Chapter 4

Causal Structure Learning I: Overview and Identifiability

4.1 Motivation

When we study policy evaluation, one question that naturally arises is "how do we know the causal graph we're using to tell us how to estimate interventional quantities?" In the examples discussed so far, we have attributed this knowledge to domain expertise: we imagine that an economist or healthcare professional has used some combination of common sense (like causes coming before effects) and domain-specific training to postulate a plausible causal graph. However, such domain expertise might not always be available, either because the system is too large and complex, or because the system involves objects with which humans have little direct familiarity.

A prototypical example of such a system is the system of genetic interactions happening inside each one of our cells. Genes regulate one another (exert causal influences on one another) through a number of biological mechanisms. For example, in *E. coli*, one gene codes for a *tryptophan repressor* protein. When this protein binds to several molecules of tryptophan, then it can bind to a DNA sequence next to the genes which encode for tryptophan. This binding prevents these genes from being transcribed, implementing a form of negative feedback control for tryptophan. Thus, if we were to intervene on this system by removing the gene which encodes for the tryptophan repressor protein (setting the expression to zero), then we would increase the expression of the genes which encode for tryptophan.

Since human cells have roughly 20,000 genes, it is inconceivable that a biologist would be able to provide "domain expertise" by writing down, for each pair of genes, whether one of them regulates the other. However, with recent advances in high-throughput measurement of gene expression and gene editing technologies such as CRISPR, it is possible to record large amounts of gene expression data from different interventions. The subfield of causality known as **causal structure learning** (also called **causal discovery**) is concerned with developing methods for such problems of learning causal models from data.

4.1.1 Approaches to causal structure learning

Figure 4.1 gives a loose taxonomy of the different types of approaches used for causal structure learning. At the first level of division, we have the following:

- **Constraint-based methods**, which tests for conditional independences or other constraints satisfied by the data, and constructs a graph which implies those constraints,
- **Score-based methods**, which score graphs according to some measure of how well they fit the data, and search for a graph which maximizes that score, and

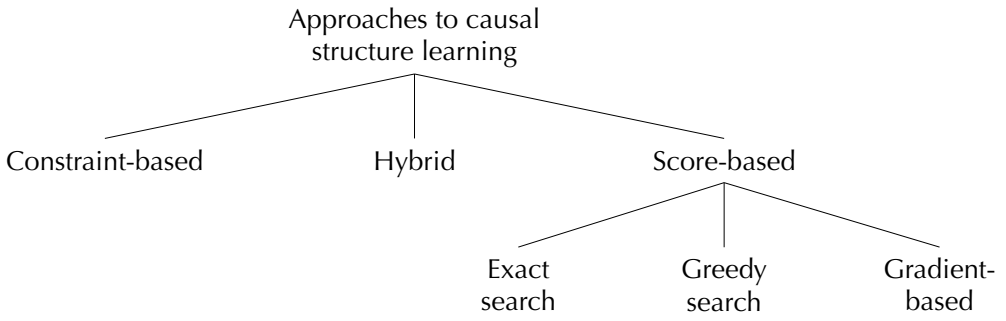

Figure 4.1: Approaches to causal structure learning.

- **Hybrid methods**, which use aspects of both approach, e.g. first restricting the search space based on constraints and then maximizing a score within the restricted search space.

Score-based methods are often based on some form of (penalized) maximum likelihood, and thus enjoy some favorable theoretical properties in terms of statistical efficiency. However, they are often developed in parametric settings, and require some work to extend to nonparametric settings. On the other hand, since constraint-based methods rely primarily on hypothesis testing (e.g. for conditional independence), and there are a variety of nonparametric hypothesis tests, they are more easily ready to be used in the nonparametric setting. Thus, it is worth discussing both types of approaches in this course.

At the second level of division, we may consider differences in the search procedure used for a score-based method:

- **Exact search** methods are guaranteed to return the graph with the highest score. However, this requires solving a challenging optimization problem over the space of causal graphs, and thus these methods tend to be less scalable.
- **Greedy search** methods search by taking "greedy steps" which each increase the score. For finite amounts of data, these methods might get stuck at local maxima and be unable to find the highest-scoring graph. However, one of the seminal results in causal structure learning shows that, in the limit of infinite data, certain greedy search methods never get stuck at a local maxima that is not the global maxima.
- **Gradient-based** methods are a newer class of approaches which relax the combinatorial optimization problem over graphs into a continuous optimization problem, which can then be optimized via gradient descent. These methods perform well in practice, but it is difficult to obtain theoretical guarantees on whether they find the global optimum even with infinite data, since the resulting problem is often highly non-convex.

4.1.2 Identifiability

Before we discuss methods for causal structure learning, we will address the question: what about the causal structure can be learned from data?

Consider two causal models M_a and M_b, where in M_a, we have

$$\begin{aligned} X_1 &= \varepsilon_1 & \varepsilon_1 &\sim \mathcal{N}(0,1) \\ X_2 &= aX_1 + \varepsilon_2 & \varepsilon_2 &\sim \mathcal{N}(0,1) \end{aligned}$$

for $a > 0$. Then $\mathbb{E}_a[X_1] = \mathbb{E}_a[X_2] = 0$ and

$$\Sigma_a = \mathrm{Cov}_a([X_1, X_2]) = \begin{bmatrix} 1 & a \\ a & a^2 + 1 \end{bmatrix}$$

In M_b, let

$$\begin{aligned} X_2 &= \varepsilon_2 & \varepsilon_2 &\sim \mathcal{N}(0, a^2+1) \\ X_1 &= \frac{a}{a^2+1}X_2 + \varepsilon_1 & \varepsilon_1 &\sim \mathcal{N}\left(0, \frac{1}{a^2+1}\right) \end{aligned}$$

Then again $\mathbb{E}_b[X_1] = \mathbb{E}_b[X_2] = 0$, and $\Sigma_b = \Sigma_a$. Since a normal distribution is completely characterized by its mean and covariance, we have $\mathbb{P}^a_{\mathcal{X}} = \mathbb{P}^b_{\mathcal{X}}$, i.e., the entailed distributions of the two models are equivalent, even though in M^a we have the causal graph $X_1 \to X_2$, and in M^b we have the causal graph $X_2 \to X_1$.

Now, consider intervening on X_1, e.g. suppose we perform a do-intervention setting $X_1 = 0$. Then M^I_a, the interventional SCM for M_a, is

$$\begin{aligned} X_1 &= 0 \\ X_2 &= aX_1 + \varepsilon_2 & \varepsilon_2 &\sim \mathcal{N}(0,1) \end{aligned}$$

and now

$$\Sigma^I_a = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Meanwhile, M^I_b is

$$\begin{aligned} X_2 &= \varepsilon_2 & \varepsilon_2 &\sim \mathcal{N}(0, a^2+1) \\ X_1 &= 0 \end{aligned}$$

and

$$\Sigma^I_b = \begin{bmatrix} 0 & 0 \\ 0 & a^2+1 \end{bmatrix}$$

In M^a, where X_2 is downstream of X_1, the intervention on X_1 changes the variance of X_2 from a^2+1 to 1, whereas in M^b, where X_2 is not downstream of X_1, the intervention on X_1 does not affect the distribution of X_2.

From this example, we have two takeaways:

- Two causal models may be indistinguishable from observational data alone, and
- Adding interventional data may help distinguish between such causal models.

This lecture will be focused on generalizing these two insights to the non-parametric setting. We will first establish what can be identified from observational data, and then use our tool of the expanded interventional SCM to extend these results to interventional data.

Remark 4.1. *A separate approach to distinguishing between causal models is to add assumptions on the causal mechanisms and/or exogenous noise appearing in the structural causal model. The canonical example of this approach to identifiability is the* **linear non-Gaussian additive noise model***, which assumes that each variable is a linear function of its parents, plus exogenous noise with a non-Gaussian distribution.*

4.2 Preview: Constraint-based learning of a graph

Let $\mathcal{G}^*$ be the graph in Figure 4.2(a), and let $\mathbb{P}_{\mathcal{X}}$ factorize according to $\mathcal{G}^*$. Suppose that, if $X_i \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_j \mid \mathbf{S}$, then we can conclude that $X_i \perp\!\!\!\perp_{\mathcal{G}^*} X_j \mid \mathbf{S}$, i.e., the only conditional independence statements in $\mathbb{P}_{\mathcal{X}}$ are those implied by d-separation in $\mathcal{G}^*$. Then we can make the following conclusions:

- Since $X_1 \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_5 \mid X_4$, there is no edge $X_1 - X_5$.

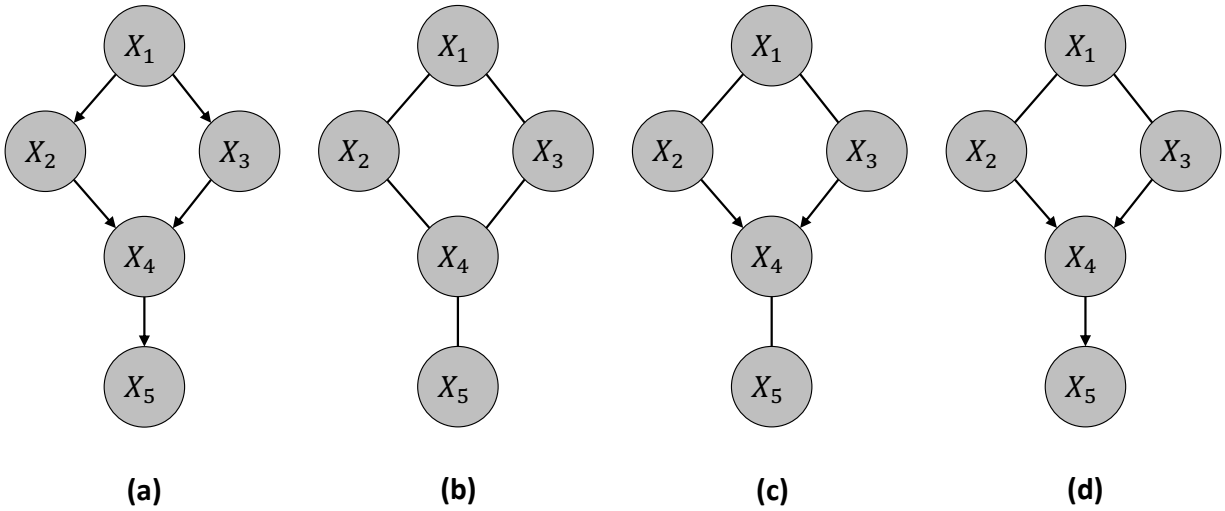

Figure 4.2: **(a)** The data-generating graph $\mathcal{G}^*$. **(b)** The skeleton of $\mathcal{G}^*$. **(c)** The skeleton of $\mathcal{G}^*$ with the immorality $X_2 \to X_4 \leftarrow X_3$ directed. **(d)** The graph from (c) with the additional orientation $X_4 \to X_5$.

- Since $X_2 \perp\!\!\!\perp_{\mathbb{P}_\mathcal{X}} X_5 \mid X_4$, there is no edge $X_1 - X_5$.
- Since $X_3 \perp\!\!\!\perp_{\mathbb{P}_\mathcal{X}} X_5 \mid X_4$, there is no edge $X_3 - X_5$.
- Since $X_2 \perp\!\!\!\perp_{\mathbb{P}_\mathcal{X}} X_3 \mid X_1$, there is no edge $X_2 - X_4$.
- Since $X_1 \perp\!\!\!\perp_{\mathbb{P}_\mathcal{X}} X_4 \mid X_2, X_3$, there is no edge $X_1 - X_4$.

If we start with an undirected complete graph and remove these edges, we obtain the graph in Figure 4.2(b). Moreover, since $X_2 \perp\!\!\!\perp_{\mathbb{P}_\mathcal{X}} X_3 \mid X_1$, and thus by assumption $X_2 \perp\!\!\!\perp_{\mathcal{G}^*} X_3 \mid X_1$, we know that X_4 must be a collider on the path $\langle X_2, X_4, X_3 \rangle$. Adding these orientations gives the graph in Figure 4.2(c). Finally, if we had $X_5 \to X_4$ in $\mathcal{G}^*$, then we would have $X_2 \perp\!\!\!\perp_{\mathbb{P}_\mathcal{X}} X_5$. Since this does not hold, we must have $X_4 \to X_5$, giving the graph in Figure 4.2(d).

In this case, we notice that we could recover two aspects of the causal graph: its skeleton and its unshielded colliders.

Definition 4.1. *The* **skeleton** *of a DAG $\mathcal{G}$ is an undirected graph with the edge $X_i - X_j$ if $X_i \to X_j$ in $\mathcal{G}$.*

Definition 4.2. *Given a DAG $\mathcal{G}$, a path $X_i \to X_k \leftarrow X_j$ with X_i and X_j non-adjacent is called an* **unshielded collider** *(also called* **immoralities***).*

4.3 Markovianity implies factorization

First, we will establish the converse to Theorem 2.2: if every d-separation statement in a DAG $\mathcal{G}$ holds as a conditional independence statement in $\mathbb{P}_\mathcal{X}$, then $\mathbb{P}_\mathcal{X}$ factorizes according to $\mathcal{G}$. In particular, if two DAGs $\mathcal{G}$ and $\mathcal{G}'$ have the same d-separation statements, and $\mathbb{P}_\mathcal{X}$ factorizes according to $\mathcal{G}$, then it also factorizes according to $\mathcal{G}'$, and we cannot distinguish between $\mathcal{G}$ and $\mathcal{G}'$ from observational data alone. This will lead us to focus on when two DAGs have the same d-separation statements.

Towards proving this result, we highlight the following particularly important d-separation statement.

Lemma 4.1. *Let $\mathcal{G}$ be a DAG. Let $\mathrm{nd}_\mathcal{G}(X_i)$ denote the* **non-descendants** *of X_i. Then*

$$X_i \perp\!\!\!\perp_\mathcal{G} \mathrm{nd}_\mathcal{G}(X_i) \setminus \mathrm{pa}_\mathcal{G}(X_i) \mid \mathrm{pa}_\mathcal{G}(X_i)$$

Proof. Let $X_j \in \mathrm{nd}_{\mathcal{G}}(X_i)$ and let γ be a path from X_i to X_j.

- If the first edge in γ is into X_i, i.e. the first edge is $X_i \leftarrow X_k$, then γ is blocked at X_k, since X_k must be a non-collider on this path and $X_k \in \mathrm{pa}_{\mathcal{G}}(X_i)$.
- If the first edge in γ is out of X_i, then the path must contain a collider, since X_j is not a descendant of X_i. Let X_k be the first collider on this path, then X_k is a descendant of X_i. The path is blocked at this collider: by acyclicity, neither X_k nor any of its descendants are in $\mathrm{pa}_{\mathcal{G}}(X_i)$.

We have shown that any path from X_i to X_j is blocked, i.e., X_j and X_i are d-separated by $\mathrm{pa}_{\mathcal{G}}(X_i)$. □

Now, we introduce a way of ordering the nodes of a DAG $\mathcal{G}$.

Definition 4.3. *Let $\mathcal{G}$ be a DAG. We say that a permutation σ is a* **topological ordering** *of the nodes in $\mathcal{G}$ if $j \in \mathrm{an}_{\mathcal{G}}(i)$ implies that $\sigma(j) < \sigma(i)$.*

Finally, we prove that Markovianity implies factorization.

Theorem 4.1. *Let $\mathcal{G}$ be a DAG. Suppose $\mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G}) \subseteq \mathcal{I}_{\perp\!\!\!\perp}(\mathbb{P}_{\mathcal{X}})$, i.e., $\mathbb{P}_{\mathcal{X}}$ is Markov to $\mathcal{G}$. Then $\mathbb{P}_{\mathcal{X}}$ factorizes according to $\mathcal{G}$, i.e.,*

$$\mathbb{P}_{\mathcal{X}}(\mathcal{X}) = \prod_{X_i \in \mathcal{X}} \mathbb{P}_{\mathcal{X}}(X_i \mid \mathrm{pa}_{\mathcal{G}}(X_i))$$

Proof. Let σ be a topological order for the nodes in the DAG. Let $\mathrm{pre}_{\sigma}(X_i)$ denote all nodes which come before X_i in the topological order. Then by the chain rule, we have

$$\mathbb{P}_{\mathcal{X}}(\mathcal{X}) = \prod_{X_i \in \mathcal{X}} \mathbb{P}_{\mathcal{X}}(X_i \mid \mathrm{pre}_{\sigma}(X_i))$$

We have $\mathrm{pre}_{\sigma}(X_i) \setminus \mathrm{pa}_{\mathcal{G}}(X_i) \subseteq \mathrm{nd}(X_i) \setminus \mathrm{pa}_{\mathcal{G}}(X_i)$, and thus by Lemma 4.1, we have $X_i \perp\!\!\!\perp_{\mathcal{G}} \mathrm{pre}_{\sigma}(X_i) \setminus \mathrm{pa}_{\mathcal{G}}(X_i) \mid \mathrm{pa}_{\mathcal{G}}(X_i)$. Thus, $\mathbb{P}_{\mathcal{X}}(X_i \mid \mathrm{pre}_{\sigma}(X_i)) = \mathbb{P}_{\mathcal{X}}(X_i \mid \mathrm{pa}_{\mathcal{G}}(X_i))$. Replacing each term in the product yields the result. □

4.4 Markov equivalence

We will now begin our characterization of when two DAGs have the same d-separation statements, i.e. when they are Markov equivalent:

Definition 4.4. *We call two DAGs $\mathcal{G}$ and $\mathcal{G}'$* **Markov equivalent** *if $\mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G}) = \mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G}')$. We denote this equivalence by $\mathcal{G} \approx_{\mathcal{M}} \mathcal{G}'$. The set of all DAGs which are Markov equivalent to a DAG $\mathcal{G}$ is called the* **Markov equivalence class** *of $\mathcal{G}$, and is denoted by $\mathcal{M}(\mathcal{G})$.*

We now state the main theorem:

Theorem 4.2. *Two DAGs $\mathcal{G}$ and $\mathcal{G}'$ are Markov equivalent if and only if they have the same skeleton and unshielded colliders.*

First, note the following corollaries of Lemma 4.1:

Corollary 4.1. *If X_i and X_j are not adjacent in $\mathcal{G}$, then either $X_i \perp\!\!\!\perp_{\mathcal{G}} X_j \mid \mathrm{pa}_{\mathcal{G}}(X_i)$ or $X_i \perp\!\!\!\perp_{\mathcal{G}} X_j \mid \mathrm{pa}_{\mathcal{G}}(X_j)$.*

Proof. If X_i and X_j are not adjacent, then either $X_j \in \mathrm{nd}_{\mathcal{G}}(X_i)$ or $X_i \in \mathrm{nd}_{\mathcal{G}}(X_j)$. □

Corollary 4.2. *If $\mathcal{G} \approx_{\mathcal{M}} \mathcal{G}'$, then $\mathcal{G}$ and $\mathcal{G}'$ have the same skeleton.*

Proof. Say that $\mathcal{G}$ and $\mathcal{G}'$ have different skeletons, without loss of generality, let $X_i - X_j$ in $\mathcal{G}$ but X_i and X_j be non-adjacent in $\mathcal{G}'$. Then X_i and X_j can be d-separated in $\mathcal{G}'$, but not in $\mathcal{G}$. □

Now we prove one direction of Theorem 4.2.

Lemma 4.2. *If $\mathcal{G} \approx_{\mathcal{M}} \mathcal{G}'$, then $\mathcal{G}$ and $\mathcal{G}'$ have the same skeleton and unshielded colliders.*

Proof. That $\mathcal{G}$ and $\mathcal{G}'$ have the same skeleton is Corollary 4.2.

Now, suppose that $\mathcal{G}$ and $\mathcal{G}'$ have the same skeleton, but different unshielded colliders. Without loss of generality, assume that $X_i \to X_k \leftarrow X_j$ is an unshielded collider in $\mathcal{G}$, while this unshielded collider is not present in $\mathcal{G}'$. Let $\mathbf{S} = \text{pa}_{\mathcal{G}}(X_i)$ or $\mathbf{S} = \text{pa}_{\mathcal{G}}(X_j)$ be such that $X_i \perp\!\!\!\perp_{\mathcal{G}} X_j \mid \mathbf{S}$. The d-separation is guaranteed to hold for at least one of these sets by Corollary 4.1. However, the path $\langle X_i, X_k, X_j \rangle$ is d-connecting given $\mathbf{S}$ in $\mathcal{G}'$, since X_k is a non-collider on this path and $X_k \notin \mathbf{S}$. □

Now, we will prove the more difficult direction. This will require that, if X_i and X_j are d-connected given $\mathbf{S}$ in $\mathcal{G}$, then we can find a d-connecting path from X_i to X_j given $\mathbf{S}$ in $\mathcal{G}'$. For any d-connected pair, we will define a "canonical" type of d-connecting path, in particular, one that cannot be made any shorter.

Definition 4.5. *A d-connecting path $\gamma = \langle \gamma_1, \dots, \gamma_M \rangle$ from γ_1 to γ_M is called* **minimal** *if no subset of nodes forms a d-connecting path from γ_1 to γ_M.*

Minimal d-connecting paths have useful properties that will make it relatively easy to transfer them from $\mathcal{G}$ to a Markov equivalent graph $\mathcal{G}'$. In particular, the following proposition shows that any "triangles" in the path, i.e., sequences $\langle \gamma_{m-1}, \gamma_m, \gamma_{m+1} \rangle$ such that γ_{m-1} and γ_{m+1} are adjacent, must satisfy certain conditions.

Proposition 4.1. *Let γ be a minimal d-connecting path given $\mathbf{S}$ in $\mathcal{G}$. If γ_{m-1} is adjacent to γ_{m+1}, then $\gamma_{m-1} \leftarrow \gamma_m \to \gamma_{m+1}$, and at least one of γ_{m-1} or γ_{m+1} is a collider.*

Proof. Suppose γ is a d-connecting path given $\mathbf{S}$, with $\gamma_{m-1} \to \gamma_m \to \gamma_{m+1}$ and γ_{m-1} adjacent to γ_{m+1}. By acyclicity, $\gamma_{m-1} \to \gamma_{m+1}$. Consider γ' where we replace this segment with $\gamma_{m-1} \to \gamma_{m+1}$. Then γ' is also d-connecting given $\mathbf{S}$, since γ_{m-1} remains a non-collider in γ', and γ_{m+1} is a collider in γ' if and only if it is a collider in γ. Thus, γ is not minimal.

The case $\gamma_{m-1} \leftarrow \gamma_m \leftarrow \gamma_{m+1}$ is symmetric.

Now, suppose γ is a d-connecting path given $\mathbf{S}$, with $\gamma_{m-1} \to \gamma_m \leftarrow \gamma_{m+1}$ and γ_{m-1} adjacent to γ_{m+1}. Then $\overline{\text{de}}_{\mathcal{G}}(\gamma_m) \cap \mathbf{S} \neq \varnothing$. Suppose (without loss of generality) that $\gamma_{m-1} \to \gamma_{m+1}$. Consider γ' where we replace this segment with $\gamma_{m-1} \to \gamma_{m+1}$. Then γ' is also d-connecting given $\mathbf{S}$: γ_{m-1} remains a non-collider in γ', if γ_{m+1} remains a non-collider, then it is unblocked, if γ_{m+1} becomes a collider, then since $\overline{\text{de}}_{\mathcal{G}}(\gamma_m) \subset \overline{\text{de}}_{\mathcal{G}}(\gamma_{m+1})$, γ_{m+1} is unblocked. Thus, γ is not minimal.

For the second part, suppose $\gamma_{m-1} \leftarrow \gamma_m \to \gamma_{m+1}$ and that neither γ_{m-1} nor γ_{m+1} are colliders. Then none of γ_{m-1}, γ_m, and γ_{m+1} are in $\mathbf{S}$, and thus the path $\gamma_{m-1} \to \gamma_{m+1}$ is unblocked, and γ is not minimal. □

Corollary 4.3. *Let γ be a minimal d-connecting path given $\mathbf{S}$ in $\mathcal{G}$. Let $\mathcal{G}'$ have the same skeleton and unshielded colliders as $\mathcal{G}'$. Then, if $\gamma_{m-1} \to \gamma_m \leftarrow \gamma_{m+1}$ in γ in $\mathcal{G}$, then $\gamma_{m-1} \to \gamma_m \leftarrow \gamma_{m+1}$ in γ in $\mathcal{G}'$, i.e. γ in $\mathcal{G}$ and γ in $\mathcal{G}'$ have the same colliders.*

Lemma 4.3. *If $\mathcal{G}$ and $\mathcal{G}'$ have the same skeleton and unshielded colliders, then $\mathcal{G} \approx_{\mathcal{M}} \mathcal{G}'$.*

Proof. Fix $\mathbf{S}$. Suppose that γ is a minimal d-connecting path between X_i and X_j in $\mathcal{G}$ given $\mathbf{S}$. We wish to construct a corresponding d-connecting path in $\mathcal{G}'$. We will break the path into segments $\gamma_{m-1} - \gamma_m - \gamma_{m+1}$, and show that there is a corresponding d-connecting path from γ_{m-1} to γ_{m+1} in $\mathcal{G}'$. For each segment we have three cases:

- **Case 1**: γ_m is a non-collider in γ, and γ_{m-1} and γ_{m+1} are not adjacent in $\mathcal{G}$.

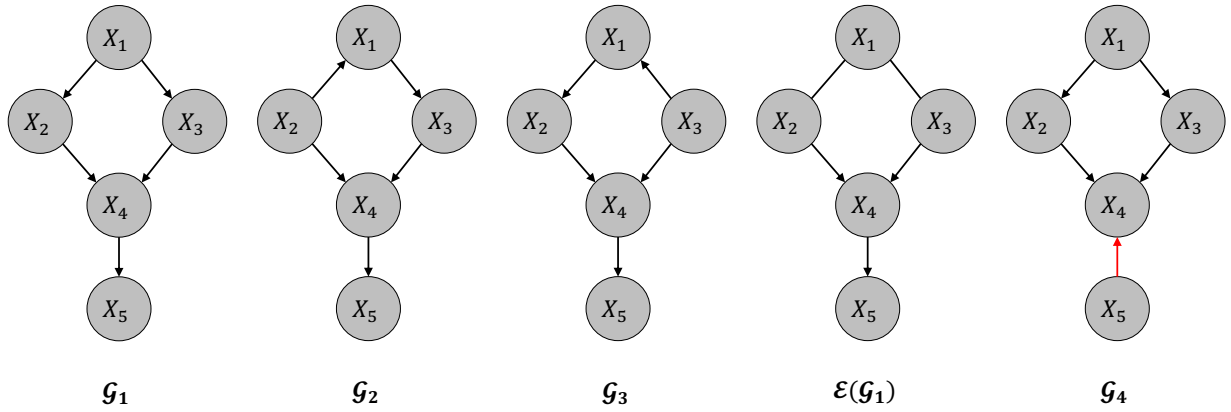

Figure 4.3: $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$ are Markov equivalent. Their essential graph is $\mathcal{E}(\mathcal{G}_1)$. They are not equivalent to $\mathcal{G}_4$, which has the same skeleton, but has two additional immoralities, $X_2 \to X_4 \leftarrow X_5$ and $X_3 \to X_4 \leftarrow X_5$.

- **Case 2**: γ_m is a non-collider in γ, and γ_{m-1} and γ_{m+1} are adjacent in $\mathcal{G}$.
- **Case 3**: γ_m is a collider in γ.

Case 1. Then $\gamma_{m-1} - \gamma_m - \gamma_{m+1}$ is a non-collider in $\mathcal{G}'$ by Corollary 4.3. Since $\gamma_{m-1} - \gamma_m - \gamma_{m+1}$ is unblocked at γ_m in $\mathcal{G}$, it is also unblocked at γ_m in $\mathcal{G}'$.

Case 2. By Proposition 4.1, at least one of γ_{m-1} or γ_{m+1} are colliders in γ in $\mathcal{G}$. Assume without loss of generality that γ_{m-1} is a collider. Then by Corollary 4.3, γ_{m-1} is a collider in γ in $\mathcal{G}'$. A path cannot have two adjacent colliders, so γ_m is not a collider in γ in $\mathcal{G}'$. Thus, $\gamma_{m-1} - \gamma_m - \gamma_{m+1}$ is unblocked at γ_m in $\mathcal{G}$.

Case 3. Let $\lambda = \langle \gamma_m, \lambda_1, \ldots, \lambda_M, S \rangle$ be the shortest path in $\mathcal{G}$ from γ_m to a descendant S in $\mathbf{S}$. Now consider the path λ in $\mathcal{G}'$.

We will show that λ has no colliders in $\mathcal{G}'$. Suppose λ in $\mathcal{G}'$ has a collider $\lambda_{u-1} \to \lambda_u \leftarrow \lambda_{u+1}$. By the assumption that $\mathcal{G}$ and $\mathcal{G}'$ have the same unshielded colliders, λ_{u-1} and λ_{u+1} must be adjacent in $\mathcal{G}'$. Since $\mathcal{G}$ has the same skeleton as $\mathcal{G}'$, λ_{u-1} and λ_{u+1} are adjacent in $\mathcal{G}$. By acyclicity, we have $\lambda_{u-1} \to \lambda_{u+1}$ in $\mathcal{G}$. However, this introduces a shorter path from γ_m to S in $\mathcal{G}$, contradicting minimality of γ in $\mathcal{G}$.

If $\lambda_1 \to \gamma_m$ in $\mathcal{G}'$, then by the assumption that $\mathcal{G}$ and $\mathcal{G}'$ have the same unshielded colliders, we must have $\lambda_1 - \gamma_{m-1}$ and $\lambda_1 - \gamma_{m+1}$ in $\mathcal{G}$. By acyclicity, $\gamma_{m-1} \to \lambda_1$ and $\gamma_{m+1} \to \lambda_1$ in $\mathcal{G}$. This gives a new minimal d-connecting path in $\mathcal{G}$ where the segment $\gamma_{m-1} \to \gamma_m \leftarrow \gamma_{m+1}$ is replaced by $\gamma_{m-1} \to \lambda_1 \leftarrow \gamma_{m+1}$. We may repeat the argument on this new path; since there are a finite number of nodes, we eventually reach a minimal d-connecting path such that no collider γ_m can be replaced by a collider on its child. □

Definition 4.6. *The* **essential graph***, of a Markov equivalence class $\mathcal{M}(\mathcal{G})$ is a mixed graph with:*

- *A directed edge $X_i \to X_j$ if $X_i \to X_j$ for all $\mathcal{G}' \in \mathcal{M}(\mathcal{G})$, and*
- *An undirected edge $X_i - X_j$ if $X_i \to X_j$ for some $\mathcal{G}_1 \in \mathcal{M}(\mathcal{G})$ and $X_i \leftarrow X_j$ for some $\mathcal{G}_2 \in \mathcal{M}(\mathcal{G})$.*

Given a DAG $\mathcal{G}$, we denote the essential graph of $\mathcal{M}(\mathcal{G})$ by $\mathcal{E}(\mathcal{G})$.

The essential graph is a form of **partially directed acyclic graph** (PDAG): a mixed graph of undirected and directed edges with no directed cycles.

Example 4.1. *An example of Markov equivalence and essential graphs are in Figure 4.3.*

.

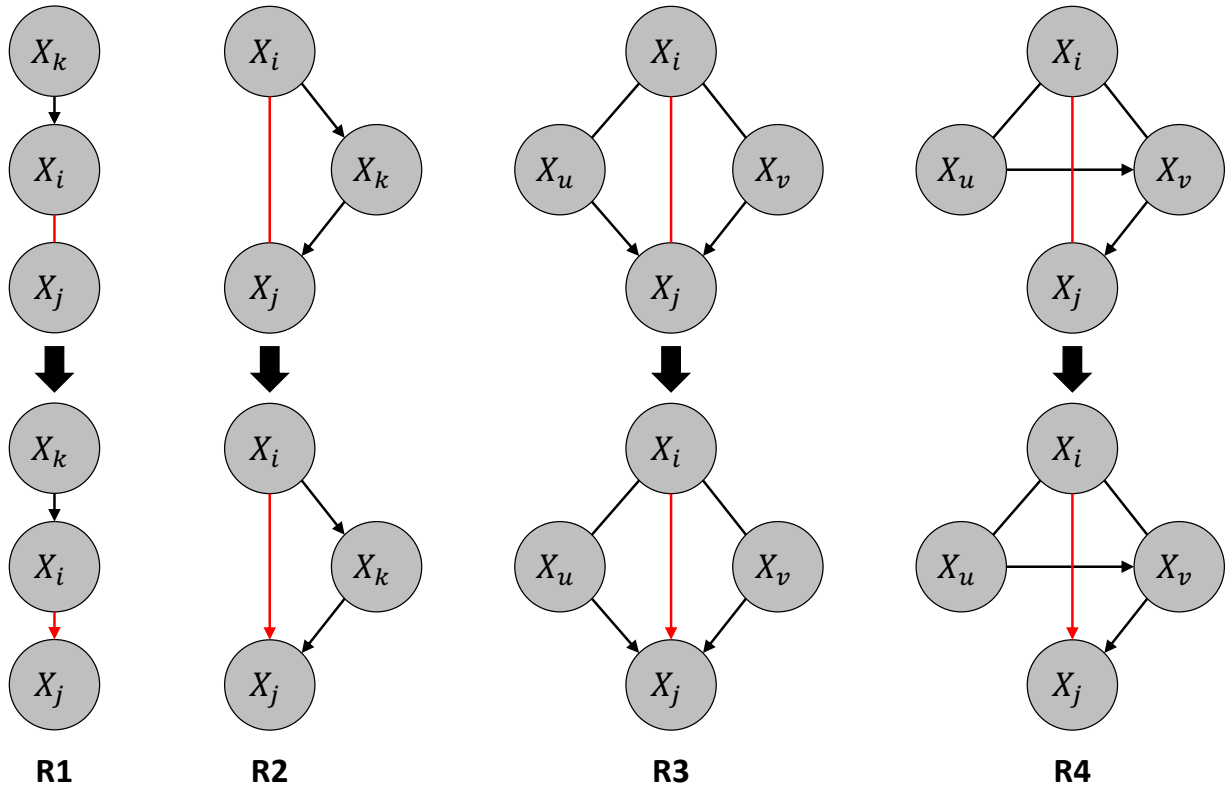

Figure 4.4: The 4 Meek orientation rules, with the altered edge $X_i \to X_j$ highlighted in red for each rule.

4.4.1 Meek's orientation rules

Since any pair $\mathcal{G} \approx_{\mathcal{M}} \mathcal{G}'$ have the same unshielded colliders, the essential graph $\mathcal{E}(\mathcal{G})$ has orientations for all edges $X_i \to X_k$ such that $X_i \to X_k$ is involved in an unshielded collider. However, as we see in Example 4.1, these may not be the only oriented edges in $\mathcal{E}(\mathcal{G})$.

There is a set of four rules, introduced by Christopher Meek, which are sufficient for recovering all orientations in the essential graph.

Definition 4.7. *Suppose that $X_i - X_j$ is in a partially directed graph $\mathcal{G}$. The* **Meek orientation rules** *are*

- ***Rule 1 (no extra unshielded colliders):*** *If $X_k \to X_i$ and X_k is not adjacent to X_j, then $X_i \to X_j$.*
- ***Rule 2 (no cycles):*** *If $X_i \to X_k$ and $X_k \to X_i$, then $X_i \to X_k$.*
- ***Rule 3:*** *If $X_i - X_u$, $X_i - X_v$, $X_u \to X_j$, $X_v \to X_j$, and X_u is not adjacent to X_v, then $X_i \to X_j$.*
- ***Rule 4:*** *If $X_i - X_u$, $X_i - X_v$, $X_u \to X_v$, $X_v \to X_j$, and X_u is not adjacent to X_j, then $X_i \to X_j$.*

Let $\mathcal{G}$ be a partially directed acyclic graph (PDAG). Then we denote by $\texttt{MPDAG}(\mathcal{G})$ *the* **maximally oriented** *PDAG $\mathcal{G}$, i.e., the graph obtained by repeatedly applying the Meek rules until none of them can be applied.*

The rules are depicted in Figure 4.4.

Lemma 4.4. *Let $\mathcal{G}$ be a DAG. Let $\mathcal{E}'$ be the partially directed graph with skeleton equal to $\mathcal{G}$, all unshielded colliders in $\mathcal{G}$ oriented, and possibly some other subset of orientations from $\mathcal{E}(\mathcal{G})$. Then, if any of the Meek rules applies to $\mathcal{E}'$, then the additional orientation is also in $\mathcal{E}(\mathcal{G})$.*

Proof. For Rule 1, any DAG which has $X_j \to X_i$ has an unshielded collider $X_j \to X_i \leftarrow X_k$. However, by assumption, all unshielded colliders from $\mathcal{G}$ are already oriented in $\mathcal{E}'$, so we must have $X_i \to X_j$ in $\mathcal{G}$.

Rule 2 simply affirms that a DAG has no cycles.

For Rule 3, let $\mathcal{G}$ be a DAG such that $X_j \to X_i$. Then, by Rule 2, we have $X_u \to X_i$ and $X_v \to X_i$. However, this introduces an unshielded collider $X_u \to X_i \leftarrow X_v$, which is ruled out by assumption.

Finally, for Rule 4, let $\mathcal{G}$ be a DAG such that $X_j \to X_i$. Then $X_i \to X_u$ by Rule 1. However, by Rule 2, we have both $X_i \to X_v$ and $X_v \to X_i$. □

Theorem 4.3. *The Meek rules are complete, i.e., if we begin with $\mathcal{E}'$ with skeleton equal to $\mathcal{G}$ and all unshielded colliders from $\mathcal{G}$ oriented, then* $\texttt{MPDAG}(\mathcal{E}') = \mathcal{E}(\mathcal{G})$.

We will not prove completeness of the Meek rules in these notes, see Meek (1995).

Bibliography

Meek, C. (1995). Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 403–410.

Chapter 5

Causal Structure Learning II: The PC Algorithm and Sparsity-based Scores

5.1 Faithfulness

Our example in Section 4.2 used the assumption that any conditional independence in $\mathbb{P}_{\mathcal{X}}$ implied a d-separation in causal graph $\mathcal{G}$. The following example shows how this constraint can be violated:

Example 5.1. *Let M be the structural causal model with*

$$\begin{aligned} X_1 &= \varepsilon_1 & \varepsilon_1 &\sim \mathcal{N}(0,1) \\ X_2 &= aX_1 + \varepsilon_2 & \varepsilon_2 &\sim \mathcal{N}(0,1) \\ X_3 &= bX_1 + cX_2 + \varepsilon_3 & \varepsilon_3 &\sim \mathcal{N}(0,1) \end{aligned}$$

We have $\mathrm{Cov}(X_1, X_3) = ab+c$. Thus, if $c = -ab$, e.g. if $a = b = 1$ and $c = -1$, then we have $\mathrm{Cov}(X_1, X_3) = 0$. In a normal distribution, zero covariance implies independence, and thus if $c = -ab$, we have $X_1 \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_3$ for the entailed distribution $\mathbb{P}_{\mathcal{X}}$. However, X_1 and X_2 are d-connected given $\mathbf{S} = \varnothing$ in the causal graph of M. Thus, $\mathbb{P}_{\mathcal{X}}$ is not faithful to $\mathcal{G}$.

Thus, the correspondence between conditional independences in $\mathbb{P}_{\mathcal{X}}$ and d-separation in $\mathcal{G}$ constitutes a substantive assumption:

Definition 5.1. *We say that $\mathbb{P}_{\mathcal{X}}$ is* **faithful** *to $\mathcal{G}$ if $\mathcal{I}_{\perp\!\!\!\perp}(\mathbb{P}_{\mathcal{X}}) = \mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G})$.*

Remark 5.1. *A typical justification for the assumption of faithfulness is that, in linear models, it holds for* **generic** *choices of parameters. Intuitively, this means that the faithfulness assumption is violated only for carefully-selected "degenerate" choices of the parameters. More formally, this means that the set of parameters for which the assumption is violated has a* **Lebesgue measure** *of zero. In particular, a violation of the faithfulness assumption implies that some polynomial of the edge weights equals zero, e.g. $ab + c = 0$ in Example 5.1. However, the zero set of a non-trivial polynomial has Lebesgue measure zero.*

Despite this justification, there might be statistical issues due to these polynomials being *close* to zero, and thus hard to discern from true zeros. Thus, in causal structure learning, we aim to precisely understand *which* conditional independences in $\mathbb{P}_{\mathcal{X}}$ should imply d-separations in $\mathcal{G}$ - often, only a small subset of conditional independences actually get used by the algorithm. Two particularly important weaker versions of the faithfulness assumptions are adjacency-faithfulness and orientation-faithfulness.

Definition 5.2. *We say that $\mathbb{P}_{\mathcal{X}}$ is* **adjacency-faithful** *to $\mathcal{G}$ if $X_i \in \mathrm{adj}_{\mathcal{G}}(X_j)$ implies that there is no set $\mathbf{S}$ for which $X_i \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_j \mid \mathbf{S}$.*

Algorithm 1 `PC-Skeleton`

Input: Conditional independence tester H
Output: Skeleton $\widehat{\mathcal{G}}$, separator function s
Let $\widehat{\mathcal{G}}$ be the complete graph and s be the empty function.
Let $d = 0$.
while $\exists\ X_i - X_j$ in $\widehat{\mathcal{G}}$ such that $|\operatorname{adj}_{\widehat{\mathcal{G}}}(X_i) \setminus \{X_j\}| \geq d$ **do**
  for $X_i - X_j$ in $\widehat{\mathcal{G}}$ **do**
    if $\exists\ \mathbf{S} \subseteq \operatorname{adj}_{\mathcal{G}}(X_i) \setminus \{X_j\}$ or $\mathbf{S} \subseteq \operatorname{adj}_{\mathcal{G}}(X_j) \setminus \{X_i\}$ such that $|\mathbf{S}| = d$ and $X_i \perp\!\!\!\perp_H X_j \mid \mathbf{S}$ **then**
      Remove $X_i - X_j$ from $\widehat{\mathcal{G}}$
      Assign $s(X_i, X_j) = \mathbf{S}$
    end if
  end for
  Let $d = d + 1$
end while
return $\widehat{\mathcal{G}}, s$

Definition 5.3. *We say that $\mathbb{P}_{\mathcal{X}}$ is* **orientation-faithful** *to $\mathcal{G}$ if, for all $X_i - X_k - X_j$ in $\mathcal{G}$ with $X_i \notin \operatorname{adj}_{\mathcal{G}}(X_j)$, we have:*

- *$X_i \to X_k \leftarrow X_j$ implies that for any $\mathbf{S}$ such that $X_i \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_j \mid \mathbf{S}$, we have $X_k \notin \mathbf{S}$ and*
- *otherwise, for any $\mathbf{S}$ such that $X_i \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_j \mid \mathbf{S}$, we have $X_k \in \mathbf{S}$*

We say that $\mathbb{P}_{\mathcal{X}}$ is **restricted-faithful** *to $\mathcal{G}$ if it is both adjacency-faithful and orientation-faithful to $\mathcal{G}$.*

5.2 The PC algorithm

We are finally ready to introduce the *PC algorithm* (PC stands for "Peter-Clark", the first names of Peter Spirtes and Clark Glymour, the developers of the algorithm). We will introduce the abstraction of a **conditional independence tester**: a function, which given two variables X_i, X_j and a set $\mathbf{S}$, returns a boolean value `True` or `False`. In the *noiseless* (also called *population*) version of the PC algorithm, which assumes direct access to the distribution $\mathbb{P}_{\mathcal{X}}$, we may define a conditional independence tester which simply checks whether $X_i \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_j \mid \mathbf{S}$. In the *sample* version of the PC algorithm, several hypothesis tests exist for testing conditional independence, with different guarantees under different sets of assumptions.

5.2.1 A description of the PC algorithm

The PC algorithm, in Algorithm 3, consists of two phases correspond to the two graphical properties that define Markov equivalence. In its first phase, found in Algorithm 1, the PC algorithm learns the skeleton of $\mathcal{G}$. In its second phase, found in Algorithm 2, the PC algorithm adds orientations to this skeleton.

To learn the skeleton of $\mathcal{G}$, the PC algorithm deletes edges in "rounds". If we assume orientation faithfulness for the conditional independence tester H, then $X_i \perp\!\!\!\perp_H X_j \mid \mathbf{S}$ implies that we can delete the edge $X_i - X_j$. For computational efficiency, we do not want to check all sets $\mathbf{S}$, so each round consider only sets of size d, where $d = 0$ in the first round and increments by one in each following round. Moreover, by Corollary 4.1, it suffices to check only those sets $\mathbf{S}$ which can possibly be parent sets of either X_i or X_j.

The first phase also saves which sets separated each pair X_i, X_j that are non-adjacent in the skeleton. The second phase uses these sets to determine unshielded colliders, orienting some edges. Then, it applies the Meek rules repeatedly to construct the essential graph.

Theorem 5.1. *Suppose that $\mathbb{P}_{\mathcal{X}}$ factorizes according to $\mathcal{G}$, and that $\mathbb{P}_{\mathcal{X}}$ is restricted-faithful to $\mathcal{G}$. Then, in the noiseless case, the PC algorithm returns $\mathcal{E}(\mathcal{G})$.*

Algorithm 2 `PC-Orient`

Input: Skeleton $\widehat{\mathcal{G}}$, separator function s
Output: Essential graph $\widehat{\mathcal{G}}$
for $X_i - X_k - X_j$ in $\widehat{\mathcal{G}}$ with X_i and X_j non-adjacent **do**
  if $X_k \notin s(X_i, X_j)$ **then**
    Let $X_i \to X_k \leftarrow X_j$ in $\widehat{\mathcal{G}}$
  end if
end for
Return $\texttt{MPDAG}(\widehat{\mathcal{G}})$

Algorithm 3 `PC`

Input: Conditional independence tester H
Output: Essential graph $\widehat{\mathcal{G}}$
Let $\widehat{\mathcal{G}}, s = \texttt{PC-Skeleton}(H)$
Let $\widehat{\mathcal{G}} = \texttt{PC-Orient}(\widehat{\mathcal{G}}, s)$
Return $\widehat{\mathcal{G}}$

Proof. First, we prove that the skeleton phase of the PC algorithm is correct. Assume that we have not incorrectly removed any edges, and that we remove the edge $X_i - X_j$. Then we have a set $\mathbf{S}$ such that $X_i \perp\!\!\!\perp_H X_j \mid \mathbf{S}$, and thus by adjacency faithfulness, X_i is not adjacent to X_j in $\mathcal{G}$. Thus, we never incorrectly remove any edges.

Conversely, assume that X_i is not adjacent to X_j in $\mathcal{G}$. Then, by Corollary 4.1, we can assume without loss of generality that $X_i \perp\!\!\!\perp_H X_j \mid \mathrm{pa}_{\mathcal{G}}(X_i)$. The algorithm will reach the round where $d = |\mathrm{pa}_{\mathcal{G}}(X_i)|$ since we never incorrectly remove an edge $X_k - X_i$ for $X_k \in \mathrm{pa}_{\mathcal{G}}(X_i)$. In this round, we will consider $\mathbf{S} = \mathrm{pa}_{\mathcal{G}}(X_i) \subseteq \mathrm{adj}_{\mathcal{G}}(X_i) \setminus \{X_j\}$ and see that $X_i \perp\!\!\!\perp_H X_j \mid \mathbf{S}$, thus removing the edge $X_i - X_j$.

Now, we show that the orientation phase is correct. If $X_i \to X_k \leftarrow X_j$, then $X_k \notin s(X_i, X_j)$ by the first condition of orientation faithfulness. Thus, we do not fail to orient any unshielded colliders. Conversely, if $\langle X_i, X_k, X_j \rangle$ is a non-collider, then $X_k \in s(X_i, X_j)$ by the second condition of orientation faithfulness.

The remaining orientations are given by appealing to Theorem 4.3. □

Lemma 5.1. *Let $\mathcal{G}$ be a DAG with maximum degree Δ. Then, in the noiseless case,* `PC-Skeleton` *terminates with $d \leq \Delta + 1$.*

Proof. Let $\Delta_{\texttt{in}}$ denote the maximum in-degree of $\mathcal{G}$. At the round when $d = \Delta_{\texttt{in}}$, we have that $\widehat{\mathcal{G}}$ equals the skeleton of $\mathcal{G}$. Thus, $|\mathrm{adj}_{\widehat{\mathcal{G}}}(X_i)| \leq \Delta$ for all X_i after this round. Thus when $d = \Delta + 1$, the while loop breaks. □

Finally, we establish the run-time of the PC algorithm. We will use the notion of **query complexity**, which asks how many times the algorithm makes a query to the conditional independence test. This allows us to abstract away what the run-time of the conditional independence tester and obtain a more general result.

Theorem 5.2. *Let $\mathcal{G}$ be a DAG on p nodes with maximum degree Δ. Then, in the noiseless case, the query complexity of the PC algorithm is $\mathcal{O}(p^{\Delta+2})$.*

Proof. At each round, we consider at most $\mathcal{O}(p^2)$ pairs of nodes. For each pair of node, we consider at most $\binom{p}{d'}$ sets $\mathbf{S}$ in the round where $d = d'$. The algorithm terminates after Δ rounds. Thus, for each pair, we consider at most $\sum_{d'=0}^{\Delta} \binom{p}{d'} = \mathcal{O}(p^{\Delta})$ sets. In total, we make at most $\mathcal{O}(p^{\Delta+2})$ queries to H. □

Conditional independence testing. In practice, one has many choices for the conditional independence tester H. If the data is assumed to be linear Gaussian, then the conditional independence $X_i \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_j \mid \mathbf{S}$ is equivalent to the partial correlation $\rho(X_i, X_j \mid \mathbf{S})$ being equal to zero. Given a sample partial correlation

$\widehat{\rho}(X_i, X_j \mid \mathbf{S})$, there is a standard hypothesis test for the null hypothesis $\rho(X_i, X_j \mid \mathbf{S}) = 0$, based on the Fisher z-transformation of $\widehat{\rho}(X_i, X_j \mid \mathbf{S})$. Note that computing the sample partial correlation for $|\mathbf{S}| = d$ takes $\mathcal{O}(d^3)$ time, which is included in a complete runtime analysis of the PC algorithm. In the nonparametric setting, there are several conditional independence tests, see e.g. Zhang et al. (2011) and Shah and Peters (2020). Many of these tests are kernel-based, and thus their computational complexity scales with the number of samples n, creating practical challenges to scaling.

5.3 Sparsity-based Scores

We now begin our discussion of score-based approaches. We discuss these approaches from the lens of *sparsity*: finding the sparsest graph which fits the observed data. Through this lens, we separate the structure learning problem into two parts: adjacencies and orientations. In the PC algorithm, we found adjacencies first and added orientations based on unshielded colliders and the Meek rules. Here, we take the reverse approach. We start with a permutation π which dictates the orientations of any edges present in the estimated graph, and find the sparsest graph consistent with π which fits the observed data. Then, we search for a permutation π^* for which the corresponding graph is as sparse as possible.

5.3.1 Minimal I-MAPs

Definition 5.4. *Let $\mathcal{G}$ be a DAG and $\mathbb{P}_{\mathcal{X}}$ be a distribution. We say that $\mathcal{G}$ is an* **independence map** *(I-MAP) of $\mathbb{P}_{\mathcal{X}}$ if $\mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G}) \subseteq \mathcal{I}_{\perp\!\!\!\perp}(\mathbb{P}_{\mathcal{X}})$, i.e., $\mathbb{P}_{\mathcal{X}}$ is Markov to $\mathcal{G}$. We say that $\mathcal{G}$ is a* **minimal I-MAP** *of $\mathbb{P}_{\mathcal{X}}$ if no strict subgraph of $\mathcal{G}$ is an I-MAP of $\mathbb{P}_{\mathcal{X}}$.*

We use the same terminology for graphs, e.g., we say that $\mathcal{G}$ is an I-MAP of $\mathcal{G}'$ if $\mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G}) \subseteq \mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G}')$.

Given a permutation, we want a procedure which constructs a minimal I-MAP of $\mathbb{P}_{\mathcal{X}}$ which is consistent with the permutation. A naïve procedure could enumerate over all 2^p graphs which are consistent with the permutation. However, we can define a much more efficient procedure, which performs one conditional independence query for each pair of nodes $X_i <_\pi X_j$. For this procedure to return a minimal I-MAP, we will introduce the assumption that $\mathbb{P}_{\mathcal{X}}$ is a *graphoid*.

Given a permutation π, we will define a procedure which takes in a pair $X_i <_\pi X_j$ and determines whether or not the edge $X_i \to X_j$ is present in the graph associated to that permutation. This procedure requires

Definition 5.5. *A* **semigraphoid** *H over elements $\mathcal{X}$ is a set of disjoint subsets $(\mathbf{A}, \mathbf{B}, \mathbf{S})$, where $(\mathbf{A}, \mathbf{B}, \mathbf{S}) \in H$ is denoted by $\mathbf{A} \perp\!\!\!\perp_H \mathbf{B} \mid \mathbf{S}$, such that the following properties hold:*

- **Symmetry**: $\mathbf{A} \perp\!\!\!\perp_H \mathbf{B} \mid \mathbf{S} \implies \mathbf{B} \perp\!\!\!\perp_H \mathbf{A} \mid \mathbf{S}$
- **Decomposition**: $\mathbf{A} \perp\!\!\!\perp_H \mathbf{B} \mid \mathbf{S} \implies \mathbf{A} \perp\!\!\!\perp_H \mathbf{B}' \mid \mathbf{S}$ *for* $\mathbf{B}' \subseteq \mathbf{B}$
- **Weak Union**: $\mathbf{A} \perp\!\!\!\perp_H \mathbf{B} \mid \mathbf{S} \implies \mathbf{A} \perp\!\!\!\perp_H \mathbf{B}_1 \mid \mathbf{S} \cup \mathbf{B}_2$ *for* $\mathbf{B}_1 \subseteq \mathbf{B}$ *where* $\mathbf{B}_2 = \mathbf{B} \setminus \mathbf{B}'$
- **Contraction**: $\mathbf{A} \perp\!\!\!\perp_H \mathbf{B}_1 \mid \mathbf{S} \cup \mathbf{B}_2$ *and* $\mathbf{A} \perp\!\!\!\perp_H \mathbf{B}_2 \mid \mathbf{S} \implies \mathbf{A} \perp\!\!\!\perp_H \mathbf{B} \mid \mathbf{S}$ *and for* $\mathbf{B} = \mathbf{B}_1 \cup \mathbf{B}_2$

A **graphoid** *is a semigraphoid such that the following property also holds:*

- **Intersection**: $\mathbf{A} \perp\!\!\!\perp_H \mathbf{B}_1 \mid \mathbf{S} \cup \mathbf{B}_2$ *and* $\mathbf{A} \perp\!\!\!\perp_H \mathbf{B}_2 \mid \mathbf{S} \cup \mathbf{B}_1 \implies \mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid \mathbf{S}$ *for* $\mathbf{B} = \mathbf{B}_1 \cup \mathbf{B}_2$

Remark 5.2. *The set $\mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G})$ for either a DAG or an undirected graph $\mathcal{G}$ is a graphoid. The set $\mathcal{I}_{\perp\!\!\!\perp}(\mathbb{P}_{\mathcal{X}})$ is always a semigraphoid, and it is a graphoid if $\mathbb{P}_{\mathcal{X}}$ is strictly positive.*

Note that $\mathbb{P}_{\mathcal{X}}$ can violate the intersection property if it is not positive, as the following example shows.

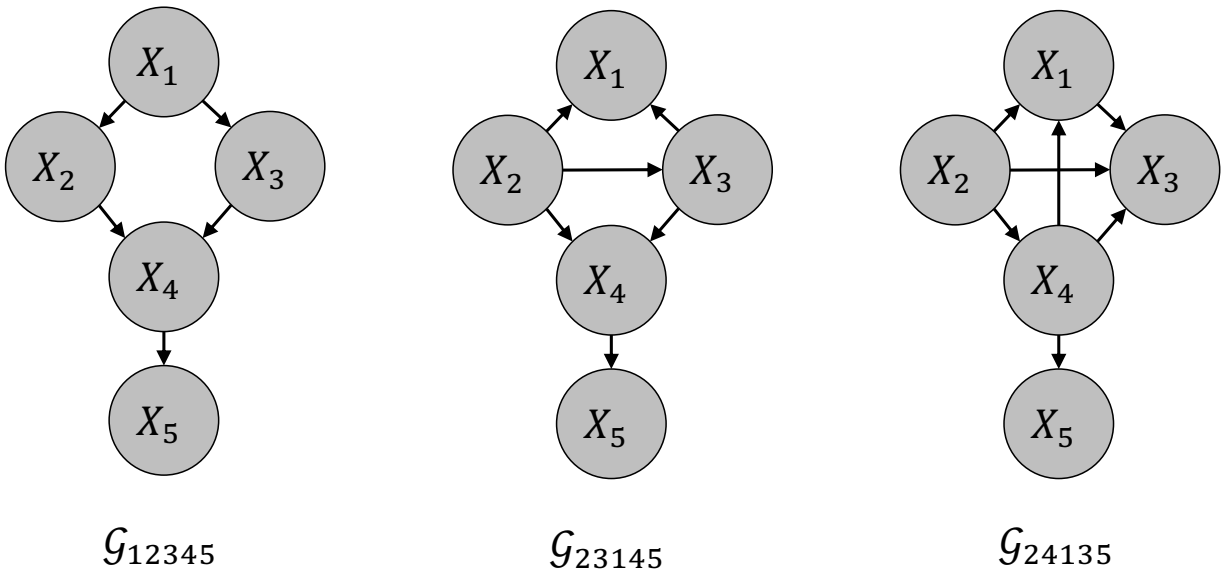

Figure 5.1: Minimal I-MAPs of a distribution $\mathbb{P}_{\mathcal{X}}$ faithful to $\mathcal{G}_{12345}$. See Example 5.3.

Example 5.2. *Let* $B_1 \sim \mathsf{Ber}(0.5)$ *and* $B_2 = B_1$. *Let* $A = \mathsf{Ber}(0.5 + .25B_2)$. *Then* $A \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} B_1 \mid B_2$, *and symmetrically,* $A \perp\!\!\!\perp_{\mathbb{P}} B_2 \mid B_1$. *However,* $A \not\perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} (B_1, B_2)$. *Note that* $\mathbb{P}_{\mathcal{X}}(B_1 = 0, B_2 = 1) = 0$, *i.e.,* $\mathbb{P}$ *is not positive.*

Proposition 5.1. *Let* $\mathbb{P}_{\mathcal{X}}$ *be a distribution and* π *a permutation. Let* $\mathcal{G}_\pi$ *be the DAG with edges* $X_i \to X_j$ *for* $X_i <_\pi X_j$ *if*

$$X_i \not\perp\!\!\!\perp X_j \mid \mathrm{pre}_\pi(X_j) \setminus \{X_i\}.$$

If $\mathbb{P}_{\mathcal{X}}$ *is a graphoid, then* $\mathcal{G}_\pi$ *is a minimal I-MAP of* $\mathbb{P}_{\mathcal{X}}$.

Proof. It suffices to show that $\mathbb{P}_{\mathcal{X}}$ factorizes according to $\mathcal{G}_\pi$. For this, it is sufficient to show that for all X_j,

$$X_j \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} \mathrm{pre}_\pi(X_j) \setminus \mathrm{pa}_{\mathcal{G}}(X_j) \mid \mathrm{pa}_{\mathcal{G}_\pi}(X_j) \tag{5.1}$$

Let $X_u, X_v \in \mathrm{pre}_\pi(X_j) \setminus \mathrm{pa}_{\mathcal{G}}(X_j)$. By the intersection property with $\mathbf{S} = \mathrm{pre}_\sigma(X_j) \setminus \{X_u, X_v\}$, $\mathbf{B}_1 = \{X_u\}$, and $\mathbf{B}_2 = \{X_v\}$, we have

$$X_j \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_u, X_v \mid \mathrm{pre}_\sigma(X_j) \setminus \{X_u, X_v\}$$

Equation (5.1) follows by induction. □

Remark 5.3. *In practice, constructing* $\mathcal{G}_\pi$ *according to the definition in Proposition 5.1 is not ideal: in a graph with* p *variables, the conditioning sets used in conditional independence tests become of size* $\mathcal{O}(p)$ *for nodes later in the permutation. More practical constructions can be made which take advantage of sparsity in the underlying graph.*

Example 5.3. *Suppose that* $\mathbb{P}_{\mathcal{X}}$ *is faithful to* $\mathcal{G}_{12345}$ *in Figure 5.1. Then* $\mathcal{G}_{23145}$ *has the extra adjacency* $X_2 \to X_3$, *since* $X_2 \not\perp\!\!\!\perp X_3$. $\mathcal{G}_{24135}$ *has two extra adjacencies,* $X_2 \to X_3$, *since* $X_2 \not\perp\!\!\!\perp X_2 \mid X_4$, *and* $X_4 \to X_1$, *since* $X_4 \not\perp\!\!\!\perp X_1 \mid X_2$.

Proposition 5.2. *Let* $\mathbb{P}_{\mathcal{X}}$ *be Markov to* $\mathcal{G}$, *and let* π *be a topological order of* $\mathcal{G}$. *Then* $\mathcal{G}_\pi \subseteq \mathcal{G}$.

Proof. Let $X_i <_\pi X_j$ with $X_i \notin \mathrm{pa}_{\mathcal{G}}(X_j)$. Let $\mathbf{V} = \overline{\mathrm{pre}}_\pi(X_j)$, which is ancestral by definition of a topological order. Then $\mathbb{P}_{\mathcal{X}}(\mathbf{V})$ factorizes according to $\mathcal{G}[\mathbf{V}]$. All paths from X_i to X_j are blocked in $\mathcal{G}[\mathbf{V}]$ by $\mathrm{pre}_\pi(X_j) \setminus \{X_i\}$, since this set includes $\mathrm{pa}_{\mathcal{G}}(X_i)$. Thus, $X_i \perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_j \mid \mathrm{pre}_\pi(X_j)$, hence G_π does not contain the edge $X_i \to X_j$. □

5.3.2 Sparsest Permutation

Lemma 5.2. *Let $\mathbb{P}_{\mathcal{X}}$ be Markov to a DAG $\mathcal{G}$. Let $\widehat{\pi} \in \arg\min_{\pi} |\mathcal{G}_{\pi}|$, where $|\mathcal{G}_{\pi}|$ denotes the number of edges in $\mathcal{G}_{\pi}$. Then*

(a) $|\mathcal{G}_{\widehat{\pi}}| \leq |\mathcal{G}|$

(b) If $\mathbb{P}_{\mathcal{X}}$ is adjacency-faithful to $\mathcal{G}$, then $\text{skel}(\mathcal{G}_{\widehat{\pi}}) = \text{skel}(\mathcal{G})$.

(c) If $\mathbb{P}_{\mathcal{X}}$ is restricted-faithful to $\mathcal{G}$, then $\mathcal{G}_{\widehat{\pi}} \in \mathcal{M}(\mathcal{G})$.

Proof.
(a) By Proposition 5.2.

(b) By (a), $\text{skel}(\mathcal{G}_{\widehat{\pi}})$ cannot have more edges than $\mathcal{G}$, so it suffices to show that $\text{skel}(\mathcal{G}) \subseteq \text{skel}(\mathcal{G}_{\widehat{\pi}})$. This is guaranteed by adjacency faithfulness and the definition of $\mathcal{G}_{\widehat{\pi}}$.

(c) By (b), we have $\text{skel}(\mathcal{G}_{\widehat{\pi}}) = \text{skel}(\mathcal{G})$. By Theorem 4.2, it suffices to show that $\mathcal{G}_{\widehat{\pi}}$ and $\mathcal{G}$ also have the same unshielded colliders.

Let $X_i \to X_k \leftarrow X_j$ be unshielded in $\mathcal{G}$. Consider π such that $X_k <_{\pi} X_j$. If $X_i <_{\pi} X_j$, then $X_k \in \mathbf{S}$ for $\mathbf{S} = \text{pre}_{\pi}(X_j) \setminus \{X_i\}$. By orientation-faithfulness, we have that $X_i \not\perp\!\!\!\perp_{\mathbb{P}_{\mathcal{X}}} X_j \mid \mathbf{S}$, so $X_i \to X_j$ in $\mathcal{G}_{\pi}$. Symmetrically, if $X_j <_{\pi} X_i$, we have $X_j \to X_i$ in $\mathcal{G}_{\pi}$. Thus, $\text{skel}(\mathcal{G}_{\pi}) \neq \text{skel}(\mathcal{G})$. The case where $X_k <_{\pi} X_i$ is symmetric. Thus, if $\text{skel}(\mathcal{G}_{\widehat{\pi}}) = \text{skel}(\mathcal{G})$, then we must have $X_i <_{\widehat{\pi}} X_k$ and $X_j <_{\pi} X_k$, i.e., any unshielded collider in $\mathcal{G}$ is an unshielded collider in $\mathcal{G}_{\widehat{\pi}}$.

Conversely, suppose $X_i - X_k - X_j$ is a non-collider in $\mathcal{G}$, with X_i and X_j non-adjacent. Consider π such that $X_i <_{\pi} X_k$ and $X_j <_{\pi} X_k$. If $X_i <_{\pi} X_j$, then $X_k \notin \mathbf{S}$ for $\mathbf{S} = \text{pre}_{\pi}(X_j) \setminus \{X_i\}$, so by orientation faithfulness, $X_i \to X_j$ in $\mathcal{G}_{\pi}$. Thus, $\text{skel}(\mathcal{G}_{\pi}) \neq \text{skel}(\mathcal{G})$. The case where $X_j <_{\pi} X_i$ is symmetric. Thus, if $\text{skel}(\mathcal{G}_{\widehat{\pi}}) = \text{skel}(\mathcal{G})$, then we must have at least one of $X_k <_{\pi} X_i$ or $X_k <_{\pi} X_j$, i.e., any unshielded non-collider in $\mathcal{G}$ is an unshielded non-collider in $\mathcal{G}_{\widehat{\pi}}$. □

5.4 Greedy Search Algorithms

Definition 5.6. *Let $\mathcal{G}$ be a DAG. We call an edge X_i to X_j in $\mathcal{G}$ a* **covered edge** *if $\overline{\text{pa}}_{\mathcal{G}}(X_i) = \text{pa}_{\mathcal{G}}(X_j)$, equivalently, $\text{pa}_{\mathcal{G}}(X_j) = \text{pa}_{\mathcal{G}}(X_i) \cup \{X_i\}$.*

Lemma 5.3. *Let $\mathcal{G}$ be a DAG and let $X_i \to X_j$ be a covered edge. Let $\mathcal{G}'$ be the graph obtained by reversing the edge $X_i \to X_j$ in $\mathcal{G}$, i.e., in $\mathcal{G}'$, we have $X_j \to X_i$. Then $\mathcal{G}'$ is Markov equivalent to $\mathcal{G}$.*

Proof. Note that, by acyclicity, there are no directed paths from X_i to X_j except for the edge $X_i \to X_j$. Thus, no new cycles are introduced by reversing $X_i \to X_j$, i.e., $\mathcal{G}'$ is a DAG.

The skeletons of $\mathcal{G}$ and $\mathcal{G}'$ are equal. So are their unshielded colliders: the only colliders in $\mathcal{G}'$ that are not in $\mathcal{G}$ are of the form $X_k \to X_i \leftarrow X_j$, but these are all shielded since by the assumption that the edge is covered. Symmetrically, the only colliders in $\mathcal{G}$ that are not in $\mathcal{G}$ are shielded. Thus, by Theorem 4.2, $\mathcal{G}$ and $\mathcal{G}'$ are Markov equivalent. □

In fact, it is well-known that for *any* $\mathcal{G}'$ that is Markov equivalent to $\mathcal{G}$, there exists a sequence of covered edge reversals from $\mathcal{G}$ to $\mathcal{G}'$. This is a special case of the upcoming result, which is one of the major results in score-based causal structure learning.

Definition 5.7. *Let $\mathcal{G}$ be a DAG and $\mathcal{H}$ be an I-MAP of $\mathcal{G}$, i.e., $\mathcal{I}_{\perp\!\!\!\perp}(\mathcal{H}) \subseteq \mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G})$. A* **Chickering sequence** *from $\mathcal{G}$ to $\mathcal{H}$ is a sequence of DAGs $\mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_M$, with $\mathcal{G}_0 = \mathcal{G}$ and $\mathcal{H} = \mathcal{G}_M$, such that $\mathcal{G}_m$ is obtained from $\mathcal{G}_{m-1}$ by either an edge addition or a covered edge reversal.*

Theorem 5.3 (Chickering (2002))**.** *Let $\mathcal{H}$ be an I-MAP of $\mathcal{G}$. Then there is a Chickering sequence from $\mathcal{G}$ to $\mathcal{H}$.*

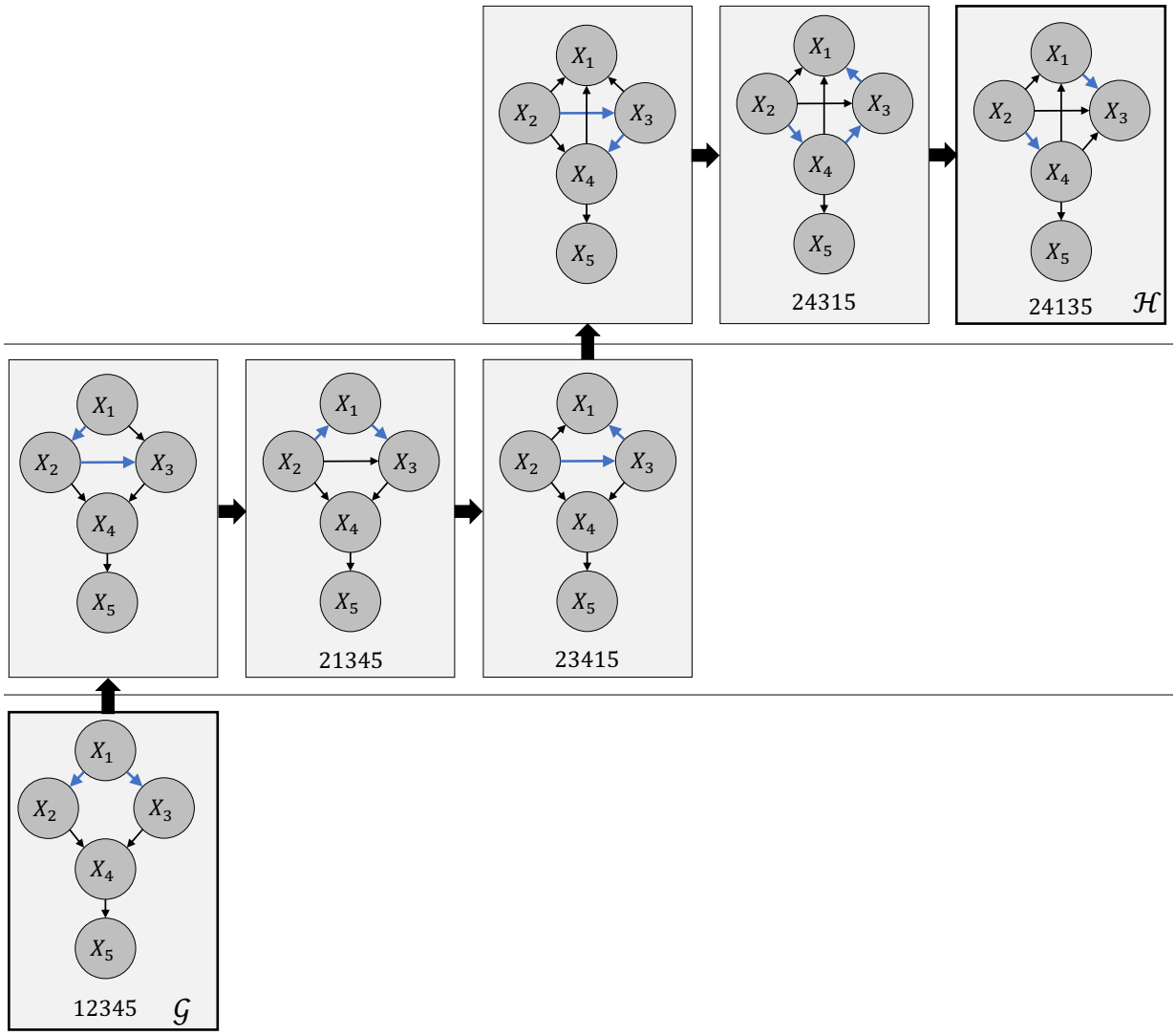

Figure 5.2: A Chickering sequence from $\mathcal{G}$ to $\mathcal{H}$. Blue edges denoted covered edges.

Remark 5.4. *Chickering (2002) offers a constructive proof of Theorem 5.3. Given a pair $\mathcal{G}_m$ and $\mathcal{H}$, the author introduces an operation which either reverses a covered edge in $\mathcal{G}_m$ or adds an edge to $\mathcal{G}_m$, obtaining a new graph $\mathcal{G}_{m+1}$ such that $\mathcal{H}$ is still an I-MAP of $\mathcal{G}_{m+1}$. The proof defines a distance $d(\mathcal{G}, \mathcal{H})$ between graphs, and shows that $d(\mathcal{G}_{m+1}, \mathcal{H}) < d(\mathcal{G}_m, \mathcal{H})$. Thus, repeated application of the operation must terminate when $d(\mathcal{G}_M, \mathcal{H}) = 0$, i.e., $\mathcal{G}_M = \mathcal{H}$. Since this proof would take a lecture of its own to go through, we will not cover it in this course.*

Example 5.4. *An example Chickering sequence can be found in Figure 5.2.*

Corollary 5.1. *Let $\mathcal{H}$ be an I-MAP of $\mathcal{G}$ such that $\mathcal{I}_{\perp\!\!\!\perp}(\mathcal{H}) \neq \mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G})$. Then there exists $\mathcal{H}' \in \mathcal{M}(\mathcal{H})$ such that $\mathcal{H}'$ is not a minimal I-MAP of $\mathcal{G}$.*

Proof. By Theorem 5.3, there exists a Chickering sequence from $\mathcal{G}$ to $\mathcal{H}$. Since $\mathcal{I}_{\perp\!\!\!\perp}(\mathcal{H}) \neq \mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G})$, we must have $|\mathcal{H}| > |\mathcal{G}|$. Let $\mathcal{G}_m$ be the last graph in this sequence such that $|\mathcal{G}_m| = |\mathcal{H}| - 1$. Then $\mathcal{H}' = \mathcal{G}_{m+1}$ is not a minimal I-MAP. □

Two greedy search procedures exploit this corollary. First, the **Greedy Sparsest Permutation** (GSP)

algorithm heuristically picks an initial permutation π_0, and uses conditional independence tests to estimate $\widehat{\mathcal{G}}_{\pi_0}$. Then, GSP uses a depth-first search (via covered edge reversals) over the Markov equivalence class of $\widehat{\mathcal{G}}_{\pi_0}$ to find some graph for which an edge can be deleted. This can be seen as a "backwards" traversal of the Chickering sequence. Further, the use of covered edge reversals allows for very few conditional independence tests to be performed at each step, i.e., the minimal I-MAP $\mathcal{G}_\pi$ does not have to be re-computed at each iteration. GSP can be seen as a "hybrid" method, using conditional independence tests for each fixed permutation, with a score-based search over permutations which looks to minimize the number of edges.

The **Greedy Equivalence Search** (GES) algorithm has both a forward and a backward phase. The forward phase, which starts from the empty graph and adds edges, is designed to end at an I-MAP of the $\mathbb{P}_{\mathcal{X}}$, while the backward phase follows a similar path "down" the Chickering sequence. Instead of depth-first search within a Markov equivalence class, GES operates over the search essential graphs, and thus involves somewhat complex rules for adding and deleting edges in the forward and backward phases. Finally, we note that GES is usually run with a score function which takes the form of either a penalized maximum likelihood or a marginal likelihood (where some prior is defined over graphs and parameters).

5.5 Additional Remarks

- **Causal structure learning with uncertainty**: The algorithms that we discussed return "point estimates", i.e., they output a single estimated graph $\widehat{\mathcal{G}}$. However, in practice, one may want to quantify the uncertainty over this estimated graph. A frequentist approach is to use a *DAG-bootstrap* (Friedman et al., 1999), which re-runs one of these algorithms with bootstrapped samples of the data (i.e., random sampling with replacement). Alternatively, one may take a Bayesian approach, defining priors $\mathbb{P}(\mathcal{G})$ and $\mathbb{P}(\Theta \mid \mathcal{G})$ over causal graphs and their parameters. You can read more about some of these approaches in Section 4.2 of my review (Squires and Uhler, 2022).

- **Causal structure learning with interventions**: Many causal structure learning algorithms can be re-purposed for learning with interventional data. As indicated by our definition of an interventional graph in Chapter 1, this can be done by adding intervention variables as nodes Mooij et al. (2020). Such an approach also allows one to learn the *targets* of an intervention, i.e. the targets do not need to be known beforehand. The main criterion for a structure learning algorithm to be re-purposed to handle interventional data is that it can incorporate *background knowledge*. In particular, the algorithm must be able to incorporate the facts that intervention variables are upstream of the "normal" variables, and that intervention variables (if there are multiple) have a deterministic relation (if $\zeta^I = 1$ for some sample, then $\zeta^{I'} = 0$ for any other intervention I'). An example of such an adaptation of the GSP algorithm is in Squires et al. (2020).

- **Causal structure learning with latent variables**: Here, we have covered causal structure learning for DAG models, i.e., in the setting of a Markovian structural causal model. There are many methods extending beyond DAGs, e.g. to handle latent variables. There are several options for graphical models which capture latent variables, including ADMGs (discussed in Chapter 1) or **maximal ancestral graphs** (MAGs). For example, the **Fast Causal Inference** (FCI) algorithm and its variants are analogues of the PC algorithm for learning MAGs. One barrier to proving the consistency of greedy search algorithms is that there is no proof that a Chickering sequence always exists between a MAG $\mathcal{G}$ and an I-MAP $\mathcal{H}$ of that MAG. For example, **Greedy Sparsest Poset** (GSPo) algorithm is a variant of the GSP algorithm for learning MAGs, but its consistency remains unproven.

Bibliography

Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554.

Friedman, N., Goldszmidt, M., and Wyner, A. (1999). Data analysis with bayesian networks: a bootstrap

approach. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 196–205.

Mooij, J. M., Magliacane, S., and Claassen, T. (2020). Joint causal inference from multiple contexts.

Shah, R. D. and Peters, J. (2020). The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, 48(3):1514–1538.

Squires, C. and Uhler, C. (2022). Causal structure learning: a combinatorial perspective. *arXiv preprint arXiv:2206.01152*.

Squires, C., Wang, Y., and Uhler, C. (2020). Permutation-based causal structure learning with unknown intervention targets. In *Conference on Uncertainty in Artificial Intelligence*, pages 1039–1048. PMLR.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. (2011). Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 804–813.