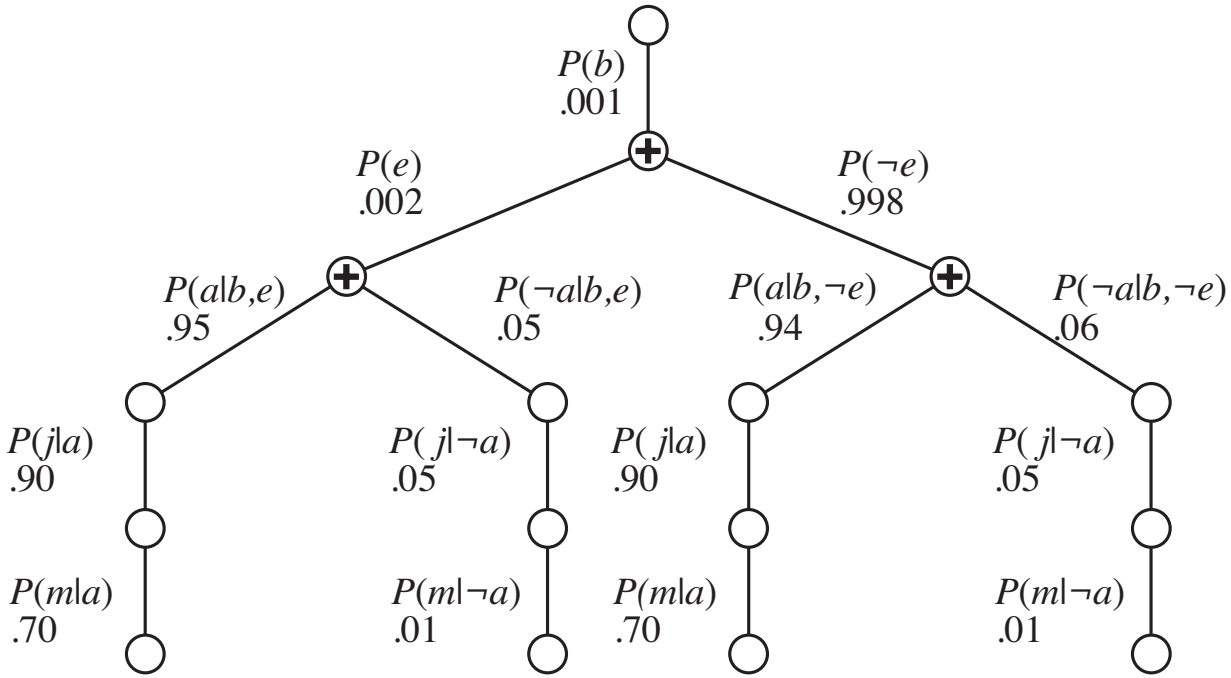
Redundancy in Probability Computation

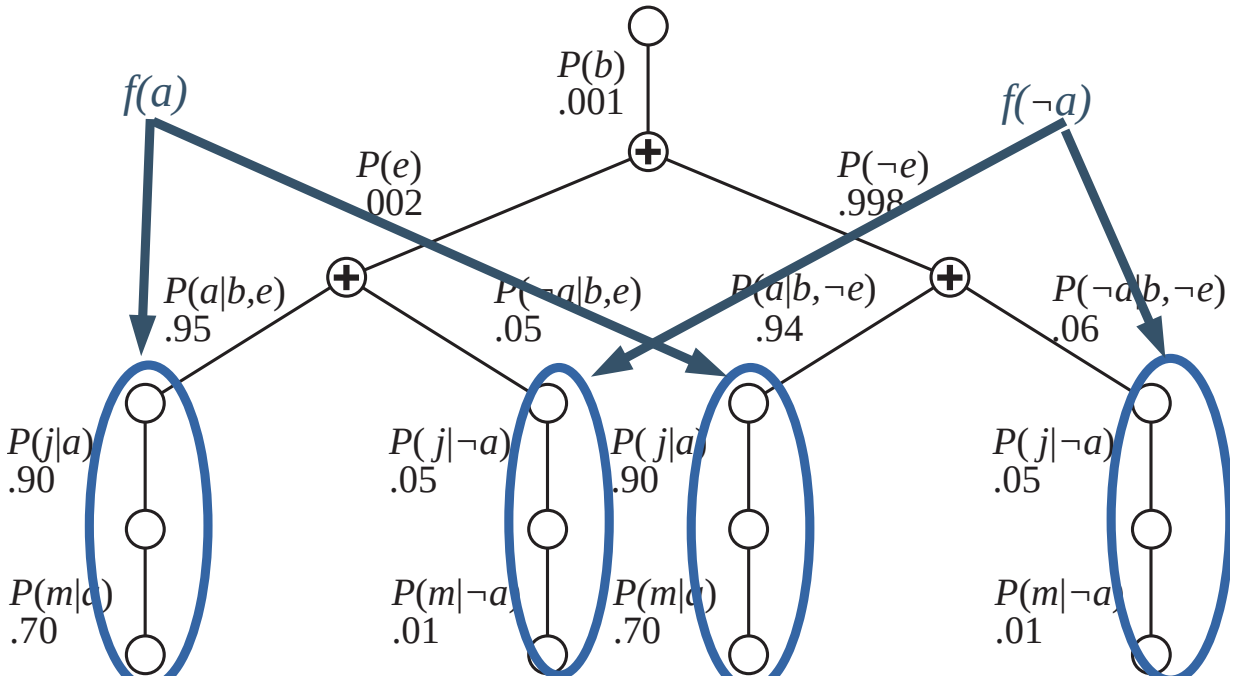
Consider $P(b \mid j, m) = \alpha P(b) \sum_E P(E) \sum_A P(A \mid b, E) P(j \mid A) P(m \mid A)$.
Observe the repeated sub-summing (branches) in its computation:



Factors

This redundancy is removed by using **factors**, which are arrays storing reusable computation results.

$$f(A) = \begin{bmatrix} f(\neg a) \\ f(a) \end{bmatrix} = \begin{bmatrix} P(j \mid \neg a)P(m \mid \neg a) \\ P(j \mid a)P(m \mid a) \end{bmatrix}$$ is an example factor.



FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Computing Probabilities with Factors

Each query generates a set **initial factors**, one for each (conditional) probability in its factorized expression. So $P(B \mid j, m)$ generates 5 initial factors:

$$P(B \mid j, m) = \alpha \underbrace{P(B)}_{f_1(B)} \sum_E \underbrace{P(E)}_{f_2(E)} \sum_A \underbrace{P(A \mid B, E)}_{f_3(A,B,E)} \underbrace{P(j \mid A)}_{f_4(A)} \underbrace{P(m \mid A)}_{f_5(A)} \quad (1)$$

They have a dimension (argument) for each *uninstantiated* (upper-case) variable and the values are given by the corresponding CPT entries. E.g.,

$$f_4(A) = \begin{bmatrix} P(j \mid \neg a) \\ P(j \mid a) \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.70 \end{bmatrix} \quad (2)$$

(Refer to the CPT shown here.)

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Computing Probabilities with Factors (cont'd)

We use the initial factors to express (1) as

$$P(B \mid j, m) = \alpha f_1(B) \times \sum_E f_2(E) \times \sum_A f_3(A, B, E) \times f_4(A) \times f_5(A) \quad (3)$$

where

- $\times$ is **factor multiplication** producing a factor from two factors (different from matrix multiplication!)
- $\sum_V$ is **variable elimination** producing a factor without the V variable from a factor with the V variable

Using these two operations, we evaluate (3) from right to left.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Factor Multiplication

Factors are multiplied **point-wise** (similar to a database *join*). Example:

$$f_3(A,B,E) \times f_4(A) =$$

$$\begin{bmatrix} 1.00 & \neg a,\neg b,\neg e \\ 0.71 & \neg a,\neg b,e \\ 0.06 & \neg a,b,\neg e \\ 0.05 & \neg a,b,e \\ 0.00 & a,\neg b,\neg e \\ 0.29 & a,\neg b,e \\ 0.94 & a,b,\neg e \\ 0.95 & a,b,e \end{bmatrix} \times \begin{bmatrix} 0.01 & \neg a \\ 0.70 & a \end{bmatrix} = \begin{bmatrix} 1.00 \cdot 0.01 \\ 0.71 \cdot 0.01 \\ 0.06 \cdot 0.01 \\ 0.05 \cdot 0.01 \\ 0.00 \cdot 0.70 \\ 0.29 \cdot 0.70 \\ 0.94 \cdot 0.70 \\ 0.95 \cdot 0.70 \end{bmatrix}$$

Blue text indicates the array indexes.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Factor Multiplication (cont'd)

For (3), we proceed as follows

$$f_6(A) = f_4(A) \times f_5(A) = \begin{bmatrix} 0.01 & \neg a \\ 0.70 & a \end{bmatrix} \times \begin{bmatrix} 0.05 & \neg a \\ 0.90 & a \end{bmatrix} = \begin{bmatrix} 0.0005 & \neg a \\ 0.63 & a \end{bmatrix}$$

(So f_6 is the example factor f from here.)

$$f_7(A,B,E) = f_3(A,B,E) \times f_6(A) = \begin{bmatrix} 1.00 & \neg a,\neg b,\neg e \\ 0.71 & \neg a,\neg b,e \\ 0.06 & \neg a,b,\neg e \\ 0.05 & \neg a,b,e \\ 0.00 & a,\neg b,\neg e \\ 0.29 & a,\neg b,e \\ 0.94 & a,b,\neg e \\ 0.95 & a,b,e \end{bmatrix} \times \begin{bmatrix} 0.0005 & \neg a \\ 0.63 & a \end{bmatrix}$$

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Variable Elimination

Now we have reduced (3) into

$$P(B \mid j, m) = \alpha f_1(B) \times \sum_E f_2(E) \times \sum_A f_7(A, B, E)$$

where $\sum_A f_7(A, B, E)$ produces a new factor $f_8(B, E)$ defined as

$$f_8(B, E) = \sum_A f_7(A, B, E) = f_7(\neg a, B, E) + f_7(a, B, E) \tag{4}$$

In general the operation $\sum_V f(V, \ldots)$ yielding a new factor without V is called **variable elimination**.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Computing Probabilities with Factors (cont'd)

Continue interleaving factor multiplication with variable elimination:

- $f_9(B, E) = f_2(E) \times f_8(B, E)$
- $f_{10}(B) = f_9(B, e) + f_9(B, \neg e)$
- $f_{11}(B) = f_1(B) \times f_{10}(B)$

Finally

$$P(B \mid j, m) = \alpha f_{11}(B) \tag{5}$$

where $\alpha = 1/(f_{11}(\neg b) + f_{11}(b))$.

Note that even with factors, complexity of inference in Bayes Networks is obviously exponential: a factor with n variables has 2^n values in it.

(Check out the first part of this exercise problem)

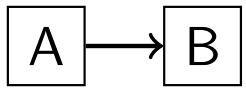
FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

MAP inference

We want to compute (??) without computing the probabilities of all combinations of values of the unobserved variables. The joint MAP state need not consist of the values maximizing their marginal probabilities!

Consider e.g.

$P(a)$
0.6



$P(b \mid A)$	A
0.90	0
0.55	1

$$\arg\max_{A} P(A) = 1$$

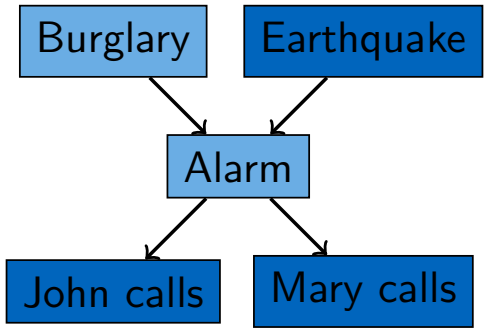
but

$$\arg\max_{A,B} P(A, B) = (0, 1)$$

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

MAP Inference (cont'd)

Principle similar to basic probability inference. Example:



First compute the maximum probability:

$$\max_{A,B} P(A, B \mid \neg e, \neg j, \neg m) =$$

$$\alpha \max_{A,B} P(B)P(\neg e)P(A \mid B, \neg e)P(\neg j \mid A)P(\neg m \mid A) =$$

$$\beta \max_{B} P(B) \max_{A} P(A \mid B, \neg e)P(\neg j \mid A)P(\neg m \mid A)$$

where $\beta = \alpha P(\neg e)$ need *not* be evaluated to compute the arg max.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

MAP Inference with Factors

Reformulate using factors:

$$\max_{A,B} P(A, B \mid \neg e, \neg j, \neg m) = \beta \max_{B} f_1(B) \times \max_{A} f_2(A, B) \times f_3(A) \times f_4(A)$$
$$= \beta \max_{B} f_1(B) \times \max_{A} f_7(A, B) = \beta \max_{B} f_8(B) \qquad (6)$$

where $\max_V f(V, \ldots)$ produces a new factor (without the V dimension) containing the maximal values over the V dimension. This is called **maximizing out** V.

The arguments $(a_{\max}, b_{\max})$ maximizing (6) are determined as

- $b_{\max} = \arg\max_B f_8(B)$
- $a_{\max} = \arg\max_A f_7(A, b_{\max})$

Note that the order of the two steps cannot be switched.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

MAP Involving Variable Elimination

Consider a case where query variables are not the full complement to evidence variables as in here.

Say we want to find the most probable state of B and E when both J and M are true and A is not observed. Now we have both maximization and summation:

$$\max_{B,E} P(B, E \mid j, m) = \alpha \max_{B,E} \sum_{A} P(B)P(E)P(A \mid B, E)P(j \mid A)P(m \mid A) \tag{7}$$

We still can push operators before the first occurrence of their arguments *but* cannot swap the order of a max operator and a $\sum$ operator since

$$\sum_{Y} \max_{X} P(X, Y) \neq \max_{X} \sum_{Y} P(X, Y)$$

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

MAP Involving Variable Elimination (cont'd)

So (7) rewrites to

$$\alpha \max_B f_1(B) \times \max_E f_2(E) \times \sum_A f_3(A, B, E) \times f_4(A) \times f_5(A) =$$

The scope of any max *is everything to the right of it!* Now multiply factors after $\sum_A$ and then eliminate A

$$\alpha \max_B f_1(B) \times \max_E f_2(E) \times \sum_A f_6(A, B, E) = \alpha \max_B f_1(B) \times \max_E f_2(E) \times f_7(B, E)$$

Multiply f_2 with f_7 and then maximize out E:

$$= \alpha \max_B f_1(B) \times \max_E f_8(B, E) = \max_B f_9(B)$$

Finally, $b_{\max} = \arg\max_B f_9(B)$, $e_{\max} = \arg\max_E f_8(b_{\max}, E)$.

(Check out the second part of this exercise problem)

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Learning a Bayes Network

We now assume that p_{k+1} is a Bayes Network with Bayes graph G and a set $\mathbf{w}$ of parameters instantiating the conditional probability tables.

A usual way to learn $p_{k+1} = (G, \mathbf{w})$ from observations $x_{\leq k}$ is to maximize the likelihood, i.e. choose a model maximizing the probability of $x_{\leq k}$:

$$p_{k+1} = \arg \max_{p=(G,\mathbf{w})} p(x_{\leq k})$$

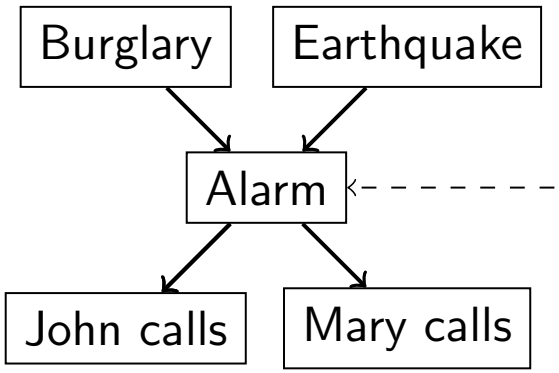
We will only consider the scenario where G is given (i.e. is agent's background knowledge) and the agent only learns $\mathbf{w}$.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Estimation of CPT Parameters of a Bayes Network

CPT parameters can be estimated using the EM algorithm as in here, except in Step **??** (M step), we estimate the conditional probabilities of the CPT by maximizing the likelihood.

Example of the M step at $k = 7$ after missing data values have been replaced by their expectations (E step) according to the current model:



$P(a \mid B, E)$	B	E
	0	0
$p(a \mid \neg b, e)$	0	1
	1	0
	1	1

	B	E	A	J	M
x_1	0	1	1	0	0
x_2	1	1	1	1	1
x_3	0	1	1	1	0
x_4	0	1	0	0	0
x_5	1	0	0	0	0
x_6	1	0	1	1	1

Here, $p_7(a \mid \neg b, e) := 2/3$ (relative frequency maximizes likelihood.)

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Fast Estimation of CPT Parameters

The previous approach has high complexity due to:

1. iterating the EM steps until convergence
2. performing MAP inference for every missing value in each E iteration
3. the need to store all of $x_{\leq k}$ in agent's memory

(1) and (2) can be avoided by estimating the CPT probabilities only from the subset of $x_{\leq k}$ where the needed components are not missing.

p_k will then converge slower to P in terms of k but may converge faster in terms of runtime.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Fast Estimation of CPT Parameters (cont'd)

For example, with observations

	B	E	A	J	M
x_1	0	1	0	0	0
x_2	?	1	1	1	1
x_3	0	1	1	1	0
x_4	0	1	?	?	0
x_5	1	0	0	0	?
x_6	0	1	0	1	1

estimate $p_6(a \mid \neg b, e) := 1/3$.

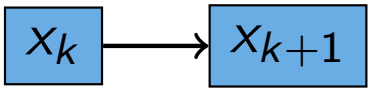
Finally, (3) is also prevented: rather than remembering all of $x_{\leq k}$, update the estimate by the cumulative moving average rule, only keeping the count K of observations used so far for the estimate. Here:

$$p_6(a \mid \neg b, e) = p_5(a \mid \neg b, e) + \frac{A_6 - p_5(a \mid \neg b, e)}{K + 1} = \frac{1}{2} + \frac{0 - \frac{1}{2}}{3} = 1/3$$

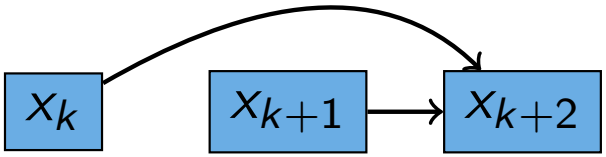
FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Temporal Bayesian Networks

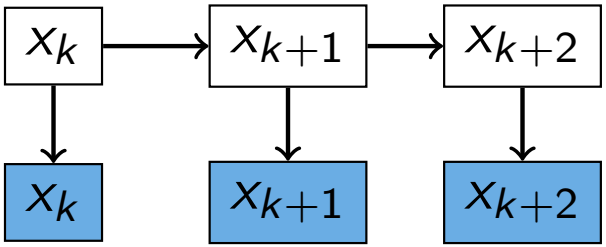
Bayesian networks generalize some well known temporal models.



Markov process (1^{st} order)



Markov process (2^{nd} order)

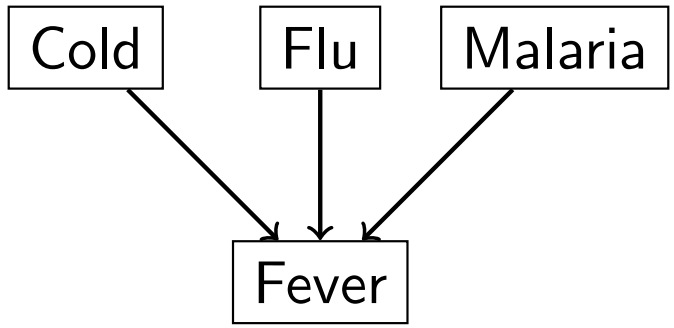


Markov hidden process (1^{st} order, 3 observations)

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

BN Encoding Logical Rules

Observe that Bayesian networks can encode propositional logic formulas. Example:



$P(\text{fever} \mid C, F, M)$	C	F	M
1	1	1	1
0	all other cases		

$$\text{fever} \leftarrow \text{cold} \wedge \text{flu} \wedge \text{malaria}$$

$P(\text{fever} \mid C, F, M)$	C	F	M
0	0	0	0
1	all other cases		

$$\text{fever} \leftarrow \text{cold} \vee \text{flu} \vee \text{malaria}$$

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Statistical Relational Learning

Bayes networks extend propositional logic by enabling to express probabilistic dependencies.

First-order logic (FOL) extends propositional logic by enabling to express structural (relational) dependencies.

These aspects are combined in the formalism of *Bayes Logic Programs* studied in the field called Statistical Relational Learning (SRL).

In SRL, also other probabilistic graphical models (e.g., Markov networks) are endowed with FOL expressivness (Markov logic networks).

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE