Problem Set 1 Solutions

Problem 1

1.

We can find that:

$P(A = 0) = 1/4 + 1/4 + 1/24 + 1/8 = 2/3$
$P(A = 1) = 1/8 + 1/8 + 1/48 + 1/16 = 1/3$
$P(B = 0) = 1/4 + 1/4 + 1/8 + 1/8 = 3/4$
$P(B = 1) = 1/24 + 1/8 + 1/48 + 1/16 = 1/4$
$P(C = 0) = 1/4 + 1/24 + 1/8 + 1/48 = 7/16$
$P(C = 1) = 1/4 + 1/8 + 1/8 + 1/16 = 9/16$

$P(A = 0, B = 0) = 1/4 + 1/4 = 1/2 = P(A = 0) * P(B = 0)$
$P(A = 0, B = 1) = 1/24 + 1/8 = 1/6 = P(A = 0) * P(B = 1)$
$P(A = 1, B = 0) = 1/8 + 1/8 = 1/4 = P(A = 1) * P(B = 0)$
$P(A = 1, B = 1) = 1/48 + 1/16 = 1/12 = P(A = 1) * P(B = 1)$

Because $P(A, B) = P(A)*P(B)$ for all A value and B value. So A and B are independent.

$P(A = 0, C = 0) = 1/4 + 1/24 = 7/24 = P(A = 0) * P(C = 0)$
$P(A = 0, C = 1) = 1/4 + 1/8 = 3/8 = P(A = 0) * P(C = 1)$
$P(A = 1, C = 0) = 1/8 + 1/48 = 7/48 = P(A = 1) * P(C = 0)$
$P(A = 1, C = 1) = 1/8 + 1/16 = 3/16 = P(A = 1) * P(C = 1)$

Because $P(A, C) = P(A)*P(C)$ for all A value and C value. So A and C are independent.

$P(B = 0, C = 0) = 1/4 + 1/8 = 3/6 \mathrel{!=} P(B = 0) * P(C = 0)$
So B and C are dependent.

So graph G is

So graph G is not the only directed graph.

(2). (a). Just apply BN semantics (node is indp of non-descendants given parents).
For A: parents are B, H, no descendant, so

$A \perp C, D, E, F, G, I, J \mid B, H$

For B: no parent, descendants are A, D, E, F, G, so

$B \perp C, H, I, J$

For C: parents are H, I , descendants are D, F, G so

$C \perp A, B, E, J \mid H, I$

For D: parents are B, C, descendants are F, G, so

$D \perp A, E, H, I, J \mid B, C$

For E: parents are B, J, no descendant, so

$E \perp A, C, D, F, G, H, I \mid J, B$

For F: parents are C, D, H, J, descendant is G, so

$F \perp A, B, E, I \mid C, D, H, J$

For G: parents are F, H, I, no descendant, so

$G \perp A, B, C, D, E, J \mid F, H, I$

For H: no parent, descendants are A, C, D, F, G, so

$H \perp B, E, I, J$

For I: no parent, descendants are C, D, F, G, so

$I \perp A, B, E, H, J$

For J: no parent, descendants are E, F, G, so

$J \perp A, B, C, D, H, I$

(b). No, the trail A, B, D, C, F, G is active

(c). Yes, all trails blocked. In particular A E D is blocked, so is the V-structure J F D.

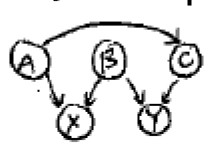
(d) No, the trail C, I, G, F, D, B, A is active.

(e) No, the trail J, E, B, A is active.

(f) Yes, all trails are blocked. In particular, neither C nor G is conditioned on, so all the V-structures leading to I are blocked.

③ No. Despite the blocked paths between A and B, and between B and C, there can still be unblocked paths between A and C.

Example:

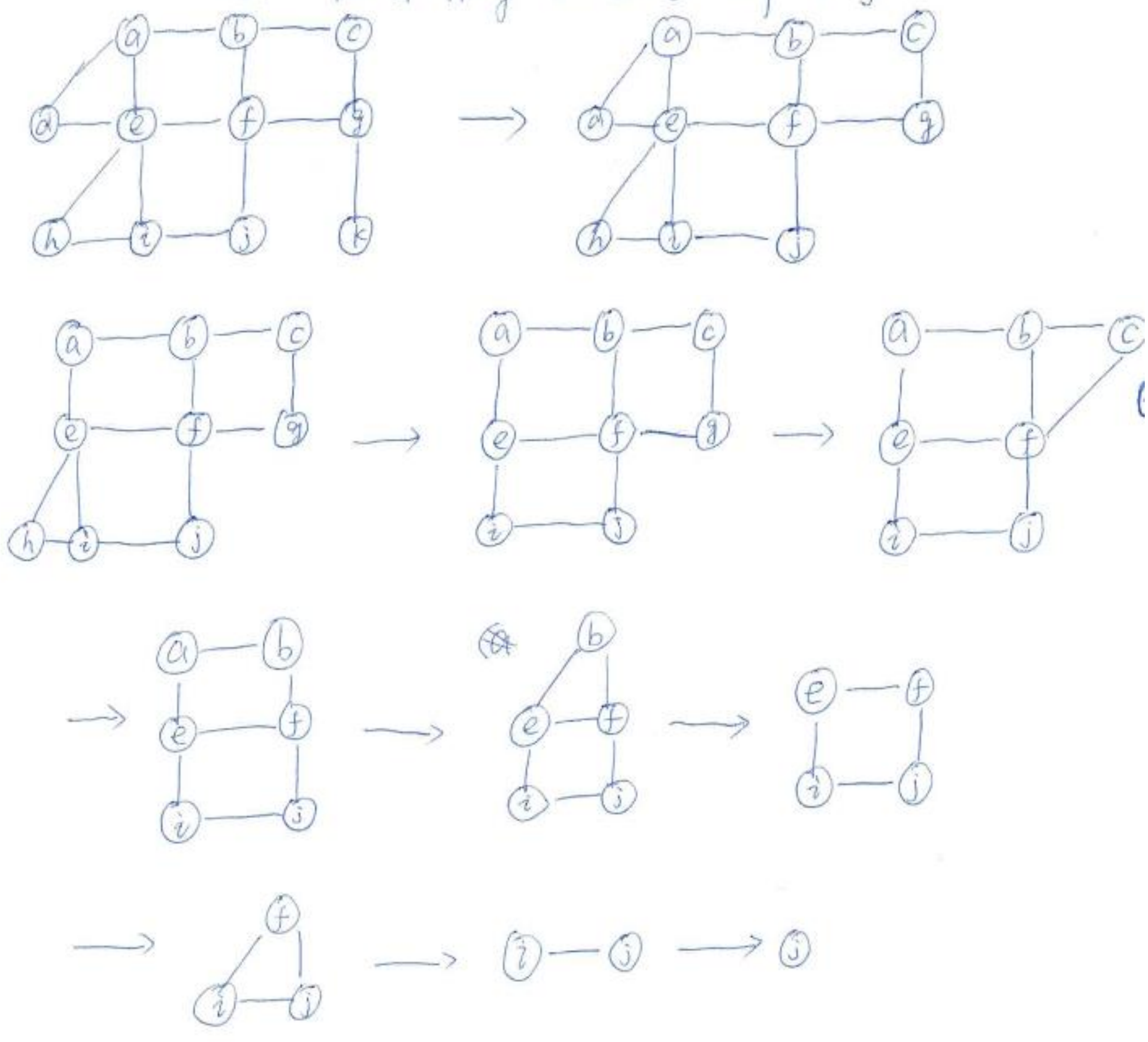


A is d-sep from B. B is d-sep from C
but A is not d-sep from C

Problem 2

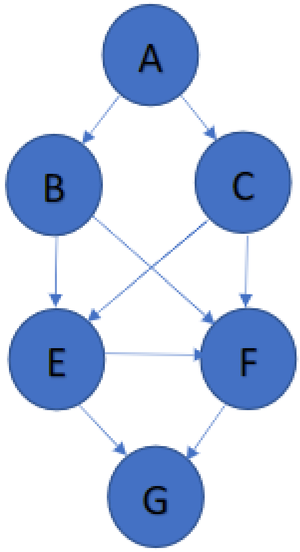
1. Largest clique created had size two so treewidth=2. Example elimination ordering:

1. The tree width is 2. An optimal elimination order. is: k-d-h-g-c-a-b-e-f-i-j



2. By definition of treewidth, this statement is equivalent to that the largest clique in the induced graph according to an optimal elimination ordering is at least as large as that in the original graph. This holds since the edge set of any induced graph is always a superset of that of the original graph (i.e. the maximal clique of the original graph will always be preserved in an induced graph, if not expanded by fill edges).
   Also see Koller text section 9.4.2.3 (p. 308) for more detailed discussions.

Problem 3.



We first remove the node D and its incident edges, then add the minimum number of new edges to preserve the original dependencies. The key here is to avoid making additional independence statements in the marginal I-map that are not present in the original network, by leaving out edges; particularly, we need enough edges to preserve all the possible ways probabilistic influence could travel between A, B, C, E, F, and G in the original network (possibly through D), in the new network where D is removed.

Removing D will not influence the relation of {A, B, C}, so they keep the same. In the original graph, there are paths: B -> D -> F, C-> D -> F, B -> D -> E, C -> D -> E, E <-D -> F, indicating that without D, F depends on B and C, E depends on B and C, and E depends on F.

A lot of people forgot about the edge between E and F, which is needed. Without it, the marginal I-map would state that E and F are independent given B and C, which doesn't generally hold in the original network (where the probabilistic influence from E to F could traveled through D).

Problem 4

1. Associate with each edge a random variable that indicates its color. E.g. let X_e represent the color of edge $e \in E_G$. Let $e, f \in E_G$ be two incident edges (i.e. there exists a shared node $i \in V_G$, $i \in e$ and $i \in f$), then we can define the potential $\phi(x_e, x_f) = \mathbf{1}_{x_e \neq x_f}$ to represent the lack of "color clash" between them.

   Let $X = [X_e, e \in E_G]$ taking values in $\chi = \{1, \dots, K\}^{|V_G|}$ be the random vector for a K-edge coloring of graph G, then the uniform distribution over valid coloring can be written as

$$p(x) = \frac{1}{Z} \prod_{e,f \in E_G, \exists i \in V_G, i \in e, i \in f} \phi(x_e, x_f) = \frac{1}{Z} \prod_{e,f \in E_G, \exists i \in V_G, i \in e, i \in f} \mathbf{1}_{x_e \neq x_f}$$

   where Z is the total number of valid coloring

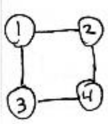
$$Z = \sum_{x \in \chi} \prod_{e,f \in E_G, \exists i \in V_G, i \in e, i \in f} \mathbf{1}_{x_e \neq x_f}$$

2. We construct a new graph G' for the above MRF. Define a function $\Phi : V_G \rightarrow V_{G'}$ that maps every edge $e \in E_G$ to a vertex $\Phi(e) \in V_{G'}$. We define an edge $e = (i, j) \in E_{G'}$ over nodes i, j, if we have that $\exists l \in V_G, l \in \Phi^{-1}(i)$ and $l \in \Phi^{-1}(j)$. In words, G' is defined by creating a node for every edge in G, and connecting two nodes if their corresponding edges in G are incident. The MRF then factorizes over G' with edge potential ϕ.

3. It's possible for $p(x)$ to be undefined if $Z = 0$, i.e., there's no valid coloring (e.g., if the edge set E_G is empty, or more edges than colors, etc.).

4. Z=18; brute-force counting works.

Possible colorings are so $Z = 18$. All other assignments violate the coloring constraint



V_1	V_2	V_3	V_4
1	2	2	1
1	2	2	3
1	2	3	1
1	3	2	1
1	3	3	1
1	3	3	2
2	1	1	2
2	1	1	3
2	1	3	2
2	3	1	2
2	3	3	1
2	3	3	2
3	1	1	2
3	1	1	3
3	1	2	3
3	2	1	3
3	2	2	1
3	2	2	3