Advanced AI Techniques

I. Bayesian Networks / 4. Constrained-based Structure Learning

(1/2)

Wolfram Burgard, Luc de Raedt, Bernhard Nebel, Lars Schmidt-Thieme

Institute of Computer Science
University of Freiburg
http://www.informatik.uni-freiburg.de/

1. Checking Probabilistic Independencies
2. Markov Equivalence and DAG patterns
3. PC Algorithm
Types of Methods for Structure Learning

There are three types of structure learning algorithms for Bayesian networks:

1. **constrained-based learning** (e.g., PC),

2. **searching with a target function** (e.g., K2),

3. **hybrid methods** (e.g., sparse candidate).

**Lemma 1 (Edge Criterion).** Let $G := (V, E)$ be a DAG and $X, Y \in V$. Then it is equivalent:

(i) $X$ and $Y$ cannot be separated by any $Z$, i.e.,

$$\neg I_G(X, Y | Z) \quad \forall Z \subseteq V \setminus \{X, Y\}$$

(ii) There is an edge between $X$ and $Y$, i.e.,

$$(X, Y) \in E \text{ or } (Y, X) \in E$$

**Definition 1.** Any $Z \subseteq V \setminus \{X, Y\}$ with $I_G(X, Y | Z)$ is called a **separator of $X$ and $Y$**.

$$Sep(X, Y) := \{Z \subseteq V \setminus \{X, Y\} | I_G(X, Y | Z)\}$$
Computing the Skeleton / Separators

1 separators-basic(set of variables $V$, independency relation $I$) :
2 Allocate $S : P^2(V) \rightarrow P(V) \cup \{\text{none}\}$
3 for ${X, Y} \subseteq V$ do
4 \hspace{1em} $S({X, Y}) := \text{none}$
5 \hspace{1em} for $T \subseteq V \setminus \{X, Y\}$ do
6 \hspace{2em} if $I(X, Y | T)$
7 \hspace{2em} $S({X, Y}) := T$
8 \hspace{2em} break
9 \hspace{1em} od
10 od
11 return $S$

Figure 1: Compute a separator for each pair of variables.

Example / 1/3 – Computing the Skeleton

Let $I$ be the following independency structure:

$I(A, D | C), \quad I(A, D | \{C, B\}), \quad I(B, D)$

Then we can compute the following separators:

$S(A, B) := \text{none}$
$S(A, C) := \text{none}$
$S(A, D) := \{C\}$
$S(B, C) := \text{none}$
$S(B, D) := \emptyset$
$S(C, D) := \text{none}$

Thus, the skeleton of the Bayesian Network representing $I$ looks like

```
D
  |
  V
C ------- B
    |
    A
```
Lemma 2 (Uncoupled Head-to-head Meeting Criterion). Let $G := (V, E)$ be a DAG, $X, Y, Z \in V$ with

$$ \xymatrix{ X \ar[r] & Y \ar[d] \ar[l] \ar[r] & Z } $$

Then it is equivalent:

(i) $X \rightarrow Z \leftarrow Y$ is an uncoupled head-to-head meeting, i.e.,

$$(X, Z), (Y, Z) \in E, (X, Y), (Y, X) \notin E$$

(ii) $Z$ is not contained in any separator of $X$ and $Y$, i.e.,

$$Z \notin S \quad \forall S \in \text{Sep}(X, Y)$$

(iii) $Z$ is not contained in at least one separator of $X$ and $Y$, i.e.,

$$Z \notin S \quad \exists S \in \text{Sep}(X, Y)$$

---

Figure 2: Compute skeleton and v-structure.

```python
vstructure(set of variables V, independency relation I) :
S := separators(V, I)
G := (V, E) with E := \{\{X, Y\} \mid S(\{X, Y\}) = \text{none}\}
for X, Y, Z \in V with X \rightarrow Z \leftarrow Y, X \not\rightarrow Y do
    if Z \notin S(X, Y)
    orient X \rightarrow Z \leftarrow Y as X \rightarrow Z \leftarrow Y
od
return G
```

Figure 3: Learn structure of a Bayesian Network (SGS/PC algorithm, [SGS00]).
Separators:

\[
\begin{align*}
S(A, B) & := \text{none} \\
S(A, C) & := \text{none} \\
S(A, D) & := \{C\} \\
S(B, C) & := \text{none} \\
S(B, D) & := \emptyset \\
S(C, D) & := \text{none}
\end{align*}
\]

Skeleton:

Checking \(A \rightarrow C \rightarrow D\):

Checking \(B \rightarrow C \rightarrow D\):

Example / 3/3 – Saturating

Skeleton and v-structure:

Saturating:
Advanced AI Techniques / 3. PC Algorithm

Number of Independency Tests

Let there be $n$ variables.
For each of the $\binom{n}{2}$ pairs of variables, there are $2^{n-2}$ candidates for possible separators.

$$\text{number of } I\text{-tests} = \binom{n}{2} 2^{n-2}$$

Example: $n = 4$:

$$\binom{n}{2} 2^{n-2} = \binom{4}{2} 2^2 = 6 \cdot 4 = 24$$

If we start with small separators and stop once a separator has been found, we still have to check

$$4 \cdot (1 + 2 + 1) + 1 \cdot (1 + 2) + 1 \cdot 1 = 20$$

Can we reduce the number of tests for a given pair of variable by reusing results for other pairs of variables?

**Lemma 3.** Let $G := (V, E)$ be a DAG and $X, Y \in V$ separated. Then

$$I(X, Y \mid \text{pa}(X)) \text{ or } I(X, Y \mid \text{pa}(Y))$$

As we do not know directions of edges at the the skeleton recovery step, we use the weaker result:

$$I(X, Y \mid \text{fan}(X)) \text{ or } I(X, Y \mid \text{fan}(Y))$$
Computing the Skeleton / Separators

```plaintext
1 separators-remove-edges(separator map S, skeleton graph G, independency relation I)
2   i := 0
3 while ∃X ∈ V : | fan_G(X) | > i do
4   for {X, Y} ∈ E with fan_G(X) | > i or fan_G(Y) | > i do
5     for T ∈ P^i(fan_G(X) \ {Y}) ∪ P^i(fan_G(Y) \ {X}) do
6       if I(X, Y | T)
7         S({X, Y}) := T
8         E := E \ {{X, Y}}
9         break
10   od
11 od
12 i := i + 1
13 od
14 return S
```

1 separators-interlaced(set of variables V, independency relation I) :
2 Allocate S : P^2(V) → P(V) ∪ {none}
3 S({X, Y}) := none ∀{X, Y} ⊆ V
4 G := (V, E) with E := P^2(V)
5 separators-remo v e-edges(S, G, I)
6 return S

Figure 4: Compute a separator for each pair of variables.

Example / Computing the Separators (1/3)

\[ I(A, D | C), \quad I(A, D | \{C, B\}), \quad I(B, D) \]

\[
\begin{align*}
  i &= 0: \\
  A, \quad B, \quad T = \emptyset: &- \\
  C, \quad T = \emptyset: &- \\
  D, \quad T = \emptyset: &- \\
  B, \quad C, \quad T = \emptyset: &- \\
  D, \quad T = \emptyset: &D(\{B, D\}) = \emptyset \\
  C, \quad D, \quad T = \emptyset: &- \\
\end{align*}
\]

initial graph:

```
D -- A -- B  
|     |     | 
D     |     | D
  
|     |     | 
  
|     |     | 

```
Example / Computing the Separators (2/3)

\[ I(A, D \mid C), \quad I(A, D \mid \{C, B\}), \quad I(B, D) \]

\( i = 1 \):
\begin{align*}
A, \quad B, \quad T = \{C\}, \{D\}: &- \\
C, \quad T = \{B\}, \{D\}: &- \\
D, \quad T = \{B\}, \{C\}: & S(\{A, D\}) = \{C\} \\
B, \quad C, \quad T = \{A\}, \{D\}: &- \\
C, \quad D, \quad T = \{A\}, \{B\}: &- \\
\end{align*}

after update for \( B, D \):

\begin{tikzpicture}
  \node (A) at (0,0) {A};
  \node (B) at (-1,-1) {B};
  \node (C) at (1,-1) {C};
  \node (D) at (0,-2) {D};
  \draw (A) -- (B);
  \draw (A) -- (C);
  \draw (B) -- (D);
  \draw (C) -- (D);
\end{tikzpicture}

after update for \( A, D \):

\begin{tikzpicture}
  \node (A) at (0,0) {A};
  \node (B) at (-1,-1) {B};
  \node (C) at (1,-1) {C};
  \node (D) at (0,-2) {D};
  \draw (A) -- (B);
  \draw (A) -- (C);
  \draw (B) -- (D);
\end{tikzpicture}

Example / Computing the Separators (3/3)

\[ I(A, D \mid C), \quad I(A, D \mid \{C, B\}), \quad I(B, D) \]

\( i = 2 \):
\begin{align*}
A, \quad C, \quad T = \{B, D\}: &- \\
B, \quad C, \quad T = \{A, D\}: &- \\
C, \quad D, \quad T = \{A, B\}: &- \\
\end{align*}

total: 19 \( I \)-tests.

after update for \( A, D \):

\begin{tikzpicture}
  \node (A) at (0,0) {A};
  \node (B) at (-1,-1) {B};
  \node (C) at (1,-1) {C};
  \node (D) at (0,-2) {D};
  \draw (A) -- (B);
  \draw (A) -- (C);
  \draw (B) -- (D);
\end{tikzpicture}
Advanced AI Techniques / 3. PC Algorithm

Algorithms – SGS vs. PC

SGS/PC with separators-basic is called SGS algorithm ([SGS00], 1990).

SGS/PC with separators-interlaced is called PC algorithm ([SGS00], 1991).

Implementations are available:

- in Tetrəd
  http://www.phil.cmu.edu/projects/tetrad/
  (class files & javadocs, no sources)

- in Hugin (commercial).

Advanced AI Techniques / 3. PC Algorithm

Another Example

Let $I$ be the following independency structure:

\[ I(A, D \mid B), \quad I(B, D) \]

PC computes the following DAG pattern:

\[ \begin{array}{c}
A \\
B \\
D
\end{array} \]

But this is not even a representation of $I$, as it implies

\[ I_G(A, D \mid \emptyset) \]
PC computes the DAG pattern of an independency relation

if there exists one at all!

Remember: not any independency relation has a faithful DAG representation.

But how do we know if an independency relation has a faithful DAG representation?

There is no easy way to decide if a given independency relation has a faithful DAG representation.

So just check ex-post if the DAG pattern we have found is a faithful representation.

Check:

1. compute all $I_G(X, Y \mid Z)$ and check if $I(X, Y \mid Z)$ (representation),

2. check for each $I(X, Y \mid Z)$ if $I_G(X, Y \mid Z)$ (faithfulness).

As there is no easy way to enumerate $I_G(X, Y \mid Z)$ for a DAG pattern $G$ directly, we draw a representative $H$ and then enumerate $I_H(X, Y \mid Z)$ (remember that $I_H = I_G$).
1 draw-representative(DAG pattern \( G \)) :
2 saturate(\( G \))
3 while \( G \) has unoriented edges do
4 draw an edge from \( G \) and orient it arbitrarily
5 saturate(\( G \))
6 od
7 return \( G \)

Figure 5: Draw a representative of a DAG pattern.

Here, rule 4 is necessary in `saturate`. Start with DAG pattern that is already saturated and choose as next edge \( X \rightarrow W \) and orient it as \( X \rightarrow W \). Then rule 4 has to be applied to saturate the resulting DAG pattern.
Draw a Representative of a DAG pattern / Example

step 1a: saturate.

step 1b: orient any unoriented edge:

step 2a: saturate.

step 2b: orient any unoriented edge:

step 3a: saturate.

step 3b: orient any unoriented edge:

done.

Representation and Faithfulness Tests

If $I = I_p$ the probabilistic independency relation of a JPD $p$, then for (1) it suffices to check the Markov property, i.e.,

for all $X$ check if $I_p(X, \text{nondesc}(X) \setminus \text{pa}(X) | \text{pa}(X))$

It even suffices to check for any topological ordering $\sigma = (X_1, \ldots, X_n)$ if

$I_p(\sigma(i), \sigma(\{1, \ldots, i - 1\}) \setminus \text{pa}(\sigma(i)) | \text{pa}(\sigma(i)))$
separators-interlaced computes separators top-down by

- starting with a complete graph and then
- successively thining the graph.

Therefore, we have to start checking with lots of candidates for possible separators.

```
1 separators-add-edges(separator map S, skeleton graph G, independency relation I) :
2 S({X, Y}) := argmin_{T ⊆ fan_G(X), T ⊆ fan_G(Y)} g(X, Y, T) \forall \{X, Y\} \in \mathcal{P}^2(V)
3 \{X_0, Y_0\} := argmax_{(X, Y) \in \mathcal{P}^2(V)} g(X, Y, S({X, Y}))
4 while ¬I(X_0, Y_0 | S({X_0, Y_0})) do
5 \{X, Y\} := argmax_{\{X, Y\} \in \mathcal{P}^2(V) \setminus E, X \in \{X_0, Y_0\}} g(X, Y, S({X, Y}))
6 S({X_0, Y_0}) := none
7 S({X, Y}) := argmin_{T \subseteq fan_G(X), T \subseteq fan_G(Y)} g(X, Y, T) \forall \{X, Y\} \in \mathcal{P}^2(V) \setminus E, X \in \{X_0, Y_0\}
8 \{X_0, Y_0\} := argmax_{(X, Y) \in \mathcal{P}^2(V) \setminus E} g(X, Y, S({X, Y}))
9 od
10 return S
```

```
1 separators-bottomup(set of variables V, independency relation I) :
2 Allocate S : \mathcal{P}^2(V) → \mathcal{P}(V) ∪ \{none\}
3 G := (V, E) with E := ∅
4 separators-add-edges(S, G, I)
5 separators-remove-edges(S, G, I)
6 return S
```

Figure 6: Compute a separator for each pair of variables [TASB03].
Embedded Faithfulness

Let $I$ be the following independency structure:

$I(A, D), \quad I(A, E), \quad I(A, E \mid B), \quad I(A, E \mid D), \quad I(B, E)$

Assume $C$ to be hidden.

d-separations between variables $A, B, D,$ and $E$:

$I(A, D), \quad I(A, E), \quad I(A, E \mid B), \quad I(A, E \mid D), \quad I(B, E)$

**Definition 2.** Let $I$ be an independency relation on the variables $V$ and $G$ be a DAG with vertices $W \supseteq V$.

$I$ is **embedded** in $G$, if all independency statements entailed by $G$ between variables from $V$ hold in $I$:

$I_G(\mathcal{X}, \mathcal{Y} \mid Z) \Rightarrow I(\mathcal{X}, \mathcal{Y} \mid Z) \quad \forall \mathcal{X}, \mathcal{Y}, Z \subseteq V$

$I$ is **embedded faithfully** in $G$, if the independency statements entailed by $G$ are exactly $I$:

$I_G(\mathcal{X}, \mathcal{Y} \mid Z) \Leftrightarrow I(\mathcal{X}, \mathcal{Y} \mid Z) \quad \forall \mathcal{X}, \mathcal{Y}, Z \subseteq V$

Many independency relations w./o. faithful DAG representation can be embedded faithfully.

But not every independency relation can be embedded faithfully.
Embedded Faithfulness / Example

Let $I$ be the independency relation [Nea03, ex. 2.13, p. 103]

$$I(X, Y), \quad I(X, Y | Z)$$

Assume, $I$ can be embedded faithfully in a DAG $G$.

- As $\neg I(X, Z)$, there must be chain $X \sim Z$ w./o. head-to-head meetings.
- As $\neg I(Z, Y)$, there must be chain $Y \sim Z$ w./o. head-to-head meetings.

Now, concatenate both chains $X \sim Z \sim Y$:

- either $X \sim \rightarrow Z \leftarrow \sim Y$, and then the chain is not blocked by $Z$, i.e., not $I(X, Y | Z)$,
- or not $X \sim \rightarrow Z \leftarrow \sim Y$, and then the chain is not blocked by $\emptyset$, i.e., not $I(X, Y)$.

There are variants of the PC algorithm for finding faithful embeddings of a given independency relation:

- Causal Inference (CI) and
- Fast Causal Inference Algorithms (FCI; [SGS00])

See also [Nea03, ch. 10.2].
References

