_{23}=0 \cap (C_{21}=0 \cup C_{13}=0))$.

$$
\begin{aligned}
p(C_{21}=1|\mathcal{C}) &= \frac{p(C_{21}=1,\mathcal{C})}{p(\mathcal{C})} \\
&= \frac{p(C_{21}=1 \cap (C_{12}=1 \cup C_{13}=1 \cap C_{32}=1) \cap C_{23}=0 \cap C_{13}=0)}{p(\mathcal{C})} \\
&= \frac{0.1 \times 0.109 \times 0.9^2}{0.097119} \\
&= \frac{1}{11} \\
p(C_{31}=1|\mathcal{C}) &= \frac{p(C_{31}=1,\mathcal{C})}{p(\mathcal{C})} \\
&= \frac{p(C_{31}=1)p(\mathcal{C})}{p(\mathcal{C})} \\
&= \frac{1}{10} = 0.1
\end{aligned}
$$

Exercise 3.15

(a)



(b) We have to find $p(inf = h|bp = n, rt = h)$. Note h represents *high*, n represents *normal* and l represents low. Using the definition of conditional probability,

$$p(inf = h|bp = n, rt = h) = \frac{p(inf = h, bp = n, rt = h)}{p(bp = n, rt = h)}.$$

Marginal probabilities are obtained from the joint probabilities by summation over eh and oil in the numerator and eh, oil and inf in the denominator,

$$p(inf = h|bp = n, rt = h) = \frac{\sum_{eh,oil} p(inf = h, bp = n, rt = h, eh, oil)}{\sum_{eh,oil,inf} p(bp = n, rt = h, eh, oil, inf)}.$$

From the distribution table and graphical dependencies in the belief network,

$$p(inf = h|bp = n, rt = h) = \frac{\sum_{eh,oil} p(bp = n|oil)p(oil|eh)p(rt = h|eh, inf = h)p(inf = h|oil, eh)p(eh)}{\sum_{eh,oil,inf} p(bp = n|oil)p(oil|eh)p(rt = h|eh, inf)p(inf|oil, eh)p(eh)}$$

Substituting in the values from the distribution table,

$$p(inf = h|bp = n, rt = h) = \frac{0.2979504 + 0.003564 + 0.00648 + 0.00162}{0.2979504 + 0.002736 + 0.003564 + 0.00036 + 0.00648 + 0.00008 + 0.00162 + 0.00162},$$

$$p(inf = h|bp = n, rt = h) = 0.985 \text{ (3 s.f.)}.$$

Exercise 3.17

1. Consider the following:

$$p(a, c) = \sum_b p(a, b, c) = \sum_b p(c|b)p(b|a)p(a) = p(a) \sum_b p(c|b)p(b|a).$$

To prove independence, we need to show $p(a,c) = p(a)p(c)$. As illustrated above, this is equivalent to showing $p(c) = \sum_b p(c|b)p(b|a)$. Since $p(b|a) = \begin{pmatrix} \frac{1}{4} & \frac{15}{40} \\ \frac{1}{12} & \frac{1}{8} \\ \frac{2}{3} & \frac{1}{2} \end{pmatrix}$ and $p(a) = \begin{pmatrix} \frac{3}{5} \\ \frac{2}{5} \end{pmatrix}$,

$$p(a,b) = p(b|a)p(a) = \begin{pmatrix} \frac{1}{4}\times\frac{3}{5} & \frac{15}{40}\times\frac{2}{5} \\ \frac{1}{12}\times\frac{3}{5} & \frac{1}{8}\times\frac{2}{5} \\ \frac{2}{3}\times\frac{3}{5} & \frac{1}{2}\times\frac{2}{5} \end{pmatrix} = \begin{pmatrix} \frac{3}{20} & \frac{3}{20} \\ \frac{1}{20} & \frac{1}{20} \\ \frac{2}{5} & \frac{1}{5} \end{pmatrix}.$$

Summing the rows gives $p(b) = \begin{pmatrix} \frac{3}{10} \\ \frac{1}{10} \\ \frac{3}{5} \end{pmatrix}$.

Since $p(c|b) = \begin{pmatrix} \frac{1}{3} & \frac{1}{2} & \frac{15}{40} \\ \frac{2}{3} & \frac{1}{2} & \frac{5}{8} \end{pmatrix}$ and $p(b) = \begin{pmatrix} \frac{3}{10} \\ \frac{1}{10} \\ \frac{3}{5} \end{pmatrix}$,

$$p(b,c) = p(c|b)p(b) = \begin{pmatrix} \frac{1}{3}\times\frac{3}{10} & \frac{1}{2}\times\frac{1}{10} & \frac{15}{40}\times\frac{3}{5} \\ \frac{2}{3}\times\frac{3}{10} & \frac{1}{2}\times\frac{1}{10} & \frac{5}{8}\times\frac{3}{5} \end{pmatrix} = \begin{pmatrix} \frac{1}{10} & \frac{1}{20} & \frac{9}{40} \\ \frac{2}{10} & \frac{1}{20} & \frac{3}{8} \end{pmatrix}.$$

Summing the rows gives $p(c) = \begin{pmatrix} \frac{3}{8} \\ \frac{5}{8} \end{pmatrix}$. We now substitute the values obtained into $\sum_b p(c|b)p(b|a)$:

$$\sum_b p(c|b)p(b|a) = \sum_b \begin{pmatrix} \frac{1}{3} & \frac{1}{2} & \frac{15}{40} \\ \frac{2}{3} & \frac{1}{2} & \frac{5}{8} \end{pmatrix} \begin{pmatrix} \frac{1}{4} & \frac{15}{40} \\ \frac{1}{12} & \frac{1}{8} \\ \frac{2}{3} & \frac{1}{2} \end{pmatrix} = \sum_b \begin{pmatrix} \frac{3}{8} & \frac{3}{8} \\ \frac{5}{8} & \frac{5}{8} \end{pmatrix} = \begin{pmatrix} \frac{3}{8} \\ \frac{5}{8} \end{pmatrix} = p(c).$$

2. $p(a,c)$ can be transformed into a marginal probability by summation over b,

$$p(a=i, c=k) = \sum_b p(a=i, b, c=k).$$

Since $p(a,b,c) = \frac{1}{Z}\phi(a,b)\psi(b,c)$,

$$\sum_b p(a=i,b,c=k) = \sum_b \frac{1}{Z}\phi(a=i,b)\psi(b,c=k) = \frac{1}{Z}\sum_b \phi(a=i,b)\psi(b,c=k).$$

Since $M_{ij} = \phi(a=i, b=j)$ and $N_{kj} = \psi(b=j, c=k)$,

$$\frac{1}{Z}\sum_b \phi(a=i,b)\psi(b,c=k) = \frac{1}{Z}\sum_b M_{ib}N_{bk} = \frac{1}{Z}[\mathbf{M}\mathbf{N}^T]_{ik}.$$

3. If $\mathbf{M}\mathbf{N}^T = \mathbf{m}_0\mathbf{n}_0^T$, by part (2.):

$$p(a=i,c=k) = \frac{1}{Z}[\mathbf{M}\mathbf{N}^T]_{ik} = \frac{1}{Z}[\mathbf{m}_o\mathbf{n}_o^T]_{ik} \text{ , by premise.}$$

Then,

$$\begin{aligned} p(a=i,c=k) &= \frac{1}{Z}[(m_{0,1}...m_{0,d}) \begin{pmatrix} n_{0,1} \\ \vdots \\ n_{0,d} \end{pmatrix}]_{ik}, \\ &= \frac{1}{Z}\Big(\phi(a=i,b=1)\psi(b=1,c=k)\Big). \end{aligned}$$

Lets define $\phi' = \phi(b=1)$ and $\psi' = \psi(b=1)$. Then,

$$p(a=i,c=k) = \frac{1}{Z}\phi'(a)\psi'(c)$$

In the last equation, ϕ' and ψ' are only dependent on a and c respectively. Then, as required, they are disjoint and $a \perp\!\!\!\perp c$.

4. Writing $\mathbf{M} = [\mathbf{m}_1 \quad \mathbf{m}_2 \quad \mathbf{m}_3]$ and $\mathbf{N} = [\mathbf{n}_1 \quad \mathbf{n}_2 \quad \mathbf{n}_3]$ for two dimensional vectors $\mathbf{m}_i$,$\mathbf{n}_i$, $i = 1, ..., 3$.

$$\begin{aligned}
\mathbf{M}\mathbf{N}^t &= [\mathbf{m}_1 \ \mathbf{m}_2 \ \mathbf{m}_3][\mathbf{n}_1 \ \mathbf{n}_2 \ \mathbf{n}_3]^T \\
&= [\mathbf{m}_1 \ \mathbf{m}_2 \ \mathbf{m}_3] \begin{bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \mathbf{n}_3 \end{bmatrix} \\
&= \mathbf{m}_1\mathbf{n}_1^T + \mathbf{m}_2\mathbf{n}_2^T + \mathbf{m}_3\mathbf{n}_3^T
\end{aligned}$$

where the final transposition necessary due to the $\mathbf{m}_i$ and $\mathbf{n}_i$ being two dimensional vectors.

5.

$$\begin{aligned}
MN^T &= \mathbf{m}_1\mathbf{n}_1^T + \mathbf{m}_2\mathbf{n}_2^T + \mathbf{m}_3\mathbf{n}_3^T \\
&= \mathbf{m}_1\mathbf{n}_1^T + (\lambda\mathbf{m}_1)\mathbf{n}_2^T + \mathbf{m}_3(\gamma(\mathbf{n}_1 + \lambda\mathbf{n}_2))^T \\
&= \mathbf{m}_1\mathbf{n}_1^T + \lambda\mathbf{m}_1 n_2^T + \gamma\mathbf{m}_3 n_1^T + \gamma\lambda\mathbf{m}_3\mathbf{n}_2^T \\
&= \mathbf{m}_1(\mathbf{n}_1^T + \lambda\mathbf{n}_2^T) + \gamma\mathbf{m}_3(\mathbf{n}_1^T + \lambda\mathbf{n}_2^T) \\
&= (\mathbf{m}_1 + \gamma\mathbf{m}_3)(\mathbf{n}_1^T + \lambda\mathbf{n}_2^T) \\
&= \mathbf{m}_0\mathbf{n}_0
\end{aligned}$$

6. Using the vector construction, as per part 4 above, and selecting random entries we are able to provide probability values for the relevant probability decomposition matrices, given in part 1, as follows:

$$p(a) = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \ p(b|a) = \begin{pmatrix} \frac{2}{8} & \frac{2}{6} \\ \frac{3}{8} & \frac{3}{6} \\ \frac{3}{8} & \frac{2}{6} \end{pmatrix} \ p(b|a) = \begin{pmatrix} \frac{2}{4} & \frac{5}{10} & \frac{2}{32} \\ \frac{2}{4} & \frac{5}{10} & \frac{30}{32} \end{pmatrix}$$

```
import brml.*

pot_assignment = {brml.array(1, [1/2 1/2]), brml.array([1 2], [2/8 2/6; 3/8 3/6; ...
    3/8 2/6]), brml.array([2 3], [2/4 5/10 2/32; 2/4 5/10 30/32])}
pot_assignment=setpotclass(pot_assignment,'array');
A=dag(pot_assignment); drawNet(A);

X=[1]; Y=[3]; Z=[2];

if condindep(A,X,Y,Z); % test if X is indep of Y given Z in the graph A
    fprintf(1,'[%s] is indep of [%s] Given [%s]\n',num2str(X),num2str(Y),num2str(Z));
else
    fprintf(1,'[%s] is NOT indep of [%s] Given ...
        [%s]\n',num2str(X),num2str(Y),num2str(Z));
end
```

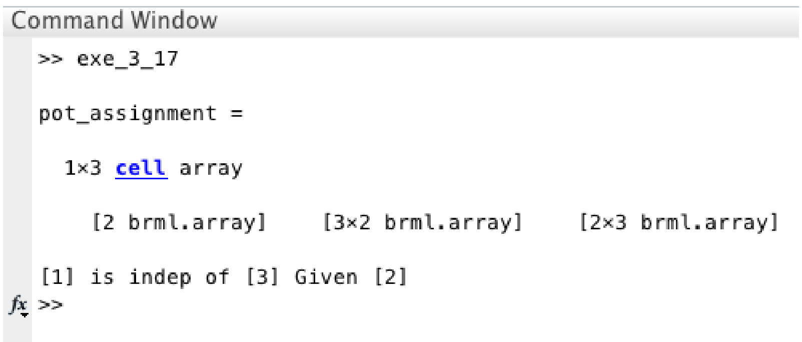


Figure 3: Output from the demo using the BRMLtoolbox

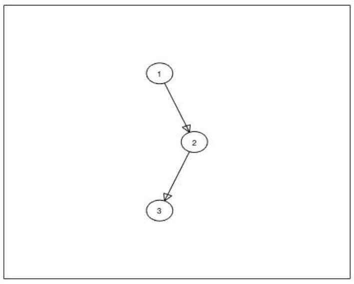
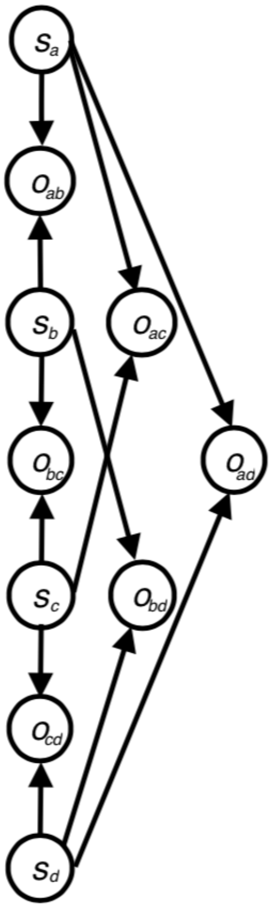


Figure 4: Graph representation from the BRMLtoolbox

Exercise 3.21

1. In the figure below, nodes labelled with an s represent the skill level of player a,b,c or d. Nodes labelled with an o represent outcomes of the game; o_{ab} is the outcome of the game between a and b, o_{ac} is the outcome of the game between a and c, etc.



2. The skill levels of the players a posteriori are independent. In the figure above, all game outcomes are colliders (have two incoming arrows) in the paths joining the players' skill levels. If the outcomes are known a posteriori and are therefore in the conditioning set, the paths between the skill levels are blocked. Skill levels are d-separated by the outcomes and hence independent given game outcomes.

3. We will use the following notation: $o_{ad} = 1$ represents Player a beating Player d and $o_{ad} = 0$ represents Player d beating Player a. From the information given in the question,

$$p(o_{ad} = 1) = \frac{1}{1 + e^{S_d - S_a}}.$$

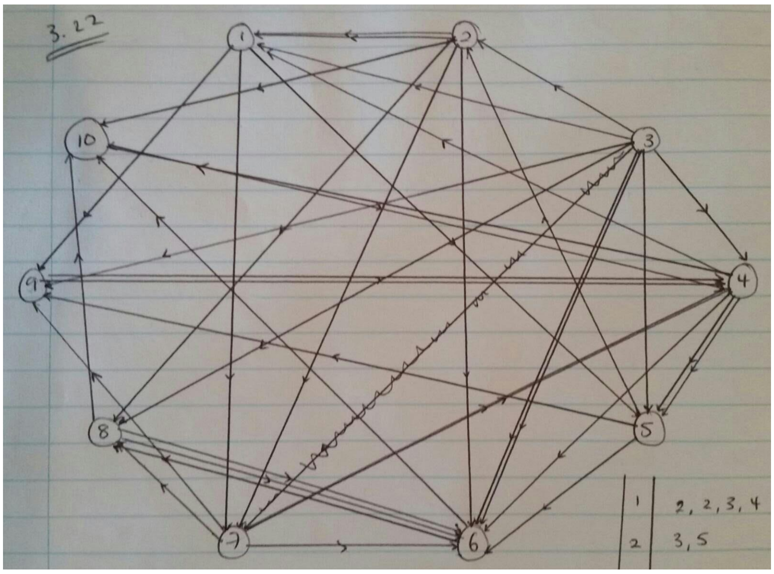
4. Let $p(X > Y) = \alpha$ denote the probability that X beats Y. Then:

$$\alpha = \frac{1}{1 + e^{s_Y - s_C}} \iff 1 + e^{s_Y - s_X} = \frac{1}{\alpha}$$
$$\iff e^{s_Y - s_X} = \frac{1}{\alpha} - 1$$
$$\iff s_Y - s_X = \ln(\alpha^{-1} - 1).$$

We then have $\binom{4}{2} = 6$ linear equations (as $p(Y > X) = 1 - \alpha$ yields the same as the above) expressed in terms of our 4 unknown variables s_a, s_b, s_c, s_d which can be solved simultaneously to give the required skill levels.

Exercise 3.22

We take a heuristic approach to solve this question. The graph below shows which adverts are clicked on in preference to others . For example, from the question we can tell that 1 was clicked on in preference to 2 (twice), 3 and 4. Hence on the graph , there are edges from 2 (two),3,4 pointing towards 1.



Let $s_i \in \{1, ...5\}$ denote the interest level for advert $i \in \{1, ..., 10\}$. We can then state that:

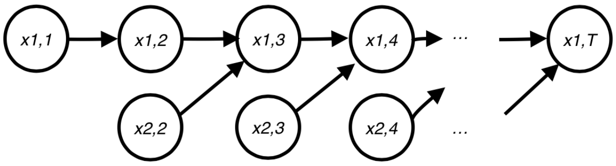
$s_6 = 5$
$s_8 = 5$
$s_3 = 1$
$s_4 = 4$
s_2 or $s_7 = 1$
s_7 or $s_2 = 2$
s_4 or $s_{10} = 4$
s_10 or $s_4 = 3$

For the remaining candidates it is clear from the belief network representing the historical data we have a number of possibilities which can be summarized as follows:

Interest level s	1	2	3	4	5
Advert	3	2/7	10/9	4	6
	7/2	1/5	5/1	9/10	8

Exercise 3.23

1. The moves of player1 can be represented with the following graph:



Exercise 3.24

1. We need to show

$$p(x_1|y_1)p(y_1)p(x_2|x_1,y_2)p(y_2)p(x_3|x_2,y_3)p(y_3) = p(x_1,y_1)p(x_2,y_2|x_1,y_1)p(x_3,y_3|x_2,y_2),$$

From the definition of conditional probability, $p(x_1,y_1) = p(x_1|y_1)p(y_1)$. Then for the LHS,

$$p(x_1|y_1)p(y_1)p(x_2|x_1,y_2)p(y_2)p(x_3|x_2,y_3)p(y_3) = p(x_1,y_1)p(x_2|x_1,y_2)p(y_2)p(x_3|x_2,y_3)p(y_3).$$

Using the chain rule for probability for the RHS,

$$p(x_1,y_1)p(x_2,y_2|x_1,y_1)p(x_3,y_3|x_2,y_2) = p(x_1,y_1)p(x_2|x_1,y_1,y_2)p(y_2)p(x_3|x_2,y_2,y_3)p(y_3).$$

Upon inspection of belief network p_1, $x_2 \perp\!\!\!\perp y_1|x_1$ and $x_3 \perp\!\!\!\perp y_2|x_2$ (eg. when x_1 is in the conditioning set, it is a non-collider in the path between y_1 and x_2). Hence the RHS can be written as the LHS,

$$p(x_1,y_1)p(x_2|x_1,y_1,y_2)p(y_2)p(x_3|x_2,y_2,y_3)p(y_3) = p(x_1,y_1)p(x_2|x_1,y_2)p(y_2)p(x_3|x_2,y_3)p(y_3),$$

and,

$$p_1(x_1,y_1,x_2,y_2,x_3,y_3) = p(x_1,y_1)p(x_2,y_2|x_1,y_1)p(x_3,y_3|x_2,y_2).$$

2. Recall $p(a|b) = \frac{p(a,b)}{p(b)} = \frac{p(b|a)p(a)}{p(b)} \Rightarrow p(a|b)p(b) = p(b|a)p(a)$. Then,

$$p(x_1,y_1)p(x_2,y_2|x_1,y_1)p(x_3,y_3|x_2,y_2) = p(x_1,y_1|x_2,y_2)p(x_2,y_2)p(x_3,y_3|x_2,y_2).$$

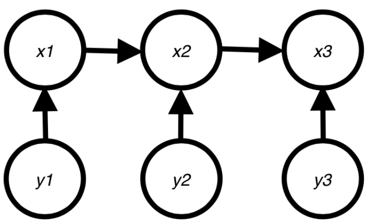
Similarly,

$$p(x_1,y_1|x_2,y_2)p(x_2,y_2)p(x_3,y_3|x_2,y_2) = p(x_1,y_1|x_2,y_2)p(x_2,y_2|x_3,y_3)p(x_3,y_3),$$

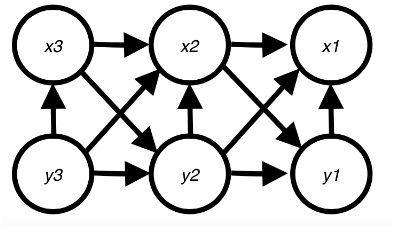
and distribution p_1 can be written as

$$p_2(x_1,y_1,x_2,y_2,x_3,y_3) = p(x_3,y_3)p(x_2,y_2|x_3,y_3)p(x_1,y_1|x_2,y_2).$$

3. The belief network for distribution p_1 is shown below and has 5 edges.

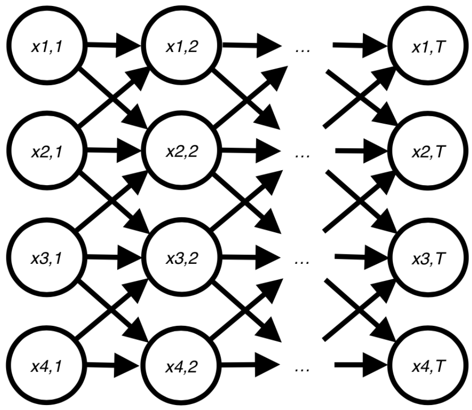


The belief network for distribution p_2 is shown below and has 11 edges. $11 > 5$ and the number of edges in p_2 is greater than that in p_1. Note that for p_2 the independence of the causes is graphically lost; the direction of the arrows between x_3 and y_3, x_2 and y_2 and x_1 and y_1 has been chosen arbitrarily .



Exercise 3.25

1.



2. This question can be answered in similar fashion to **3.24b**. The dynamics of the physical system can be encoded as

$$p(\mathbf{x}_{1:T}) = p(\mathbf{x}_1)\prod_{t=2}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}),$$

i.e.

$$p(\mathbf{x}_{1:T}) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_3|\mathbf{x}_2)...p(\mathbf{x}_T|\mathbf{x}_{T-1}).$$

Rearranging Bayes' Theorem , $p(A|B) = \frac{p(B|A)p(A)}{p(B)}$, to $p(A|B)p(B) = p(B|A)p(A)$,

$$p(\mathbf{x}_{1:T}) = p(\mathbf{x}_1|\mathbf{x}_2)p(\mathbf{x}_2)p(\mathbf{x}_3|\mathbf{x}_2)...p(\mathbf{x}_T|\mathbf{x}_{T-1}),$$

$$p(\mathbf{x}_{1:T}) = p(\mathbf{x}_1|\mathbf{x}_2)p(\mathbf{x}_2|\mathbf{x}_3)p(\mathbf{x}_3)...p(\mathbf{x}_T|\mathbf{x}_{T-1}),$$

$$\cdot\ \cdot\ \cdot$$

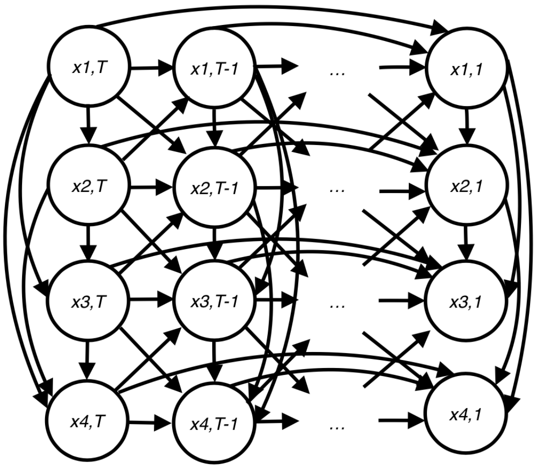
$$p(\mathbf{x}_{1:T}) = p(\mathbf{x}_1|\mathbf{x}_2)p(\mathbf{x}_2|\mathbf{x}_3)...p(\mathbf{x}_{T-1}|\mathbf{x}_T)p(\mathbf{x}_T).$$

Hence $p(\mathbf{x}_{1:T})$ can written in time-reversed form,

$$p(\mathbf{x}_{1:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T-1} p(\mathbf{x}_t|\mathbf{x}_{t+1}).$$

A belief network is drawn below for this alternative representation of $p(\mathbf{x}_{1:T})$
Note that for this belief network causal independence has been graphically lost. The direction of arrows has been chosen arbitrarily.

3. As seen in the belief network drawn for $p(\mathbf{x}_{1:T})$ in (1.), spatial variables $x_{1,t}$ and $x_{4,t}$ (at the ends of the network) only have two "nearest" neighbours at $t+1$. On the other hand, spatial variables $x_{2,t}$ and $x_{3,t}$ have three. Hence for a given timestep, x_1 and x_4 contribute two directed edges and x_2 and x_3 contribute three. Since there are $T-1$ timesteps between $t=1$ and $t=T$, the 'causal' representation has $(2+2+3+3)\times(T-1) = 10\times(T-1)$ edges.



Note that in the belief network drawn for $p(\mathbf{x}_{1:T})$ in (2.), some arrows have been omitted for simplicity. In the time-reversed form, spatial variables at a given time t are graphically dependent on **all** their (spatially) "nearest" neighbours whose time variable is greater than t. With spatially "nearest" we mean $x_{1,t-1}$ and $x_{2,t-1}$ for $x_{1,t}$, $x_{1,t-1}$, $x_{2,t-1}$ and $x_{3,t-1}$ for $x_{2,t}$, etc. We know from looking at the belief network that 6 edges will be contributed between variables that are at the same time instance. For T time instances, that is a total of $6 \times T$. We also know from the previous part of the question that those connections which are shared with the causal representation will contribute a total of $10 \times (T-1)$ edges. Now there are additional edges we have to take into account: these are the edges between "nearest" spatial variables separated by more than one time instance. These contribute a total of $6(T-1)$ additional edges(x_2 and x_3 contribute $2(T-1)$ each, x_1 and x_4 contribute $(T-1)$ each). Therefore, the total number of edges sums to:

$$\begin{aligned} 10(T-1) + 6T + 6(T-1) &= 10T - 10 + 6T + 6T - 6 \\ &= 22 \times T - 16. \end{aligned}$$