Advanced AI Techniques

I. Bayesian Networks / 3. Parameter Learning with Missing Values

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

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

1. Incomplete Data
2. Incomplete Data for Parameter Learning (EM algorithm)
3. An Example
Let $V$ be a set of variables. A **complete case** is a function
\[ c : V \to \bigcup_{v \in V} \text{dom}(V) \]
with $c(v) \in \text{dom}(V)$ for all $v \in V$.

A **incomplete case** (or a **case with missing data**) is a complete case $c$ for a subset $W \subseteq V$ of variables. We denote $\text{var}(c) := W$ and say, the values of the variables $V \setminus W$ are **missing** or **not observed**.

A data set $D \in \text{dom}(V)^*$ that contains complete cases only, is called **complete data**; if it contains an incomplete case, it is called **incomplete data**.

---

**Missing value indicators**

For each variable $v$, we can interpret its missing of values as new random variable $M_v$,
\[
M_v := \begin{cases} 
1, & \text{if } v_{\text{obs}} = ., \\
0, & \text{otherwise}
\end{cases}
\]
called **missing value indicator of** $v$.

---

**Figure 1:** Complete data for $V := \{F, L, B, D, H\}$.

**Figure 2:** Incomplete data for $V := \{F, L, B, D, H\}$, Missing values are marked by a dot.

**Figure 3:** Incomplete data for $V := \{F, L, B, D, H\}$ and missing value indicators.
A variable \( v \in V \) is called \textbf{missing completely at random} (MCAR), if the probability of a missing value is (unconditionally) independent of the (true, unobserved) value of \( v \), i.e., if
\[
I(M_v, v_{\text{true}})
\]
(MCAR is also called \textbf{missing unconditionally at random}).

**Example:** think of an apparatus measuring the velocity \( v \) of wind that has a loose contact \( c \). When the contact is closed, the measurement is recorded, otherwise it is skipped. If the contact \( c \) being closed does not depend on the velocity \( v \) of wind, \( v \) is MCAR.

If a variable is MCAR, for each value the probability of missing is the same, and, e.g., the sample mean of \( v_{\text{obs}} \) is an unbiased estimator for the expectation of \( v_{\text{true}} \); here
\[
\hat{\mu}(v_{\text{obs}}) = \frac{1}{10}(2 \cdot 1 + 4 \cdot 3 + 2 \cdot 3 + 2 \cdot 4) = \frac{1}{10}(3 \cdot 1 + 6 \cdot 3 + 3 \cdot 3 + 3 \cdot 4) = \hat{\mu}(v_{\text{true}})
\]

\[
\begin{array}{c|c|c}
\text{case} & v_{\text{true}} & v_{\text{obs}} \\
\hline
1 & 1 & . \\
2 & 2 & 2 \\
3 & 2 & . \\
4 & 4 & 4 \\
5 & 3 & 3 \\
6 & 2 & 2 \\
7 & 1 & 1 \\
8 & 4 & . \\
9 & 3 & 3 \\
10 & 2 & . \\
11 & 1 & 1 \\
12 & 3 & . \\
13 & 4 & 4 \\
14 & 2 & 2 \\
15 & 2 & 2 \\
\end{array}
\]

**Figure 4:** Data with a variable \( v \) MCAR. Missing values are stroken through.

A variable \( v \in V \) is called \textbf{missing at random} (MAR), if the probability of a missing value is conditionally independent of the (true, unobserved) value of \( v \), i.e., if
\[
I(M_v, v_{\text{true}} | W)
\]
for some set of variables \( W \subseteq V \setminus \{v\} \) (MAR is also called \textbf{missing conditionally at random}).

**Example:** think of an apparatus measuring the velocity \( v \) of wind. If we measure wind velocities at three different heights \( h = 0, 1, 2 \) and say the apparatus has problems with height not recording
1/3 of cases at height 0,
1/2 of cases at height 1,
2/3 of cases at height 2,

\[
\begin{array}{c|c|c}
\text{case} & h & \text{obs} \\
\hline
1 & 1 & . \\
2 & 2 & 2 \\
3 & 3 & . \\
4 & 4 & 4 \\
5 & 1 & 1 \\
6 & 3 & 3 \\
7 & 1 & 1 \\
8 & 2 & . \\
9 & 2 & 2 \\
10 & 2 & . \\
11 & 3 & 3 \\
12 & 4 & . \\
13 & 3 & 3 \\
14 & 4 & 4 \\
15 & 5 & 5 \\
16 & 3 & 3 \\
17 & 4 & . \\
18 & 3 & 3 \\
19 & 5 & . \\
20 & 3 & 3 \\
21 & 4 & . \\
22 & 5 & 5 \\
\end{array}
\]

**Figure 5:** Data with a variable \( v \) MAR (conditionally on \( h \)).

then \( v \) is missing at random (conditionally on \( h \)).
Types of missingness / MAR

If \( v \) depends on variables in \( W \), then, e.g., the sample mean is not an unbiased estimator, but the weighted mean w.r.t. \( W \) has to be used; here:

\[
\sum_{h=0}^{2} \mu(v|H = h)p(H = h) = 2 \cdot \frac{9}{22} + 3.5 \cdot \frac{4}{22} + 4 \cdot \frac{9}{22}
\]

\[
\neq \frac{1}{11} \sum_{\substack{i=1\ldots22 \ \text{if} \ v_i \neq \cdot}} v_i
\]

\[
= 2 \cdot \frac{6}{11} + 3.5 \cdot \frac{2}{11} + 4 \cdot \frac{3}{11}
\]

A variable \( v \in V \) is called missing systematically (or not at random), if the probability of a missing value does depend on its (unobserved, true) value.

Example: if the apparatus has problems measuring high velocities and say, e.g., misses

- 1/3 of all measurements of \( v = 1 \),
- 1/2 of all measurements of \( v = 2 \),
- 2/3 of all measurements of \( v = 3 \),

i.e., the probability of a missing value does depend on the velocity, \( v \) is missing systematically.
A variable \( v \in V \) is called **hidden**, if the probability of a missing value is 1, i.e., it is missing in all cases.

**Example:** say we want to measure intelligence \( I \) of probands but cannot do this directly. We measure their level of education \( E \) and their income \( C \) instead. Then \( I \) is hidden.

<table>
<thead>
<tr>
<th>case</th>
<th>( I_{\text{true}} )</th>
<th>( I_{\text{obs}} )</th>
<th>( E )</th>
<th>( C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 7: Data with a hidden variable \( I \).

![Diagram](intelligence/education/income)

- **intelligence**
- education
- income

Figure 8: Suggested dependency of variables \( I, E, \) and \( C \).

![Diagram](types_of_missingness)

- **variable X**
  - missing at random (MAR)
    \( I(M_X, X \mid Z) \)
  - missing completely at random (MCAR)
    \( I(M_X, X) \)
  - hidden
    \( p(M_X = 1) = 1 \)

Figure 9: Types of missingness.

MAR/MCAR terminology stems from [LR87].
The simplest scheme to learn from incomplete data $D$, e.g., the vertex potentials $(p_v)_{v \in V}$ of a Bayesian network, is **complete case analysis** (also called **casewise deletion**): use only complete cases

$$D_{\text{compl}} := \{ d \in D \mid d \text{ is complete} \}$$

<table>
<thead>
<tr>
<th>case</th>
<th>F</th>
<th>L</th>
<th>B</th>
<th>D</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>.</td>
<td>0</td>
<td>.</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>.</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 10: Incomplete data and data used in complete case analysis (highlighted).

If $D$ is MCAR, estimations based on the subsample $D_{\text{compl}}$ are unbiased for $D_{\text{true}}$.

But for higher-dimensional data (i.e., with a larger number of variables), complete cases might become rare.

Let each variable have a probability for missing values of 0.05, then for 20 variables the probability of a case to be complete is

$$(1 - 0.05)^{20} \approx 0.36$$

for 50 variables it is $\approx 0.08$, i.e., most cases are deleted.
A higher case rate can be achieved by **available case analysis**. If a quantity has to be estimated based on a subset $W \subseteq V$ of variables, e.g., the vertex potential $p_v$ of a specific vertex $v \in V$ of a Bayesian network ($W = \text{fam}(v)$), use only complete cases of $D|_W$

$$(D|_W)_{\text{compl}} = \{ d \in D|_W \mid d \text{ is complete} \}$$

If $D$ is MCAR, estimations based on the subsample $(D_W)_{\text{compl}}$ are unbiased for $(D_W)_{\text{true}}$.

<table>
<thead>
<tr>
<th>case</th>
<th>F</th>
<th>L</th>
<th>B</th>
<th>D</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>.</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>.</td>
<td>0</td>
<td>.</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>.</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 11: Incomplete data and data used in available case analysis for estimating the potential $p_L(L \mid F)$ (highlighted).
Let $V$ be a set of variables and $d$ be an incomplete case. A (complete) case $\bar{d}$ with

$$\bar{d}(v) = d(v), \quad \forall v \in \text{var}(d)$$

is called a **completion of** $d$.

A probability distribution

$$\bar{d} : \text{dom}(V) \to [0,1]$$

with

$$\bar{d}^{\text{var}(d)} = \text{epd}_d$$

is called a **distribution of completions** of $d$ (or a **fuzzy completion of** $d$).

**Example** If $V := \{F, L, B, D, H\}$ and $d := (2,.,0,1,.)$ an incomplete case, then

$$\bar{d}_1 := (2,1,0,1,1)$$

$$\bar{d}_2 := (2,2,0,1,0)$$

etc. are possible completions, but $e := (1,1,0,1,1)$ is not.

Assume $\text{dom}(v) := \{0,1,2\}$ for all $v \in V$. The potential

$$\bar{d} : \text{dom}(V) \to [0,1]$$

\[ (x_v)_{v \in V} \mapsto \begin{cases} \frac{1}{5}, & \text{if } x_F = 2, x_B = 0, \text{ and } x_D = 1 \\ 0, & \text{otherwise} \end{cases} \]

is the uniform distribution of completions of $d$.

Given a bayesian network structure $G := (V, E)$ on a set of variables $V$ and a "fuzzy data set" $D \in \text{pdf}(V)^*$ of "fuzzy cases" (pdfs $q$ on $V$). **Learning the parameters of the bayesian network from "fuzzy cases"** $D$ means to find vertex potentials $(p_v)_{v \in V}$ s.t. the **maximum likelihood criterion**, i.e., the probability of the data given the bayesian network is maximal:

$$\text{find } (p_v)_{v \in V} \text{ s.t. } p(D) \text{ is maximal,}$$

where $p$ denotes the JPD build from $(p_v)_{v \in V}$. Here,

$$p(D) := \prod_{q \in D} \prod_{v \in V} \prod_{x \in \text{dom}(\text{fam}(v))} (p_v(x)) q^{\text{fam}(v)}(x) q^{\text{pa}(v)}(y)$$

\[ (x_v)_{v \in V} \mapsto \begin{cases} \frac{1}{5}, & \text{if } x_F = 2, x_B = 0, \text{ and } x_D = 1 \\ 0, & \text{otherwise} \end{cases} \]

\[ (x_v)_{v \in V} \mapsto \begin{cases} \frac{1}{5}, & \text{if } x_F = 2, x_B = 0, \text{ and } x_D = 1 \\ 0, & \text{otherwise} \end{cases} \]

\[ (x_v)_{v \in V} \mapsto \begin{cases} \frac{1}{5}, & \text{if } x_F = 2, x_B = 0, \text{ and } x_D = 1 \\ 0, & \text{otherwise} \end{cases} \]

\[ (x_v)_{v \in V} \mapsto \begin{cases} \frac{1}{5}, & \text{if } x_F = 2, x_B = 0, \text{ and } x_D = 1 \\ 0, & \text{otherwise} \end{cases} \]
Maximum likelihood estimates

If $D$ is incomplete data, in general we are looking for
(i) distributions of completions $\bar{D}$ and
(ii) vertex potentials $(p_v)_{v \in V}$,
that are
(i) compatible, i.e.,
$$\bar{d} = \operatorname{infer}_{(p_v)_{v \in V}}(d)$$
for all $\bar{d} \in \bar{D}$ and s.t.
(ii) the probability, that the completed data $\bar{D}$ has been generated from the bayesian network specified by $(p_v)_{v \in V}$, is maximal:
$$p((p_v)_{v \in V}, \bar{D}) := \prod_{d \in D} \prod_{v \in V} \prod_{x \in \operatorname{dom}(\operatorname{fam}(v))} (p_v(x))^{d(\operatorname{fam}(v))(x)}$$
(with the usual constraints that $\operatorname{Im} p_v \subseteq [0, 1]$ and $\sum_{y \in \operatorname{dom}(\operatorname{pa}(v))} P_v(x|y) = 1$ for all $v \in V$ and $x \in \operatorname{dom}(v)$).

Unfortunately this is
• a non-linear,
• high-dimensional,
• for bayesian networks in general even non-convex optimization problem without closed form solution.

Any non-linear optimization algorithm (gradient descent, Newton-Raphson, BFGS, etc.) could be used to search local maxima of this probability function.
Let the following bayesian network structure and training data given.

<table>
<thead>
<tr>
<th>case</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{A} \rightarrow \text{B}
\]

Optimization Problem (1/3)

<table>
<thead>
<tr>
<th>case</th>
<th>A</th>
<th>B</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>(\alpha_4)</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>(1 - \alpha_4)</td>
</tr>
<tr>
<td>5,6</td>
<td>1</td>
<td>0</td>
<td>(2 \alpha_5)</td>
</tr>
<tr>
<td>5,6</td>
<td>0</td>
<td>0</td>
<td>(2 (1 - \alpha_5))</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>(\beta_{10})</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>(1 - \beta_{10})</td>
</tr>
</tbody>
</table>

\[
\theta = p(A = 1) \\
\eta_1 = p(B = 1 | A = 1) \\
\eta_2 = p(B = 1 | A = 0)
\]

\[
p(D) = \theta^{4+\alpha_4+2 \alpha_5} (1 - \theta)^3 (1 - \theta^{\alpha_4}) (1 - \theta^{\alpha_5}) \\
\eta_1^{1+\alpha_4+\beta_{10}} (1 - \eta_1)^{2+2 \alpha_5} (1 - \beta_{10}) \\
\eta_2^{2+\alpha_5} (1 - \eta_2)^{1+2 (1-\alpha_5)}
\]
From parameters

\[ \theta = p(A = 1) \]
\[ \eta_1 = p(B = 1 | A = 1) \]
\[ \eta_2 = p(B = 1 | A = 0) \]

we can compute distributions of completions:

\[ \alpha_4 = p(A = 1 | B = 1) = \frac{p(B = 1 | A = 1) p(A = 1)}{\sum_{a \in A} p(B = 1 | A = a) p(A = a)} = \frac{\theta \eta_1}{\theta \eta_1 + (1 - \theta) \eta_2} \]

\[ \alpha_5 = p(A = 1 | B = 0) = \frac{p(B = 0 | A = 1) p(A = 1)}{\sum_{a \in A} p(B = 0 | A = a) p(A = a)} = \frac{\theta (1 - \eta_1)}{\theta (1 - \eta_1) + (1 - \theta) (1 - \eta_2)} \]

\[ \beta_{10} = p(B = 1 | A = 1) = \eta_1 \]

Substituting \( \alpha_4, \alpha_5 \) and \( \beta_{10} \) in \( p(D) \), finally yields:

\[ p(D) = \theta^{4 + \frac{\theta \eta_1}{\eta_1 + (1 - \theta) \eta_2} + 2 \frac{\theta (1 - \eta_1)}{\theta (1 - \eta_1) + (1 - \theta) (1 - \eta_2)}} \]
\[ \cdot (1 - \theta)^{6 - \frac{\theta \eta_1}{\eta_1 + (1 - \theta) \eta_2} - 2 \frac{\theta (1 - \eta_1)}{\theta (1 - \eta_1) + (1 - \theta) (1 - \eta_2)}} \]
\[ \cdot \eta_1^{1 + \frac{\theta \eta_1}{\eta_1 + (1 - \theta) \eta_2} + \eta_1} \]
\[ \cdot (1 - \eta_1)^{3 + 2 \frac{\theta (1 - \eta_1)}{\theta (1 - \eta_1) + (1 - \theta) (1 - \eta_2)}} \]
\[ \cdot \eta_2^{3 - \frac{\theta \eta_1}{\eta_1 + (1 - \theta) \eta_2}} \]
\[ \cdot (1 - \eta_2)^{3 - 2 \frac{\theta (1 - \eta_1)}{\theta (1 - \eta_1) + (1 - \theta) (1 - \eta_2)}} \]
EM algorithm

For bayesian networks a widely used technique to search local maxima of the probability function $p$ is

**Expectation-Maximization** (EM, in essence a gradient descent).

At the beginning, $(p_v)_{v \in V}$ are initialized, e.g., by complete, by available case analysis, or at random.

Then one computes alternating

**expectation or E-step:**

$$\bar{d} := \text{infer}_{(p_v)_{v \in V}}(d), \quad \forall d \in D$$

(forcing the compatibility constraint) and

**maximization or M-step:**

$$(p_v)_{v \in V}$ with maximal $p((p_v)_{v \in V}, \bar{D})$$

keeping $\bar{D}$ fixed.

The E-step is implemented using an inference algorithm, e.g., clustering [Lau95]. The variables with observed values are used as evidence, the variables with missing values form the target domain.

The M-step is implemented using lemma 2:

$$p_v(x|y) := \frac{\sum_{q \in D} q^{\text{fam}(v)}(x, y)}{\sum_{q \in D} q^{\text{pa}(v)}(y)}$$

See [BKS97] and [FK03] for further optimizations aiming at faster convergence.
Example

Let the following Bayesian network structure and training data given.

\[
\begin{array}{c|cc}
\text{case} & A & B \\
1 & 0 & 0 \\
2 & 0 & 1 \\
3 & 0 & 1 \\
4 & . & 1 \\
5 & . & 0 \\
6 & . & 0 \\
7 & 1 & 0 \\
8 & 1 & 0 \\
9 & 1 & 1 \\
10 & 1 & .
\end{array}
\]

Using complete case analysis we estimate (1st M-step)

\[p(A) = (0.5, 0.5)\]

and

\[p(B|A) = \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0.333 & 0.667 \\ 1 & 0.667 & 0.333 \end{bmatrix}\]

Then we estimate the distributions of completions (1st E-step)

\[
\begin{array}{c|cc|cc}
\text{case} & B & p(A=0) & p(A=1) \\
4 & 1 & 0.667 & 0.333 \\
5,6 & 0 & 0.333 & 0.667 \\
\end{array}
\]

\[
\begin{array}{c|cc|cc}
\text{case} & A & p(B=0) & p(B=1) \\
10 & 1 & 0.667 & 0.333 \\
\end{array}
\]

From that we estimate (2nd M-step)

\[p(A) = (0.433, 0.567)\]

and

\[p(B|A) = \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0.385 & 0.706 \\ 1 & 0.615 & 0.294 \end{bmatrix}\]

Then we estimate the distributions of completions (2nd E-step)

\[
\begin{array}{c|cc|cc}
\text{case} & B & p(A=0) & p(A=1) \\
4 & 1 & 0.615 & 0.385 \\
5,6 & 0 & 0.294 & 0.706 \\
\end{array}
\]

\[
\begin{array}{c|cc|cc}
\text{case} & A & p(B=0) & p(B=1) \\
10 & 1 & 0.706 & 0.294 \\
\end{array}
\]

From that we estimate (3rd M-step)

\[p(A) = (0.420, 0.580)\]

and

\[p(B|A) = \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0.378 & 0.710 \\ 1 & 0.622 & 0.290 \end{bmatrix}\]

etc.

Figure 12: Convergence of the EM algorithm (black p(A=1), red p(B=1|A=0), green p(B=1|A=1)).
1. Incomplete Data

2. Incomplete Data for Parameter Learning (EM algorithm)

3. An Example

---

**Definition 1.** Let $\mathcal{V}$ be a set of variables and let $C \in \mathcal{V}$ be a variable called **target variable**.

The Bayesian network structure on $\mathcal{V}$ defined by the set of edges

$$E := \{(C, X) \mid X \in \mathcal{V}, X \neq C\}$$

is called **naive Bayesian network with target** $C$.

Naive Bayesian networks typically are used as classifiers for $C$ and thus called **naive Bayesian classifier**.
Naive Bayesian Network

A naive bayesian network encodes both,

- **strong dependency assumptions:**
  there are no two variables that are independent, i.e.,
  \[ \neg I(X, Y) \ \forall X, Y \]

- **strong independency assumptions:**
  each pair of variables is conditionally independent given a very small set of variables:
  \[ I(X, Y | C) \ \forall X, Y \neq C \]

\[
\begin{align*}
C & \to \ X_1, \ X_2, \ X_3, \ldots, \ X_n \\
\end{align*}
\]

Learning a Naive Bayesian Network means to estimate

\[ p(C) \quad \text{and} \quad p(X_i | C) \]

Inferencing in a Naive Bayesian Network means to compute

\[ p(C | X_1 = x_1, \ldots, X_n = x_n) \]

which is due to Bayes formula:

\[
\begin{align*}
p(C | X_1 = x_1, \ldots, X_n = x_n) &= \frac{p(X_1 = x_1, \ldots, X_n = x_n | C) p(C)}{p(X_1 = x_1, \ldots, X_n = x_n)} \\
&= \frac{\prod_i p(X_i = x_i | C) p(C)}{p(X_1 = x_1, \ldots, X_n = x_n)} \\
&= \left( \prod_i p(X_i = x_i | C) p(C) \right)^C
\end{align*}
\]

Be careful,

\[ p(X_1 = x_1, \ldots, X_n = x_n) \neq \prod_i p(X_i = x_i) \]

in general and we do not have access to this probability easily.
UCI Mushroom Data

The UCI mushroom data contains 23 attributes of 8124 different mushrooms.

<table>
<thead>
<tr>
<th>edible</th>
<th>cap-shape</th>
<th>cap-surface</th>
<th>cap-color</th>
<th>bruises</th>
<th>gill-attachment</th>
<th>gill-spacing</th>
<th>gill-size</th>
<th>stalk-shape</th>
<th>stalk-root</th>
<th>stalk-surface-above-rings</th>
<th>stalk-surface-below-rings</th>
<th>stalk-color-above-rings</th>
<th>stalk-color-below-rings</th>
<th>veil-type</th>
<th>veil-color</th>
<th>ring-number</th>
<th>ring-type</th>
<th>spore-print-color</th>
<th>population</th>
<th>habitat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>p</td>
<td>x</td>
<td>s</td>
<td>n</td>
<td>t</td>
<td>p</td>
<td>f</td>
<td>c</td>
<td>e</td>
<td>e</td>
<td>s</td>
<td>w</td>
<td>w</td>
<td>p</td>
<td>o</td>
<td>p</td>
<td>k</td>
<td>s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>e</td>
<td>x</td>
<td>s</td>
<td>y</td>
<td>t</td>
<td>a</td>
<td>f</td>
<td>c</td>
<td>b</td>
<td>k</td>
<td>e</td>
<td>c</td>
<td>s</td>
<td>w</td>
<td>p</td>
<td>w</td>
<td>p</td>
<td>n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>e</td>
<td>b</td>
<td>s</td>
<td>w</td>
<td>t</td>
<td>l</td>
<td>f</td>
<td>c</td>
<td>b</td>
<td>n</td>
<td>e</td>
<td>c</td>
<td>s</td>
<td>w</td>
<td>p</td>
<td>w</td>
<td>o</td>
<td>p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>p</td>
<td>x</td>
<td>y</td>
<td>w</td>
<td>t</td>
<td>p</td>
<td>f</td>
<td>c</td>
<td>n</td>
<td>e</td>
<td>e</td>
<td>s</td>
<td>s</td>
<td>w</td>
<td>p</td>
<td>w</td>
<td>o</td>
<td>p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>e</td>
<td>x</td>
<td>s</td>
<td>g</td>
<td>f</td>
<td>n</td>
<td>f</td>
<td>w</td>
<td>b</td>
<td>k</td>
<td>t</td>
<td>e</td>
<td>s</td>
<td>w</td>
<td>p</td>
<td>w</td>
<td>o</td>
<td>e</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>e</td>
<td>x</td>
<td>y</td>
<td>y</td>
<td>t</td>
<td>a</td>
<td>f</td>
<td>c</td>
<td>b</td>
<td>n</td>
<td>e</td>
<td>c</td>
<td>s</td>
<td>w</td>
<td>p</td>
<td>w</td>
<td>o</td>
<td>p</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

edible: e = edible, p = poisonous

cap-shape: b=bell, c=conical, x=convex, f=flat, k=knobbed, s=sunken
e tc

Mushroom has missing values:

- in variable $X_{11} = stalk-root$, starting at case 3985.
Learning Task

We want to learn target $C = \text{edible}$ based on all the other attributes, $X_1, \ldots, X_{22} = \text{cap-shape}, \ldots, \text{habitat}$. We split the dataset randomly in

7124 training cases plus 1000 test cases

class distribution:

<table>
<thead>
<tr>
<th>actual = e</th>
<th>529</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>471</td>
</tr>
</tbody>
</table>

Accuracy of constant classifier (always predicts majority class e):

$$\text{acc} = 0.529$$

Complete Case Analysis

Learning only from the 4942 complete cases (out of 7124), we are quite successful on the 702 complete test cases:

confusion matrix:

<table>
<thead>
<tr>
<th>predicted =</th>
<th>e</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual = e</td>
<td>433</td>
<td>3</td>
</tr>
<tr>
<td>p</td>
<td>0</td>
<td>266</td>
</tr>
</tbody>
</table>

$$\text{acc} = 0.9957$$
Complete Case Analysis

But the classifier deteriorates dramatically, once evaluated on all 1000 cases, thereof 298 containing missing values:

confusion matrix:

<table>
<thead>
<tr>
<th>predicted = e</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual = e</td>
<td>516 13</td>
</tr>
<tr>
<td></td>
<td>p 201 270</td>
</tr>
</tbody>
</table>

\[ \text{acc} = 0.786 \]

Diagnostics:

6937 p k y n f f f c n b t s s p w p w o e w v d

\[ p(X_9 = b | C) = 0 \]
as \( X_9 = b \) occurs only with \( X_{11} = . \)

For the whole dataset:

<table>
<thead>
<tr>
<th>( X_9 = )</th>
<th>b</th>
<th>e</th>
<th>g</th>
<th>h</th>
<th>k</th>
<th>n</th>
<th>o</th>
<th>p</th>
<th>r</th>
<th>u</th>
<th>w</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_{11} = ) false</td>
<td>0</td>
<td>0</td>
<td>656</td>
<td>720</td>
<td>408</td>
<td>984</td>
<td>0</td>
<td>1384</td>
<td>24</td>
<td>480</td>
<td>966</td>
<td>22</td>
</tr>
<tr>
<td>= true</td>
<td>1728</td>
<td>96</td>
<td>96</td>
<td>12</td>
<td>0</td>
<td>64</td>
<td>64</td>
<td>108</td>
<td>0</td>
<td>12</td>
<td>236</td>
<td>64</td>
</tr>
</tbody>
</table>
Available Case Analysis

If we use available case analysis, this problem is fixed.

<table>
<thead>
<tr>
<th>predicted</th>
<th>e</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual</td>
<td>e</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td></td>
<td>p</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>p</td>
</tr>
</tbody>
</table>

acc = 0.994

EM for predictor variables in Naive Bayesian Networks always converges to the available case estimates (easy exercise; compute the update formula).

---

### Variable Importance / Mutual Information

**Definition 2. mutual information** of two random variables $X$ and $Y$:

$$\text{MI}(X, Y) := \sum_{x \in \text{dom } X, y \in \text{dom } Y} p(X = x, Y = y) \log \frac{p(X = x, Y = y)}{p(X = x) p(Y = y)}$$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$\text{MI}(X, C)$</th>
<th>$X$</th>
<th>$\text{MI}(X, C)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>0.04824</td>
<td>X12</td>
<td>0.28484</td>
</tr>
<tr>
<td>X2</td>
<td>0.02901</td>
<td>X13</td>
<td>0.27076</td>
</tr>
<tr>
<td>X3</td>
<td>0.03799</td>
<td>X14</td>
<td>0.24917</td>
</tr>
<tr>
<td>X4</td>
<td>0.19339</td>
<td>X15</td>
<td>0.24022</td>
</tr>
<tr>
<td>X5</td>
<td><strong>0.90573</strong></td>
<td>X16</td>
<td><strong>0.00000</strong></td>
</tr>
<tr>
<td>X6</td>
<td>0.01401</td>
<td>X17</td>
<td>0.02358</td>
</tr>
<tr>
<td>X7</td>
<td>0.10173</td>
<td>X18</td>
<td>0.03863</td>
</tr>
<tr>
<td>X8</td>
<td>0.23289</td>
<td>X19</td>
<td><strong>0.31982</strong></td>
</tr>
<tr>
<td>X9</td>
<td><strong>0.41907</strong></td>
<td>X20</td>
<td>0.48174</td>
</tr>
<tr>
<td>X10</td>
<td>0.00765</td>
<td>X21</td>
<td>0.20188</td>
</tr>
<tr>
<td>X11</td>
<td>0.09716</td>
<td>X22</td>
<td>0.15877</td>
</tr>
<tr>
<td>X12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X22</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
If we use the 4 variables with highest mutual information only,

- $X_5 = \text{odor}$
- $X_{20} = \text{spore-print-color}$
- $X_9 = \text{gill-color}$
- $X_{19} = \text{ring-type}$

we still get very good results.

Confusion matrix:

<table>
<thead>
<tr>
<th></th>
<th>e</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>actual</strong> = e</td>
<td>529</td>
<td>0</td>
</tr>
<tr>
<td>p</td>
<td>6</td>
<td>465</td>
</tr>
</tbody>
</table>

$\text{acc} = 0.994$
Fresh random split.

<table>
<thead>
<tr>
<th></th>
<th>Predicted</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>X5, X9, X19, and X20:</td>
<td>e p</td>
<td>e 541 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>p 1 454</td>
</tr>
<tr>
<td>Accuracy</td>
<td>.995</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Predicted</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1, X2, X3, and X4:</td>
<td>e p</td>
<td>e 419 126</td>
</tr>
<tr>
<td></td>
<td></td>
<td>p 101 354</td>
</tr>
<tr>
<td>Accuracy</td>
<td>.773</td>
<td></td>
</tr>
</tbody>
</table>

**Naive Bayesian Network / Cluster Analysis**

Naive Bayesian Networks also could be used for cluster analysis.

The unknown cluster membership is modelled by a hidden variable \( C \) called **latent class**.

EM algorithm is used to "learn" fuzzy cluster memberships.

Naive Bayesian Networks used this way are a specific instance of so called **model-based clustering**.
Each cluster contains "similar cases", i.e., cases that contain cooccuring patterns of values.
Summary

- To learn parameters from data with missing values, sometimes simple heuristics as complete or available case analysis can be used.

- Alternatively, one can define a joint likelihood for distributions of completions and parameters.

- In general, this gives rise to a nonlinear optimization problem. But for given distributions of completions, maximum likelihood estimates can be computed analytically.

- To solve the ML optimization problem, one can employ the expectation maximization (EM) algorithm:
  - parameters $\rightarrow$ completions (expectation; inference)
  - completions $\rightarrow$ parameters (maximization; parameter learning)

References


