Bayesian Networks

10. Parameter Learning / Missing Values

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Business Economics and Information Systems
& Institute for Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de

1. Incomplete Data

2. Incomplete Data for Parameter Learning (EM algorithm)

3. An Example

Complete and incomplete cases

Let V be a set of variables. A **complete case** is a function

$$c : V \to \bigcup_{v \in V} \mathrm{dom}(V)$$

with $c(v) \in \mathrm{dom}(V)$ 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 $\mathrm{var}(c) := W$ and say, the values of the variables $V \setminus W$ are **missing** or **not observed**.

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

case	F	L	B	D	H
1	0	0	0	0	0
2	0	0	0	0	0
3	1	1	1	1	0
4	0	0	1	1	1
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	1	1
8	0	0	0	0	0
9	0	0	1	1	1
10	1	1	0	1	1

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

case	F	L	B	D	H
1	0	0	0	0	0
2	.	0	0	0	0
3	1	1	1	1	0
4	0	0	.	1	1
5	0	0	0	0	0
6	0	0	0	0	0
7	0	.	0	.	1
8	0	0	0	0	0
9	0	0	1	1	1
10	1	1	.	1	1

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

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.

case	F	M_F	L	M_L	B	M_B	D	M_D	H	M_H
1	0	0	0	0	0	0	0	0	0	0
2	.	1	0	0	0	0	0	0	0	0
3	1	0	1	0	1	0	1	0	0	0
4	0	0	0	0	.	1	1	0	1	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	.	1	0	0	.	1	1	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	1	0	1	0	1	0
10	1	0	1	0	.	1	1	0	1	0

Figure 3: Incomplete data for $V := \{F, L, B, D, H\}$ and missing value indicators.

Types of missingness / MCAR

A variable $v \in V$ is called **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 **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_{obs} is an unbiased estimator for the expectation of v_{true}; here

case	v_{true}	v_{observed}
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

Figure 4: Data with a variable v MCAR. Missing values are stroken through.

$$\hat{\mu}(v_{\text{obs}}) = \frac{1}{10}(2 \cdot 1 + 4 \cdot 3 + 2 \cdot 3 + 2 \cdot 4)$$
$$= \frac{1}{15}(3 \cdot 1 + 6 \cdot 3 + 3 \cdot 3 + 3 \cdot 4) = \hat{\mu}(v_{\text{true}})$$

Types of missingness / MAR

A variable $v \in V$ is called **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 **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,

case	v_{true}	v_{observed}	h
1	1	.	0
2	2	2	0
3	3	.	0
4	3	3	0
5	1	1	0
6	3	3	0
7	1	1	0
8	2	.	0
9	2	2	0

case	v_{true}	v_{observed}	h
10	3	.	1
11	4	4	1
12	4	.	1
13	3	3	1

case	v_{true}	v_{observed}	h
14	3	.	2
15	4	4	2
16	4	.	2
17	5	5	2
18	3	.	2
19	5	.	2
20	3	3	2
21	4	.	2
22	5	.	2

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:

$$\begin{aligned}
&\sum_{h=0}^{2} \hat{\mu}(v|H=h)p(H=h) \\
=&2 \cdot \frac{9}{22} + 3.5 \cdot \frac{4}{22} + 4 \cdot \frac{9}{22} \\
\neq&\frac{1}{11} \sum_{\substack{i=1,\dots,22 \\ v_i \neq .}} v_i \\
=&2 \cdot \frac{6}{11} + 3.5 \cdot \frac{2}{11} + 4 \cdot \frac{3}{11}
\end{aligned}$$

case	v_{true}	$v_{observed}$	h
1	1	.	0
2	2	2	0
3	3	.	0
4	3	3	0
5	1	1	0
6	3	3	0
7	1	1	0
8	2	.	0
9	2	2	0
10	3	.	1
11	4	4	1
12	4	.	1
13	3	3	1
14	3	.	2
15	4	4	2
16	4	.	2
17	5	5	2
18	3	.	2
19	5	.	2
20	3	3	2
21	4	.	2
22	5	.	2

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

Types of missingness / missing systematically

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.

case	v_{true}	$v_{observed}$
1	1	.
2	1	1
3	2	.
4	3	.
5	3	3
6	2	2
7	1	1
8	2	.
9	3	.
10	2	2

Figure 6: Data with a variable v missing systematically.

Again, the sample mean is not unbiased; expectation can only be estimated if we have background knowledge about the probabilities of a missing value dependend on its true value.

Types of missingness / hidden variables

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.

case	I_{true}	I_{obs}	E	C
1	1	.	0	0
2	2	.	1	2
3	2	.	2	1
4	2	.	2	2
5	1	.	0	2
6	2	.	2	0
7	1	.	1	2
8	0	.	2	1
9	1	.	2	2
10	2	.	2	1

Figure 7: Data with a hidden variable I.

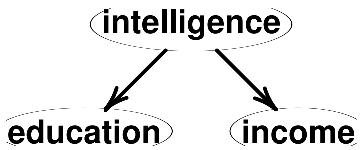


Figure 8: Suggested dependency of variables I, E, and C.

types of missingness

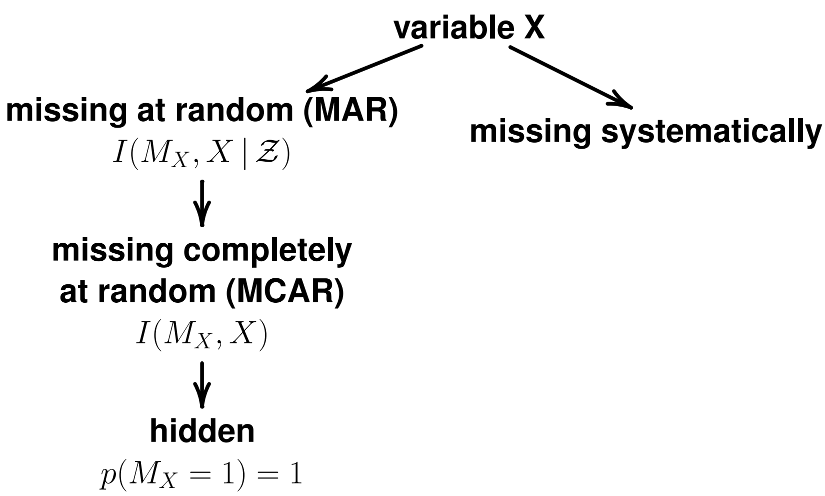


Figure 9: Types of missingness.

MAR/MCAR terminology stems from [LR87].

complete case analysis

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

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

case	F	L	B	D	H
1	0	0	0	0	0
2	.	0	0	0	0
3	1	1	1	1	0
4	0	0	.	1	1
5	0	0	0	0	0
6	0	0	0	0	0
7	0	.	0	.	1
8	0	0	0	0	0
9	0	0	1	1	1
10	1	1	.	1	1

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

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

complete case analysis (2/2)

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 ≈ 0.08, i.e., most cases are deleted.

available case analysis

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 vertext potential p_v of a specific vertex $v \in V$ of a Bayesian network ($W = \text{fam}(v)$), use only complete cases of $D|_W$

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

case	F	L	B	D	H
1	0	0	0	0	0
2	.	0	0	0	0
3	1	1	1	1	0
4	0	0	.	1	1
5	0	0	0	0	0
6	0	0	0	0	0
7	0	.	0	.	1
8	0	0	0	0	0
9	0	0	1	1	1
10	1	1	.	1	1

Figure 11: Incomplete data and data used in available case analysis for estimating the potential $p_L(L \mid F)$ (highlighted).

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

1. Incomplete Data

2. Incomplete Data for Parameter Learning (EM algorithm)

3. An Example

completions

Let V be a set of variables and d be an incomplete case. A (complete) case $\bar{d}$ with

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

is called a **completion of** d.

A probability distribution

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

with

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

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

$$\bar{d} : \mathrm{dom}(V) \to [0,1]$$
$$(x_v)_{v \in V} \mapsto \begin{cases} \frac{1}{9}, & \text{if } x_F = 2, x_B = 0, \\ & \quad \text{and } x_D = 1 \\ 0, & \text{otherwise} \end{cases}$$

is the uniform distribution of completions of d.

learning from "fuzzy cases"

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:

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

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

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

Lemma 1. *$p(D)$ is maximal iff*

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

(if there is a $q \in D$ with $q^{\downarrow \mathrm{pa}(v)} > 0$, otherwise $p_v(x|y)$ can be choosen arbitrarily – $p(D)$ does not depend on it).

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.,

$$\bar{d} = \text{infer}_{(p_v)_{v \in V}}(d)$$

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:

$$p((p_v)_{v \in V}, \bar{D}) := \prod_{\bar{d} \in \bar{D}} \prod_{v \in V} \prod_{x \in \text{dom}(\text{fam}(v))} (p_v(x))^{\bar{d}^{\downarrow \text{fam}(v)}(x)}$$

(with the usual constraints that Im$p_v \subseteq [0,1]$ and $\sum_{y \in \text{dom}(\text{pa}(v))} p_v(x|y) = 1$ for all $v \in V$ and $x \in \text{dom}(v)$).

Maximum likelihood estimates

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.

Example

Let the following bayesian network structure and training data given.

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	.

Optimization Problem (1/3)

case	A	B	weight
1	0	0	1
2	0	1	1
3	0	1	1
7	1	0	1
8	1	0	1
9	1	1	1
4	1	1	α_4
4	0	1	$1-\alpha_4$
5,6	1	0	$2\,\alpha_5$
5,6	0	0	$2\,(1-\alpha_5)$
10	1	1	β_{10}
10	1	0	$1-\beta_{10}$

$$\begin{aligned}\theta =& p(A=1)\\ \eta_1 =& p(B=1\,|\,A=1)\\ \eta_2 =& p(B=1\,|\,A=0)\end{aligned}$$

$$\begin{aligned}p(D) =& \theta^{4+\alpha_4+2\,\alpha_5}\,(1-\theta)^{3+(1-\alpha_4)+2\,(1-\alpha_5)}\,\eta_1^{1+\alpha_4+\beta_{10}}\,(1-\eta_1)^{2+2\,\alpha_5+(1-\beta_{10})}\\ &\cdot\eta_2^{2+(1-\alpha_4)}\,(1-\eta_2)^{1+2\,(1-\alpha_5)}\end{aligned}$$

Optimization Problem (2/3)

From parameters

$$\theta = p(A=1)$$
$$\eta_1 = p(B=1 \mid A=1)$$
$$\eta_2 = p(B=1 \mid A=0)$$

we can compute distributions of completions:

$$\alpha_4 = p(A=1 \mid B=1) = \frac{p(B=1 \mid A=1)\, p(A=1)}{\sum_{a \in A} p(B=1 \mid A=a)\, p(A=a)} = \frac{\theta\, \eta_1}{\theta\, \eta_1 + (1-\theta)\, \eta_2}$$

$$\alpha_5 = p(A=1 \mid B=0) = \frac{p(B=0 \mid A=1)\, p(A=1)}{\sum_{a \in A} p(B=0 \mid A=a)\, p(A=a)} = \frac{\theta\,(1-\eta_1)}{\theta\,(1-\eta_1) + (1-\theta)\,(1-\eta_2)}$$

$$\beta_{10} = p(B=1 \mid A=1) = \eta_1$$

Optimization Problem (3/3)

Substituting α_4, α_5 and β_{10} in $p(D)$, finally yields:

$$p(D) = \theta^{4 + \frac{\theta\,\eta_1}{\theta\,\eta_1 + (1-\theta)\eta_2} + 2\,\frac{\theta\,(1-\eta_1)}{\theta\,(1-\eta_1)+(1-\theta)(1-\eta_2)}}$$
$$\cdot\, (1-\theta)^{6 - \frac{\theta\,\eta_1}{\theta\,\eta_1 + (1-\theta)\eta_2} - 2\,\frac{\theta\,(1-\eta_1)}{\theta\,(1-\eta_1)+(1-\theta)(1-\eta_2)}}$$
$$\cdot\, \eta_1^{1 + \frac{\theta\,\eta_1}{\theta\,\eta_1 + (1-\theta)\eta_2} + \eta_1}$$
$$\cdot\, (1-\eta_1)^{3 + 2\,\frac{\theta\,(1-\eta_1)}{\theta\,(1-\eta_1)+(1-\theta)(1-\eta_2)} - \eta_1}$$
$$\cdot\, \eta_2^{3 - \frac{\theta\,\eta_1}{\theta\,\eta_1 + (1-\theta)\eta_2}}$$
$$\cdot\, (1-\eta_2)^{3 - 2\,\frac{\theta\,(1-\eta_1)}{\theta\,(1-\eta_1)+(1-\theta)(1-\eta_2)}}$$

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:

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

keeping $\bar{D}$ fixed.

EM algorithm

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:

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

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

Example

Let the following bayesian network structure and training data given.

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	.

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

$$p(A) = (0.5, 0.5)$$

and

$$p(B|A) = \begin{array}{r|ll} A & 0 & 1 \\ \hline B=0 & 0.333 & 0.667 \\ 1 & 0.667 & 0.333 \end{array}$$

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

case	B	p(A=0)	p(A=1)
4	1	0.667	0.333
5,6	0	0.333	0.667
case	**A**	**p(B=0)**	**p(B=1)**
10	1	0.667	0.333

example / second & third step

From that we estimate (2nd M-step)

$$p(A) = (0.433, 0.567)$$

and

$$p(B|A) = \begin{array}{r|ll} A & 0 & 1 \\ \hline B=0 & 0.385 & 0.706 \\ 1 & 0.615 & 0.294 \end{array}$$

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

case	B	p(A=0)	p(A=1)
4	1	0.615	0.385
5,6	0	0.294	0.706
case	**A**	**p(B=0)**	**p(B=1)**
10	1	0.706	0.294

From that we estimate (3rd M-step)

$$p(A) = (0.420, 0.580)$$

and

$$p(B|A) = \begin{array}{r|ll} A & 0 & 1 \\ \hline B=0 & 0.378 & 0.710 \\ 1 & 0.622 & 0.290 \end{array}$$

etc.

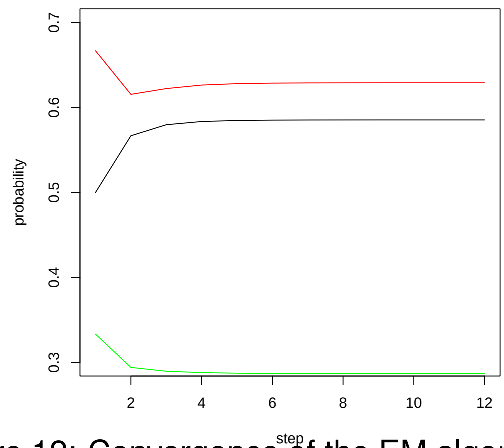


Figure 12: Convergence of the EM algorithm (black p(A=1), red p(B=1|A=0), green p(B=1|A=1)).

1. **Incomplete Data**

2. **Incomplete Data for Parameter Learning (EM algorithm)**

3. **An Example**

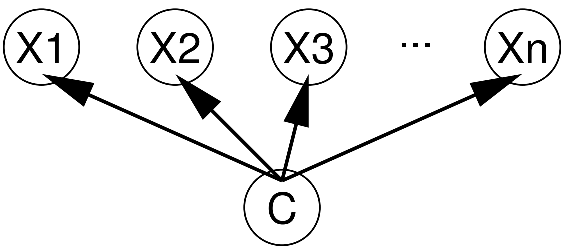
Naive Bayesian Network

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

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

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

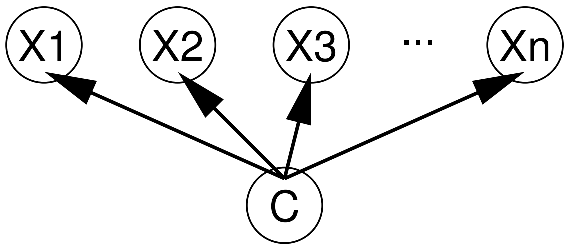
A naive bayesian network encodes both,

- strong dependency assumptions:
  there are no two variables that are independent, i.e.,

$$\neg I(X, Y) \quad \forall X, Y$$

- strong independency assumptions:
  each pair of variables is conditionally independent
  given a very small set of variables:

$$I(X, Y | C) \quad \forall X, Y \neq C$$



Naive Bayesian Network

Learning a Naive Bayesian Network means to estimate

$$p(C) \qquad \text{and} \qquad p(X_i \,|\, C)$$

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:

$$\begin{aligned} p(C \,|\, X_1 = x_1, \ldots, X_n = x_n) &= \frac{p(X_1 = x_1, \ldots, X_n = x_n \,|\, C)\, p(C)}{p(X_1 = x_1, \ldots, X_n = x_n)} \\ &= \frac{\prod_i p(X_i = x_i \,|\, C)\, p(C)}{p(X_1 = x_1, \ldots, X_n = x_n)} \\ &= \Big(\prod_i p(X_i = x_i \,|\, C)\, p(C)\Big)\Big|^{C} \end{aligned}$$

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.

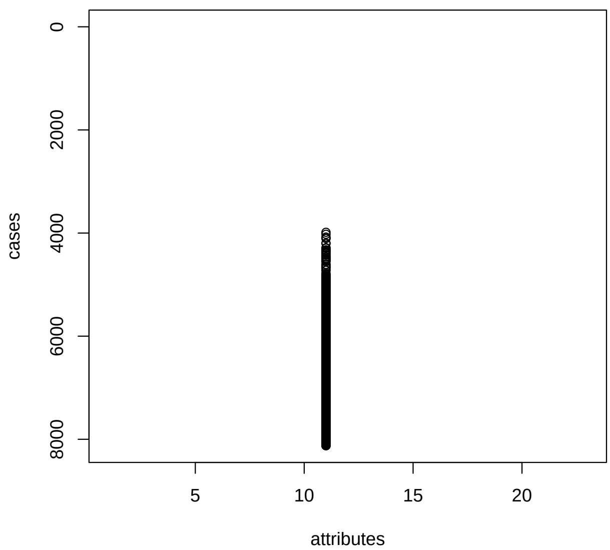
	edible	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above-ring	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
1	p	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	k	s	u
2	e	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	n	g
3	e	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	m
4	p	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	k	s	u
5	e	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	n	a	g
6	e	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	n	g
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

edible: e = edible, p = poisonous
cap-shape: b=bell, c=conical, x=convex, f=flat, k=knobbed, s=sunken
etc.

UCI Mushroom Data / Missing Values

Mushroom has missing values:

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



Learning Task

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

We split the dataset randomly in

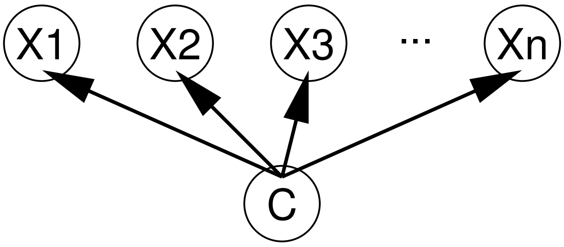
7124 training cases plus 1000 test cases

class distribution:

actual = e	529
p	471

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:

predicted =	e	p
actual = e	433	3
p	0	266

$$\text{acc} = 0.9957$$

Complete Case Analysis

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

confusion matrix:

predicted =	e	p
actual = e	516	13
p	201	270

$$\mathrm{acc} = 0.786$$

Complete Case Analysis

Diagnostics:

	edible	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above	stalk-surface-below	stalk-color-above	stalk-color-below	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
6937	p	k	y	n	f	f	f	c	n	b	t	.	s	s	p	w	p	w	o	e	w	v	d

$$p(X_9 = b \mid C) = 0$$

as $X_9 = b$ occurrs only with $X_{11} = .$!

For the whole dataset:

$X_9 =$	b	e	g	h	k	n	o	p	r	u	w	y
$M_{11} =$ false	0	0	656	720	408	984	0	1384	24	480	966	22
= true	1728	96	96	12	0	64	64	108	0	12	236	64

Available Case Analysis

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

confusion matrix:

predicted =	e	p
actual = e	523	6
p	0	471

$$\text{acc} = 0.994$$

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:

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

X	$\text{MI}(X,C)$
X1	0.04824
X2	0.02901
X3	0.03799
X4	0.19339
X5	0.90573
X6	0.01401
X7	0.10173
X8	0.23289
X9	0.41907
X10	0.00765
X11	0.09716

X	$\text{MI}(X,C)$
X12	0.28484
X13	0.27076
X14	0.24917
X15	0.24022
X16	0.00000
X17	0.02358
X18	0.03863
X19	0.31982
X20	0.48174
X21	0.20188
X22	0.15877

Pruned Network

If we use the 4 variables with highest mutual information only,

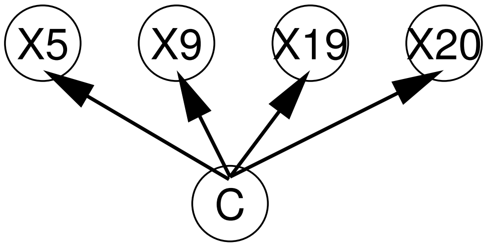
- X5 = odor
- X20 = spore-print-color
- X9 = gill-color
- X19 = ring-type

we still get very good results.

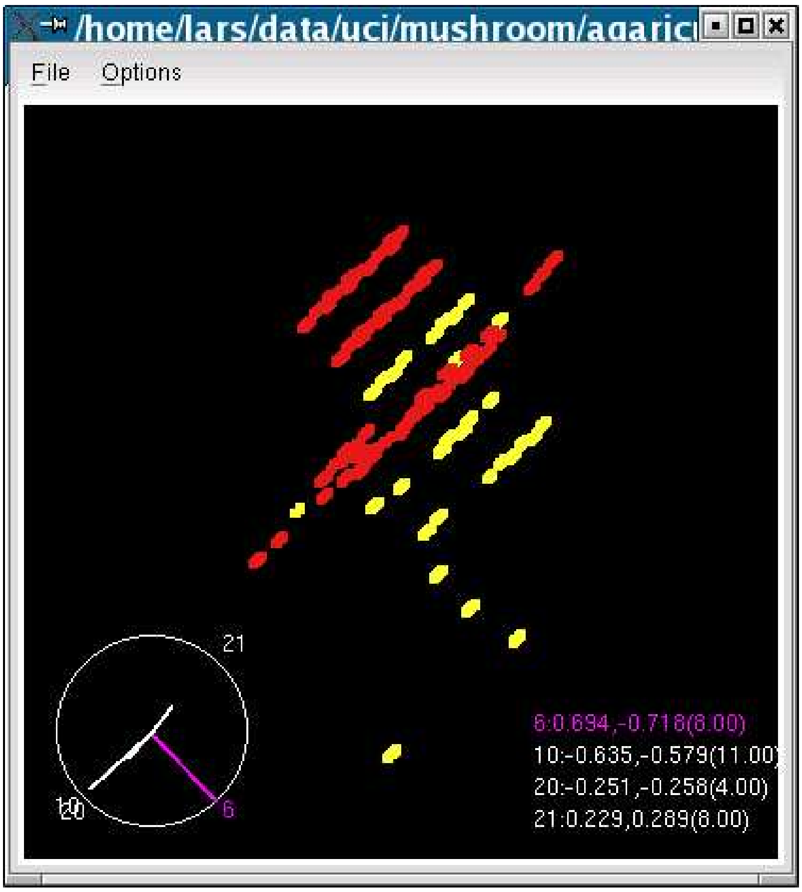
confusion matrix:

predicted =	e	p
actual = e	529	0
p	6	465

$$\text{acc} = 0.994$$



Pruned Network



Pruned Network

Fresh random split.

all variables:

predicted =	e	p
actual = e	541	4
p	1	454

$\text{acc} = .995$

X_5, X_9, X_{19}, and X_{20}:

predicted =	e	p
actual = e	544	0
p	8	447

$\text{acc} = .992$

X_1, X_2, X_3, and X_4:

predicted =	e	p
actual = e	419	126
p	101	354

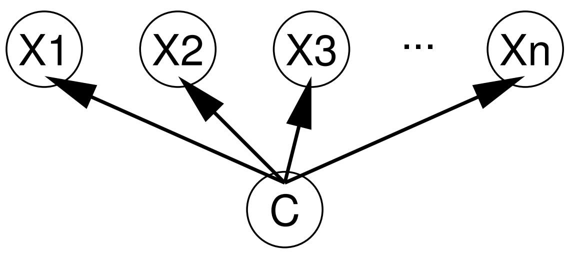
$\text{acc} = .773$

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**.

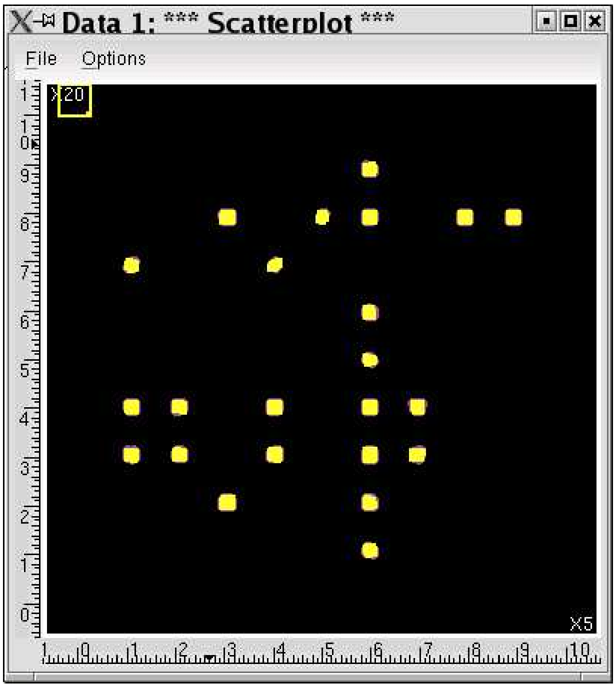
EM algorithm is used to "learn" fuzzy cluster memberships.



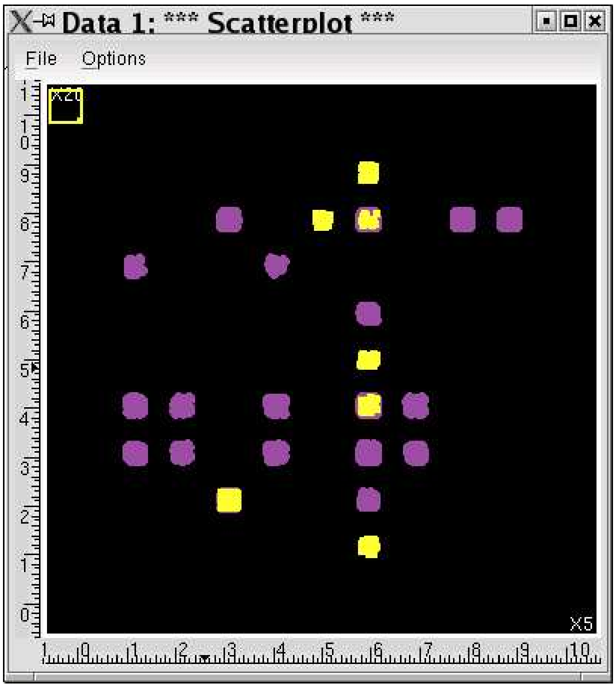
Naive Bayesian Networks used this way are a specific instance of so called **model-based clustering**.

Naive Bayesian Network / Cluster Analysis

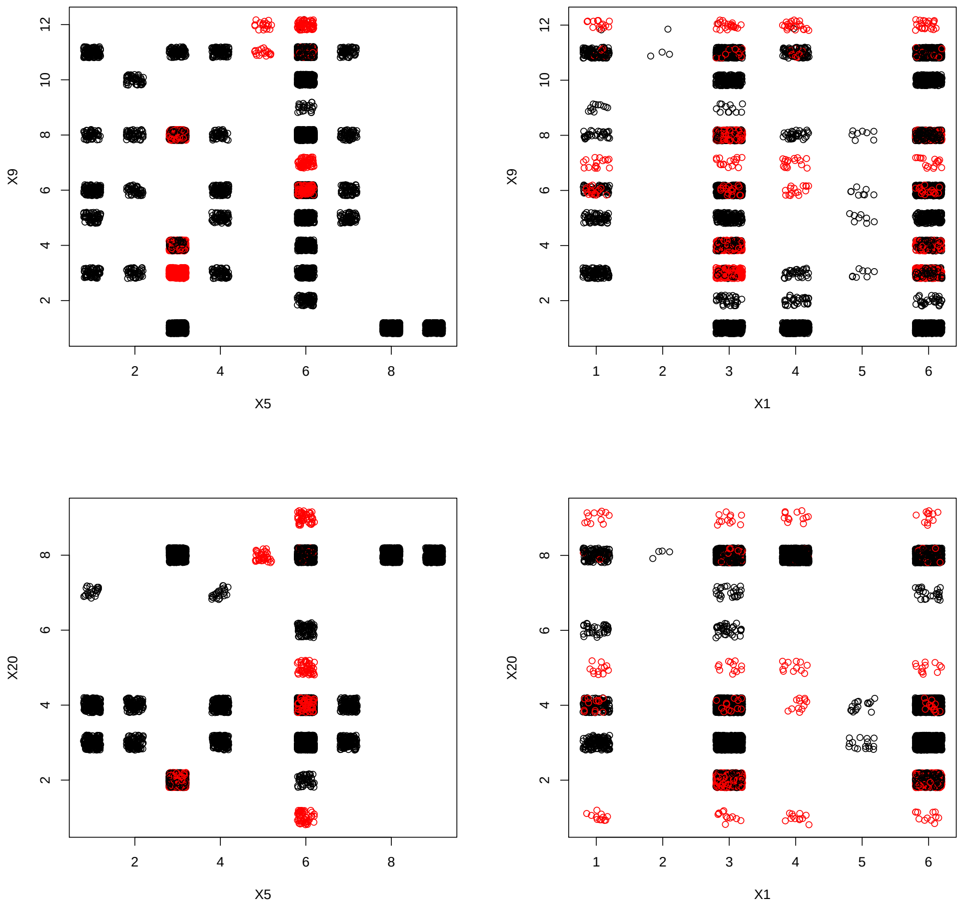
Each cluster contains "similar cases", i.e., cases that contain cooccurring patterns of values.



random



clustered



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 → completions (expectation; inference)
  - completions → parameters (maximization; parameter learning)

References

[BKS97] E. Bauer, D. Koller, and Y. Singer. Update rules for parameter estimation in Bayesian networks. In *Proceedings of the 13th Annual Conference on Uncertainty in AI (UAI)*, 1997.

[FK03] J. Fischer and K. Kersting. Scaled cgem: A fast accelerated em. In N. Lavrac, D. Gamberger, H. Blockeel, and L. Todorovski, editors, *Proceedings of the Fourteenth European Conference on Machine Learning (ECML-2003)*, pages 133–144, Cavtat, Croatia, 2003.

[Lau95] S. L. Lauritzen. The em algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis*, 19:191–201, 1995.

[LR87] R. J. A. Little and D. B. Rubin. *Statistical analysis with missing data*. Wiley, New York, 1987.