Probabilistic Graphical Models
Part 2: Bayesian Networks – Syntax and Semantics

M. Jaeger

Aalborg University
Syntax

Let $X_1, \ldots, X_n$ be random variables. A *Bayesian network* for $X_1, \ldots, X_n$ consists of

- a *directed acyclic graph* (DAG) $\mathcal{G}$ whose nodes are the variables $X_1, \ldots, X_n$,
- a *conditional probability table* (cpt) for each node $X_i$ specifying a conditional distribution $P(X_i | \text{Pa}(X_i))$, where $\text{Pa}(X_i)$ are the parents of $X_i$ in the graph.

Notation

We write $\mathcal{N} = (\mathcal{G}, \text{cpt}(X_i)_{i=1,\ldots,n})$ for a BN, or more succinctly, $\mathcal{N} = (\mathcal{G}, \theta)$ where $\theta$ is a parameter vector containing all the entries of all $\text{cpt}(X_i)$.

Semantics

A Bayesian network $\mathcal{N}$ with nodes $X_1, \ldots, X_n$ defines the joint distribution

$$P_{\mathcal{N}}(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \text{Pa}(X_i))$$
Given

- an ordered set of variables $X_1, \ldots, X_n$ with joint distribution $P$
- conditional independencies

$$P(X_i \mid X_1, \ldots, X_{i-1}) = P(X_i \mid \text{Pa}(X_i)) \quad (\text{Pa}(X_i) \subseteq \{X_1, \ldots, X_{i-1}\})$$

one obtains a Bayesian network representing $P$: 

- the parents of $X_i$ in the dag are the variables in $\text{Pa}(X_i)$
- the conditional probability table of $X_i$ specifies the conditional distribution $P(X_i \mid \text{Pa}(X_i))$
BN could have been constructed using

**Variable ordering:** $B, C, A, E, D$

**Conditional Independencies:**

\[
P(A \mid B, C) = P(A \mid B)
\]
\[
P(E \mid B, C, A) = P(E \mid A)
\]
\[
P(D \mid B, C, A, E, D) = P(D \mid A)
\]

Which of these conditional independencies are actually true in the distribution defined by BN?

All!
BN could have been constructed using

Variable ordering: $B, C, A, E, D$
Conditional Independencies: 
\[
\begin{align*}
P(A | B, C) &= P(A | B) \\
P(E | B, C, A) &= P(E | A) \\
P(D | B, C, A, E, D) &= P(D | A)
\end{align*}
\]

or

Variable ordering: $B, A, C, D, E$
Conditional Independencies: 
\[
\begin{align*}
P(C | B, A) &= P(C | B) \\
P(D | B, A, C) &= P(D | A) \\
P(E | B, A, C, D) &= P(E | A)
\end{align*}
\]
BN could have been constructed using

**Variable ordering: B, C, A, E, D**

**Conditional Independencies:**

- \( P(A \mid B, C) = P(A \mid B) \)
- \( P(E \mid B, C, A) = P(E \mid A) \)
- \( P(D \mid B, C, A, E, D) = P(D \mid A) \)

or

**Variable ordering: B, A, C, D, E**

**Conditional Independencies:**

- \( P(C \mid B, A) = P(C \mid B) \)
- \( P(D \mid B, A, C) = P(D \mid A) \)
- \( P(E \mid B, A, C, D) = P(E \mid A) \)

or

...
BN could have been constructed using

**Variable ordering:** $B, C, A, E, D$

**Conditional Independencies:**

\[
\begin{align*}
P(A | B, C) & = P(A | B) \\
P(E | B, C, A) & = P(E | A) \\
P(D | B, C, A, E, D) & = P(D | A)
\end{align*}
\]

or

**Variable ordering:** $B, A, C, D, E$

**Conditional Independencies:**

\[
\begin{align*}
P(C | B, A) & = P(C | B) \\
P(D | B, A, C) & = P(D | A) \\
P(E | B, A, C, D) & = P(E | A)
\end{align*}
\]

or

\[
\begin{align*}
P(C | B, A) & = P(C | B) \\
P(D | B, A, C) & = P(D | A) \\
P(E | B, A, C, D) & = P(E | A)
\end{align*}
\]

Which of these conditional independencies are actually true in the distribution defined by BN?

**All!**
In the distribution $P$ defined by the BN the following independence relation holds:

$$P(X_i \mid \text{Pa}(X_i), \text{rest}(X_i)) = P(X_i \mid \text{Pa}(X_i))$$

"$X_i$ is independent of its non-descendants given its parents"

This is the independence relation corresponding to a variable ordering in which $X_i$ is in the last possible position (just before all its descendants).
Independence and Causal Reasoning
Val(Rainfall) = \{no, light, medium, heavy\}, Val(Water level) = \{low, medium, high\}, Val(Flooding) = \{yes, no\},

\[
P(Flooding \mid Water level, Rainfall) = P(Flooding \mid Water level)
\]

but not

\[
P(Flooding \mid Rainfall) = P(Flooding)
\]

**Causal model:** cause-effect chain.

If state of immediate cause is known, then state of more indirect cause does not affect probabilities for effect.
Causal Reasoning

3 Variables Diverging Connection

\[ P(\text{Hair length} \mid \text{Stature, Sex}) = P(\text{Hair length} \mid \text{Sex}) \]

but not

\[ P(\text{Hair length} \mid \text{Stature}) = P(\text{Hair length}) \]

**Causal model**: a common cause with two effects.

- if state of cause is not known:
  knowing that one effect has happened makes the second effect also more probable
- if state of cause is known:
  knowing about one effect does not affect the probabilities for the other effect
Val(Fuel)=\{yes,no\}, Val(Spark Plugs)=\{clean,dirty\}, Val(Start)=\{yes,no\},

\[
P(Fuel \mid Spark Plugs) = P(Fuel)
\]

\textbf{but not}

\[
P(Fuel \mid Spark Plugs, Start) = P(Fuel \mid Start)
\]

\textbf{Causal model:} two distinct causes for one effect

- if state of effect is not known:
  state of one cause does not affect probabilities for other cause
- if state of effect is known:
  state of one cause changes probabilities of other cause (\textit{explaining away})
Constructing BNs using causal dependencies:

- specify dag so that edges represent direct cause-effect relations
- specify $P(X_i \mid \text{Pa}(X_i))$ by assessing the probability for $X_i$ for all combinations of its possible (direct) causes $\text{Pa}(X_i)$. 
Constructing BNs using causal dependencies:

- specify dag so that edges represent direct cause-effect relations
- specify $P(X_i \mid Pa(X_i))$ by assessing the probability for $X_i$ for all combinations of its possible (direct) causes $Pa(X_i)$.

This is very often how BNs are constructed. But:

Causal dependencies need not be acyclic: There can be dependencies without causality:

- Inflation
- Salaries
- Rain
- Temperature
The Naive Bayes Classifier
Naive Bayes

Classification Problems

**Example** (Poole & Mackworth, Artificial Intelligence, 2010)

User preference data (will a user read a new posting on an online forum):

<table>
<thead>
<tr>
<th>Example</th>
<th>Author</th>
<th>Thread</th>
<th>Length</th>
<th>WhereRead</th>
<th>UserAction</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>known</td>
<td>new</td>
<td>long</td>
<td>home</td>
<td>skips</td>
</tr>
<tr>
<td>$e_2$</td>
<td>unknown</td>
<td>new</td>
<td>short</td>
<td>work</td>
<td>reads</td>
</tr>
<tr>
<td>$e_3$</td>
<td>unknown</td>
<td>follow up</td>
<td>long</td>
<td>work</td>
<td>skips</td>
</tr>
<tr>
<td>$e_4$</td>
<td>known</td>
<td>follow up</td>
<td>long</td>
<td>home</td>
<td>skips</td>
</tr>
<tr>
<td>$e_5$</td>
<td>known</td>
<td>new</td>
<td>short</td>
<td>home</td>
<td>reads</td>
</tr>
<tr>
<td>$e_6$</td>
<td>known</td>
<td>new</td>
<td>short</td>
<td>work</td>
<td>skips</td>
</tr>
<tr>
<td>$e_7$</td>
<td>unknown</td>
<td>follow up</td>
<td>long</td>
<td>work</td>
<td>skips</td>
</tr>
<tr>
<td>$e_8$</td>
<td>unknown</td>
<td>new</td>
<td>short</td>
<td>work</td>
<td>reads</td>
</tr>
<tr>
<td>$e_9$</td>
<td>known</td>
<td>follow up</td>
<td>long</td>
<td>home</td>
<td>skips</td>
</tr>
<tr>
<td>$e_{10}$</td>
<td>known</td>
<td>new</td>
<td>long</td>
<td>work</td>
<td>skips</td>
</tr>
<tr>
<td>$e_{11}$</td>
<td>unknown</td>
<td>follow up</td>
<td>short</td>
<td>home</td>
<td>skips</td>
</tr>
<tr>
<td>$e_{12}$</td>
<td>known</td>
<td>new</td>
<td>long</td>
<td>work</td>
<td>skips</td>
</tr>
<tr>
<td>$e_{13}$</td>
<td>known</td>
<td>follow up</td>
<td>short</td>
<td>home</td>
<td>reads</td>
</tr>
<tr>
<td>$e_{14}$</td>
<td>known</td>
<td>new</td>
<td>short</td>
<td>work</td>
<td>reads</td>
</tr>
<tr>
<td>$e_{15}$</td>
<td>known</td>
<td>new</td>
<td>short</td>
<td>home</td>
<td>reads</td>
</tr>
<tr>
<td>$e_{16}$</td>
<td>known</td>
<td>follow up</td>
<td>short</td>
<td>work</td>
<td>?</td>
</tr>
<tr>
<td>$e_{17}$</td>
<td>known</td>
<td>new</td>
<td>short</td>
<td>home</td>
<td>?</td>
</tr>
<tr>
<td>$e_{18}$</td>
<td>unknown</td>
<td>new</td>
<td>short</td>
<td>work</td>
<td>?</td>
</tr>
<tr>
<td>$e_{19}$</td>
<td>unknown</td>
<td>new</td>
<td>long</td>
<td>work</td>
<td>?</td>
</tr>
<tr>
<td>$e_{20}$</td>
<td>unknown</td>
<td>follow up</td>
<td>long</td>
<td>home</td>
<td>?</td>
</tr>
</tbody>
</table>

Task: from the training data, learn a predictor for (future) test cases.
**Classification Model**

Joint probability distribution $P$ on $\text{Val}(A_1) \times \ldots \times \text{Val}(A_n) \times \text{Val}(C)$

**Classification Rule**

$$C(a_1, \ldots, a_n) := \arg \max_{c \in \text{Val}(C)} P(C = c \mid A_1 = a_1, \ldots, A_n = a_n)$$

or for a specified loss function $\text{Loss}(c, c')$:

$$C(a_1, \ldots, a_n) := \arg \min_{c \in \text{Val}(C)} \sum_{c' \in \text{Val}(C)} P(C = c' \mid A_1 = a_1, \ldots, A_n = a_n)\text{Loss}(c, c')$$

This also works with incompletely observed instances: denote by $\tilde{A} \subseteq A$ the attributes for which values $A_i = a_i$ are observed. Then classify according to

$$P(C = c \mid A_i = a_i (A_i \in \tilde{A}))$$
Probabilistic classifiers differ in what classes of joint distributions $P$ can be represented – and how. One possibility: $P$ is represented using a (restricted type of) Bayesian network.

- Bayesian network can contain much information that is irrelevant for classification task. Only relevant: $P(C \mid A_1, \ldots, A_n)$.
- For classification typically restricted types of BNs are used.
The Naive Bayes Model is given by the BN structure:

![BN Structure](image)

- Attributes are independent, given the class label.

\[ P(C, A_1, \ldots, A_n) = P(C) \cdot \prod_{i=1}^{n} P(A_i \mid C) \]

### Learning Naive Bayes

- Model parameters simply learned by counting:

\[ P(A_i = a \mid C = c) = \frac{\# \text{training ex. with } A_i = a, C = c}{\# \text{training ex. with } C = c} \]
Predicting with Naive Bayes

\[ P(C = c \mid \mathbf{A} = \mathbf{a}) = \frac{P(C = c, \mathbf{A} = \mathbf{a})}{P(\mathbf{A} = \mathbf{a})} \]

\[ \arg \max_c P(C = c \mid \mathbf{A} = \mathbf{a}) = \arg \max_c P(C = c, \mathbf{A} = \mathbf{a}) = \arg \max_c P(C) \cdot \prod_{i=1}^{n} P(A_i = a_i \mid C = c) \]

For incomplete data:

\[ P(C = c, \tilde{\mathbf{A}} = \tilde{\mathbf{a}}) = P(C) \cdot \prod_{i: A_i \in \tilde{\mathbf{a}}} P(A_i = a_i \mid C = c) \]

In both cases: classification computable in time \( O(n) \).
Class: Symbol $\in \{A, \ldots, Z, 0, \ldots, 9\}$

Predictors: Cell-1, \ldots, Cell-9 $\in \{b, w\}$.

For example:

$$P(\text{Cell-2} = b \mid \text{Cell-5} = b, \text{Symbol} = 1) > P(\text{Cell-2} = b \mid \text{Symbol} = 1)$$

Attributes not independent given Symbol=1!
Class: $Spam \in \{y, n\}$

Predictors:
$Subject\text{-}all\text{-}caps, Known\text{-}spam\text{-}server, \ldots, Contains\text{'}Money', Contains\text{'}Watch', \ldots \in \{y, n\}$.

For example:

$$P(Contains\text{'Nigeria' } = y \mid Contains\text{'}Confidential' = y, Spam = y) \gg P(Contains\text{'Nigeria' } = y \mid Spam = y)$$

Attributes not independent given $Spam=\text{yes}$!

$\therefore$ Naive Bayes assumption often not realistic. Nevertheless, Naive Bayes often successful.
D-Separation
The non-descendant criterion:

\[ P_{\mathcal{N}}(X_i \mid \text{Pa}(X_i), \text{rest}(X_i)) = P_{\mathcal{N}}(X_i \mid \text{Pa}(X_i)) \]

Questions:

- What other conditional independence relations can be inferred from the graph structure of a Bayesian network?
- Is there a single, most general criterion?
- Given a distribution \( P \) and a BN graph \( \mathcal{G} \), can \( P \) be represented via a suitable parameterization of \( \mathcal{G} \)?
$A, B, C \subseteq V$ disjoint subsets of variables in a Bayesian network $G$. $C$ d-separates $A$ from $B$ if the following holds:

every undirected path that connects a node $A \in A$ with a node $B \in B$ satisfies at least one of the following two conditions:

1. the path contains a node $C \in C$, and the edges that connect $C$ are serial ($\ldots \rightarrow C \rightarrow \ldots$) or divergent ($\ldots \leftarrow C \rightarrow \ldots$).

2. the path contains a node $U$, the edges that connect $U$ are convergent ($\ldots \rightarrow U \leftarrow \ldots$), and neither $U$ nor any of the descendants of $U$ belong to $C$. 
Pa(A) d-separates A from rest(A):
Correctness

For all pairwise disjoint sets $A, B, C$ of nodes in a Bayesian network $\mathcal{N} = (\mathcal{G}, \theta)$:

$C$ d-separates $A$ from $B$ in $\mathcal{G} \Rightarrow P_{\mathcal{N}}(A \mid B, C) = P_{\mathcal{N}}(A \mid C)$.

(d-separation is a correct criterion to identify independencies)

Completeness

For all BN structures $\mathcal{G}$ there exists a parameterization $\theta$, so that for $\mathcal{N} = (\mathcal{G}, \theta)$:

$C$ d-separates $A$ from $B$ in $\mathcal{G} \iff P_{\mathcal{N}}(A \mid B, C) = P_{\mathcal{N}}(A \mid C)$
A graph $\mathcal{G}$ over variables $X$ is an $I$-Map for a distribution $P(X)$, if for all $A, B, C \subseteq X$:

$C$ d-separates $A$ from $B$ in $\mathcal{G} \Rightarrow P(A | B, C) = P(A | C)$.

$\mathcal{G}$ is a perfect map for $P$, if

$C$ d-separates $A$ from $B$ in $\mathcal{G} \iff P(A | B, C) = P(A | C)$.

- For $\mathcal{N} = (\mathcal{G}, \theta)$: $\mathcal{G}$ is an $I$-map for $P_{\mathcal{N}}$.
- If $\mathcal{G}$ is an $I$-map for $P$, then there exist parameters $\theta$, such that $P_{(\mathcal{G}, \theta)} = P$.
- For all $\mathcal{G}$ exists a parameterization $\theta$, such that $\mathcal{G}$ is a perfect map for $P_{(\mathcal{G}, \theta)}$.

But:

- $\mathcal{G}$ is not always a perfect map for $P_{(\mathcal{G}, \theta)}$ (one can have parameterizations $\theta$ that lead to independencies in $P_{(\mathcal{G}, \theta)}$ which cannot be read off the structure $\mathcal{G}$).
- There exist distributions $P$, for which no $\mathcal{G}$ is a perfect map.
Why is D-separation important?

- Gaining insight: given a (correct) Bayesian network model, can derive insight into the dependencies among the variables
- Debugging a model: given a Bayesian network model, check whether entailed independence relations are plausible
- Correctness of algorithms: certain computational procedures depend on validity of special independence relations
Independence properties extracted from the designed model:

\(pos_t\) is independent of \(cont_{t-2}, sens_{t-1}\) given \(pos_{t-1}\)

\(pos_t\) is \textit{not} independent of \(sens_{t-1}\) given \(pos_{t-1}\) \textit{and} \(sens_t\)
Independence relations discovered from the learned network:

AA7 is independent of everything else

NQO2 is independent of *everything except* MNW7 and KID1 given RDE6
Example: two binary RV’s $A, B$. Space of all joint distributions of $A, B$ is

$$\Delta^4 = \{(p_1, p_2, p_3, p_4) \in [0, 1]^4 \mid \sum_i p_i = 1\}$$

$\Delta^4$ is a 3-dimensional tetrahedron in 4-dimensional space:
Example: two binary RV’s $A, B$. Space of all joint distributions of $A, B$ is

$$\Delta^4 = \{(p_1, p_2, p_3, p_4) \in [0, 1]^4 \mid \sum_i p_i = 1\}$$

$\Delta^4$ is a 3-dimensional tetrahedron in 4-dimensional space:

Distributions in which $A, B$ are independent form a 2-dimensional manifold in $\Delta^4$
D-Separation

The distributions for which $G$ is an $I$-map form a low-dimensional manifold in the probability polytope $\Delta^n$.

These are the distributions that can be represented using $G$ with a suitable parameterization $\theta$.

If $G'$ is obtained from $G$ by adding edges, then the distributions representable with $G'$ are a superset of the distributions representable with $G$.

For any maximal DAG $G^{max}$: $\{P(G^{max}, \theta) \mid \theta\} = \Delta^n$