Overview of Methods

1. Constraint-based structure learning
   – Based on tests for conditional independencies in data

2. Score-based structure learning
   – Optimization problem: find structure that optimizes a score (typically using heuristic search)

3. Bayesian model averaging approaches
   – Generates an ensemble of possible structures
   – Can be done efficiently for special cases
Constraint-Based Approaches

• Based on variants of algorithms for building I-maps and perfect maps
• Need some way to answer independence queries eg. \((X \perp Y | Z)\)

Build-Minimal-I-Map
To find a minimal I-map for a distribution \(P\):
• Pick a variable ordering
• For each variable \(X_i\) in the ordering:
  • Find some minimal subset \(U\) of \(\{X_1, ..., X_{i-1}\}\) to be \(X_i\)'s parents in \(\mathcal{G}\) such that
    \(\{X_i \perp \{X_1, ..., X_{i-1}\} - U | U\}\)
Constraint-Based Approaches

Recall this algorithm from Section 3.4.1

Build-Minimal-I-Map
To find a minimal I-map for a distribution $P$:
• Pick a variable ordering
• For each variable $X_i$ in the ordering:
  • Find some minimal subset $U$ of $\{X_1, \ldots, X_{i-1}\}$ to be $X_i$'s parents in $\mathcal{G}$ such that
    \[ \{X_i \perp \{X_1, \ldots, X_{i-1}\} \mid U \} \]

Problem #1: Final structure is sensitive to the ordering
Problem #2: Conditional independence query involves a large number of variables – hard to estimate from empirical data
Problem #3: Lots of subsets to search over

Constraint-Based Approaches

• Won’t learn a single network
• Instead, we will learn an I-equivalence class

Two graph structures $\kappa_1$ and $\kappa_2$ are I-equivalent if $I(\kappa_1) = I(\kappa_2)$
Constraint-Based Approaches

- Will use a class Partially Directed Acyclic Graph (PDAG) to represent this class
- A PDAG is an acyclic graph with both directed and undirected edges e.g.

\[ \begin{align*}
X & \rightarrow Y \\
Y & \rightarrow Z
\end{align*} \]

Goal: reconstruct the network that best matches the domain without a prespecified variable ordering and using a polynomial number of independence tests that involve a bounded number of variables

Constraint-Based Approaches

Assumptions:
- Each node has \( \leq d \) parents
- Independence procedure can answer any query involving up to \( 2d + 2 \) variables
- The underlying distribution \( P^* \) is faithful to \( G^* \)

Recall that faithfulness means that any independence in the distribution \( P^* \) is reflected in the \( d \)-separation properties of the graph \( G^* \).
Learning PDAGs

• Goal: learn a DAG $\mathcal{g}^*$ that is a perfect map (P-map) of distribution $P$
• $\mathcal{g}^*$ is not unique: a distribution can have many P-maps, but they are all I-equivalent
e.g. $(X \perp Y \mid Z)$

\[X \rightarrow Z \rightarrow Y\]
\[X \leftarrow Z \rightarrow Y\]
\[X \rightarrow Z \leftarrow Y\]

Can’t learn a single network

Learning PDAGs

• Want to return the entire equivalence class with some compact representation
• Theorem 3.8: two DAGs are I-equivalent if they share the same undirected skeleton and the same set of immoralities
• Can identify I-equivalence class by:
  1. Identify the undirected skeleton
  2. Identify independence properties
Learning PDAGs

Identifying the undirected skeleton

- **Intuition**: If X and Y are adjacent in $G^*$ then we cannot make them conditionally independent given some set of variables $U$

- Suppose you do find $U$ such that $P \models (X \perp Y | U)$. We call set $U$ a **witness** of their independence

- If $G^*$ has bounded in-degree $d$, then we do not need to consider witness sets larger than $d$

---

**Procedure** Build-PMap-Skeleton

\[
\begin{align*}
\mathcal{X} &= \{X_1, ..., X_n\}, \quad \text{// Set of random variables} \\
P, \quad \text{// Distribution over } \mathcal{X} \\
d \quad \text{// Bound on witness set} \\
\end{align*}
\]

Let $\mathcal{H}$ be the complete undirected graph over $\mathcal{X}$

for $X_i, X_j$ in $\mathcal{X}$

\[
U_{X_i,X_j} \leftarrow \emptyset
\]

for $U \in \text{Witnesses}(X_i, X_j, \mathcal{H}, d)$

\[
\text{// Consider } U \text{ as a witness set for } X_i, X_j
\]

if $P \models (X_i \perp X_j | U)$ then

\[
U_{X_i,X_j} \leftarrow U
\]

Remove $X_i, X_j$ from $\mathcal{H}$

break

return $(\mathcal{H}, U_{X_i,X_j}; i, j \in \{1, ..., n\})$
Learning PDAGs

**Procedure** Build-PMap-Skeleton (  
\[ X = \{X_1, \ldots, X_n\}; \] // Set of random variables  
\[ P, \] // Distribution over \( X \)  
\[ d \] // Bound on witness set  
)

Let \( \mathcal{H} \) be the complete undirected graph over \( X \) for \( X_i, X_j \in X \) 
\[ U_{X_i, X_j} \leftarrow \emptyset \] for \( U \in \text{Witnesses}(X_i, X_j, \mathcal{H}, d) \) 
  // Consider \( U \) as a witness set for \( X_i, X_j \)  
  if \( P \models (X_i \perp X_j | U) \) then 
    \[ U_{X_i, X_j} \leftarrow U \] 
    Remove \( X_i - X_j \) from \( \mathcal{H} \) 
  break 

return \( (\mathcal{H}, \{U_{X_i, X_j}; i, j \in \{1, \ldots, n\}) \) 


Several speedups possible here e.g. restrict size of \( U \) to be \( d \)

Learning PDAGs

• **Build-PMap-Skeleton** has complexity \( O(n^{d+2}) \):
  – Considers \( O(n^2) \) pairs
  – For each pair, we perform \( O((n - 2)^d) \) independence tests

• Note: **Build-PMap-Skeleton** may fail if \( P \) does not have a P-map
Learning PDAGs

Identifying Immoralities
• We have the undirected skeleton
• Need to determine edge directions
• Use immoralities to inform us about edge directions

---

Learning PDAGs

• Consider potential immoralities in the skeleton eg. $X \rightarrow Z \rightarrow Y$
• A potential immorality is an immorality if and only if $Z$ is not in any witness set $U$ for $X$ and $Y$.

![Diagram](diagram.png)

• If $X \rightarrow Z \rightarrow Y$ is not an immorality, then $Z$ must be in every witness set $U$. 
Learning PDAGs

Procedure Mark-Immoralities (  
\( X = \{X_1, \ldots, X_n\} \),  
\( S \) \, // Skeleton  
\( \{U_{X_i,X_j}; 1 \leq i,j \leq n\} \) \, // Witnesses found by Build-PMap-Skeleton  
)  
\( \mathcal{K} \leftarrow S \)  
for \( X_i, X_j, X_k \) such that \( X_i - X_j - X_k \in S \) and \( X_i - X_k \notin S \)  
\// \( X_i - X_j - X_k \) is a potential immorality  
if \( X_j \notin U_{X_i,X_k} \) then  
Add the orientations \( X_i \rightarrow X_j \) and \( X_j \leftarrow X_k \) to \( \mathcal{K} \)  
return \( \mathcal{K} \)

Note: \( \mathcal{K} \) has directed and undirected edges (called a chain graph or partially directed acyclic graph)

Learning PDAGs

Let \( \mathcal{G} \) be a DAG. A chain graph \( \mathcal{K} \) is a class PDAG of the equivalence class of \( \mathcal{G} \) if it shares the same skeleton as \( \mathcal{G} \), and contains a directed edge \( X \rightarrow Y \) if and only if all \( \mathcal{G}' \) that are I-equivalent to \( \mathcal{G} \) contain the edge \( X \rightarrow Y \)

Represented by chain graph:

\[
X \rightarrow Z \rightarrow Y
\]

Note: this has no directed edges because not all edge orientations are in the equivalence class eg. \( X \rightarrow Z \leftarrow Y \)
Learning PDAGs

Rules for orienting edges in a PDAG

Rule 1:

Rule 2:

Rule 3:

Learning PDAGs

**Procedure** Build-PDAG (  
\[ X = \{X_1, \ldots, X_n\} \] // A specification of the random variables  
\[ P \] // Distribution of interest  
)  

\[ S, \{U_{X_i,X_j}\} \leftarrow \text{Build-PMap-Skeleton}(X, P) \]  
\[ K \leftarrow \text{Find-Immoralities}(X, S, \{U_{X_i,X_j}\}) \]  

**while** not converged  

Find a subgraph in \( K \) matching the left-hand side of a rule (Rules 1-3)  
Replace the subgraph with the right-hand side of the rule  

**return** \( K \)
Independence Tests

How to determine independence?
• Hypothesis tests eg. with two variables X and Y
• Null Hypothesis $H_0$: X and Y are independent
• Alternate Hypothesis $H_1$: X and Y are not independent

Independence Tests

• Accept / Reject the null hypothesis
• False rejection: wrongly rejecting the null hypothesis when it is correct
Independence Tests

Measuring deviance from the null hypothesis
eg.

• Chi-squared statistic

\[ d_{X^2}(D) = \sum_{x,y} \frac{(M[x,y] - M \cdot \hat{p}(x) \cdot \hat{p}(y))^2}{M \cdot \hat{p}(x) \cdot \hat{p}(y)} \]

• Mutual Information

\[ d_I(D) = I_{p,D}(X; Y) = \frac{1}{M} \sum_{x,y} M[x,y] \log \frac{M[x,y]}{M[x]M[y]} \]

Independence Tests

Rule for accepting/rejecting the null hypothesis

\[ R_{d,t}(D) = \begin{cases} 
\text{Accept if } d(D) \leq t \\
\text{Reject if } d(D) > t 
\end{cases} \]

Threshold \( t \) determines the false rejection rate.

\[ p - \text{value}(t) = P(D: d(D) > t|H_0,M) \]

Typically, threshold \( t \) set to give p-value \( \leq 0.05 \)
Summary

Main problem with constraint-based approaches: independence tests aren’t perfect
• Threshold-dependent results
• Multiple hypothesis testing: number of incorrect results can grow large

Leads to errors in resulting PDAG