Learning the Structure of Bayesian Networks

Kaur Alasoo, Aleksandr Tkatsenko, Jüri Reimand

08.04.09
Learning the Structure of Bayesian Networks

- Reconstructing the Bayesian network from samples of cases.

- Assumptions:
  - The sample is fair
  - all links in the BN are essential
Space of BN Structures is Extremely Large

<table>
<thead>
<tr>
<th>Nodes</th>
<th>DAGs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>543</td>
</tr>
<tr>
<td>5</td>
<td>29281</td>
</tr>
<tr>
<td>6</td>
<td>3800000</td>
</tr>
</tbody>
</table>
Outline

- Constraint-based learning
- Score-based learning
  - Score functions
  - Search procedures
Constraint-Based Learning

- Establish a set of **conditional independencies**.
- Build a network with **d-separation** properties corresponding to the conditional independencies.
3 States of d-separation

\[ m \in X \]

I(A,B,X) - A is d-separated from B given X.
2 Steps of Constraint-Based Learning

- Determine the **skeleton of the network**.
  - Ask the oracle: “*is the variable A d-separated from the variable B given the set X?*”.
  - $I(A,B,X)$ - A is d-separated from B given X.
  - $A \leftarrow B$ is part of the skeleton $\iff \neg I(A,B,X)$ for all X not containing A or B.

- Direct the links.
Directing the Links
Directing the links

* Only conditional independence found is $I(A,B)$

* A and B are not independent given C
Rule 1 [Introduction of v-structures]: If you have three nodes, A, B, C, such that A — B and B — C, but not A — B, then introduce the v-structure \( A \rightarrow C \leftarrow B \) if there exists an \( X \) (possibly empty) such that \( I(A,B,X) \) and \( C \notin X \).
Rules for directing the links

* Rule 1 [Introduction of v-structures]: If you have three nodes, A, B, C, such that A → B and B → C, but not A → B, then introduce the v-structure A → C ← B if there exists an X (possibly empty) such that I(A,B,X) and C ∉ X.
Rules for directing the links

* Rule 1 [Introduction of v-structures]: If you have three nodes, A, B, C, such that A — B and B — C, but not A — B, then introduce the v-structure \( A \to C \leftarrow B \) if there exists an \( X \) (possibly empty) such that \( I(A,B,X) \) and \( C \notin X \).
Rules for directing the links

* Rule 2 [Avoid new v-structures]: When Rule 1 has been exhausted, and you have $A \rightarrow C \leftarrow B$ (and no link between $A$ and $B$), then direct $C \rightarrow B$.

* Rule 3 [Avoid cycles]: If $A \rightarrow B$ introduces a directed cycle in the graph then do $A \leftarrow B$.

* Rule 4 [Choose randomly]: If none of the rules 1-3 can be applied anywhere in the graph, choose an undirected link and give it an arbitrary direction.
Rules for directing the links

- Rule 1 [Introduce v-structures]
- Rule 2 [Avoid new v-structures]
- Rule 3 [Avoid cycles]
- Rule 4 [Choose randomly]
Rules for directing the links

- Rule 1 [Introduce v-structures]
- Rule 2 [Avoid new v-structures]
- Rule 3 [Avoid cycles]
- Rule 4 [Choose randomly]
Applying the rules

- Rule 1 [Introduce v-structures]
- Rule 2 [Avoid new v-structures]
- Rule 3 [Avoid cycles]
- Rule 4 [Choose randomly]
Applying the rules

- Rule 1 [Introduce v-structures]
- Rule 2 [Avoid new v-structures]
- Rule 3 [Avoid cycles]
- Rule 4 [Choose randomly]
Applying the rules

- Rule 1 [Introduce v-structures]
- Rule 2 [Avoid new v-structures]
- Rule 3 [Avoid cycles]
- Rule 4 [Choose randomly]
Applying the rules

- Rule 1 [Introduce v-structures]
- Rule 2 [Avoid new v-structures]
- Rule 3 [Avoid cycles]
- Rule 4 [Choose randomly]
Applying the rules

• Rule 1 [Introduce v-structures]
• Rule 2 [Avoid new v-structures]
• Rule 3 [Avoid cycles]
• Rule 4 [Choose randomly]
Applying the rules

• Rule 1 [Introduce v-structures]
• Rule 2 [Avoid new v-structures]
• Rule 3 [Avoid cycles]
• Rule 4 [Choose randomly]
Applying the rules

- Rule 1 [Introduce v-structures]
- Rule 2 [Avoid new v-structures]
- Rule 3 [Avoid cycles]
- Rule 4 [Choose randomly]
Applying the rules

- Rule 1 [Introduce v-structures]
- Rule 2 [Avoid new v-structures]
- Rule 3 [Avoid cycles]
- Rule 4 [Choose randomly]
There can be several equally good results
There can be several equally good results

Rule 1

Rule 2
There can be several equally good results.
There can be several equally good results
There can be several equally good results.
There can be several equally good results
There can be several equally good results
From Independence Tests to Skeleton
Asking the Oracle has a price

Theorem 7.1. The nodes $A$ and $B$ are not linked in $N$ if and only if $I(A,B,pa(A))$ or $I(A,B,pa(B))$.

$\Rightarrow$ It is sufficient to ask questions of the form $I(A,B,X)$, where $X$ is a subset of $A$’s or $B$’s neighbors.
The PC algorithm

1. Start with a complete graph;

\[ i := 0; \]

\[
\textbf{while} \ a \text{ node has at least } i + 1 \text{ neighbors} \\
\quad \textbf{for all} \ A \text{ with at least } i + 1 \text{ neighbors} \\
\quad \quad \textbf{for all} \ B \text{ of } A \\
\quad \quad \quad \textbf{for all} \ X \text{ such that } |X| = i \text{ and } X \subseteq (\text{nb}(A) \setminus \{B\}) \\
\quad \quad \quad \quad \textbf{if } I(A,B,X) \text{ then remove link } A \rightarrow B \text{ and store } \text{“I}(A,B,X)\text{“}. \\
\quad \quad i := i + 1
\]
Example
Example

I(A,B)?, I(A,C)?, I(A,D)?, I(A,E)?, I(B,C)?, I(B,D)?, I(B,E)?, I(C,D)?, I(C,E)®, I(D,E)?
Example

\[ I(A,B)?, I(A,C)?, I(A,D)?, I(A,E)?, I(B,C)?, I(B,D)?, I(B,E)?, I(C,D)?, I(C,E)?, I(D,E)? \]
Example

I(A,B)?, I(A,C)?, I(A,D)?, I(A,E)?, I(B,C)?, I(B,D)?, I(B,E)?, I(C,D)?, I(C,E)?, I(D,E)?
Example

I(A,C,E)?, I(B,C,D)?, I(B,C,E)?, I(B,D,C)?, I(B,D,E)?, I(B,E,C)?, I(B,E,D)?, I(C,B,A)?, I(C,D,A)?
Example

I(A,C,E)?, I(B,C,D)?, I(B,C,E)?, I(B,D,C)?, I(B,D,E)?, I(B,E,C)?, I(B,E,D)?, I(C,B,A)?, I(C,D,A)?
Example

I(A,C,E)?, I(B,C,D)?, I(B,C,E)?, I(B,D,C)?, I(B,D,E)?, I(B,E,C)?, I(B,E,D)?, I(C,B,A)?, I(C,D,A)?
Example

\[ I(B,E,\{C,D\})? , I(A,E, \{C,D\})? \]
Example

Original

I(B,E,\{C,D\})?, I(A,E, \{C,D\})?
Example

\[ I(B,E,\{C,D\})? , I(A,E, \{C,D\})? \]
Conditional independencies:

\[ I(A,B), I(B,C), I(C,D,A), I(B,E,\{C,D\}), I(A,E, \{C,D\}) \]
Properties of PC algorithm

* **Property 1:** If the case set is faithful sample from a Bayesian Network, $N$, then the graph resulting from the PC-algorithm is the skeleton of $N$.

* **Property 2:** The conditional independencies found by the PC-algorithm are sufficient for determining the v-structures.
Constraint-Based Learning on Data Sets
We Don’t Have an Oracle

**Definition 7.2.** $D$ is a faithful sample from $N$ if the following holds: $A$ and $B$ are $d$-separated in $N$ given $X$ if and only if $I_D(A,B,X)$.

If $D$ is faithful to $N$, we can use **conditional mutual information** to test independence.

\[
CMI(X,Y \mid Z) = \sum_{x,y,z} p(z)p(x,y\mid z) \log \frac{p(x,y\mid z)}{p(x\mid z)p(y\mid z)}
\]

It holds that $ID(A,B,X) \Leftrightarrow CMI(A,B \mid X) = 0$

Finally, $\chi^2$-test on the hypothesis $CMI(A,B \mid X) = 0$
No Test is Perfect

Uncertain regions

¬I(A,B)
¬I(A,C)
¬I(B,C)
I(A,B,C)
I(A,C,B)
I(B,C,A)
Data Might Be Incomplete

Hidden variables

I(A,C), I(A,D), I(B,D)
Data Might Be Incomplete

Hidden variables

I(A,C), I(A,D), I(B,D)
Data Might Be Incomplete

Hidden variables

I(A,C), I(A,D), I(B,D)