Bayesian Networks and Decision Graphs

Chapter 6

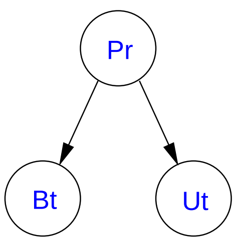
Learning probabilities from a database

We have:

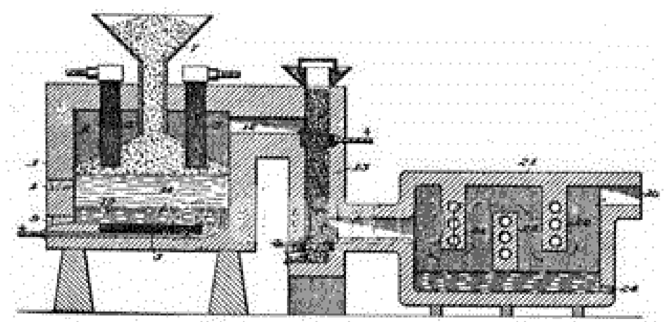
- A Bayesian network structure.
- A database of cases over (some of) the variables.

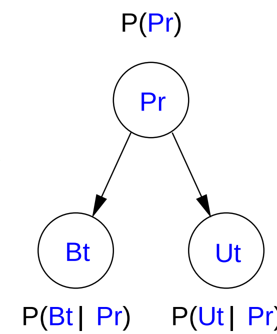
We want:

- A Bayesian network model (with probabilities) representing the database.



Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

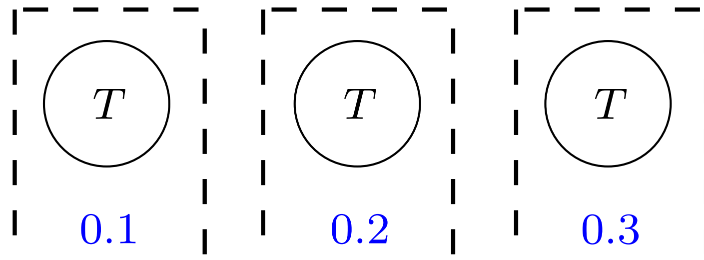




Complete data: Maximum likelihood estimation

We have tossed a thumb tack 100 times. It has landed pin up 80 times, and we now look for the model that best fits the observations/data:

Structure ....



Probability, $P($pin up$) =$

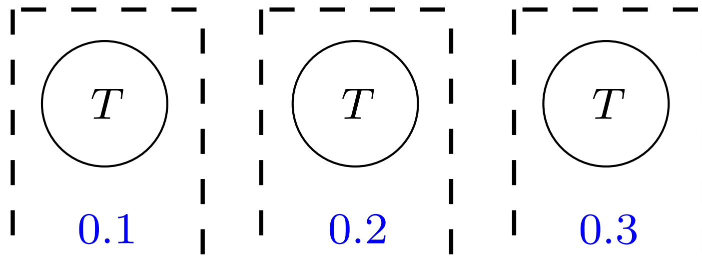
Model .... $M_{0.1}$ $M_{0.2}$ $M_{0.3}$ ....

Complete data: Maximum likelihood estimation

We have tossed a thumb tack 100 times. It has landed pin up 80 times, and we now look for the model that best fits the observations/data:

Structure ....

Probability, $P(\text{pin up}) =$



....

Model .... $M_{0.1}$ $M_{0.2}$ $M_{0.3}$ ....

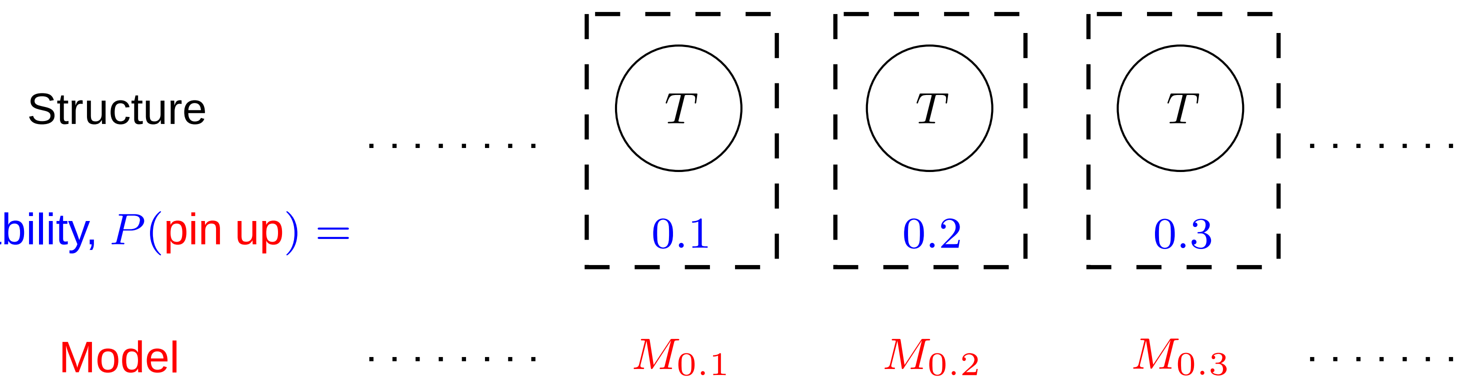
We can measure how well a model fits the data using:

$$P(\mathcal{D}|M_\theta) = P(\text{pin up}, \text{pin up}, \text{pin down}, \ldots, \text{pin up}|M_\theta)$$
$$= P(\text{pin up}|M_\theta)P(\text{pin up}|M_\theta)P(\text{pin down}|M_\theta)\cdot\ldots\cdot P(\text{pin up}|M_\theta)$$

$\boxed{\text{This is also called the likelihood of } M_\theta \text{ given } \mathcal{D}.}$

Complete data: Maximum likelihood estimation

We have tossed a thumb tack 100 times. It has landed pin up 80 times, and we now look for the model that best fits the observations/data:



We select the parameter $\hat{\theta}$ that maximizes:

$$
\begin{aligned}
\hat{\theta} &= \arg\max_{\theta} P(\mathcal{D}|M_{\theta}) \\
&= \arg\max_{\theta} \prod_{i=1}^{100} P(d_i|M_{\theta}) \\
&= \arg\max_{\theta} \mu \cdot \theta^{80}(1-\theta)^{20}.
\end{aligned}
$$

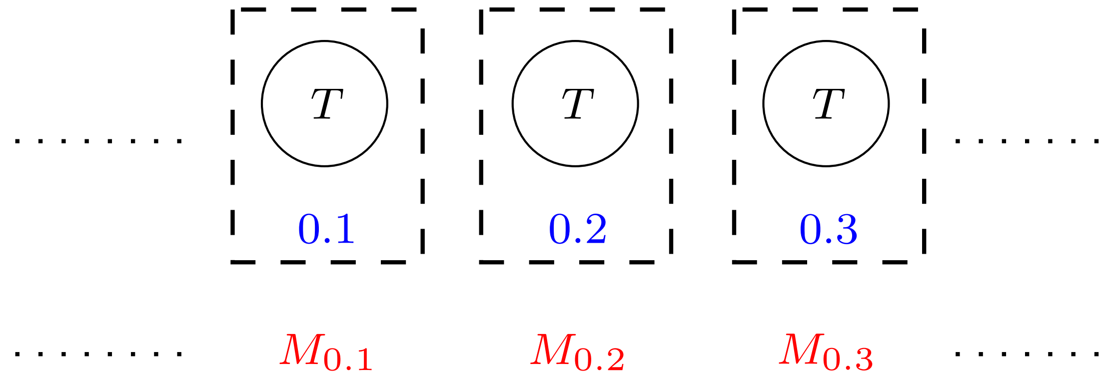
Complete data: Maximum likelihood estimation

We have tossed a thumb tack 100 times. It has landed pin up 80 times, and we now look for the model that best fits the observations/data:

Structure

Probability, $P(\text{pin up}) =$

Model



By setting:

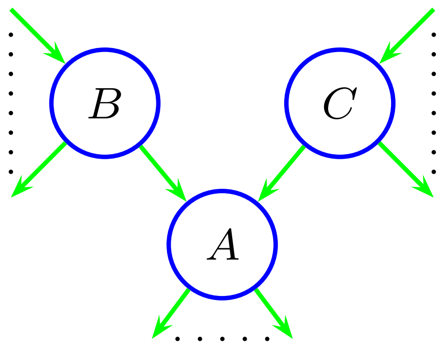
$$\frac{d}{d\theta}\mu \cdot \theta^{80}(1-\theta)^{20} = 0$$

we get the maximum likelihood estimate:

$$\hat{\theta} = 0.8.$$

Complete data: maximum likelihood estimation

In general, you get a maximum likelihood estimate as the fraction of counts over the total number of counts.



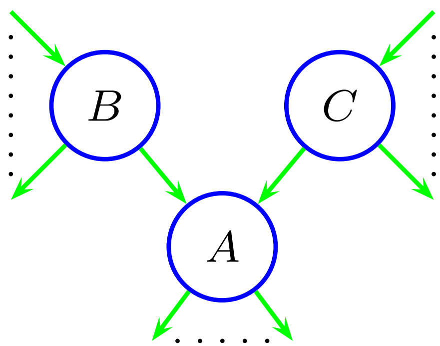
We want $P(A = a \mid B = b, C = c)$!

To find the maximum likelihood estimate $\hat{P}(A = a \mid B = b, C = c)$ we simply calculate:

$$\hat{P}(A = a \mid B = b, C = c) =$$

Complete data: maximum likelihood estimation

In general, you get a maximum likelihood estimate as the fraction of counts over the total number of counts.



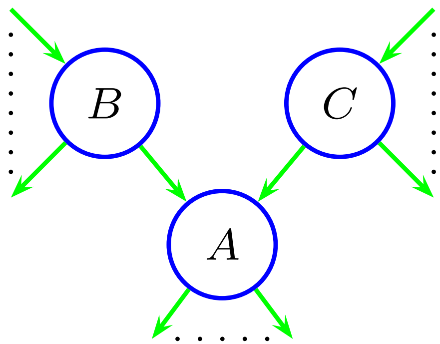
We want $P(A = a \mid B = b, C = c)$!

To find the maximum likelihood estimate $\hat{P}(A = a \mid B = b, C = c)$ we simply calculate:

$$\hat{P}(A = a \mid B = b, C = c) = \frac{\hat{P}(A = a, B = b, C = c)}{\hat{P}(B = b, C = c)}$$

Complete data: maximum likelihood estimation

In general, you get a maximum likelihood estimate as the fraction of counts over the total number of counts.



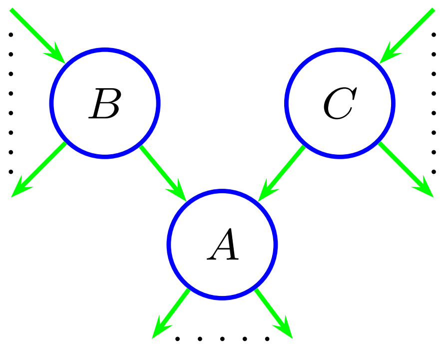
We want $P(A = a \mid B = b, C = c)$!

To find the maximum likelihood estimate $\hat{P}(A = a \mid B = b, C = c)$ we simply calculate:

$$\hat{P}(A = a \mid B = b, C = c) = \frac{\hat{P}(A = a, B = b, C = c)}{\hat{P}(B = b, C = c)} = \frac{\left[\frac{N(A=a, B=b, C=c)}{N}\right]}{\left[\frac{N(B=b, C=c)}{N}\right]}$$

Complete data: maximum likelihood estimation

In general, you get a maximum likelihood estimate as the fraction of counts over the total number of counts.



We want $P(A = a \mid B = b, C = c)$!

To find the maximum likelihood estimate $\hat{P}(A = a \mid B = b, C = c)$ we simply calculate:

$$\hat{P}(A = a \mid B = b, C = c) = \frac{\hat{P}(A = a, B = b, C = c)}{\hat{P}(B = b, C = c)} = \frac{\left[\frac{N(A=a, B=b, C=c)}{N}\right]}{\left[\frac{N(B=b, C=c)}{N}\right]}$$

$$= \frac{N(A = a, B = b, C = c)}{N(B = b, C = c)}.$$

So we have a simple counting problem!

Complete data: maximum likelihood estimation

Unfortunately, maximum likelihood estimation has a drawback:

		Last three letters							
		aaa	aab	aba	abb	baa	bba	bab	bbb
First two letters	aa	2	2	2	2	5	7	5	7
	ab	3	4	4	4	1	2	0	2
	ba	0	1	0	0	3	5	3	5
	bb	5	6	6	6	2	2	2	2

By using this table to estimate e.g. $P(T_1 = b, T_2 = a, T_3 = T_4 = T_5 = a)$ we get:

$$\hat{P}(T_1 = b, T_2 = a, T_3 = T_4 = T_5 = a) = \frac{N(T_1 = b, T_2 = a, T_3 = T_4 = T_5 = a)}{N} = 0$$

This is not reliable!

Complete data: maximum likelihood estimation

An even prior distribution corresponds to adding a virtual count of 1:

		Last three letters							
		aaa	aab	aba	abb	baa	bba	bab	bbb
First two letters	aa	2	2	2	2	5	7	5	7
	ab	3	4	4	4	1	2	0	2
	ba	0	1	0	0	3	5	3	5
	bb	5	6	6	6	2	2	2	2

From this table we get:

$$
\begin{array}{cc|cc}
 & & \multicolumn{2}{c}{T_1} \\
 & & a & b \\
\hline
T_2 & a & 32 & 17 \\
 & b & 20 & 31
\end{array}
\Rightarrow
\begin{array}{cc|cc}
 & & \multicolumn{2}{c}{T_1} \\
 & & a & b \\
\hline
T_2 & a & 32+1 & 17+1 \\
 & b & 20+1 & 31+1
\end{array}
\Rightarrow
\begin{array}{cc|cc}
 & & \multicolumn{2}{c}{T_1} \\
 & & a & b \\
\hline
T_2 & a & \left(\frac{33}{54}\right) & \left(\frac{18}{50}\right) \\
 & b & \left(\frac{21}{54}\right) & \left(\frac{32}{50}\right)
\end{array}
$$

$N(T_1, T_2)$ $\qquad$ $N'(T_1, T_2)$ $\qquad$ $P(T_2 \mid T_1) = \frac{N'(T_1, T_2)}{N'(T_1)}$

Incomplete data

How do we handle cases with missing values:

- ➤ Faulty sensor readings.
- ➤ Values have been intentionally removed.
- ➤ Some variables may be unobservable.

$\boxed{\text{Why don't we just throw away the cases with missing values?}}$

Incomplete data

How do we handle cases with missing values:

- Faulty sensor readings.
- Values have been intentionally removed.
- Some variables may be unobservable.

Why don't we just throw away the cases with missing values?

A	B	A	B
a_1	b_1	a_2	b_1
a_1	b_1	a_2	b_1
a_1	b_1	a_2	b_1
a_1	b_1	a_2	b_1
a_1	b_1	a_2	b_1
a_1	b_1	a_2	?
a_1	b_1	a_2	?
a_1	b_1	a_2	?
a_1	b_1	a_2	?
a_1	b_1	a_2	?

$\Rightarrow$

Using the entire database:

$$\hat{P}(a_1) = \frac{N(a_1)}{N(a_1) + N(a_2)} = \frac{10}{10 + 10} = 0.5.$$

Having removed the cases with missing values:

$$\hat{P}'(a_1) = \frac{N'(a_1)}{N'(a_1) + N'(a_2)} = \frac{10}{10 + 5} = 2/3.$$

How is the data missing?

We need to take into account how the data is missing:

Missing completely at random The probability that a value is missing is independent of both the observed and unobserved values.

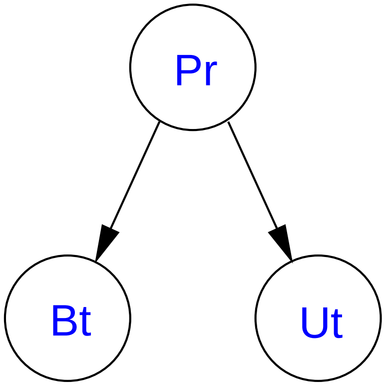
Missing at random The probability that a value is missing depends only on the observed values.

Non-ignorable Neither MAR nor MCAR.

What is the type of missingness:

- In an exit poll, where an extreme right-wing party is running for parlament?
- In a database containing the results of two tests, where the second test has only performed (as a “backup test”) when the result of the first test was negative?
- In a monitoring system that is not completely stable and where some sensor values are not stored properly?

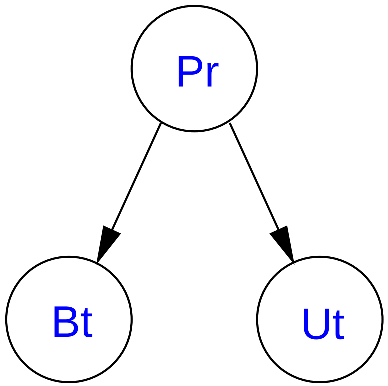
The EM algorithm



Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

Estimate the required probability distributions for the network

The EM algorithm



Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

If the database was complete we would estimate the required probabilities, $P(\text{Pr})$, $P(\text{Ut} \mid \text{Pr})$ and $P(\text{Bt} \mid \text{Pr})$ as:

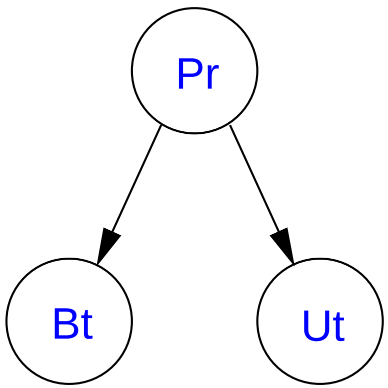
$$P(\text{Pr} = \text{yes}) = \frac{N(\text{Pr} = \text{yes})}{N}$$

$$P(\text{Ut} = \text{yes} \mid \text{Pr} = \text{yes}) = \frac{N(\text{Ut} = \text{yes}, \text{Pr} = \text{yes})}{N(\text{Pr} = \text{yes})}$$

$$P(\text{Bt} = \text{yes} \mid \text{Pr} = \text{no}) = \frac{N(\text{Bt} = \text{yes}, \text{Pr} = \text{no})}{N(\text{Pr} = \text{no})}$$

So estimating the probabilities is basically a counting problem!

The EM algorithm



Cases	Pr	Bt	Ut
$1.$	?	pos	pos
$2.$	yes	neg	pos
$3.$	yes	pos	?
$4.$	yes	pos	neg
$5.$	?	neg	?

Estimate $P(\text{Pr})$ from the database above:

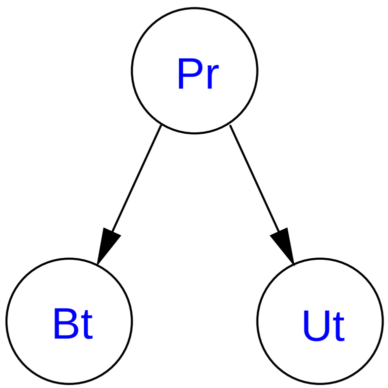
Case 2, 3 and 4 contributes with a value 1 to $N(\text{Pr} = \text{yes})$, but what is the contribution from case 1 and 5?

- Case 1 contributes with $P(\text{Pr} = \text{yes} | \text{Bt} = \text{pos}, \text{Ut} = \text{pos})$.
- Case 5 contributes with $P(\text{Pr} = \text{yes} | \text{Bt} = \text{neg})$.

To find these probabilities we assume some initial distributions, $P_0(\cdot)$, have been assigned to the network.

$\boxed{\text{We are basically calculating the expectation for } N(\text{Pr} = \text{yes}) \text{, denoted } \mathbb{E}[N(\text{Pr} = \text{yes})]}$

The EM algorithm



Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

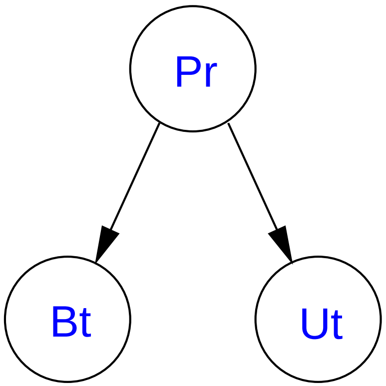
Using $P_0(\text{Pr}) = (0.5, 0.5)$, $P_0(\text{Bt} \,|\, \text{Pr} = \text{yes}) = (0.5, 0.5)$ etc., as starting distributions we get:

$$\begin{aligned}\mathbb{E}[N(\text{Pr} = \text{yes})] &= P_0(\text{Pr} = \text{yes} \,|\, \text{Bt} = \text{Ut} = \text{pos}) + 1 + 1 + 1 + P_0(\text{Pr} = \text{yes} \,|\, \text{Bt} = \text{neg}) \\ &= 0.5 + 1 + 1 + 1 + 0.5 = 4 \\ \mathbb{E}[N(\text{Pr} = \text{no})] &= P_0(\text{Pr} = \text{no} \,|\, \text{Bt} = \text{Ut} = \text{pos}) + 0 + 0 + 0 + P_0(\text{Pr} = \text{no} \,|\, \text{Bt} = \text{neg}) \\ &= 0.5 + 0 + 0 + 0 + 0.5 = 1\end{aligned}$$

So we e.g. get:

$$\hat{P}_1(\text{Pr} = \text{yes}) = \frac{\mathbb{E}[N(\text{Pr} = \text{yes})]}{N} = \frac{4}{5} = 0.8$$

The EM algorithm



Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

To estimate $\hat{P}_1(\text{Ut} \mid \text{Pr}) = \mathbb{E}[N(\text{Ut}, \text{Pr})] / \mathbb{E}[N(\text{Pr})]$ we e.g. need:

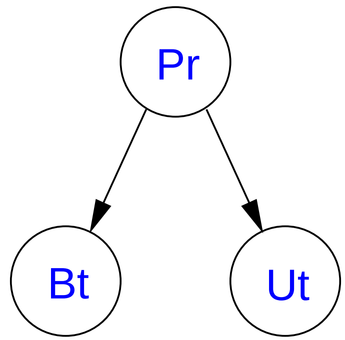
$$\begin{aligned}
\mathbb{E}[N(\text{Ut} = \text{p}, \text{Pr} = \text{y})] =& P_0(\text{Ut} = \text{p}, \text{Pr} = \text{y} \mid \text{Bt} = \text{Ut} = \text{p}) + 1 + P_0(\text{Ut} = \text{p}, \text{Pr} = \text{y} \mid \text{Bt} = \text{p}, \text{Pr} = \text{y} \\
& + 0 + P_0(\text{Ut} = \text{p}, \text{Pr} = \text{y} \mid \text{Bt} = \text{n}) = 0.5 + 1 + 0.5 + 0 + 0.25 = 2.25 \\
\mathbb{E}[N(\text{Pr} = \text{yes})] =& P_0(\text{Pr} = \text{yes} \mid \text{Bt} = \text{Ut} = \text{pos}) + 1 + 1 + 1 + P_0(\text{Pr} = \text{yes} \mid \text{Bt} = \text{neg}) \\
=& 0.5 + 1 + 1 + 1 + 0.5 = 4
\end{aligned}$$

So we e.g. get:

$$\hat{P}_1(\text{Ut} = \text{pos} \mid \text{Pr} = \text{yes}) = \frac{\mathbb{E}[N(\text{Ut} = \text{p}, \text{Pr} = \text{y})]}{\mathbb{E}[N(\text{Pr} = \text{yes})]} = \frac{2.25}{4} = 0.5625$$

The EM algorithm

$P_0(Pr) = (0.5, 0.5)$
$P_0(Ut = \text{pos} \mid Pr) = (0.5, 0.5)$
$P_0(Bt = \text{pos} \mid Pr) = (0.5, 0.5)$

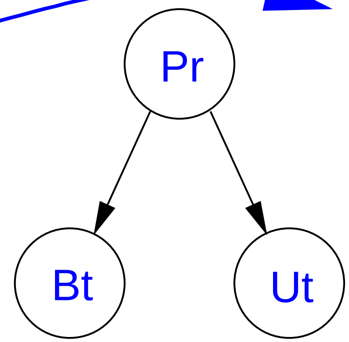


Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

The EM algorithm

E-step 1

$$P_0(Pr) = (0.5, 0.5)$$
$$P_0(Ut = \text{pos} \mid Pr) = (0.5, 0.5)$$
$$P_0(Bt = \text{pos} \mid Pr) = (0.5, 0.5)$$



$$\mathbb{E}_0[N(Pr)] = (4, 1)$$
$$\mathbb{E}_0[N(Ut = \text{pos}, Pr)] = (2.25,$$
$$0.5 + 0 + 0 + 0 + 0.25)$$
$$\mathbb{E}_0[N(Bt = \text{pos}, Pr)] = (0.5 + 0 + 1 + 1 + 0 = 2.5$$
$$, 0.5 + 0 + 0 + 0 + 0 = 0.$$

Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

The EM algorithm

E-step 1

$\mathbb{E}_0[N(Pr)] = (4, 1)$

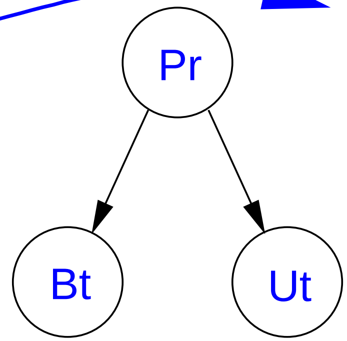
$\mathbb{E}_0[N(Ut = \text{pos}, Pr)] = (2.25,$
$0.5 + 0 + 0 + 0 + 0.25)$

$\mathbb{E}_0[N(Bt = \text{pos}, Pr)] = (0.5 + 0 + 1 + 1 + 0 = 2.5$
$, 0.5 + 0 + 0 + 0 + 0 = 0.$

$P_0(Pr) = (0.5, 0.5)$

$P_0(Ut = \text{pos} \mid Pr) = (0.5, 0.5)$

$P_0(Bt = \text{pos} \mid Pr) = (0.5, 0.5)$



M-step 2

$P_1(Pr) = (\frac{4}{5}, \frac{1}{5})$

$P_1(Ut = \text{pos} \mid Pr) = (\frac{2.25}{4}, \frac{0.75}{1})$

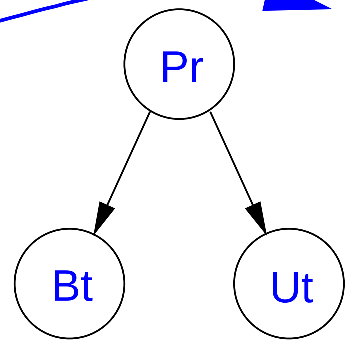
$P_1(Bt = \text{pos} \mid Pr) = (\frac{2.5}{4}, \frac{0.5}{1})$

Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

The EM algorithm

E-step 1

$P_0(Pr) = (0.5, 0.5)$
$P_0(Ut = \text{pos} \mid Pr) = (0.5, 0.5)$
$P_0(Bt = \text{pos} \mid Pr) = (0.5, 0.5)$



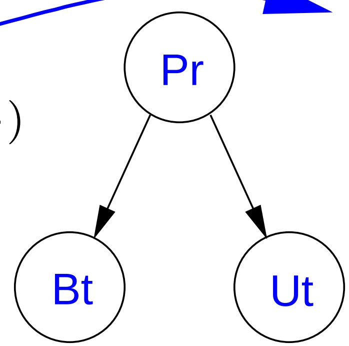
$\mathbb{E}_0[N(Pr)] = (4, 1)$
$\mathbb{E}_0[N(Ut = \text{pos}, Pr)] = (2.25,$
$0.5 + 0 + 0 + 0 + 0.25)$

$\mathbb{E}_0[N(Bt = \text{pos}, Pr)] = (0.5 + 0 + 1 + 1 + 0 = 2.5$
$, 0.5 + 0 + 0 + 0 + 0 = 0.$

M-step 2

E-step 3

$P_1(Pr) = (\frac{4}{5}, \frac{1}{5})$
$P_1(Ut = \text{pos} \mid Pr) = (\frac{2.25}{4}, \frac{0.75}{1})$
$P_1(Bt = \text{pos} \mid Pr) = (\frac{2.5}{4}, \frac{0.5}{1})$



$\mathbb{E}_1[N(Pr)] = (\quad , \quad)$
$\mathbb{E}_1[N(Ut = \text{pos}, Pr)] = (\quad , \quad)$
$\mathbb{E}_1[N(Bt = \text{pos}, Pr)] = (\quad , \quad)$

Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

The EM algorithm

E-step 1

$\mathbb{E}_0[N(Pr)] = (4, 1)$

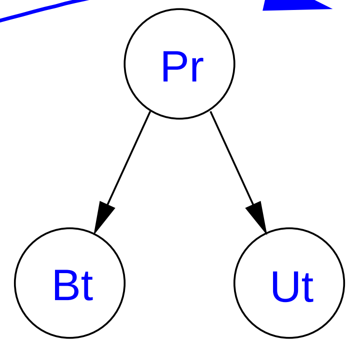
$\mathbb{E}_0[N(Ut = \text{pos}, Pr)] = (2.25,$
$\quad 0.5 + 0 + 0 + 0 + 0.25)$

$\mathbb{E}_0[N(Bt = \text{pos}, Pr)] = (0.5 + 0 + 1 + 1 + 0 = 2.5$
$\quad , 0.5 + 0 + 0 + 0 + 0 = 0.$

$P_0(Pr) = (0.5, 0.5)$

$P_0(Ut = \text{pos} \mid Pr) = (0.5, 0.5)$

$P_0(Bt = \text{pos} \mid Pr) = (0.5, 0.5)$



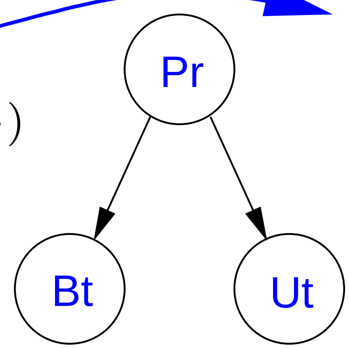
M-step 2

E-step 3

$P_1(Pr) = (\frac{4}{5}, \frac{1}{5})$

$P_1(Ut = \text{pos} \mid Pr) = (\frac{2.25}{4}, \frac{0.75}{1})$

$P_1(Bt = \text{pos} \mid Pr) = (\frac{2.5}{4}, \frac{0.5}{1})$



$\mathbb{E}_1[N(Pr)] = (\qquad , \qquad)$

$\mathbb{E}_1[N(Ut = \text{pos}, Pr)] = (\qquad , \qquad)$

$\mathbb{E}_1[N(Bt = \text{pos}, Pr)] = (\qquad , \qquad)$

M-step 4

$P_2(Pr) = (\cdot, \cdot)$

$P_2(Ut = \text{pos} \mid Pr) = (\cdot, \cdot)$

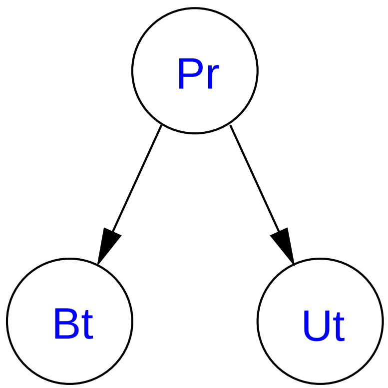
$P_2(Bt = \text{pos} \mid Pr) = (\cdot, \cdot)$

$\vdots$

Until convergence

Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

The EM algorithm



Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

1. Let $\boldsymbol{\theta}^0 = \{\theta_{ijk}\}$ be some start estimates ($P(X_i = j \mid \mathrm{pa}(X_i = k) = \theta_{ijk}$).
4. Repeat until convergence:

E-step: For each variable X_i calculate the table of expected counts:

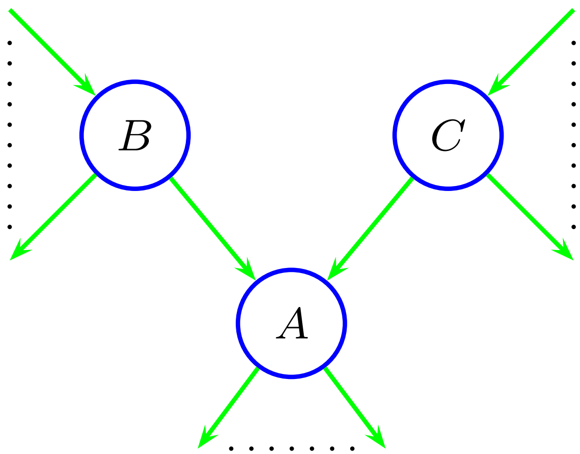
$$\mathbb{E}_{\boldsymbol{\theta}^t}[N(X_i, \mathrm{pa}(X_i) \mid \mathcal{D}] = \sum_{\mathbf{d} \in \mathcal{D}} P(X_i, \mathrm{pa}(X_i) \mid \mathbf{d}, \boldsymbol{\theta}^t).$$

M-step: Use the expected counts as if they were actual counts:

$$\hat{\theta}_{ijk} = \frac{\mathbb{E}_{\boldsymbol{\theta}^i}[N(X_i = k, \mathrm{pa}(X_i) = j \mid \mathcal{D}]}{\sum_{k=1}^{|\mathrm{sp}(X_i)|} \mathbb{E}_{\boldsymbol{\theta}^i}[N(X_i = k, \mathrm{pa}(X_i) = j \mid \mathcal{D}]}.$$

Adaptation

Adapt the tables to experience (cases):



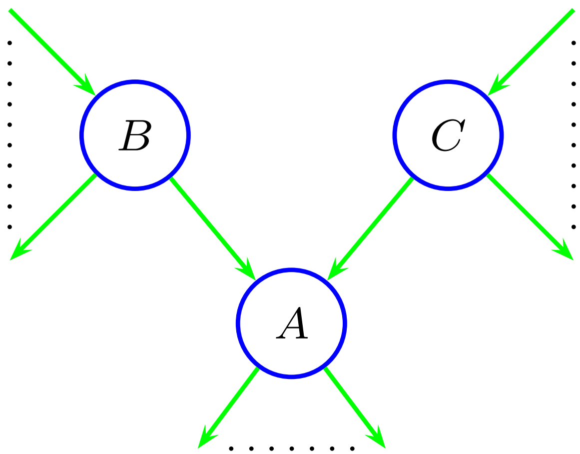
Social env. (or expert) t_1: $P_1(A|B,C)$

⋮

Social env. (or expert) t_k: $P_k(A|B,C)$

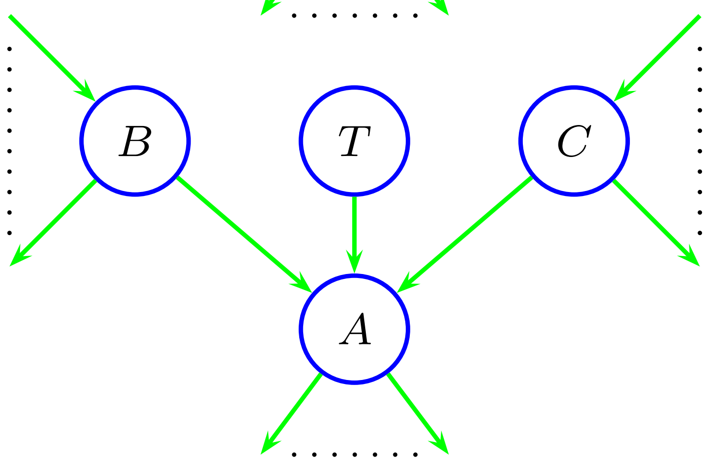
Adaptation

Adapt the tables to experience (cases):



Social env. (or expert) t_1: $P_1(A|B,C)$

Social env. (or expert) t_k: $P_k(A|B,C)$



Variable T: $t_1, \ldots, t_k$

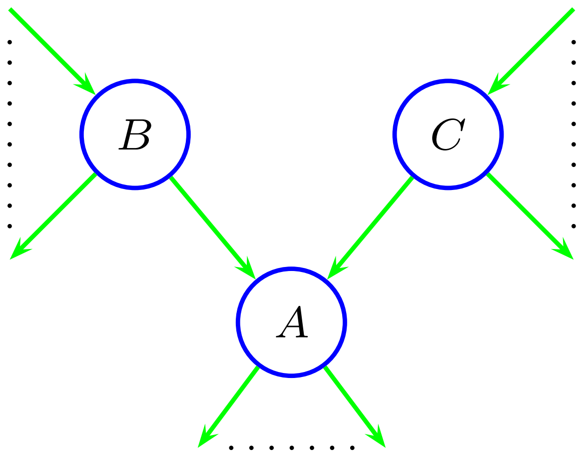
$P(T)$ reflects the credibility of $t_1, \ldots, t_k$

$P(A|B,C,T=t_i) = P_i(A|B,C)$

Any case e will yield a $P(T|e)$:

This is used as prior for the next case.

Fractional updating

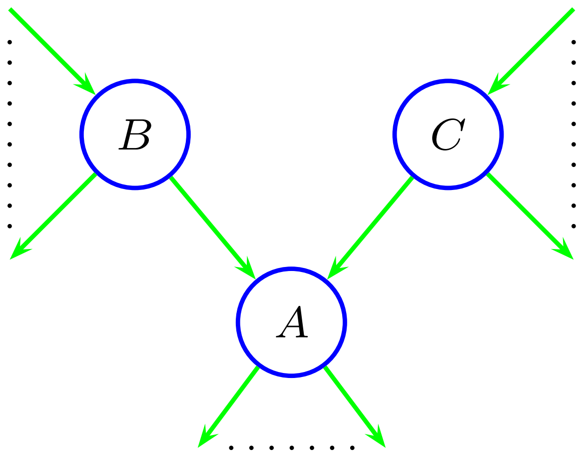


I am uncertain about $P(A|B, C)$.

Idea: I can represent my uncertainty by assuming that $P(A|b_i, c_j)$ are frequencies from a virtual sample of n cases.

- ▶The larger I put n, the more certain I am, i.e.,$P(A|b_i, c_j) = \left(\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_m}{n}\right)$.

Fractional updating



I am uncertain about $P(A|B,C)$.

Idea: I can represent my uncertainty by assuming that $P(A|b_i, c_j)$ are frequencies from a virtual sample of n cases.

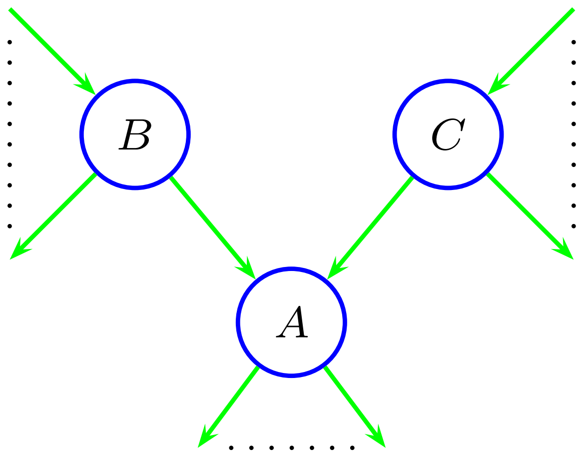
▶The larger I put n, the more certain I am, i.e.,$P(A|b_i, c_j) = \left(\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_m}{n}\right)$.

I update $P(A|b_i, c_j)$ when a new case arrives:

a) New case: $B = b_i, C = c_j, A = a_1, \ldots$:

$$P^*(A|b_i, c_j) = \left(\frac{n_1 + 1}{n+1}, \frac{n_2}{n+1}, \ldots, \frac{n_m}{n+1}\right)$$

Fractional updating



I am uncertain about $P(A|B,C)$.

Idea: I can represent my uncertainty by assuming that $P(A|b_i, c_j)$ are frequencies from a virtual sample of n cases.

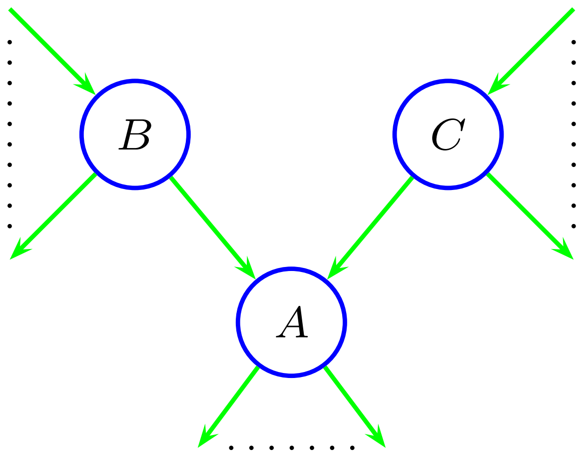
▶The larger I put n, the more certain I am, i.e.,$P(A|b_i, c_j) = \left(\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_m}{n}\right)$.

I update $P(A|b_i, c_j)$ when a new case arrives:

b) New case: $B = b_i, C = c_j, \ldots$. and $P(A|case) = (x_1, \ldots, x_m)$:

$$P^*(A|b_i, c_j) = \left(\frac{n_1 + x_1}{n+1}, \frac{n_2 + x_2}{n+1}, \ldots, \frac{n_m + x_m}{n+1}\right)$$

Fractional updating



I am uncertain about $P(A|B,C)$.

Idea: I can represent my uncertainty by assuming that $P(A|b_i,c_j)$ are frequencies from a virtual sample of n cases.

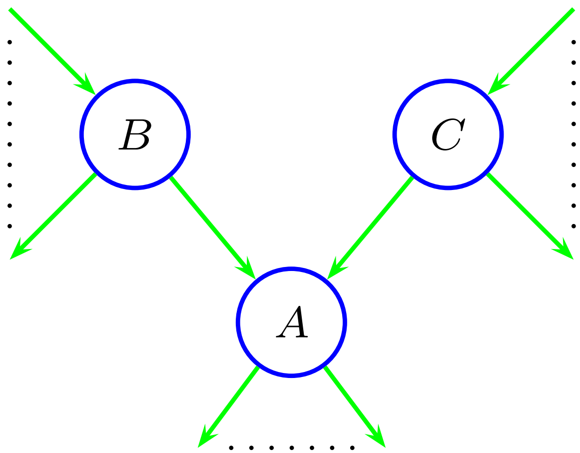
▶The larger I put n, the more certain I am, i.e.,$P(A|b_i,c_j) = \left(\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_m}{n}\right)$.

I update $P(A|b_i,c_j)$ when a new case arrives:

c) New case: $\ldots, A = a_1, \ldots.$ and $P(b_i, c_j|case) = x$:

$$P^*(A|b_i,c_j) = \left(\frac{n_1+x}{n+x}, \frac{n_2}{n+x}, \ldots, \frac{n_m}{n+x}\right)$$

Fractional updating



I am uncertain about $P(A|B,C)$.

Idea: I can represent my uncertainty by assuming that $P(A|b_i,c_j)$ are frequencies from a virtual sample of n cases.

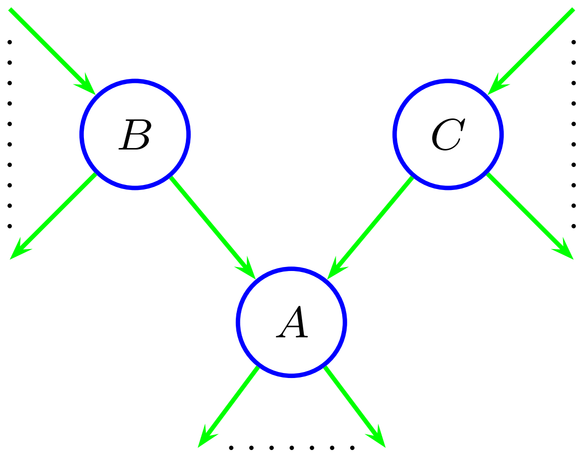
▶The larger I put n, the more certain I am, i.e.,$P(A|b_i,c_j) = (\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_m}{n})$.

I update $P(A|b_i,c_j)$ when a new case arrives:

d) New (general) case: $\ldots \Rightarrow P(A|case) = (x_1, \ldots, x_m)$ and $P(b_i, c_j|case) = x$:

$$P^*(A|b_i,c_j) = \left(\frac{n_1 + x \cdot x_1}{n+x}, \frac{n_2 + x \cdot x_2}{n+x}, \ldots, \frac{n_m + x \cdot x_m}{n+x}\right)$$

Fractional updating



I am uncertain about $P(A|B,C)$.

Idea: I can represent my uncertainty by assuming that $P(A|b_i,c_j)$ are frequencies from a virtual sample of n cases.

▶The larger I put n, the more certain I am, i.e.,$P(A|b_i,c_j) = \left(\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_m}{n}\right)$.

I update $P(A|b_i,c_j)$ when a new case arrives:

e) New case: $B = b_i, C = c_j$ and this is all!!

$$P^*(A|b_i,c_j) = \left(\frac{n_1 + \left(\frac{n_1}{n}\right)}{n+1}, \frac{n_2 + \left(\frac{n_2}{n}\right)}{n+1}, \ldots, \frac{n_m + \left(\frac{n_m}{n}\right)}{n+1} \right) = \left(\frac{n_1}{n}, \ldots, \frac{n_m}{n}\right)$$

Unjustified, we thereby confirm our belief in our present distribution.

Assumptions

What is the situation?

- We are uncertain about $P(A|B,C)$.
- We get a new case with $B = b_1$ and $C = c_2$.

When updating we have that:

- $P(A|b_1, c_2)$ is changed.
- All other $P(A|b_i, c_j)$ are unaffected.

This involves the following two assumptions:

Local independence: The (second order) uncertainty on $P(A|b_i, c_j)$ is independent of the (second order) uncertainty on $P(A|b'_i, c'_j)$.

Global independence: The (second order) uncertainty for the various variables is independent.

Example: Spoofing

Estimate: $P(\#\text{chosen}|\#\text{in-hand} = 2) = (0.2, 0.6, 0.2)$
Virtual sample size $= 20$ (corresponding to (4,12,4)).

New case: $\#$chosen$= 0$
$P(\#\text{chosen}|\#\text{in-hand} = 2) = (\frac{5}{21}, \frac{12}{21}, \frac{4}{21})$

23 new cases: $(7, 8, 8)$
$P(\#\text{chosen}|\#\text{in-hand} = 2) = (\frac{12}{44}, \frac{20}{44}, \frac{12}{44}) = (0.27, 0.46, 0.27)$

Apparently, she plays $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$!!

Do I have to take the (wrong) past with me?

We have two situations:

- The initial probabilities are wrong.
- The probabilities change over time.

Fading: Multiply the old set of counts with a fading factor $q < 1$.

$$\left(\frac{n_1}{n}, \frac{n_2}{n}, \frac{n_3}{n}\right) \bigoplus (x_1, x_2, x_3)$$

Do I have to take the (wrong) past with me?

We have two situations:

- The initial probabilities are wrong.
- The probabilities change over time.

Fading: Multiply the old set of counts with a fading factor $q < 1$.

$$\left(\frac{n_1}{n}, \frac{n_2}{n}, \frac{n_3}{n}\right) \bigoplus (x_1, x_2, x_3) \qquad \text{(with } x = \sum_i x_i\text{)}$$

Updating proceeds as follows:

The counts:

$$(n_1, n_2, n_3) \longrightarrow (n_1 \cdot q, n_2 \cdot q, n_3 \cdot q)$$

$$n \longrightarrow n \cdot q$$

The probabilities:

$$P(\cdot) = \left(\frac{n_1 \cdot q + x_1}{n \cdot q + x}, \frac{n_2 \cdot q + x_2}{n \cdot q + x}, \frac{n_3 \cdot q + x_3}{n \cdot q + x}\right)$$

This technique is very efficient for implementing adaptive agents for games with perfect recall.

Interpreting the fading factor

With the fading factor we have:

$$(n_1, n_2, n_3) \longrightarrow (n_1 \cdot q, n_2 \cdot q, n_3 \cdot q)$$
$$n \longrightarrow n \cdot q$$

If all counts will be updated with the value 1, then the past will fade away exponentially and the limit (the effective sample size) will be:

$$n^* = \frac{1}{1-q}$$

If $n = n^*$ and a new case arrives, we get:

$$n := n^* \cdot q + 1 = \frac{q}{1-q} + 1 = \frac{1}{1-q} = n^*$$

So instead of declaring a fading factor we can specify an effective sample size, and the fading factor is then:

$$q = \frac{n^* - 1}{n^*}$$