Bayesian Theory and Graphical Models

— Lecture Notes —

Laurenz Wiskott
Institut für Neuroinformatik
Ruhr-Universität Bochum, Germany, EU

14 December 2016

Contents

1 Bayesian inference 2
  1.1 Discrete random variables and basic Bayesian formalism .................. 2
  1.2 Partial evidence ........................................................................... 5
  1.3 Expectation values ..................................................................... 6
  1.4 Continuous random variables ....................................................... 7
  1.5 A joint as a product of conditionals ............................................ 7
  1.6 Marginalization .......................................................................... 8
  1.7 Application: Visual attention + ................................................... 9

2 Inference in Bayesian networks 13
  2.1 Graphical representation .............................................................. 13
  2.2 Conditional (in)dependence .......................................................... 14
  2.3 Inverting edges ........................................................................... 16
  2.4 d-Separation in Bayesian networks .............................................. 16
  2.5 Calculating marginals and message passing in Bayesian networks .... 16
  2.6 Message passing with evidence .................................................... 19
  2.7 Application: Image super-resolution .......................................... 19

© 2006, 2007, 2009, 2010, 2012, 2013, 2016 Laurenz Wiskott (ORCID http://orcid.org/0000-0001-6237-740X, homepage https://www.ini.rub.de/PEOPLE/wiskott/). This work (except for all figures from other sources, if present) is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License, see http://creativecommons.org/licenses/by-sa/4.0/. If figures are not included for copyright reasons, they are uni colored, but the word 'Figure', 'Image', or the like in the reference is often linked to a freely available copy.

Core text and formulas are set in dark red, one can repeat the lecture notes quickly by just reading these; ♦ marks important formulas or items worth remembering and learning for an exam; ◊ marks less important formulas or items that I would usually also present in a lecture; + marks sections that I would usually skip in a lecture.

More teaching material is available at https://www.ini.rub.de/PEOPLE/wiskott/Teaching/Material/.
3 Inference in Gibbsian networks

3.1 Introduction

3.2 Moral graph

3.3 Junction tree and message passing in a Gibbsian network

3.4 Most probable explanation

3.5 Triangulated graphs

4 Approximate inference

4.1 Introduction

4.2 Gibbs sampling

5 Bayesian learning for binary variables

5.1 Levels of Bayesian learning

5.2 A simple example

5.3 The Beta distribution

6 Learning in Bayesian networks

6.1 Breaking down the learning problem

6.2 Learning with complete data

6.3 Learning with incomplete data

Sections 1, 2, 3, and 4 are largely based on (Bishop, 2006; Cowell, 1999; Jensen and Lauritzen, 2001; Jordan and Weiss, 2002). Sections 5 and 6 are largely based on (Heckerman, 1999).

1 Bayesian inference

1.1 Discrete random variables and basic Bayesian formalism

Random variables (D: Zufallsgrößen, Zufallsvariablen) shall be indicated by capital letters A, B, C, ..., the values they assume by lower case letters a, b, c, ..., and the set of possible values by calligraphic letters A, E, C, .... If a variable A has assumed a concrete value a, it is called instantiated (for convenience, however, I will often use the phrase “A is known”); if certain values can be excluded (but it it might still not be clear which value the variable assumes) it is said to have received evidence. The reduced set of possible values for A after having received evidence α is A_α. For simplicity we will consider discrete random variables, which can only assume a discrete and finite set of values.

A probability distribution (D: Wahrscheinlichkeitsverteilung) for a random variable A is indicated by P_A(A) or simply P(A). The probability (D: Wahrscheinlichkeit) for A assuming a particular value then is P(A = a) or P(a). If a variable assumes a concrete value, say 5, one can write P_A(5) or P(5) but should avoid writing P(5), since it is not clear which variable and therefore which probability distribution is referred to. For convenience I will allow swapping the positions of variables, so that P(A, B) = P(B, A). Thus, the variables are identified by names, not by positions. The word ‘distribution’
is sometimes dropped for brevity, so that one speaks of a probability \( P(A) \) even though it is actually a distribution.

There are some basic definitions and rules for probabilities which the reader should be familiar with. **Any probability distribution must be normalized to one**

\[
\sum_a P(a) = 1. \quad \text{(normalization)} \tag{1}
\]

This also holds for conditional probabilities, \( \sum_a P(a|B) = 1 \), but not for marginals of joint probabilities, of course, i.e. in general \( \sum_a P(a, B) \neq 1 \), see below for the definitions.

The **joint probability distribution** (D: Verbundwahrscheinlichkeitsverteilung) for \( A \) and \( B \) is \( P(A, B) \). **If a joint probability distribution is given, the probability distribution of a single variable can be obtained by summing over the other variables, a process called marginalization** (D: Marginalisieren), resulting in the marginal distribution (D: Randverteilung)

\[
\sum_b P(A, b) \quad \text{(marginal distribution)} \tag{2}
\]

The **conditional probability distribution** (D: bedingte Wahrscheinlichkeitsverteilung) for \( A \) given \( B \) and not knowing anything about other variables is \( P(A|B) \). **Joint and conditional probabilities are related by**

\[
\begin{align*}
\sum P(A, B) &= P(A|B)P(B) \quad \text{(3)} \\
\iff P(A|B) &= \frac{P(A, B)}{P(B)}. \quad \text{(4)}
\end{align*}
\]

\((P(A|B)P(B)\) is often written from left to right, \( P(B)P(A|B) \). Somehow, I prefer writing it from right to left, maybe because I am so used to matrix notation in linear algebra. I think it is more elegant.)

If \( P(A|B) \) and \( P(B) \) are given, **the total probability** (D: totale Wahrscheinlichkeit) of \( A \) is

\[
P(A) = \sum_b P(A|b)P(b). \tag{5}
\]

**Example:** Imagine you have a red and a blue die. The red one is a normal die with six faces with the numbers 1 to 6; the blue die has six faces but only the numbers 1 to 3, each twice. In any event, you pick one die at random, the red die twice as often as the blue one, and then you role it. It is easy to see that the probabilities for color and number are overall like in Table 1.

<table>
<thead>
<tr>
<th>( P(C, N) )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>any number</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>6/9</td>
</tr>
<tr>
<td>blue</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3/9</td>
</tr>
<tr>
<td>any color</td>
<td>2/9</td>
<td>2/9</td>
<td>2/9</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>9/9</td>
</tr>
</tbody>
</table>

**Table 1:** Example of a red and a blue die. Joint and marginal probabilities. Bold face indicates the normalization condition.

The joint probabilities \( P(\text{color}, \text{number}) \), or \( P(C, N) \) for short, are shown in the center area, e.g. \( P(\text{red}, 5) = 1/9 \). The marginal probabilities \( P(C) \) and \( P(N) \) are shown to the right and bottom, respectively, and can be calculated by summing the joint probabilities over the other variable, e.g. \( P(\text{red}) = \sum_{N=1}^6 P(\text{red}, N) = 2/3 \) as we would expect since we pick the red die twice as often as the blue one. The sum over all joint probabilities can be calculated in different ways and should always be 1, i.e. \( \sum_{C,N} P(C, N) = \sum_C P(C) = \sum_N P(N) = 1 \) as one can verify in the table.
| $P(C|N)$ | 1  | 2  | 3  | 4  | 5  | 6  | $\sum_{N} P(C|N)$ |
|-------|----|----|----|----|----|----|-------------------|
| red   | 1/2| 1/2| 1/2| 2/2| 2/2| 2/2| 9/2              |
| blue  | 1/2| 1/2| 1/2| 0  | 0  | 0  | 3/2              |
| any color | 2/2| 2/2| 2/2| 2/2| 2/2| 2/2| 3/2              |
| $P(N|C)$ | 1  | 2  | 3  | 4  | 5  | 6  | $\sum_{C} P(N|C)$ |
| red   | 1/6| 1/6| 1/6| 1/6| 1/6| 1/6| 6/6              |
| blue  | 2/6| 2/6| 2/6| 0  | 0  | 0  | 6/6              |
| any number | 2/2| 2/2| 2/2| 2/2| 2/2| 2/2| 3/2              |

Table 2: Example of a red and a blue die. Conditional probabilities. Bold face indicates normalization conditions.

To get the conditional probabilities one has to renormalize the joint probabilities according to (4) resulting in Table 2.

The conditional probabilities tell us, for instance, that the probability for rolling 1 is greater if we use the blue die than if we use the red die. Or is we have rolled a 1, it comes equally likely from the red and the blue die, but if we role a 4, it must come from the red die.

Notice that the conditional probabilities summed over the variable conditioned on does not yield 1. □

**Bayes’ rule** (D: Bayes-Formel) tells us how to invert conditional probabilities,

$$P(A, B) = P(A|B)P(B)$$

(6)

$$= P(B|A)P(A)$$

(7)

$$\implies P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$ (Bayes’ rule).

(8)

In this equation $P(B)$ is called the *a priory* probability, or simple the *prior*, $P(A|B)$ is the *likelihood* of $B$ for a fixed $A$, and $P(B|A)$ is the *a posteriori* probability of $B$ given $A$.

Two variables are **statistically independent** iff (if and only if) $P(A|B) = P(A)$ or, equivalently, $P(B|A) = P(B)$, which also implies that their joint probability distribution factorizes, i.e.

$$\implies P(A|B) = P(A)$$ (statistical independence)

(9)

$$\implies P(B|A) = P(B)$$ (statistical independence)

(10)

$$\iff P(A|B) = P(A)P(B)$$

(11)

$$\iff P(B|A) = P(A)P(B)$$

(12)

$$\iff P(A, B) = P(A)P(B)$$ (statistical independence).

(13)

All the rules above generalize to more random variables, i.e. one could, for instance, write $P(A, B, C, D) = P(A, B|C, D)P(C, D)$ or $P(A, B|C, D) = P(B, C|A, D)P(A, D)/P(C, D)$. One can also condition everything on another random variable, e.g. Bayes rule could read $P(B|A, C) = \frac{P(A, B|C)P(B|C)}{P(A|C)}$.

It is important to realize that Bayesian formalism is not about causality but about beliefs what knowing the value of some variables tells us about the unknown variables. Imagine you like to take a long walk ($B = 'long walk'$) on Saturday mornings if and only if you woke up early in the morning ($A = 'woke up early'$). That is your general habit, but there is still some chance that you take a long walk if you got up late ($A = 'woke up late'$) or that you do not take a walk ($B = 'no walk'$) even though you got up early. Now, your walk causally depends on the time you wake up and not vice versa. Taking a walk does not make you wake up early. However, if you tell your brother at the phone that you have taken a walk
in the morning, he can guess that probably you got up early in the morning. Thus, Bayesian inference is about knowing or believing things, not about causality, although causality might play an important role in defining the system and knowing the (conditional) probability distributions in the first place.

Further readings: (Cowell, 1999, secs. 2, 3).

### 1.2 Partial evidence

So far, the value of a random variable was either known, i.e. the variable was instantiated, or completely unknown. However, it may also happen that one has partial evidence (D: Teilevidenz?) \(E\) about a variable. Imagine I role a die and you have to guess the result. Initially, you would assume that the numbers from 1 to 6 all have the probability 1/6. But then I give you partial evidence by telling you that I have gotten an even number. Then of course you would correct your expectation to a probability of 1/3 for 2, 4, and 6 and to zero for the odd numbers. Thus, partial evidence changes probabilities.

It is quite obvious, how joint probabilities change with partial evidence. You simply discard those cases that are not possible anymore and renormalize the probabilities to one. For the example above, one could make the following table, with \(A\) indicating the number I might have roled and \(E_A = \{2, 4, 6\}\) indicating the set of values that are still possible, thereby indicating the evidence about \(A\) that I have roled an even number.

<table>
<thead>
<tr>
<th>(A)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>possible values of (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P(A))</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>original probabilities of (A)</td>
</tr>
<tr>
<td>(P(A</td>
<td>E_A))</td>
<td>-</td>
<td>1/6</td>
<td>-</td>
<td>1/6</td>
<td>-</td>
<td>1/6</td>
</tr>
<tr>
<td>(P(A</td>
<td>E_A))</td>
<td>-</td>
<td>1/3</td>
<td>-</td>
<td>1/3</td>
<td>-</td>
<td>1/3</td>
</tr>
</tbody>
</table>

I have invented the notation \(P(A|E_A)\) here to indicate the original probabilities masked by the evidence. Formally one could write

\[
P(A|E_A) = \begin{cases} \frac{P(A)}{\sum_{a' \in E_A} P(a')} & \text{for } A = a \in E_A, \\ 0 & \text{otherwise} \end{cases},
\]

or, for short,

\[
\diamond \quad P(A|E_A) = \frac{P(A)}{\sum_{a' \in E_A} P(a')},
\]

with the understanding that the probability is zero for discarded values \(a \notin E_A\).

The same applies to joint probabilities, e.g.

\[
\diamond \quad P(A, B|E_{AB}) = \frac{P(A, B)}{\sum_{(a', b') \in E_{AB}} P(a', b')},
\]

where \(E_{AB}\) is a set of duplets \((a, b)\).

To simplify matters a bit, we assume that evidence is independently given for the different variables at hand, so that we can write

\[
\blacklozenge \quad P(A, B|E_A, E_B) = \frac{P(A, B)}{\sum_{a' \in E_A, b' \in E_B} P(a', b')},
\]
A marginal probability is then

\[ P(A|\mathcal{E}_A, \mathcal{E}_B) = \sum_{b \in \mathcal{E}_B} P(A, b|\mathcal{E}_A, \mathcal{E}_B) \quad (18) \]

\[ = \frac{\sum_{b \in \mathcal{E}_B} P(A, b)}{\sum_{a', b' \in \mathcal{E}_A, \mathcal{E}_B} P(a', b')} \quad (19) \]

A conditional probability is

\[ P(B|A, \mathcal{E}_A, \mathcal{E}_B) = P(A, B|\mathcal{E}_A, \mathcal{E}_B) \frac{P(A|\mathcal{E}_A, \mathcal{E}_B)}{P(A, B)} \quad (20) \]

\[ = \frac{\sum_{a' \in \mathcal{E}_A, b' \in \mathcal{E}_B} P(a', b') P(A|\mathcal{E}_A, \mathcal{E}_B)}{\sum_{b \in \mathcal{E}_B} P(A, b)} \quad (21) \]

\[ = \frac{P(A, B)}{\sum_{b \in \mathcal{E}_B} P(A, b)} \quad (22) \]

### 1.3 Expectation values

One is often not interested in the probabilities of individual outcomes but in the average outcome. This can be determined by averaging over all outcomes weighted with the respective probabilities, which is the **expectation value** (D: Erwartungswert) or **mean (value)** (D: Mittelwert)

\[ E\{A\} := \bar{A} := \langle a \rangle_a := \sum_a a P(a). \quad (23) \]

You see, there are several different ways to write the expectation value. I will usually use the angle brackets notation. The average over a random variable, of course, only exists, if it is a numerical value or any other value for which a weighted sum is defined.

Equation (23) generalizes to the **expectation value of functions** of random variables,

\[ \langle f(a) \rangle_a := \sum_a f(a) P(a). \quad (24) \]

which is particularly useful if one cannot take the average over a directly, because it is a qualitative variable rather than a quantitative variable. The range of \( f(a) \) may be real, complex, vectorial, or whatever can be averaged over. (24) includes (23) as a special case, if the function is the identity.

When taking the **expectation value of a function in two** or more random variables one can play around with the order of summation and with conditional probabilities.

\[ \langle f(a, b) \rangle_{a, b} = \sum_{a, b} f(a, b) P(a, b) \quad (25) \]

\[ = \langle \langle f(a, b) \rangle_a \rangle_b = \sum_b \left( \sum_a f(a, b) P(a|b) \right) P(b) \quad (26) \]

\[ = \langle \langle f(a, b) \rangle_b \rangle_a = \sum_a \left( \sum_b f(a, b) P(b|a) \right) P(a) \quad (27) \]
Taking the expectation value is a linear operation, meaning that

\[ \langle \gamma f(a) \rangle_a = \gamma \langle f(a) \rangle_a , \quad (28) \]
\[ \langle f(a) + g(a) \rangle_a = \langle f(a) \rangle_a + \langle g(a) \rangle_a , \quad (29) \]

for scalars \( \gamma \), which follows directly from the linearity of the sum in the definition (24) of the expectation value.

Besides the mean \( \bar{a} = \langle a \rangle_a \), another particularly important expectation value is the squared difference between \( a \) and its mean, which is the variance \( \langle (a - \bar{a})^2 \rangle_a \). In this context I often use the bar notation besides the angle brackets notation.

### 1.4 Continuous random variables

The formalism above can be readily generalized to continuous random variables. Basically, one simply has to replace sums by integrals and think in terms of probability densities instead of probabilities. The notation also changes a bit. One uses a lower case \( p \) instead of upper case \( P \) for the densities, and the random variable is usually lower case as well, e.g. \( x \) instead of \( A \).

The normalization rule (1) becomes

\[ \int_x p(x) \, dx = 1 . \quad (30) \]

The marginal distribution (2) becomes

\[ p(x) = \int_y p(x,y) \, dy . \quad (31) \]

Bayes rule (8) is still the same

\[ p(y|x) = \frac{p(x|y)p(y)}{p(x)} . \quad (32) \]

The expectation value (24) becomes

\[ \langle f(x) \rangle_x := \int_x f(x)p(x) \, dx . \quad (33) \]

Thinking in terms of probability densities means that one should not ask “What is the probability that \( x \) assumes a particular value?” because that probability is always zero (except if the probability density function is a \( \delta \)-functions). One can only ask “What is the probability that \( x \) lies within a certain interval?”, which would be

\[ P(x_1 \leq x \leq x_2) = \int_{x_1}^{x_2} p(x) \, dx . \quad (34) \]

Asking for a particular value would mean that \( x_1 = x_2 \) and the integral would vanish.

### 1.5 A joint as a product of conditionals

Assume a probability distribution for five random variables is given by \( P(A, B, C, D, E) \). Every statistical expression involving these five variables can be inferred from this joint probability distribution by the rules given above. However, computationally it is rather expensive, because if each variable can assume ten values then 100,000 probabilities have to be stored to define \( P(A, B, C, D, E) \). Furthermore, computing the marginal distribution \( P(A) \), for instance, would require summation over all other variables, requiring 9,999 additions. Also from an analytical point of view is the expression \( P(A, B, C, D, E) \)
not very helpful, because it does not reflect any of the structure you might know about the variables. **Thus,** it would be nice to get a more efficient and informative representation that takes advantage of the structure of the problem at hand.

In many cases the **structure can be inferred from the causal relationships** between the variables. For instance, if you consider a sequence of events and you know that event $A$ has a direct causal influence on $B$, $B$ and $D$ together influence $E$, $B$ and $E$ together influence $C$, and that these are all causal relationships, there are, then you can simply define

$$
\begin{align*}
P(A, B, C, D, E) &= P(C|B, E)P(E|D, B)P(D)P(B|A)P(A). \\
&\quad \left(\begin{array}{c}
100,000 \otimes \\
1,000 \otimes \\
1,000 \otimes \\
10 \otimes 100 \otimes 10 \otimes \\
2,120 \otimes
\end{array}\right)
\end{align*}
$$

and go from there.

**The figures under the braces indicate the numbers of original probabilities to be stored with $\otimes$ indicating the cost of storage.** They show that writing the joint probability distribution as a product of conditional probability distributions is much more efficient in terms of memory consumption.

But keep in mind that **Bayesian formalism is not about causality**, e.g. you cannot say that $P(A|B) = P(A)$ because $B$ does not have any causal influence on $A$, because $B$ still tells you something about $A$. You must also be careful with feedback loops. If $A$ influences $B$, $B$ influences $C$, and $C$ influences $A$, you might be tempted to define $P(A, B, C) := P(A|C)P(C|B)P(B|A)$. However, that can cause all kinds of trouble. For instance, since probabilities have to be normalized to one, we have $\sum_a P(a|C) = \sum_c P(c|B) = \sum_b P(b|A) = 1$, but from that does generally not follow $\sum_{a,b,c} P(a, b, c) = 1$ as you would expect. So, **do not use feedback loops in your product-of-conditionals expression**.

Further readings: (Cowell, 1999, sec. 6).

### 1.6 Marginalization

The next thing we might be interested in are the marginal distributions of subsets of variables. Knowing the marginals, for instance, would allow us to compute conditional probabilities with (4), e.g. $P(C|B) = P(B, C)/P(B)$.

Formally, to **compute the marginal of one variable one simply has to sum the joint probability over all other variables**. The computational costs for doing that, however, can vary dramatically depending on the way one does that. Consider the example of the previous section and three different ways of computing the marginal distribution $P(E)$ from $P(A, B, C, D, E)$.

$$
\begin{align*}
P(E) &= \sum_{a,b,c,d} P(a, b, c, d, E) \\
&\quad \left(\begin{array}{c}
9,999 \oplus
\end{array}\right)
\end{align*}
$$

$$
\begin{align*}
\phi &= \sum_{a,b,c,d} P(c|b, E)P(E|d, b)P(d)P(b|a)P(a) \\
&\quad \left(\begin{array}{c}
40,000 \otimes + 9,999 \oplus
\end{array}\right)
\end{align*}
$$

$$
\begin{align*}
\phi &= \sum_b \left(\sum_c P(c|b, E)\right) \left(\sum_d P(E|d, b)P(d)\right) \left(\sum_a P(b|a)P(a)\right) \\
&\quad \left(\begin{array}{c}
=1 \\
10 \otimes + 9 \oplus \\
10 \otimes + 9 \oplus \\
210 \otimes + 189 \oplus
\end{array}\right)
\end{align*}
$$
where $\otimes$ and $\oplus$ indicate the expenses of a multiplication and a summation operation, respectively. We see here that not only the memory consumption but also the number of operations needed reduces dramatically if we use the product of conditionals and compute the sums efficiently, i.e. if we apply the sums only to those factors that depend on the variable summed over. Computing the sum efficiently in this way is referred to as variable elimination. Summing over $a$ in the example above, for instance, eliminates $A$ and leaves the term $P(b) = \sum_a P(b|a)P(a)$, which only depends on $b$. (Variable elimination is usually explained as rearranging the factors and shifting the sums as much to the right side as possible. I find it more intuitive to say that one rearranges the factors (which was not necessary in the example above) and focuses the summations by (possibly nested) parentheses.)

If one wants to compute the marginal distribution of all different random variables, one should store and reuse some of the intermediate sums for greater efficiency. We will see below that bookkeeping of this reuse of sums can be conveniently done in a graph structure reflecting the dependencies between the variables.

Further readings: (Jensen and Lauritzen, 2001, sec. 3.1; Jordan and Weiss, 2002, sec. 3.1).

1.7 Application: Visual attention

Imagine you are a subject in a psychophysical experiment and have to fixate the center of a blank computer screen. In each trial a little stimulus in the shape of a light Gaussian spot is presented to you with equal probability at a particular location on the left or on the right of your fixation point or, in half of the cases, there is no stimulus presented at all. Your task is to respond if a stimulus is being presented. As a hint you get a precue, which tells you with a reliability of 80% whether the stimulus is going to be on the left or on the right, if it comes up at all. See Figure 1 for an illustration of the trial conditions. The task is difficult because the stimulus is barely visible and its perception corrupted by noise, so you have to concentrate and probably you will tend to attend to the cued location.

Figure 1: Protocol for the different types of trials in the attention experiment.

Figure: (Shimozaki et al., 2003, Fig. 6, URL)¹

Attention in this kind of experiment is usually conceptualized as a way to optimally use limited resources, the idea being that your visual system is not able to dedicate full computational power to both locations simultaneously. So it has to make a decision where to concentrate the computational resources in order to optimize the performance. Shimozaki et al. (2003) have used Bayesian theory to model visual attention in a rather different way. They do not assume limited computational resources but
assume that you trust your results differently because of prior information, the precue in this case.

The experiment sketched above can be formalized as follows. For symmetry reasons we do not distinguish left and right but only cued and uncued location, which spares us one variable. Let $V$ be the discrete random variable for the visual input with subscripts $c$ and $u$ indicating the cued and uncued location, respectively. $V$ can assume the values $v \in \{s, n\}$ where $s$ indicates 'stimulus present' and $n$ indicates 'no stimulus present'. We can distinguish four cases and assign the following probabilities to them.

\[ P(V_c = s, V_u = s) := 0, \]
\[ P(V_c = s, V_u = n) := 0.4, \]
\[ P(V_c = n, V_u = s) := 0.1, \]
\[ P(V_c = n, V_u = n) := 0.5. \]

The first case does not occur, i.e. has probability zero, but has been added for the sake completeness to make later summations easier.

The perceptual accuracy at the cued and uncued location are assumed to be equal and independent of each other. Thus, there are no perceptual resources that could be distributed more to the cued location. However, perception is assumed to be corrupted by Gaussian noise. If $A$ indicates the continuous random variable for the internal response or activity to the visual input, we have

\[ p(A = a|V = v) := \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{(a - \mu_v)^2}{2} \right), \]
\[ \mu_n := 0 \]
\[ \mu_s \geq 0. \]

Add a $c$ or $u$ as a subscript if you want to use this for the cued or uncued location, respectively. The variance of the Gaussians is assumed to be one for simplicity. If no stimulus is present, the Gaussian is centered at zero; if a stimulus is present, the Gaussian is centered at $\mu_s \geq 0$ and $a$ is greater on average than if the stimulus is not present. However, if $\mu_s$ is small, the distributions overlap so much that the value of $A$ does not reliably indicate whether the stimulus is present or not.

By simple Bayesian calculus we can derive

\[ p(A_c, A_u|V_c, V_u) \]
\[ \overset{(46)}{=} \frac{p(A_c|V_c)p(A_u|V_u)}{p(A_c, A_u)}, \]
\[ P(V_c, V_u|A_c, A_u) \]
\[ \overset{(47)}{=} \frac{p(A_c, A_u|V_c, V_u)P(V_c, V_u)}{p(A_c, A_u)}, \]
\[ \overset{(48)}{=} \sum_{v_c, v_u \in \{s, n\}} p(A_c, A_u|v_c, v_u)P(v_c, v_u). \]

The conditional probability for a stimulus being present or absent given the responses then is

\[ P(s|A_c, A_u) = P(V_c = s, V_u = n|A_c, A_u) + P(V_c = n, V_u = s|A_c, A_u), \]
\[ P(n|A_c, A_u) = P(V_c = n, V_u = n|A_c, A_u), \]

respectively, where $s$ stands for $(V_c = s, V_u = n) \lor (V_c = n, V_u = s)$ and $n$ stands for $(V_c = n, V_u = n)$. Optimal performance is obviously achieved if the subject responds with 'stimulus present' when

\[ P(s|A_c, A_u) > P(n|A_c, A_u) \]
\[ \overset{(51)}{=} 0.5 \]
\[ \overset{(52)}{=} 0.5 \]
\[ \overset{(53)}{=} 0.5. \]
It should be emphasized that this Bayesian formalism gives the optimal performance under the assumptions given above. **No better performance can be achieved by any other model.** All information that is available at the cued as well as at the uncued location is being used. There is no particular distribution of computational capacity in favor of the cued location.

Now, if $R$ indicates the response of the subject, which like $V$ can assume the values $r \in \{s, n\}$, and if we assume that the subject responds deterministically according to (51) we have

$$P(R = s|A_c, A_u) = \Theta[P(s|A_c, A_u) - P(n|A_c, A_u)],$$

(54)

where $\Theta(\cdot)$ is the Heaviside step function (with $\Theta(x) := 0$ if $x \leq 0$ and $\Theta(x) := 1$ otherwise).

In (Shimozaki et al., 2003) the performance is quantified with the correct hit and false alarm rate, the former being separately computed for valid cues and invalid cues.

- **Correct hit rate given valid cue:** $P(R = s|V_c = s, V_u = n),$  
- **Correct hit rate given invalid cue:** $P(R = s|V_c = n, V_u = s),$  
- **False alarm rate:** $P(R = s|V_c = n, V_u = n).$

These expressions are not easily computed analytically, because they involve difficult to solve integrals, namely

$$P(R = s|V_c, V_u) = \int \int P(R = s|a_c, a_u) p(a_c, a_u|V_c, V_u) da_c da_u.$$  

(58)

which can be further resolved with equations (54; 49; 47; 40; 41; 42; 46; 43), in that order, resulting in a quite complicated integral. **The authors have therefore determined the correct hit and false alarm rates numerically** with Monte Carlo methods, i.e. by random sampling. Figure 2-left shows these rates as a function of the signal-to-noise ratio (SNR), which is $\mu_s$ in this case. They also performed psychophysical experiments for comparison, see Figure 2-right. For these experiments the SNR has been estimated from the image contrast. **Figure 3 illustrates the effect the cueing has on the correct hit rate.**

**Figure 2:** Correct hit rates given a valid cue (pHv) or an invalid cue (pHi) and false alarm rates (pFA) for different signal-to-noise ratios (SNR). Simulation results for the Bayesian model are shown on the left, experimental results for the first of three subject are shown on the right.

Figure: (Shimozaki et al., 2003, Figs. 4, 7, URL)²

Apart from a certain misfit between the scaling of the abscissas, **the fit between the Bayesian model and human performance is rather good.** The scaling of the abscissas might be due to the fact that human sensitivity to the stimuli is worse than the theoretical optimum by about a factor of two, which is reasonable.
Figure 3: Cueing effect defined as the difference between pHv and pHi for three models (left) and three subjects (right). The sum-of-likelihoods model is the Bayesian model discussed here (bottom, most wiggly line). The authors also considered a more traditional attentional-switching model (top line) and a linear-weighting model (middle line), both of which did not explain the experimental data well.

Figure: (Shimozaki et al., 2003, Figs. 5, 8, URL)³

So we know now that the Bayesian model performs optimally and that human subjects perform similarly to the Bayesian model except for the scaling of the SNR-axis. The pHv- and pFA-curves in Figure 2-left are intuitively clear. But why is the pHi-curve always below the pHv-curve even though all information is used at both locations? And why does the pHi-curve first drop and only then rise to approach one as the SNR gets large?

To get a formal understanding for why the pHi-curve is below the pHv-curve consider Equation (51) for a concrete pair of values $a_c$ and $a_u$.

\[
P(s|a_c, a_u) - P(n|a_c, a_u) < 0 \quad (51)
\]

\[
P(V_c = s, V_u = n|a_c, a_u) + P(V_c = n, V_u = s|a_c, a_u) - P(V_c = n, V_u = n|a_c, a_u)\]  
\[
\approx 0 < 0.4 p(a_c, a_u|V_c = s, V_u = n) + 0.1 p(a_c, a_u|V_c = n, V_u = s) - 0.5 p(a_c, a_u|V_c = n, V_u = n) \quad (59)
\]

\[
0 < 0.4 \exp \left( -\frac{(a_c - \mu_s)^2 + a_u^2}{2} \right) + 0.1 \exp \left( -\frac{a_u^2 + (a_u - \mu_s)^2}{2} \right) - 0.5 \exp \left( -\frac{a_u^2}{2} \right) \quad (60)
\]

\[
\approx 0 < \exp \left( -\frac{(a_c - \mu_s)^2 + a_u^2}{2} \right) \quad (61)
\]

\[
0 < \exp \left( -\frac{(a_c - \mu_s)^2 + a_u^2}{2} \right) \quad (62)
\]

If the stimulus is at the cued location, $a_c$ is typically greater than $a_u$ and therefore also $\exp \left( -\frac{(a_c - \mu_s)^2 + a_u^2}{2} \right)$ is greater than $\exp \left( -\frac{(a_u - \mu_s)^2 + a_u^2}{2} \right)$. If we take the same values $a_c$ and $a_u$ but exchange them, which would be the case if the cued and uncued location swapped roles, then (62) becomes smaller, because of the different weighting factors 0.4 and 0.1, and the system is less likely to report ‘stimulus present’. Intuitively one might phrase it as follows: If $a_c$ is relatively large and greater than $a_u$ then this is an indication for the stimulus being present at the cued location and it is consistent with the general expectation that the stimulus appears at the cued location; if $a_u$ is relatively large and greater than $a_c$ then this is an indication for the stimulus being present at the uncued location, but it is in conflict with our general expectation that the stimulus appears at the cued location. Thus, we trust the evidence less and tend to believe that it is just noise taking large values that we see.

The second question why the pHi-curve first drops is more difficult to answer and I did not come up with a nice intuitive explanation. Suggestions are welcome.

I like this model for two reasons. Firstly, it models attention in a very principled way based on Bayesian theory and the notion of optimality. No ad-hoc assumptions are necessary. Secondly, the model shows that attentional effects can arise without assuming limited resources. This opens a new
way of thinking about attention. I am sure that such a Bayesian approach does not work for all aspects of attention. Limited resources probably play an important role in some attentional tasks. But this model shows that not everything has to be due to limited resources.

2 Inference in Bayesian networks

2.1 Graphical representation

A Bayesian network is a graphical representation of a probability distribution that can be written as a product of conditional probabilities. It is a graph with nodes representing random variables (in the following I will use the terms variable and node largely interchangeably) and directed edges representing statistical dependencies. For each conditional probability in a product of conditionals you have to draw arrows pointing from the variables on the right side to the one variable on the left side of the ‘|’. Since there are no feedback loops allowed, the result is a directed acyclic graph (DAG). The example (35) considered above translates into the graph shown in Figure 4.

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

for instance, translates into the graph shown in Figure 4.

![Figure 4: A Bayesian network.](https://example.com/figure4.png)

Because of the acyclic nature of Bayesian networks it makes sense to use terms of family trees to relate nodes. If node A points at node B, A is called a parent of B, which in turn is called a child of A. All the nodes that one can reach following the edges in their forward direction are the descendants of a node and those that can be reached in the other direction are the ancestors.

Note that the Bayesian network structure is not unique for a given joint probability, since a joint probability can be factorized into a product of conditionals in different ways and edges can be inverted (Sec. 2.3).

### 2.2 Conditional (in)dependence

A random variable $A$ is said to be **conditionally independent** (D: bedingt unabhängig) of $C$ given $B$ (sometimes written as $A \perp \!\!\!\!\!\!\perp C \mid B$) if

\[ P(A \mid B, C) = P(A \mid B) \quad \text{(conditional independence)} . \tag{64} \]

This is trivial if $A$ does not depend on $C$ in any case, but it is an interesting statement if $A$ and $C$ are statistically dependent if $B$ is unknown, i.e. if $P(A \mid C) \neq P(A)$.

One can develop a good intuition for conditional independences if one considers Bayesian networks with a causal interpretation. Consider first the three simple examples in Figure 5, in which the nodes $A$ and $C$ are related via $B$ by a **serial**, a **diverging**, and a **converging** connection.

![Figure 5: Simple graphs representing statistical relationships between random variables.](cc-by-sa-4.0)

**In figure 5.a** $C$ is causally determined by $B$ alone. If $B$ is not known, $A$ can tell us something about $B$ which in turn tells us something about $C$. If, however, $B$ is known, $A$ does not tell us anything about $C$ that is not already contained in $B$. Thus, **if $B$ is known, $C$ is independent of $A$**, $P(A \mid C, B) = P(C \mid B)$, which also implies $P(A, C \mid B) = P(A \mid B)P(C \mid B)$ and $P(A \mid B, C) = P(A \mid B)$.

**In figure 5.b one arrives at a similar conclusion.** If $B$ is not known, $C$ tells us something about $B$, which in turn tells us something about $A$. If $B$ is known, $C$ cannot tell us anything about $B$ that we do not know already. And since $A$ does not depend on $C$ directly but only on $B$, $A$ does not tell us anything about $C$. Thus, if $B$ is known, $A$ is independent of $C$.

The situation in **figure 5.c is different**. Neither $A$ nor $C$ does causally depend on any other variable. **If $B$ is not(!) known, $A$ and $C$ are statistically independent** (sometimes written as $A \perp \!\!\!\!\!\!\perp C \mid \emptyset$). However, if $B$ is known, $A$ can tell us something about $C$, because only certain combinations of $A$ and $B$ might yield certain values of $C$, thus $A$ and $C$ become conditionally dependent if $B$ is known (sometimes written as $A \perp \!\!\!\!\!\!\perp C \mid B$).

**To understand this example intuitively, consider eye color.** $A$ and $B$ shall be the eye color of the parents and $C$ that of the child (now I am speaking of true humans). If you don’t know the eye color of the child, knowing the eye color of one parent does not tell you anything about the eye color of the other parent.
However, if you know that the child has brown eyes and one parent has blue eyes, you can infer that the other parent has brown eyes, because brown is dominant and blue is recessive.

Of course, one can derive the conditional (in)dependencies also formally from the product expression of the joint probability. For the example of Figure 5.a one gets

\[ P(A|B,C) \]
\[ = \frac{P(A,B,C)}{P(B,C)} \]
\[ = \frac{P(C|B)P(B|A)P(A)}{\sum_a P(C|B)P(B|a)P(a)} \]
\[ = \frac{P(B|A)P(A)}{\sum_a P(a|B)P(B)} \]
\[ = \frac{P(B|A)P(A)}{P(B)} \]
\[ = \frac{P(A,B)}{P(B)} \]
\[ = P(A|B) \]

Similarly, one can derive

\[ P(C|A,B) = P(C|B). \]

Notice, however, that if B is not known, A and C can depend on each other. No such simplification is possible for \( P(B|A,C) \). Thus, A and C are conditionally independent of each other given B, but B always depends on A as well as C. The Example of Figure 5.b is equivalent to 5.a, which follows directly from

\[ P(A|B)P(B) = P(B|A)P(A) \] (7).

For the example of Figure 5.c we get

\[ P(A,C) \]
\[ = \sum_b P(b|C,A)P(C)P(A) \]
\[ = \frac{P(C)P(A)}{P(A)} \]

but no simplification for \( P(A|B,C) \) or \( P(C|A,B) \). Thus, in contrast to the previous two examples, A and C are independent if B is not known and become dependent if B is known. This is a somewhat surprising result.

Further readings: (Cowell, 1999, secs. 5, 6.2; Jensen and Lauritzen, 2001, secs. 1.2, 2.1).

Note that

\[ P(A|B,C) = P(A|B) \]
\[ \Rightarrow \quad P(A|B,C,D) = P(A|B,D). \]
As a counter example consider the joint probability \( P(A, B, C, D) = P(D|C, A)P(C|B)P(B|A)P(A) \).

### 2.3 Inverting edges

We have mentioned in passing that the networks \( (a) \) and \( (b) \) of Figure 5 are equivalent, because \( P(A|B)P(B) = P(B|A)P(A) \). Replacing \( P(B|A)P(A) \) by \( P(A|B)P(B) \) to turn network \( (a) \) into the equivalent network \( (b) \) is actually an interesting operation, because it implies an inversion of an edge. This relates back to the statement that the Bayesian formalism is about statistical dependencies but not about causality. A causal relationship cannot formally be inverted, but a statistical dependency can. However, this works only as long as nodes have (and get) no more than one parent.

### 2.4 d-Separation in Bayesian networks

In the first two examples of Figure 5 knowing \( B \) separated the node \( A \) and \( C \) from each other, and in the third example not knowing \( B \) separated the node \( A \) and \( C \) from each other. Such cases are referred to as d-separation. One can generalize the findings above and say that two nodes \( A \) and \( C \) in a Bayesian network are d-separated iff for all paths between \( A \) and \( C \) there is an intermediate node \( B \) such that either the connection through \( B \) is serial or diverging and \( B \) is instantiated or the connection is converging and neither \( B \) nor any of its descendants has received evidence (Jensen and Lauritzen, 2001).

Note that for a serial or diverging connection \( B \) has to be instantiated to completely separate \( A \) and \( C \), and for a converging connection \( B \) has to be completely unknown. If there is any evidence for \( B \), may it be direct or through its descendants, then \( A \) and \( C \) become connected through \( B \) if it is in a converging connection. Further readings: (Jensen and Lauritzen, 2001, secs. 1.2, 2.4).

### 2.5 Calculating marginals and message passing in Bayesian networks

Consider the Bayesian network in figure 6 with the probability

\[
\]  

\(1\) We find that

\[
P(A, B, C, D) \quad (\text{76})
\]

\[
P(A|B, C) \quad (\text{4})
\]

\[
P(A|B, C, D) \quad (\text{4})
\]

\[
P(A|B, D) \quad (\text{2,76})
\]

\[
= P(A|B, D) \quad (\text{4})
\]

\[
P(A, B, C, D) = P(D|C, A)P(C|B)P(B|A)P(A) \quad (\text{82})
\]

\[
P(A, B, C, D) \quad (\text{76})
\]

\[
P(A|B, C) \quad (\text{4}) = \frac{P(A, B, C)}{P(B, C)} = \frac{P(A, B, C)}{\sum_a P(d|C, A)P(C|B)P(B|A)P(A)} \quad (\text{77})
\]

\[
P(A|B, C, D) \quad (\text{4}) = \frac{P(A, B, C, D)}{P(B, C, D)} = \frac{P(A, B, C, D)}{\sum_a P(D|C, A)P(C|B)P(B|A)P(A)} \quad (\text{79})
\]

\[
\neq \frac{\sum_a P(D|c, A)P(c|B)P(B|A)P(A)}{\sum_{a,c} P(D|c, a)P(c|B)P(B|a)P(a)} \quad (\text{in general}) \quad (\text{80})
\]

\[
P(A|B, D) \quad (\text{2,76}) = \frac{P(A, B, D)}{P(B, D)} \quad (\text{4}) \quad (\text{81})
\]
\[ P(A,B,C,D) = P(D|B) \ P(C|B) \ P(B|A) \ P(A) \]

**Figure 6:** Message passing in a Bayesian network. The messages needed to compute the marginal \( P(C) \) are set in black, the others in gray.
© CC BY-SA 4.0

It is often of interest to know what the probability distribution of one particular variable is, which is known
as the marginal and can be calculated by summing over all other variables.

\[ P(A) \equiv \sum_{b,c,d} P(d|b)P(c|b)P(b|A)P(A) \]  
\[ P(B) \equiv \sum_{a,c,d} P(d|B)P(c|B)P(B|a)P(a) \]  
\[ P(C) \equiv \sum_{a,b,d} P(d|b)P(c|b)P(b|a)P(a) \]  
\[ P(D) \equiv \sum_{a,b,c} P(D|b)P(c|b)P(b|a)P(a) \]

Many sums actually add up to 1 due to the normalization of probabilities and the expressions could be simplified correspondingly, but this will change if we take evidence into account, see Section 2.6. So we leave the terms as they are. For the same reason the marginals are all normalized properly as well, but this too will change with evidence.

We see that the four calculations have some terms in common, such as \( \sum_{a,b,c} P(d|B) \) or \( \sum_{a} P(B|a)P(a) \). It is therefore good to store these as intermediate sums and reuse them rather than recalculating them. We also see that marginalizing over a variable can be done directly at the leaves of the Bayesian network, such as for \( a, c, \) and \( d \), but that for \( b \) which is more in the middle of the tree, one first has to do the marginalization over two of the three neighbors, namely those that are currently not of interest. Thus, inner nodes must ‘wait’ for the marginals over the outer nodes.

This successive summation and taking advantage of intermediate results can be efficiently organized in a message passing algorithm. The messages are indicated in the equations above and indicated in Figure 6. \( m_{AB}(B) \), for instance, is the message from \( A \) to \( B \) that only depends on variable \( B \) because variable \( A \) has been summed over already. More generally, a message a node \( A \) sends to node \( B \) is the sum over \( A \) of the product over all incoming messages from the other neighbors of \( A \) (excluding \( B \)) times the conditional that depends on \( A \) and \( B \) (either \( P(A|B) \) or \( P(B|A) \) depending on whether \( B \) is parent or child of \( A \), respectively) possibly times a prior over \( A (P(A)) \) if \( A \) has no parents.
message passing has to start at the leaves of the network, which only have one neighbor. The marginal probability distribution for one variable is the product over all incoming messages possibly times its prior. Notice that messages are not single values but functions of the variable of the sending node. You can also view them as a probability tables.

Unfortunately, this nice and efficient scheme does not work if a node has more than one parent. Consider, for instance, the example of Figure 5.c with \( P(A, B, C) = P(B|C, A)P(C)P(A) \). In a direct generalization of the message-passing scheme above one might be tempted to define the message from \( A \) to \( B \) to be \( m_{AB}(B) = \sum_a P(B|C, a)P(a) \), but this obviously also depends on \( C \) and is therefore problematic. The message from \( C \) to \( B \) might similarly be defined as \( m_{CB}(B) = \sum_c P(B|C, a)P(c) \).

According to the rules of message passing, the marginal \( P(B) \) would be

\[
P(B) = m_{CB}(B)m_{AB}(B) = (\sum_c P(B|C, a)P(c))(\sum_a P(B|C, a)P(a)),
\]

which is complete nonsense, because it still depends on \( A \) and \( C \).

Further readings: (Jordan and Weiss, 2002, sec. 3.2).

### 2.6 Message passing with evidence

More interesting than simply asking for the marginal of a variable in a Baysian network is to ask for a marginal if we have evidence, either we know the value of some nodes or we have partial evidence for some or even all nodes. Consider the latter case, i.e. partial evidence for all nodes, which is the most general case. The general rule is that variable combinations that are inconsistent with the evidence become impossible and that the probabilities of the other combinations are simply renormalized versions of the original ones, e.g.

\[
P(A, B|\mathcal{E}_A, \mathcal{E}_B) = \frac{P(A, B)}{\sum_{a\in\mathcal{E}_A, b\in\mathcal{E}_B} P(a, b)}
\]

for all \( A, B \) still possible. The normalization constant in the denominator is called partition function (D: Zustandssumme) and denoted by \( Z \) (from the German word ‘Zustandssumme’).

In a Bayesian network the numerator can be calculated exactly as before by message passing, just summing over the valid values rather than all values. The denominator can be calculated by summing over all valid values of the remaining variable. If in the network above the evidences \( \mathcal{E}_A, \mathcal{E}_B, \mathcal{E}_C, \mathcal{E}_D \) were given, we could calculate the marginal for \( D \), for instance, using (90) and renormalizing the result

\[
\diamond \quad \hat{P}(D|\mathcal{E}_A, \mathcal{E}_B, \mathcal{E}_C, \mathcal{E}_D) \overset{(90)}{=} \left( \sum_{b\in\mathcal{E}_B} \left( \sum_{c\in\mathcal{E}_C} P(c|b) \right) \left( \sum_{a\in\mathcal{E}_A} P(b|a)P(a) \right) \right) \frac{m_{CB}(b)}{m_{BD}(D)}
\]

\[
\diamond \quad Z := \sum_{d\in\mathcal{E}_D} \hat{P}(d|\mathcal{E}_A, \mathcal{E}_B, \mathcal{E}_C, \mathcal{E}_D),
\]

\[
P(D|\mathcal{E}_A, \mathcal{E}_B, \mathcal{E}_C, \mathcal{E}_D) = \frac{1}{Z} \hat{P}(D|\mathcal{E}_A, \mathcal{E}_B, \mathcal{E}_C, \mathcal{E}_D),
\]

where \( \hat{P} \) indicates an unnormalized probability. Notice that the message \( m_{CB}(b) \) is not normalized to 1 anymore.

### 2.7 Application: Image super-resolution

Dalley et al. (2004) have used a Bayesian network to infer high-resolution images from low-resolution images. Figure 7 illustrates the graphical model.
Figure 7: A Bayesian network representing the generation of a $2 \times 2$ pixel high-resolution (upper branch) and a $5 \times 5$ pixel low-resolution binary image-patch (lower branch) from a true shape $S$ (left). Figure: (Dalley et al., 2004, Fig. 2, URL)\(^4\)

$S$ represents the true shape, $L$ the $5 \times 5$ pixel low-resolution image patch, and $H$ the $2 \times 2$ pixel high-resolution image patch of the center pixel of $L$. $T$ and $F$ represent the binarized versions of $L$ and $H$, respectively. Thus the joint probability distribution can be written as

\[
P(F, H, T, L, S) = P(F|H)P(H|S)P(T|L)P(L|S)P(S).\tag{95}
\]

$f$, $h$, $t$, $l$, and $s$ indicate instantiations of $F$, $H$, $T$, $L$, and $S$, respectively, with 0 indicating black and 1 indicating white. Individual pixels of the low- and high-resolution patches are indexed with $i$ and $j$, respectively.

The values of $L$ and $H$ are deterministically derived from $S$ with some functions $\mathcal{L}$ and $\mathcal{H}$, e.g. by low-pass filtering and (sub-)sampling, i.e.

\[P(L|S) := \delta(L, \mathcal{L}(S)),\tag{96}\]
\[P(H|S) := \delta(H, \mathcal{H}(S)).\tag{97}\]

$\mathcal{L}$, $\mathcal{H}$, and $P(S)$ are not known, or at least difficult to formalize mathematically. Instead, a large set of training data $h^\mu$ (at 400 dpi) and $l^\mu$ (at 200 dpi) with $\mu = 1, ..., N$ is given, that has been derived algorithmically from synthetically rendered images of letters and other symbols. From this we can estimate all we need to know about $P(H|S)P(L|S)P(S)$. It is actually interesting that we do not need any explicit information about $P(S)$ but only samples from $P(H, L) = \sum_s P(H|s)P(L|s)P(s)$.

The values of $T$ and $F$ are derived from $L$ and $H$ by flipping the pixels to white with a probability of $P(t_i = 1|l_i) = l_i$ and $P(f_i = 1|h_i) = h_i$, respectively, i.e.

\[P(T|L) := \prod_{i=1}^{|L|}(L_i\delta_{1T_i} + (1 - L_i)\delta_{0T_i}),\tag{98}\]
\[P(F|H) := \prod_{j=1}^{|H|}(H_j\delta_{1F_j} + (1 - H_j)\delta_{0F_j}).\tag{99}\]

The task is to infer the most likely high-resolution image patch $f$ from a concrete low-resolution
This means we have to estimate

\[ P(F|T) = \frac{P(F,T)}{P(T)} \]  \hspace{1cm} (100)

\[ = \frac{\sum_h l s P(F,h,T,l,s)}{\sum_f h l s P(f,h,T,l,s)} \]  \hspace{1cm} (101)

\[ = \frac{\sum_h l s P(F|h)P(T|h,s)P(T|l)P(l|s)P(s)}{\sum_f h l s P(f|h)P(h|s)P(T|l)P(l|s)P(s)} \]  \hspace{1cm} (102)

\[ = \frac{\sum_h l s P(F|h)P(T|l)P(l|s)P(s)}{\sum_f h l s P(T|l)P(l|s)P(s)} \]  \hspace{1cm} (103)

\[ = \sum_s P(F|H(s))P(T|L(s))P(s) \]  \hspace{1cm} (104)

\[ \approx \frac{\sum_{\mu} P(F|h^{\mu})P(T|l^{\mu})}{\sum_{\mu} P(T|l^{\mu})} \]  \hspace{1cm} (105)

(if we approximate \( P(S) \) by the samples \( s^{\mu} \) given,

i.e. if we set \( \sum_s g(s)P(s) \approx \frac{1}{N} \sum_s g(s^{\mu}) \) for some function \( g \))

\[ \approx \prod_j \frac{\sum_{\mu} P(F_j|h^{\mu})P(T|l^{\mu})}{\sum_{\mu} P(T|l^{\mu})} \]  \hspace{1cm} (106)

(if we assume independence between the \( F_j \) given \( T \)).

The \( \hat{P}(F_j|T) \) can be estimated from the training set and stored in \( 2^5 \times 5 \) tables (one for each possible binary low-resolution pattern) of size \( 2 \times 2 \) (one cell for each high-resolution pixel).

In order to derive a high-resolution image from a low-resolution image we proceed as follows. For each pixel in the low-resolution image we determine its surrounding \( 5 \times 5 \) binary image patch \( t \) and use (106) to get the probabilities \( \hat{P}(F_j = 1|t) \) for the four high-resolution pixels representing the center pixel of the low-resolution pixel. This results in a high-resolution ‘image’ of probabilities for getting white pixels. Simple thresholding (preferably at 0.5) results in a binary high-resolution image. Figure 8 shows some results of the algorithm for Latin and Japanese (Kanji) characters.

![Figure 8](image-url)
3 Inference in Gibbsian networks

3.1 Introduction

We have seen above that the direction of edges does not really matter so much, but that the relationship of being parents of the same child matters a lot (of course the direction of edges matters a lot as soon as it affects this parental relationships). We have also seen that certain nodes can become conditionally independent of each other if certain other nodes are instantiated and/or others are completely unknown. Thus, Bayesian networks are a bit complicated to deal with. In the following we will see that things become easier if we turn the directed graphs into undirected graphs, but this comes at a price of losing information about statistical dependencies.

3.2 Moral graph

The goal of this section is to get rid of the directionality of the edges while somehow preserving information about the special relationship between parents of a common child. We proceed in two steps, first we insert undirected edges between parents of a common child (a process called marrying parents) and then we drop the directionality of the edges. The result is called a moral-graph. Jensen and Lauritzen (2001) call such an undirected graph a Gibbsian network, which acknowledges its conceptual origin in statistical physics. Figure 9 shows a simple example. Other names are Markov random field or Markov network (Bishop, 2006). Note that by construction each child node and its parents now form a complete subgraph, i.e. a set of nodes that are completely mutually connected.

On the mathematical side we can simply redefine the conditional probabilities, such as \( P(E|D,B) \), as potential functions \( \phi(B, D, E) \) without a directed relationship between the variables, i.e. \( \phi(B, D, E) := P(E|D,B) \). We can simplify this further by combining potential functions within the same clique (a clique is a complete subgraph that is not a subgraph of another complete subgraph, in that sense it has maximal size). This is not necessarily unique. For a Bayesian

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

\[
P(A,B,C,D,E) = \phi(A,B) \ \phi(B,C,E) \ \phi(B,D,E)
\]
network with \( P(A, B, C) = P(A|B)P(C|B)P(B) \), for instance, one could define \( \phi(A, B) := P(A|B) \) and \( \phi(B, C) := P(C|B)P(B) \) or one could define \( \phi(A, B) := P(A|B)P(B) \) and \( \phi(B, C) := P(C|B) \). Either way we have \( P(A, B, C) = \phi(A, B)\phi(B, C) \). It is also generally not possible to do a back-translation from the moral graph to its generating Bayesian network, since a potential function like \( \phi(B, C) \) could have been derived from \( P(B|C) \) or from \( P(C|B) \).

In the example above we can define

\[
\begin{align*}
\phi(A, B) & := P(B|A)P(A), \\
\phi(B, C, E) & := P(C|E, B), \\
\phi(B, D, E) & := P(E|D, B)P(D), \\
\diamond P(A, B, C, D, E) & \quad \text{[fig.9.a]} \quad \overset{(107-109)}{=} \quad P(C|E, B)P(E|D, B)P(D)P(B|A)P(A) \\
\diamond & \quad \overset{(107-109)}{=} \quad \phi(A, B)\phi(B, C, E)\phi(B, D, E).
\end{align*}
\]

In replacing conditionals by potential functions we loose the directionality information, which corresponds to dropping the directionality of the edges in the graph, and the parents now have the same relationship among each other as the parents have with their children, which corresponds to adding edges between parents of a common child in the graph. In fact the terms 'parents' and 'child' are not appropriate anymore in a moral-graph, we simple speak of \textit{neighbors}.

What we have gained is greater simplicity of the network, most importantly \textbf{two nodes A and C in a Gibbsian network are conditionally independent if (not iff!) for all paths between A and C there is an intermediate node B that is instantiated} (Jensen and Lauritzen, 2001). Thus finding conditional independences is much easier for a Gibbsian than for a Bayesian network, but one might miss some that were present in the Bayesian network and got lost in the process of turning the Bayesian into a Gibbsian network by generating the moral-graph.

It is also possible to define Gibbsian networks directly instead of deriving them from a Bayesian network. One can draw a graph and define a non-negative potential function for each clique. However, one usually has to normalize the product of all potential functions to a total sum of one to obey (1).

Further readings: (Cowell, 1999, sec. 7; Jensen and Lauritzen, 2001, sec. 2.5.1; Jordan and Weiss, 2002, sec. 2).

### 3.3 Junction tree and message passing in a Gibbsian network

The idea of message passing from Section 2.5 can be generalized to Gibbsian networks that do not have a tree structure. To simplify bookkeeping, it is useful to first construct a \textit{junction tree}. A \textit{junction tree} is a graph that has nodes representing the cliques of a Gibbsian network and edges connecting the nodes to a tree, such that nodes with a given variable form single sub-trees. A given Gibbsian network may have different junction trees or it may have none, see Section 3.5. Figure 10 shows a simple example of a Gibbsian network and one of its two possible junction trees, which are both maximal spanning trees.

Now, messages can be passed between the nodes of the junction tree or, for short, the cliques to compute the marginals of the variables efficiently. The difference to the procedure described in Section 2.5 is that the messages now may depend on more than one variable. The notation \( m_{BDE, BCE}(B, E) \), for instance, indicates that the message is sent from clique \( [BDE] \) to \( [BCE] \) and depends on \( B \) and \( E \). A message always depends on all variables that are common to the sender and receiver clique. All other variables of the sender clique must be marginalized over, so that they disappear.
\[ P(A,B,C,D,E) = \phi(A,B) \phi(B,C,E) \phi(B,D,E) \]

**Figure 10:** A junction tree (boxes and dashed lines) for the Gibbsian network of Figure 9 (circles and solid lines). The junction tree [\(AB\)] - [\(BCE\)] - [\(BDE\)] would be equally valid, but [\(BDE\)] - [\(AB\)] - [\(BCE\)] would not, because variable \(E\) would be contained in disconnected nodes.

© CC BY-SA 4.0

For the example in Figure 10 we get the messages

\[
m_{AB,BDE}(B) := \sum_a \phi(a,B), \quad (112)
\]

\[
\diamond m_{BCE,BDE}(B,E) := \sum_c \phi(B,c,E), \quad (113)
\]

\[
\diamond m_{BDE,AB}(B) := \sum_e m_{BCE,BDE}(B,e) \sum_d \phi(B,d,e), \quad (114)
\]

\[
m_{BDE,BCE}(B,E) := m_{AB,BDE}(B) \sum_d \phi(B,d,E), \quad (115)
\]

and the marginals

\[
\diamond P(A, B) = m_{BDE,AB}(B) \phi(A,B) \quad \text{(this is the result, now follows a verification)} \quad (116)
\]

\[
\diamond (114) \Rightarrow \sum_e m_{BCE,BDE}(B,e) \sum_d \phi(B,d,e) \phi(A,B) \quad (117)
\]

\[
\diamond (113) \Rightarrow \sum_e \sum_c \phi(B,c,e) \sum_d \phi(B,d,e) \phi(A,B) \quad (118)
\]

\[
\diamond (107-109.) \Rightarrow \sum_e \sum_c P(c|e,B) \sum_d P(e|d,B) P(d) P(B|A) P(A) \quad (119)
\]

\[
= \sum_d \sum_e P(e|d,B) P(d) P(B|A) P(A) \quad (120)
\]

\[
= P(A, B), \quad (121)
\]

\[
P(B, D, E) = m_{AB,BDE}(B) m_{BCE,BDE}(B,E) \phi(B,D,E), \quad (122)
\]

\[
P(B, C, E) = m_{BDE,BCE}(B,E) \phi(B,C,E). \quad (123)
\]
Marginals of just one variable can be derived by further marginalization, e.g.

$$P(E) \overset{(2)}{=} \sum_{b,d} P(B,D,E)$$

$$\overset{(122)}{=} \sum_b m_{AB,BDE}(b) m_{BCE,BDE}(b,E) \sum_d \phi(b,d,E)$$

$$\overset{(112,113)}{=} \sum_b \left( \sum_a \phi(a,b) \right) \left( \sum_c \phi(b,c,E) \right) \left( \sum_d \phi(b,d,E) \right)$$

$$\overset{(127)}{=} \sum_b \left( \sum_a P(b|a)P(a) \right) \left( \sum_c P(c|E,b) \right) \left( \sum_d P(E|d,b)P(d) \right) .$$

Compare the result with (38). The resubstitution of the probabilities $P$ for the potential functions $\phi$ is done here only for didactic purposes and not necessary to actually calculate the messages and marginals.

The partition function can be calculated in different ways

$$Z := \sum_{a,b,c,d,e} P(a,b,c,d,e)$$

$$\overset{(111)}{=} \sum_{a,b,c,d,e} \phi(a,b)\phi(b,c,e)\phi(b,d,e)$$

$$= \sum_{b,c,e} \left( \sum_a \phi(a,b) \right) \left( \sum_c \phi(b,c,e) \right) \phi(b,d,e) \quad AB \rightarrow BDE \leftarrow BCE \quad (130)$$

$$= \sum_{b,c,e} \left( \sum_d \phi(b,d,e) \right) \phi(b,c,e) \phi(a,b) \quad AB \rightarrow BDE \rightarrow BCE \quad (131)$$

$$= \sum_{a,b} \left( \sum_{d,e} \phi(b,c,e) \phi(b,d,e) \right) \phi(a,b) \quad AB \leftarrow BDE \leftarrow BCE \quad (132)$$

$$= \sum_b \left( \sum_a \phi(a,b) \right) \left( \sum_c \phi(b,c,e) \right) \left( \sum_d \phi(b,d,e) \right) . \quad (133)$$

Equations (130–132) show the order in which $Z$ would be calculated for the three different ways in which messages could be passed to one node. Equation (133) shows an order that would be even more efficient but does not correspond to a message passing algorithm.

Message passing with partial evidence is analogous to that in a Bayesian network. One simply sums only over valid values of the random variables, i.e. one replaces expressions like $\sum_{b,c,e}$ by $\sum_{b \in E_b,c \in E_c,e \in E_e}$, and renormalizes by the partition function $Z$, also calculated with partial evidence. Further readings: (Cowell, 1999, secs. 9, 10; Jensen and Lauritzen, 2001, sec. 3.2; Jordan and Weiss, 2002, sec. 3.4).
3.4 Most probable explanation

We have seen above how one can calculate marginal probabilities in a graphical model, with or without partial evidence. Another question that is often of interest is: What is the most probable state of the network? Notice that this is not necessarily the set of most probable individual values. Thus calculating all marginals and then maximizing the probabilities of all random variables individually would not work. We have to search for the combination of values that maximize the overall probability. If one does that with some evidence given, the most probable state can be interpreted as the most probable explanation (MPE) of the variables with partial evidence. Fortunately, this can again be calculated efficiently by message passing, simply by replacing the sums in the calculation of the partition function, see (130–132) above, by maxima. This works because \( \max_{ij}(x_i, y_j) = \max_i(x_i) \max_j(y_j) \) if \( x_i, y_j \geq 0 \), which is the case for potential functions. For the graph in figure 10 we find for the three different message passing schedules

\[
\max_{a,b,c,d,e} P(a,b,c,d,e)
\]

\[
= \max_{a,b,c,d,e} \phi(a,b) \phi(b,c,e) \phi(b,d,e)
\]

\[
= \max_{b,d,e} \left( \max_a \phi(a,b) \right) \left( \max_c \phi(b,c,e) \right) \phi(b,d,e)
\]

\[
= \max_{b,c,e} \left( \max_d \left( \max_a \phi(a,b) \right) \phi(b,d,e) \phi(b,c,e) \right)
\]

\[
= \max_{a,b} \left( \max_d \left( \max_c \phi(b,c,e) \phi(b,d,e) \phi(a,b) \right) \right)
\]

Notice, that the messages \( e \) are again not real numbers but tables. The last line shows again the even more efficient calculation that does not correspond to a message passing schedule. Unfortunately, we are not done yet, because we only have the probability of the most probable state but not the state itself. We do not know yet, which values the random variables would have to assume. However, this is easy to trace back if we have not only stored the maximum values in the messages but also which values of the random variables gave rise to the maximum values. As an example consider equation (137) and assume the random variables are all binary with values \( \{0, 1\} \). Let the maximum over \( B,C,E \) be assumed for \( B = 0, C = 1, E = 1 \). We can then look into message \( e_{BDE,BCE}(BE) \) and ask which value of \( D \) gave rise to the maximum for \( B = 0, E = 1 \), let say \( D = 1 \). We can finally look into message \( e_{AB,BDE}(B) \) and ask which value of \( A \) gave rise to the maximum for \( B = 0, E = 1 \), let say \( A = 0 \). Then we know \( A = 0, B = 0, C = 1, D = 1, E = 1 \) is the most probable state.

Finding the most probable explanation in Bayesian networks works exactly the same way. If evidence has to be taken into account, maxima are taken only over valid values, i.e. expressions like \( \max_{b,c,e} \) are replaced by \( \max_{b \in E_b, c \in E_c, e \in E_e} \), and if we really need to know the probability, the result must be normalized by the partition function with evidence. However, for just finding the most probable explanation we do not need the value of the partition function, because that is only a global factor, which does not change the location of the maximum.

26
3.5 Triangulated graphs

If a Gibbsian network has loops longer than three nodes/edges without a shortcut, a junction tree would have disconnected subtrees with the same variable. Consider Figure 11.

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

triangulate by removing C, A, B, D, E

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

triangulate by removing E, D, B, A, C

**Figure 11**: Triangulation of a graph to permit junction trees. (a) shows a Gibbsian network (circles and solid lines) and nodes (boxes) of a potential junction tree. The tree structure indicated (dashed lines) does not form a junction tree, because nodes \[AB\] and \[BCE\] are disconnected even though both contain \(B\). The network in (b) has been triangulated with one edge added (dotted line) so that a junction tree can be formed. (c) shows an unfavorable triangulation where all nodes are connected with each other, resulting in a junction tree with only one node \[ABCDE\].

© CC BY-SA 4.0

The graph on the left has a loop of length four without shortcuts. The cliques would form the nodes of the junction tree, but it is not possible to connect them to a tree such that the cliques involving a particular random variable are always connected to a single sub-tree, which is known as the *running intersection property*. In the suggested connectivity in Figure 11.a, the cliques involving \(B\), namely \[AB\] and \[BCE\], are disconnected. **This** has fatal consequences, because it renders message passing impossible, since one cannot marginalize over \(B\) at node \[AB\] without taking into account node \[BCE\]. Thus, loops longer than three nodes/edges must be avoided.

**This** problem can always be solved by *triangulation*, which means to add shortcut edges such that there are no loops without shortcuts greater than three. The added edges do not really introduce statistical dependencies, but one pretends that there are additional dependencies. So once again one loses information about conditional independences to simplify the structure of the graph.

A simple procedure to triangulate a graph is to successively remove nodes and their edges to other nodes, but for each node being removed add shortcut edges between all of its neighboring nodes if they are not connected already. The resulting graph is empty, of course, because you have removed all nodes. But if you add all shortcut edges to the original graph again, you get a triangulated graph. The graph in Figure 11.a can be triangulated by removing the nodes in the order \(A, D, B, E, C\), which results in the graph (b), for which a junction tree can be formed. **The trick is to remove the nodes in such an order that the cliques remain small**, so that message passing in the junction tree can be efficient. Figure 11.c shows an unfavorable trianulation, where the nodes have been removed in the order \(E, D, B, A, C\), so that the junction tree has only one node with all variables, which is quite inefficient.

Further readings: (Cowell, 1999, secs. 11, 12; Jensen and Lauritzen, 2001, secs. 3.2; Jordan and Weiss, 2002, sec. 3.4).
4 Approximate inference

4.1 Introduction

So far we have done exact inference, i.e. we have computed the true marginals. This might be computationally too expensive, so cheaper approximate methods are needed.

4.2 Gibbs sampling

The idea of Monte Carlo algorithms is to produce a large number of samples from the joint probability distribution instead of computing its distribution or marginals exactly. The Marginals of interest can then be estimated from the samples. A simple example of a Monte Carlo algorithm is the Gibbs sampling algorithm.

In Gibbs sampling one repeatedly selects a random variable at random and assigns to it a new value according to its conditional probability distribution given the current values of all other random variables. The interesting property of this procedure is, that even though one updates only one unit at a time the samples produced that way reproduce the probability distribution of the network as a whole (under fairly general conditions, see below). Computing the conditional is simplified by the fact that only the terms that include the selected random variable have to be taken into account. For instance, the conditional of \( D \) given all other variables in the example of Figure 4 would be

\[
P(D|A, B, C, E) = \frac{P(A, B, C, D, E)}{P(A, B, C, E)}
\]

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

Once a great number of value combinations for all random variables have been generated, marginals can be estimated from that.

As a very simple example consider a system with two random variables, \( A \) and \( B \), which both can assume the values 0 and 1 with the probabilities shown in Table 3. (Imagine you first role a red die. If it shows 1, 2, or 3 then \( A = 0 \) otherwise \( A = 1 \). If \( A = 0 \) then \( B = 0 \), otherwise you role a blue die. If it shows 1 or 2 then \( B = 0 \) otherwise \( B = 1 \).)

| \( A \) | \( B \) | \( P_A \) | \( P_{A|B} = \frac{P_B}{P_B|A}P_A \) | \( P_B = \sum_a P_{A,B} \) | \( P_{A|B} = \frac{P_{A,B}}{P_B} \) | exact \( P_{A,B} \) | est. \( P_{A,B} \) |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1/2 | 1 | 4/6 | 3/4 | 0.5 | 0.36 |
| 0 | 1 | 1/2 | 0 | 2/6 | 0 | 0.0 | 0.0 |
| 1 | 0 | 1/2 | 1/3 | 4/6 | 1/4 | 0.166 | 0.15 |
| 1 | 1 | 1/2 | 2/3 | 2/6 | 1 | 0.333 | 0.48 |

Table 3: Probabilities of a very simple example system. The last two columns show the exact (cf. fifth column) and the approximate values of \( P(A, B) \) estimated by Gibbs sampling, see Table 4.

To do Gibbs sampling start with \( A = 0, B = 0 \). Then repeat the following two steps: (i) randomly select \( A \) or \( B \) with equal probability (flip a coin); (ii) choose a new (but not necessarily different) value for the selected variable according to its conditional probability given its neighbors (Role a die. For \( P_{B|A}(B|1) \) one could decide to choose \( B = 0 \) if the die shows 1 or 2 and \( B = 1 \) for 3,
4, 5, and 6; for $P_{A|B}(A|0)$ one could decide to choose $A = 0$ for 1, 2, and 3, choose $A = 1$ for 4, and role the die again if it shows 5 or 6.). **Table 4** shows an example of such an experiment.

<table>
<thead>
<tr>
<th>coin</th>
<th>die</th>
<th>$A$</th>
<th>$B$</th>
<th>coin</th>
<th>die</th>
<th>$A$</th>
<th>$B$</th>
<th>coin</th>
<th>die</th>
<th>$A$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5,4</td>
<td>1</td>
<td>0</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>B</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>5,1</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>6,5,2</td>
<td>0</td>
<td>0</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>6,6,4</td>
<td>1</td>
<td>0</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>B</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>A</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>A</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>B</td>
<td>0</td>
<td>0</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>B</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>0</td>
<td>0</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>6,6,4</td>
<td>1</td>
<td>0</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>B</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>5,3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>5,1</td>
<td>0</td>
<td>0</td>
<td>B</td>
</tr>
</tbody>
</table>

**Table 4:** Gibbs sampling of a simple Gibbsian network with just two nodes. A coin is used to select a node and a die is used to choose its value according to its conditional probability, which is $P(A = 0|B = 0) = 3/4$ (die values 1, 2, and 3), $P(A = 1|B = 0) = 1/4$ (die value 4), $P(A = 0|B = 1) = 0$, $P(A = 1|B = 1) = 1$ for $A$ and $P(B = 0|A = 0) = 1$, $P(B = 1|A = 0) = 0$, $P(B = 0|A = 1) = 1/3$ (die values 1 and 2), $P(B = 1|A = 1) = 2/3$ for $B$ (die values 3, 4, 5, and 6), see Table 3. Each row shows the selected node, the result of the die-rolling, and the new values of $A$ and $B$. Because of $(A = 0) \Rightarrow (B = 0)$ and $(B = 1) \Rightarrow (A = 1)$ no die rolling is necessary in these cases; if $B = 0$ the die values 5 and 6 are not used to determine the value of $A$ and the die has to be rolled again.

The last column of **Table 3** shows the probabilities of $P_{A,B}$ as estimated from the results of the Gibbs sampling, from which also the marginals can be easily derived. The deviation from the exact values is actually fairly large, which might be just bad luck (I have not made any attempt to get good results but have taken the first results I have got), but it is probably also due to the fact that the system has two states, $(A = 0, B = 0)$ and $(A = 1, B = 1)$, that are connected via states of low or even zero probability. Thus, the system tends to stick to each of the likely states for a long time, as is also obvious from the values in Table 4 and is illustrated in Figure 12. If both intermediate states, i.e. $(A = 0, B = 1)$ and $(A = 1, B = 0)$, had zero probability, the system would have stuck to one of the possible states forever. This illustrates two conditions that must be met for Gibbs sampling to work (Heckerman, 1999). Firstly, there must be a significant (theoretically non-zero) probability to get from any instantiation of all random variables to any other instantiation. Secondly, one must select all random variables very often (theoretically infinitely often).

Further readings: (Jordan and Weiss, 2002, sec. 4.1).
Figure 12: Transition probabilities between the four different states \((A, B) = (0, 0), (0, 1), (1, 0), \) and \((1, 1)\) for the system given in Table 4.

© CC BY-SA 4.0
5 Bayesian learning for binary variables

5.1 Levels of Bayesian learning

A Bayesian network has a structure (nodes and edges representing random variables and direct statistical dependencies, respectively), local probability distributions for the random variables, and values of the random variables. In practice one has to decide on a particular parameterization of the local probability distributions and then has to find the right parameter values for it. Thus in a Bayesian network we can distinguish four levels:

1. a structure,
2. a parameterization of the local probability distributions,
3. values of the parameters and probability distributions thereof,
4. values of the random variables and probability distributions thereof.

Inference in a Bayesian network is about level four if the other three levels are given; learning in a Bayesian network is about estimating what is appropriate on the first three levels. In this and the next section I focus mainly on level three, namely on the learning of the parameters, assuming that the structure and the parameterization are given.

5.2 A simple example

As the simplest possible example consider the case of a binary random variable $X$ that can assume the values $x \in \{0, 1\}$ with unknown probabilities $P(x = 0)$ and $P(x = 1)$. It could, for instance, be a biased coin or a thumbtack that we toss. Since $P(x = 0) + P(x = 1) = 1$ we have only one unknown probability to estimate and take this as the parameter $\theta$, i.e. we set the probability of heads to $P(x = 1) := \theta$; the other can be derived from that. At this point one could distinguish in the notation between the true probability and the estimated (parameterized) one, but I do not do that for simplicity. Note also that the parameterization in this case is rather trivial, so that you might even not have noticed that I have just introduced a parameterization. However, alternative parameterizations might have been $P(x = 0) := \theta$ or $P(x = 1) := \theta^2$ (not particularly sensible, I admit, but possible).

As mentioned above, in Bayesian learning such parameters $\theta$ that define the probability distributions are treated as instantiations of random variables themselves, in this case $\Theta$. Even if we do not have any data available we have some prejudice about the probability distribution of $\Theta$. This is the a priori probability distribution (D: a priori Wahrscheinlichkeitsverteilung) or, for short, the prior $p(\theta)$ (note that I write $p(\cdot)$ instead of $P(\cdot)$, because $\Theta$ is a continuous variable and not a discrete one; I sometimes write $p_\theta$ to indicate the prior without a concrete argument). Figure 13 shows an example of such a prior.

If $\theta$ were fixed and known, one would not worry about it much and simply set $P(x = 1) := \theta$. However, since $\theta$ is not known and might vary from coin to coin (not from toss to toss) we have to take it into account in our calculations explicitely and write

\[ P(x = 1|\theta) := \theta \] (143)
with a variable $\theta$. To estimate the probability of heads, we have to integrate over all possible values of $\Theta$ weighted by its probability density function.

\[
\begin{align*}
\bullet \quad P(x = 1 | p_\theta) &= \int_0^1 P(x = 1 | \theta) p(\theta) d\theta \\
\bullet \quad &= \int_0^1 \theta p(\theta) d\theta. 
\end{align*}
\]

(144) (145)

This is what we can say if we just rely on our prejudices, i.e. prior, about $\Theta$ and have not tossed the coin once.

It will later be useful to know also the probabilities of combinations of tosses. Given we know $\theta$, if we toss the coin $N$ times altogether, what is the probability to get exactly $H$ times heads ($x = 1$) and $T$ times tails ($x = 0$)? Easier to answer is the question of what the probability of a particular sequence of heads and tails is, because that is simple the product of $\theta$ for each heads and $(1 - \theta)$ for each tails. The Sequence heads-tails-tails-heads-tails, for instance has the probability $\theta (1 - \theta) (1 - \theta) \theta (1 - \theta) = \theta^2 (1 - \theta)^3$.

Now if we do not care about the exact order but only about how often we got heads, we have to multiply the probability of the sequence with the number of different sequences with the same number of heads and tails, which is $(H + T)!/(H!T!)$, i.e.

\[
\begin{align*}
\bullet \quad P(H,T | \theta) &= \frac{(H + T)!}{H!T!} \theta^H (1 - \theta)^T.
\end{align*}
\]

(146)

How does the probability of heads change if we have tossed the coin $N$ times and have gotten $H$ heads and $T$ tails? In other words what is $p(\theta|H,T,p_\theta)$? Note that summarizing the results of the tosses just by the numbers of heads and tails is justified by the fact (or assumption) that all tosses are statistically independent, so that the order of results does not matter at all. It is clear that now the probability (our belief) for getting heads in the next toss does not only depend on our prior $p_\theta$ but also on the results of the previous tosses, which give us additional information. Figure 14 illustrates how our pdf of $\theta$ might change with an increasing number of tosses.

Formally, let us first update our probability distribution over $\theta$ by taking the results of the tosses into
account. Bayes formula yields

\[
p(\theta|H, T, p_\theta) = \frac{P(H, T|\theta, p_\theta) p(\theta|p_\theta)}{P(H, T|p_\theta)} \tag{147}
\]

\[
p = \frac{P(H, T|\theta) p(\theta|p_\theta)}{P(H, T|p_\theta)} \quad \text{(because } p_\theta \text{ does not matter if } \theta \text{ is known)} \tag{148}
\]

\[
p = \frac{P(H, T|\theta) p_\theta(\theta)}{P(H, T|p_\theta)} \quad \text{(because } \theta \text{ is trivially distributed according to } p_\theta(\theta) \text{ if only } p_\theta \text{ is given)} \tag{149}
\]

\[
p = \frac{P(H, T|\theta) p_\theta(\theta)}{\int_0^1 P(H, T|\theta') p_\theta(\theta') \, d\theta'} \quad \text{(analogous to (144))} \tag{150}
\]

\[
p = \frac{\theta^H (1-\theta)^T p_\theta(\theta)}{\int_0^1 \theta'^H (1-\theta')^T p_\theta(\theta') \, d\theta'} \tag{151}
\]

This is the Bayesian learning part. We have first started with a prior probability distribution \( p_\theta(\theta) \) and as we get some data we update it and learn a new distribution \( p(\theta|H, T, p_\theta) \). The rest is plain inference to derive the probability of heads in the next toss analogous to (145).

\[
P(x = 1|H, T, p_\theta) = \int_0^1 P(x = 1|\theta) p(\theta|H, T, p_\theta) \, d\theta \quad \tag{152}
\]

\[
= \int_0^1 \theta p(\theta|H, T, p_\theta) \, d\theta \quad \tag{153}
\]

\[
= \frac{\int_0^1 \theta^H (1-\theta)^T p_\theta(\theta) \, d\theta}{\int_0^1 \theta'^H (1-\theta')^T p_\theta(\theta') \, d\theta'} \quad \tag{154}
\]

5.3 The Beta distribution

In the simple example given above expressions like \( \theta^H (1-\theta)^T p_\theta(\theta) \) and integrals like \( \int_0^1 \theta'^H (1-\theta')^T p(\theta') \, d\theta' \) occur repeatedly. Since we have some freedom in defining \( p(\theta) \) we can use that to simplify the

\[
\begin{align*}
\text{Figure 14: How our pdf of } \theta \text{ might change as more and more tosses become available. The two additional curves have been computed after 16 and 64 tosses, respectively, with 3/4 heads and 1/4 tails.} \\
\text{© CC BY-SA 4.0}
\end{align*}
\]
calculations. One convenient choice is to use the beta distribution (Wikipedia, 2006)

\[
\beta(h; t) := \frac{\theta^{h-1}(1 - \theta)^{t-1}}{\int_0^1 \theta^{h-1}(1 - \theta)^{t-1} \, d\theta'} \tag{155}
\]

with \( h, t > 0 \). The denominator is simply a normalization factor that can also be written in terms of the beta-function

\[
B(h, t) := \int_0^1 \theta^{h-1}(1 - \theta)^{t-1} \, d\theta' \tag{156}
\]

or in terms of the gamma-function as \( \Gamma(h)\Gamma(t)/\Gamma(h + t) \). See Figure 15 for examples of the beta distribution. The beta distribution is so convenient, because if we set

\[
p(\theta) := \beta(\theta; h, t),
\]

then equations (151) and (154) simplify to

\[
p(\theta|H, T, p_\theta) \overset{(151,157)}{=} \frac{\theta^H (1 - \theta)^T \beta(\theta; h, t)}{\int_0^1 \theta^H (1 - \theta)^T \beta(\theta'; h, t) \, d\theta'} \tag{158}
\]

\[
= \frac{\theta^H (1 - \theta)^T \theta^{h-1}(1 - \theta)^{t-1}}{\int_0^1 \theta^H (1 - \theta)^T \theta^{h-1}(1 - \theta)^{t-1} \, d\theta'}, \tag{159}
\]

(since the normalization constants of the two \( \beta \) cancel out)

\[
= \frac{\theta^{h + T - 1}}{\int_0^1 \theta^{h + T - 1} \, d\theta'}, \tag{160}
\]

\[
\beta(\theta; H + h, T + t), \tag{161}
\]

\[
P(x = 1|H, T, p_\theta) \overset{(153,158)}{=} \frac{\int_0^1 \theta^H (1 - \theta)^T \beta(\theta; h, t) \, d\theta}{\int_0^1 \theta^H (1 - \theta)^T \beta(\theta'; h, t) \, d\theta'} \tag{162}
\]

\[
= \frac{\int_0^1 \theta^{H + h + 1 - 1}(1 - \theta)^{T + t - 1} \, d\theta}{\int_0^1 \theta^{H + h + T - 1} \, d\theta'} \tag{163}
\]

\[
= \frac{B(H + h + 1, T + t)}{B(H + h, T + t)}, \tag{164}
\]

\[
= \frac{H + h}{H + h + T + t} \tag{165}
\]

(since \( B(h + 1, t) = B(h, t)h/(h + t) \)).

Equation (165) suggests a simple interpretation of the beta distribution. The ratio \( H/(H + T) \) is a reasonable estimate of \( \theta \) if we have no prior, or rather a flat prior. The ratio of (165) can be interpreted as if we had added some virtual tosses with \( h \) heads and \( t \) tails. Thus the beta distribution reflects our belief that if we had tossed the coin before, we would have gotten \( h \) heads and \( t \) tails, so that \( h/(h + t) \) is the most likely value of \( \Theta \) and \( h + t \) reflects the confidence of this belief.

The beta distribution is a unimodal distribution, which means it has only one peak. If that is not suitable to represent our prior, we can combine a few beta distributions. For instance,

\[
p(\theta) := 0.7 \beta(\theta; 10, 10) + 0.3 \beta(\theta; 15, 5) \tag{166}
\]

could reflect our belief that with probability 0.7 we have a fair coin and with probability 0.3 we have a coin that has been biased towards heads (note that 0.3 and 0.7 have to add up to 1 to make \( p(\theta) \) a proper probability distribution). With sums of beta distributions the calculations are still fairly simple.

There is also a straightforward extension of the beta distribution to random variables with more than two possible values. It is called Dirichlet distribution.
\[ h/t = \frac{1}{3} \]

<table>
<thead>
<tr>
<th>( h, t )</th>
<th>( \beta(\theta; h, t) )</th>
<th>( h, t = (1, 1) )</th>
<th>( h + t = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 3)</td>
<td>[ \beta(\theta; h, t) ]</td>
<td>( (1, 1) )</td>
<td>( (2, 2) )</td>
</tr>
<tr>
<td>(2, 6)</td>
<td>[ \beta(\theta; h, t) ]</td>
<td>( (3, 1) )</td>
<td>( (4, 4) )</td>
</tr>
<tr>
<td>(4, 12)</td>
<td>[ \beta(\theta; h, t) ]</td>
<td>( (6, 2) )</td>
<td>( (8, 8) )</td>
</tr>
<tr>
<td>(8, 24)</td>
<td>[ \beta(\theta; h, t) ]</td>
<td>( (12, 4) )</td>
<td>( (16, 16) )</td>
</tr>
<tr>
<td>(16, 48)</td>
<td>[ \beta(\theta; h, t) ]</td>
<td>( (24, 8) )</td>
<td>( (32, 32) )</td>
</tr>
<tr>
<td>(32, 64)</td>
<td>[ \beta(\theta; h, t) ]</td>
<td>( (48, 16) )</td>
<td>( (64, 64) )</td>
</tr>
</tbody>
</table>

**Figure 15:** Beta distributions \( \beta(\theta; h, t) \) for different parameter values \( (h, t) \). The scales of all plots are identical with an abscissa range of \([0, 1]\) and an ordinate range of \([0, 8]\). All distributions are normalized to \( \int \beta(\theta; h, t) \, d\theta = 1 \).

© CC BY-SA 4.0
6 Learning in Bayesian networks

6.1 Breaking down the learning problem

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

Figure 16: A Bayesian network. © CC BY-SA 4.0

Now that we know how Bayesian learning works in a simple example of one single random variable we can consider learning in Bayesian networks. Recall that we assume that the structure of the network is given (and accurate). Then the probability distribution over all variables of the network can be written as a product of conditionals according to the network structure. For instance, the Bayesian network of Figure 16 can be written as

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

(167)

This also means we can decompose the problem of learning the overall probability distribution \(P(A,B,C,D,E)\) into the subproblems of learning the conditional probabilities \(P(C|E,B)\), \(P(E|D,B)\), etc.

For simplicity we again only consider the case of binary variables and we assume that a separate parameter \(\theta\) is given for each possible combination of values for the parent nodes. For node/variable \(C\), for instance, we can write \(\theta_{C|E,B}\), which represents four different probabilities, namely \(P(c=1|e=0,b=0) = \theta_{C|0,0}\), \(P(c=1|e=0,b=1) = \theta_{C|0,1}\) etc. (you can think of \(\theta_{C|E,B}\) as a little matrix with elements \(\theta_{C|e,b}\)). For the network of Figure 16 we have overall 12 scalar parameters, namely \(\theta_A, \theta_D, \theta_{B|0}, \theta_{B|1}, \theta_{E|0,0}, \theta_{E|0,1}, \theta_{E|1,0}, \theta_{E|1,1}, \theta_{C|0,0}, \theta_{C|0,1}, \theta_{C|1,0}, \text{ and } \theta_{C|1,1}\). If there were a node with three parents, we would have parameters like \(\theta_{F|0,0,0}, \theta_{F|0,0,1}, \text{ etc.}\)

6.2 Learning with complete data

The true or optimal values of the parameters \(\theta\) are not known; they must be learned, or rather their probability distributions must be learned. Now the trick is that learning one of these parameters works exactly like described in Section 5. Of course, we need data to learn the parameters. So we assume we have a number of cases \(\mu = 1, \ldots, M\) for which the values of all nodes in the network are given, \(a^\mu, b^\mu, c^\mu, d^\mu, \text{ and } e^\mu\). For learning the distribution over \(\theta_{C|0,1}\), for instance, we ignore \(a^\mu\) and \(d^\mu\), because \(A\) and \(D\) do not matter for \(P(C|E,B)\) and therefore \(\theta_{C|0,1}\), and we only take the cases \(\mu\) with \(e^\mu = 0\) and \(b^\mu = 1\), because these are the only relevant cases for \(\theta_{C|0,1}\). The number of these cases in which \(c^\mu = 0\) and \(c^\mu = 1\) constitutes our \(T\) and \(H\) of Section 5, respectively, see Table 5. Of course we also need a prior \(p_\theta\) for each \(\theta\), which we have to make up before learning.
Take all data, ignore \(a^\mu\) and \(d^\mu\) and select the cases with \(c^\mu = 0\) and \(b^\mu = 1\), count tails and heads for \(c^\mu\), and consult Section 5 to determine

\[
\begin{align*}
T_{C|0,1} &= 3 \\
H_{C|0,1} &= 4
\end{align*}
\]

\[
p(\theta_{C|0,1}|H_{C|0,1}, T_{C|0,1}, p_{\theta_{C|0,1}})
\]

<table>
<thead>
<tr>
<th>(\mu)</th>
<th>(a^\mu)</th>
<th>(b^\mu)</th>
<th>(c^\mu)</th>
<th>(d^\mu)</th>
<th>(e^\mu)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</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>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5: Learning of \(p(\theta_{C|0,1}|\Delta, p_{\theta_{C|0,1}})\) given all the data \(\Delta\) is done by selecting only the relevant cases, counting heads and tails, and then applying the methods of Section 5. The true parameters for this example were \(\theta_A = 0.5\), \(\theta_D = 0.5\), \(\theta_{B|0} = 0.25\), \(\theta_{B|1} = 0.75\), \(\theta_{E|0,0} = 0.25\), \(\theta_{E|0,1} = 0.75\), \(\theta_{E|1,0} = 0.75\), \(\theta_{E|1,1} = 0.25\), \(\theta_{C|0,0} = 0.25\), \(\theta_{C|0,1} = 0.75\), \(\theta_{C|1,0} = 0.75\), and \(\theta_{C|1,1} = 0.25\). Thus, \(A\) and \(D\) simulate fair coins, \(B\) copies \(A\) with some noise, and \(E\) and \(C\) realize a noisy XOR of their parents.
6.3 Learning with incomplete data

If the data is not complete, i.e. if for some cases the value of some variables is not known, we cannot apply the method above directly. We first have to complete the data set by, e.g., Gibbs sampling, a method we have learned about in context of inference in graphical models 4. After having filled in the missing values, we can proceed as if the data were complete, see above. See Table 6 for an example. The problem, of course, is that initially we do not have the probabilities available yet to fill in the missing data. Thus, one has to fill in some arbitrary values first and then iterate the whole process several times. Repeating the iteration even further and averaging over the results improves the estimates.

Take the incomplete data, fill in missing values by Gibbs sampling,

\[
\begin{array}{cccccc}
\mu & a^\mu & b^\mu & c^\mu & d^\mu & e^\mu \\
1 & 1 & 1 & 1 & 0 \\
2 & 0 & 1 & 1 & 0 & 0 \\
3 & 1 & 1 & 1 & 0 & 0 \\
4 & 0 & 0 & 1 & 0 & 0 \\
5 & 1 & 1 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 \\
7 & 0 & 0 & 0 & 0 & 0 \\
8 & 0 & 1 & 0 & 1 & 0 \\
9 & 0 & 0 & 0 & 0 & 0 \\
10 & 1 & 0 & 0 & 0 & 0 \\
11 & 1 & 1 & 0 & 1 & 0 \\
12 & 1 & 0 & 0 & 0 & 0 \\
13 & 0 & 0 & 0 & 0 & 0 \\
14 & 0 & 0 & 0 & 0 & 0 \\
15 & 1 & 1 & 0 & 0 & 0 \\
16 & 1 & 0 & 0 & 0 & 0 \\
17 & 1 & 0 & 0 & 0 & 0 \\
18 & 0 & 0 & 0 & 0 & 0 \\
19 & 0 & 1 & 0 & 0 & 0 \\
20 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
\mu & a^\mu & b^\mu & c^\mu & d^\mu & e^\mu \\
1 & 1 & 1 & 0 & 1 & 0 \\
2 & 0 & 1 & 1 & 0 & 0 \\
3 & 1 & 0 & 0 & 1 & 0 \\
4 & 0 & 1 & 0 & 1 & 0 \\
5 & 1 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 \\
5 & 1 & 1 & 1 & 0 & 0 \\
7 & 1 & 1 & 0 & 0 & 0 \\
8 & 0 & 0 & 0 & 0 & 0 \\
9 & 1 & 1 & 1 & 1 & 0 \\
10 & 0 & 1 & 0 & 0 & 0 \\
11 & 1 & 1 & 1 & 1 & 0 \\
12 & 1 & 0 & 0 & 0 & 0 \\
13 & 0 & 1 & 1 & 0 & 0 \\
14 & 0 & 0 & 0 & 0 & 0 \\
15 & 0 & 0 & 0 & 0 & 0 \\
16 & 0 & 0 & 0 & 0 & 0 \\
17 & 0 & 0 & 0 & 0 & 0 \\
18 & 0 & 1 & 0 & 0 & 0 \\
19 & 0 & 0 & 0 & 0 & 0 \\
20 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

and proceed as in Table 5.

Table 6: Learning of \(p(\theta_{C|0,1}|\Delta,\theta_{0,1}|\Delta)\) given incomplete data \(\Delta\) is done by first filling in the missing data, e.g. by Gibbs sampling, and then proceeding as if the data were complete.

Acknowledgment

I thank Gerald Dalley for his help with Section 2.7.

References


Notes


4Dalley, Freeman, & Marks, 2004, Intl. Conf. on Image Proc., Fig. 2, https://pdfs.semanticscholar.org/31f7/c871ec632db5ea04df0dd7708dc16ac6a5a.pdf

5Dalley, Freeman, & Marks, 2004, Intl. Conf. on Image Proc., Fig. 4, https://pdfs.semanticscholar.org/31f7/c871ec632db5ea04df0dd7708dc16ac6a5a.pdf