Structure Learning 1

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. $(X \perp Y | Z)$

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 $\boldsymbol{U}$ of $\{X_1, \ldots, X_{i-1}\}$ to be X_i's parents in $\mathcal{G}$ such that $\{X_i \perp \{X_1, \ldots, X_{i-1}\} - \boldsymbol{U} | \boldsymbol{U}\}$

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 $\boldsymbol{U}$ of $\{X_1, \ldots, X_{i-1}\}$ to be X_i's parents in $\mathcal{G}$ such that $\{X_i \perp \{X_1, \ldots, X_{i-1}\} - \boldsymbol{U} \mid \boldsymbol{U}\}$

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

5

Constraint-Based Approaches

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

Two graph structures κ_1 and κ_2 are I-equivalent if $I(\kappa_1) = I(\kappa_2)$

6

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.

$$X \rightarrow Y - Z$$

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

7

Constraint-Based Approaches

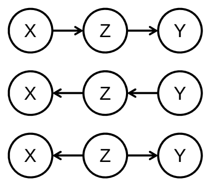
Assumptions:
- Each node has $\leq$ d parents
- Independence procedure can answer any query involving up to 2d + 2 variables
- The underlying distribution P^* is faithful to $\mathcal{G}^*$

Recall that faithfulness means that any independence in the distribution P^* is reflected in the d-separation properties of the graph $\mathcal{G}^*$.

8

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 eg. $(X \perp Y \mid Z)$



Can't learn a single network

9

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

10

Learning PDAGs

<u>Identifying the undirected skeleton</u>

- Intuition: If X and Y are adjacent in $\mathcal{G}^*$ then we cannot make them conditionally independent given some set of variables **U**
- Suppose you do find **U** such that $P \models (X \perp Y \mid \boldsymbol{U})$. We call set **U** a witness of their independence
- If $\mathcal{G}^*$ has bounded in-degree *d*, then we do not need to consider witness sets larger than *d*

11

Learning PDAGs

Procedure Build-PMap-Skeleton (
    $\mathcal{X} = \{X_1, \ldots, X_n\}$, // Set of random variables
    P,   // Distribution over $\mathcal{X}$
    d // Bound on witness set
)
Let $\mathcal{H}$ be the complete undirected graph over $\mathcal{X}$
for X_i, X_j in $\mathcal{X}$
    $\boldsymbol{U}_{X_i, X_j} \leftarrow \emptyset$
    for $\boldsymbol{U} \in Witnesses(X_i, X_j, \mathcal{H}, d)$
        // Consider $\boldsymbol{U}$ as a witness set for X_i, X_j
        if $P \models (X_i \perp X_j \mid \boldsymbol{U})$ **then**
            $\boldsymbol{U}_{X_i, X_j} \leftarrow \boldsymbol{U}$
            Remove $X_i - X_j$ from $\mathcal{H}$
            break
    return $(\mathcal{H}, \{\boldsymbol{U}_{X_i, X_j} : i, j \in \{1, \ldots, n\}\})$

12

Learning PDAGs

Procedure Build-PMap-Skeleton (
  $\mathcal{X} = \{X_1, \ldots, X_n\}$, // Set of random variables
  P, // Distribution over $\mathcal{X}$
  d // Bound on witness set
)
Let $\mathcal{H}$ be the complete undire[cted graph over $\mathcal{X}$]
for X_i, X_j in $\mathcal{X}$
  $\boldsymbol{U}_{X_i,X_j} \leftarrow \emptyset$
  for $\boldsymbol{U} \in Witnesses(X_i, X_j, \mathcal{H}, d)$
    // Consider U as a witness set for X_i, X_j
    if $P \models (X_i \perp X_j \mid \boldsymbol{U})$ **then**
      $\boldsymbol{U}_{X_i,X_j} \leftarrow \boldsymbol{U}$
      Remove $X_i - X_j$ from $\mathcal{H}$
      break
  return $(\mathcal{H}, \{\boldsymbol{U}_{X_i,X_j} : i,j \in \{1, \ldots, n\})$

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. $X - Z - Y$
- A potential immorality is an immorality if and only if Z is not in any witness set $\boldsymbol{U}$ for X and Y.

$$X \rightarrow Z \leftarrow Y$$

- If $X - Z - Y$ is not an immorality, then Z must be in every witness set $\boldsymbol{U}$.

Learning PDAGs

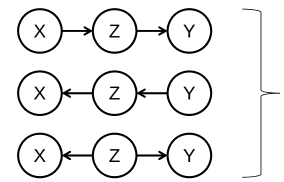
Procedure Mark-Immoralities (
$\mathcal{X} = \{X_1, \dots X_n\}$,
S // Skeleton
$\{\boldsymbol{U}_{X_i,X_j}: 1 \le i, j \le n\}$ // Witnesses found by Build-PMap-Skeleton
)
$\mathcal{K} \leftarrow S$
for X_i,X_j,X_k such that $X_i - X_j - X_k \in S$ and $X_i - X_k \notin S$
// $X_i - X_j - X_k$ is a potential immorality
if $X_j \notin \boldsymbol{U}_{X_i,X_k}$ **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)

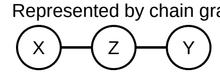
17

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:



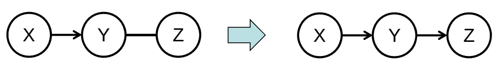
Note: this has no directed edges because not all edge orientations are in the equivalence class eg. $X \rightarrow Z \leftarrow Y$

18

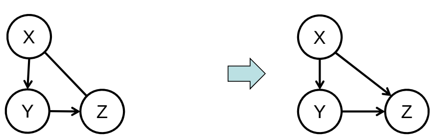
Learning PDAGs

Rules for orienting edges in a PDAG

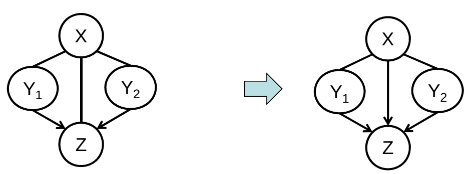
Rule 1:



Rule 2:



Rule 3:



19

Learning PDAGs

Procedure Build-PDAG (
$\mathcal{X} = \{X_1, \dots, X_n\}$ // A specification of the random variables
P // Distribution of interest
)
$S, \{\boldsymbol{U}_{X_i,X_j}\} \leftarrow \text{Build-PMap-Skeleton}(\mathcal{X}, P)$
$\mathcal{K} \leftarrow \text{Find-Immoralities}(\mathcal{X}, S, \{\boldsymbol{U}_{X_i,X_j}\})$
while not converged
Find a subgraph in $\mathcal{K}$ matching the left-hand side of a rule (Rules 1-3)
Replace the subgraph with the right-hand side of the rule
return $\mathcal{K}$

20

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

$$d_{\chi^2}(\mathcal{D}) = \sum_{x,y} \frac{\left(M[x,y] - M \cdot \hat{P}(x) \cdot \hat{P}(y)\right)^2}{M \cdot \hat{P}(x) \cdot \hat{P}(y)}$$

- Mutual Information

$$d_{\boldsymbol{I}}(\mathcal{D}) = \boldsymbol{I}_{\hat{P}_D}(X;Y) = \frac{1}{M} \sum_{x,y} M[x,y] log \frac{M[x,y]}{M[x]M[y]}$$

Independence Tests

Rule for accepting/rejecting the null hypothesis

$$R_{d,t}(\mathcal{D}) = \begin{cases} Accept \;\; if \; d(\mathcal{D}) \leq t \\ Reject \; if \; d(\mathcal{D}) > t \end{cases}$$

Threshold t determines the false rejection rate.

$$p - value(t) = P(\{\mathcal{D}: d(\mathcal{D}) > t\} | H_0, M)$$

Typically, threshold t set to give p-value ≤ 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