Exercise 1. ***Moralisation: Converting DAGs to undirected minimal I-maps***

In the lecture, we had the following recipe to construct undirected minimal I-maps for $\mathcal{I}(p)$:

- *Determine the Markov blanket for each variable x_i*
- *Construct a graph where the neighbours of x_i are given by its Markov blanket.*

We can adapt the recipe to construct an undirected minimal I-map for the independencies $\mathcal{I}(G)$ encoded by a DAG G. What we need to do is to use G to read out the Markov blankets for the variables x_i rather than determining the Markov blankets from the distribution p.

Show that this procedure leads to the following recipe to convert DAGs to undirected minimal I-maps:

1. *For all immoralities in the graph: add edges between* all *parents of the collider node.*
2. *Make all edges in the graph undirected.*

The first step is sometimes called "moralisation" because we "marry" all the parents in the graph that are not already directly connected by an edge. The resulting undirected graph is called the moral graph of G, sometimes denoted by $\mathcal{M}(G)$.

Solution. The Markov blanket of a variable x is the set of its parents, children, and co-parents, as shown in the graph below in sub-figure (a). The parents and children are connected to x in the directed graph, but the co-parents are not directly connected to x. Hence, according to "Construct a graph where the neighbours of x_i are its Markov blanket.", we need to introduce edges between x and all its co-parents. This gives the intermediate graph in sub-figure (b).

Now, considering the top-left parent of x, we see that for that node, the Markov blanket includes the other parents of x. This means that we need to connect all parents of x, which gives the graph in sub-figure (c). This is sometimes called "marrying" the parents of x. Continuing in this way, we see that we need to "marry" all parents in the graph that are not already married.

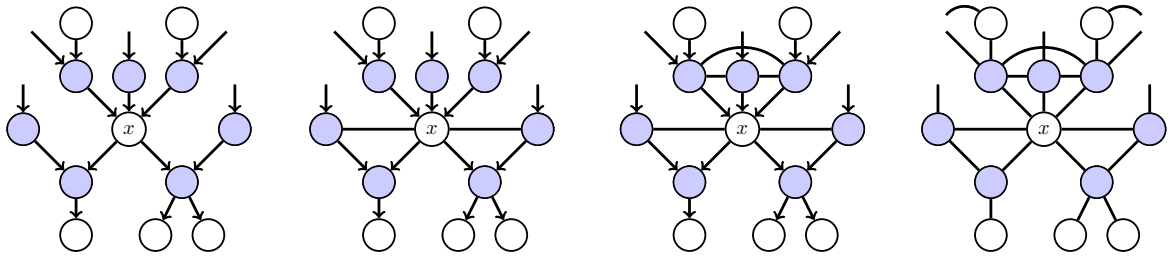
Finally, we need to make all edges in the graph undirected, which gives sub-figure (d).

A simpler approach is to note that the DAG specifies the factorisation $p(\mathbf{x}) = \prod_i p(x_i|\text{pa}_i)$. We can consider each conditional $p(x_i|\text{pa}_i)$ to be a factor $\phi_i(x_i, \text{pa}_i)$ so that we obtain the Gibbs distribution $p(\mathbf{x}) = \prod_i \phi_i(x_i|\text{pa}_i)$. Visualising the distribution by connecting all variables in the same factor $\phi_i(x_i|\text{pa}_i)$ leads to the "marriage" of all parents of x_i. This corresponds to the first step in the recipe because x_i is in a collider configuration with respect to the parent nodes. Not all parents form an immorality but this does not matter because those that do not form an immorality are already connected by a covering edge in the first place.

Exercise 2. ***Moralisation exercise***

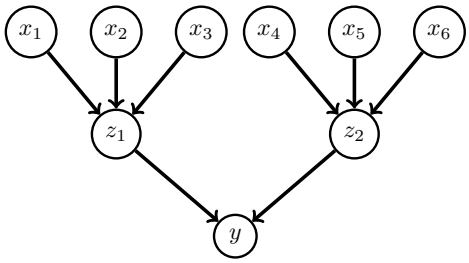
Consider the DAG G:

©Gutmann, University of Edinburgh, 2026 CC BY 4.0

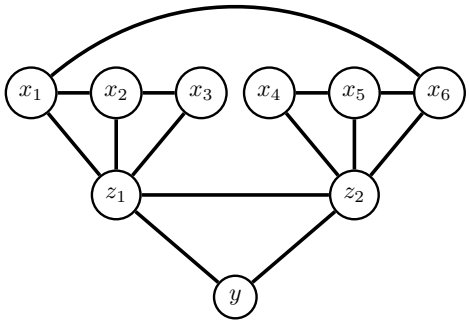


(a) DAG (b) Intermediate step 1 (c) Intermediate step 2 (d) Undirected graph

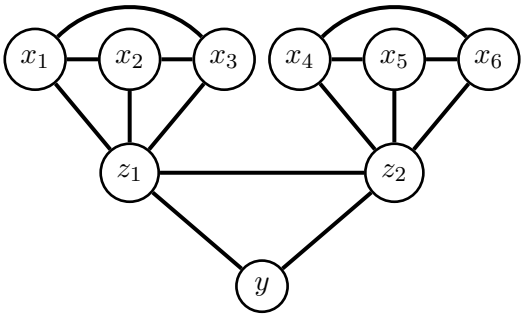
Figure 1: Answer to Exercise 1: Illustrating the moralisation process



A friend claims that the undirected graph below is the moral graph $\mathcal{M}(G)$ of G. Is your friend correct? If not, state which edges needed to be removed or added, and explain, in terms of represented independencies, why the changes are necessary for the graph to become the moral graph of G.



Solution. The moral graph $\mathcal{M}(G)$ is an undirected minimal I-map of the independencies represented by G. Following the procedure of connecting "unmarried" parents of colliders, we obtain the following moral graph of G:



We can thus see that the friend's undirected graph is not the moral graph of G.

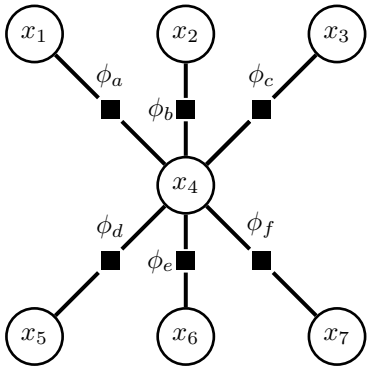
©Gutmann, University of Edinburgh, 2026 CC BY 4.0

The edge between x_1 and x_6 can be removed. This is because for G, we have e.g. the independencies $x_1 \perp\!\!\!\perp x_6|z_1$, $x_1 \perp\!\!\!\perp x_6|z_2$, $x_1 \perp\!\!\!\perp x_6|z_1, z_2$ which is not represented by the drawn undirected graph.

We need to add edges between x_1 and x_3, and between x_4 and x_6. Otherwise, the undirected graph makes the wrong independency assertion that $x_1 \perp\!\!\!\perp x_3|x_2, z_1$ (and equivalent for x_4 and x_6).

Exercise 3. ***Choice of elimination order in factor graphs***

We would like to compute the marginal $p(x_1)$ by variable elimination for a joint pmf represented by the following factor graph. All variables x_i can take K different values.



(a) A friend proposes the elimination order $x_4, x_5, x_6, x_7, x_3, x_2$, i.e. to do x_4 first and x_2 last. Explain why this is computationally inefficient.

Solution. According to the factor graph, $p(x_1, \ldots, x_7)$ factorises as

$$p(x_1, \ldots, x_7) \propto \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\phi_e(x_6, x_4)\phi_f(x_7, x_4) \tag{S.1}$$

If we choose to eliminate x_4 first, i.e. compute

$$p(x_1, x_2, x_3, x_5, x_6, x_7) = \sum_{x_4} p(x_1, \ldots, x_7) \tag{S.2}$$

$$\propto \sum_{x_4} \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\phi_e(x_6, x_4)\phi_f(x_7, x_4) \tag{S.3}$$

we cannot pull any of the factors out of the sum since each of them depends on x_4. This means the cost to sum out x_4 for all combinations of the six variables $(x_1, x_2, x_3, x_5, x_6, x_7)$ is K^7. Moreover, the new factor

$$\tilde{\phi}(x_1, x_2, x_3, x_5, x_6, x_7) = \sum_{x_4} \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\phi_e(x_6, x_4)\phi_f(x_7, x_4) \tag{S.4}$$

does not factorise anymore so that subsequent variable eliminations will be expensive too.

(b) Propose an elimination ordering that achieves $O(K^2)$ computational cost per variable elimination and explain why it does so.

©Gutmann, University of Edinburgh, 2026 CC BY 4.0

Solution. Any ordering where x_4 is eliminated last will do. At any stage, elimination of one of the variables x_2, x_3, x_5, x_6, x_7 is then a $O(K^2)$ operation. This is because e.g.

$$p(x_1, \ldots, x_6) = \sum_{x_7} p(x_1, \ldots, x_7) \tag{S.5}$$

$$\propto \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\phi_e(x_6, x_4) \underbrace{\sum_{x_7} \phi_f(x_7, x_4)}_{\tilde{\phi}_7(x_4)} \tag{S.6}$$

$$\propto \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\phi_e(x_6, x_4)\tilde{\phi}_7(x_4) \tag{S.7}$$

where computing $\tilde{\phi}_7(x_4)$ for all values of x_4 is $O(K^2)$. Further,

$$p(x_1, \ldots, x_5) = \sum_{x_6} p(x_1, \ldots, x_6) \tag{S.8}$$

$$\propto \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\tilde{\phi}_7(x_4) \sum_{x_6} \phi_e(x_6, x_4) \tag{S.9}$$

$$\propto \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\tilde{\phi}_7(x_4)\tilde{\phi}_6(x_4), \tag{S.10}$$

where computation of $\tilde{\phi}_6(x_4)$ for all values of x_4 is again $O(K^2)$. Continuing in this manner, one obtains

$$p(x_1, x_4) \propto \phi_a(x_1, x_4)\tilde{\phi}_2(x_4)\tilde{\phi}_3(x_4)\tilde{\phi}_5(x_4)\tilde{\phi}_6(x_4)\tilde{\phi}_7(x_4). \tag{S.11}$$

where each derived factor $\tilde{\phi}$ has $O(K^2)$ cost. Summing out x_4 and normalising the pmf is again a $O(K^2)$ operation.

©Gutmann, University of Edinburgh, 2026 CC BY 4.0