Bayesian Networks Part 1

CS 760@UW-Madison

Goals for the lecture

you should understand the following concepts

- the Bayesian network representation
- inference by enumeration

- the parameter learning task for Bayes nets
- the structure learning task for Bayes nets
- maximum likelihood estimation
- Laplace estimates
- *m*-estimates

Bayesian network example

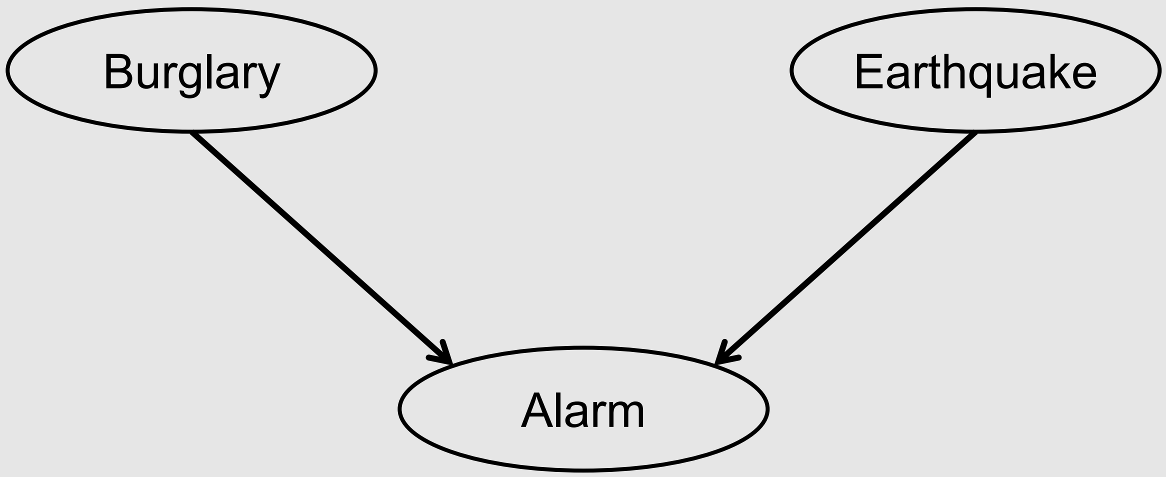
- Consider the following 5 binary random variables:

  B = a burglary occurs at the house
  E = an earthquake occurs at the house
  A = the alarm goes off
  J = John calls to report the alarm
  M = Mary calls to report the alarm

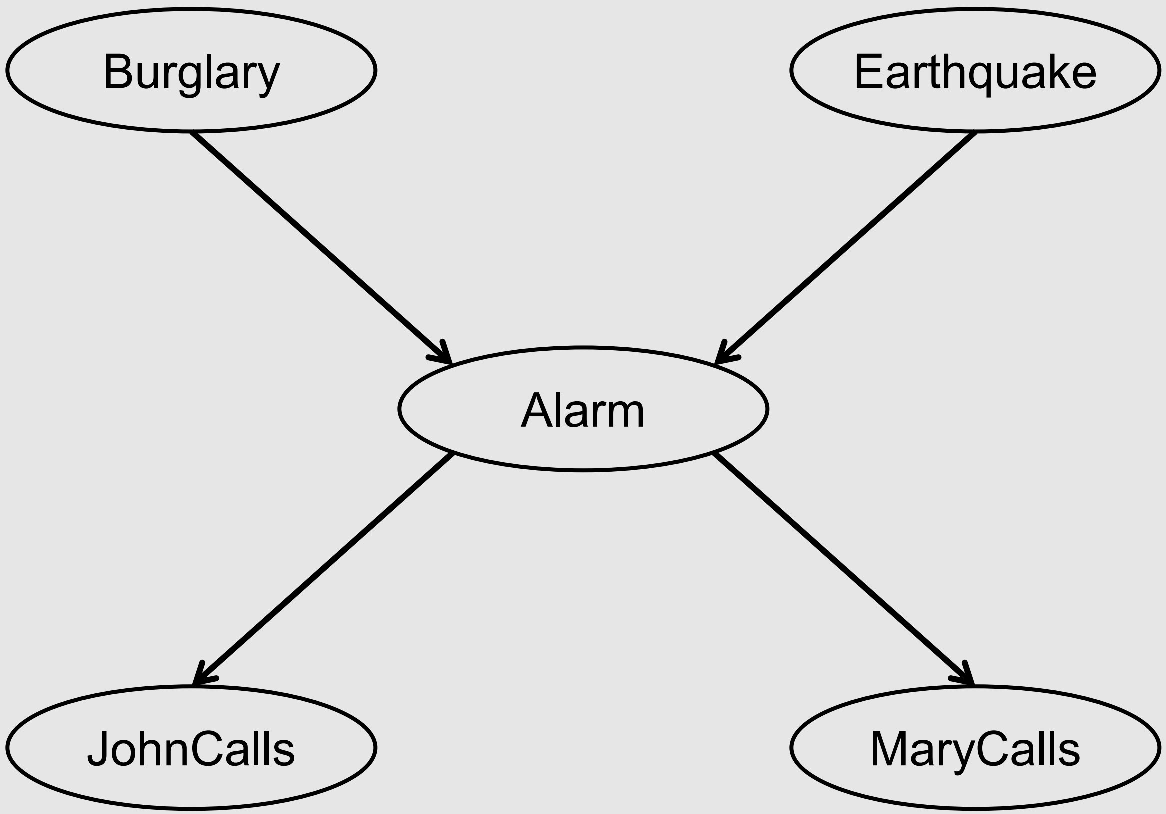
- Suppose Burglary or Earthquake can trigger Alarm, and Alarm can trigger John's call or Mary's call

- Now we want to answer queries like what is $P(B \mid M, J)$?

Bayesian network example



Bayesian network example



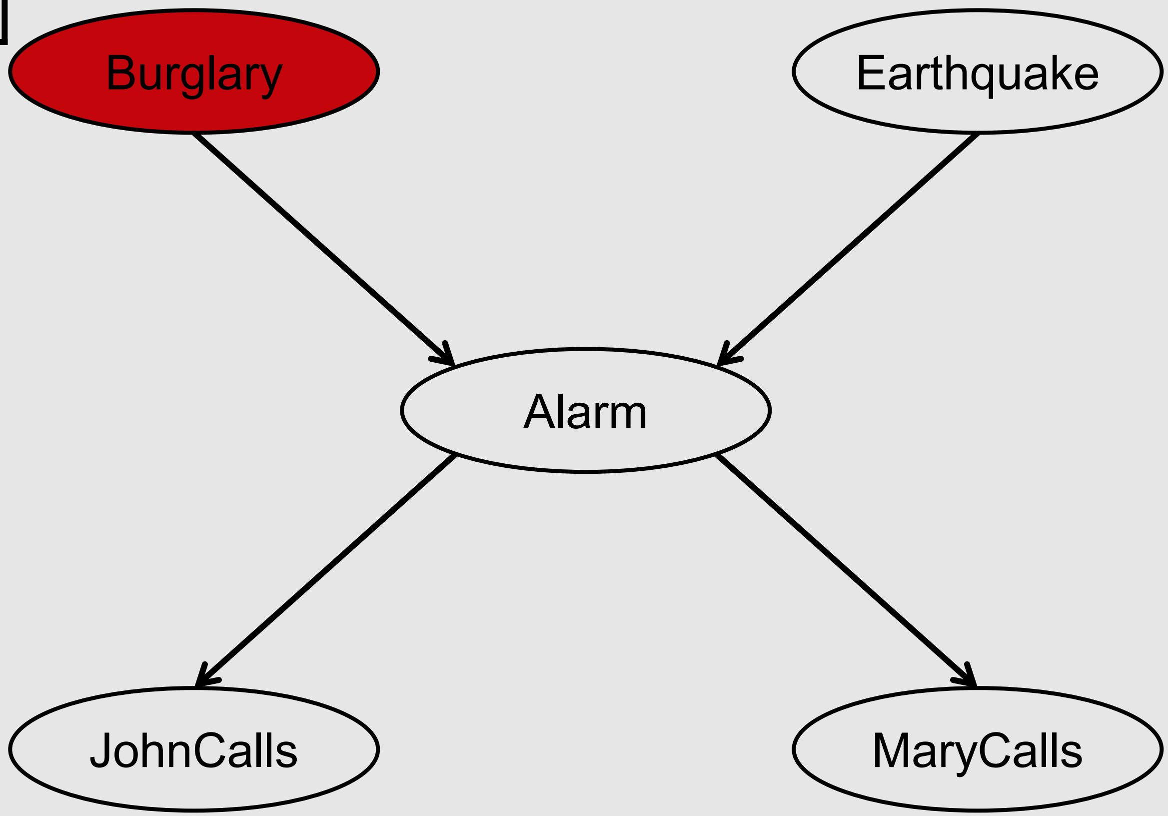
Bayesian network example

P (*B*)

t	f
0.001	0.999

P (*E*)

t	f
0.001	0.999



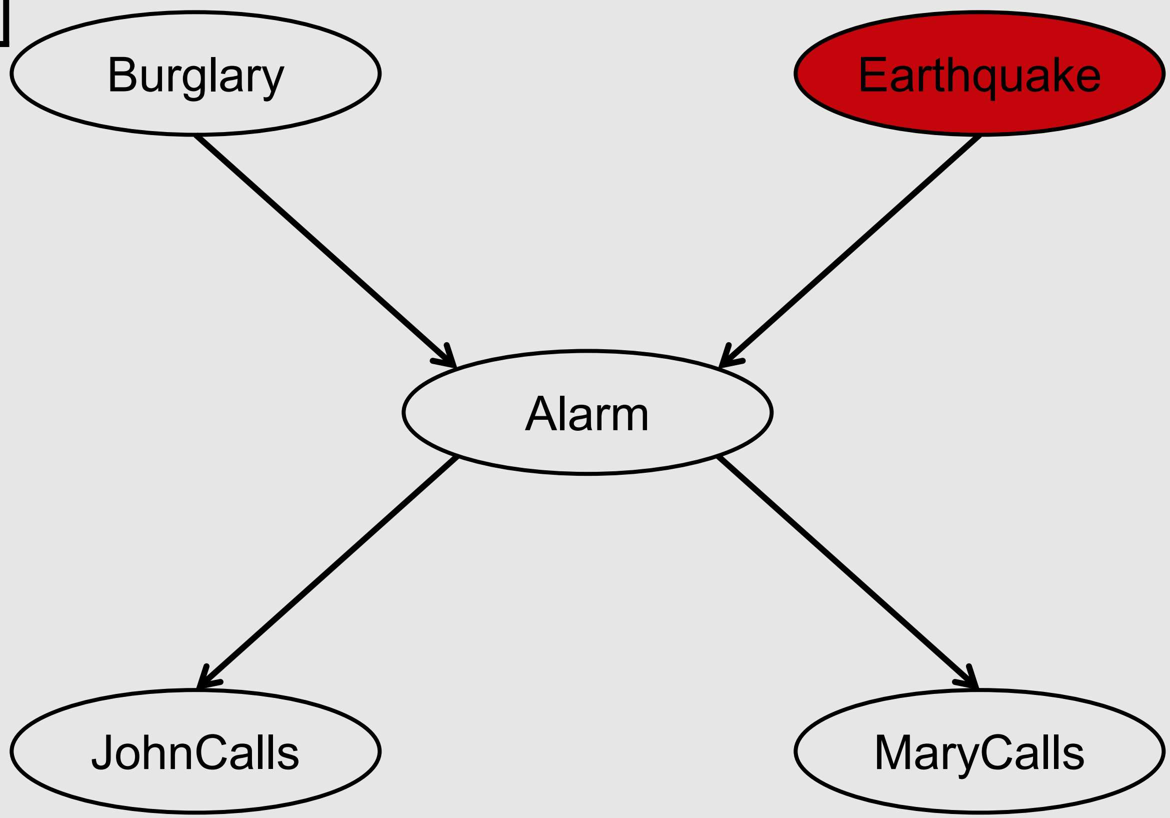
Bayesian network example

P (*B*)

t	f
0.001	0.999

P (*E*)

t	f
0.001	0.999



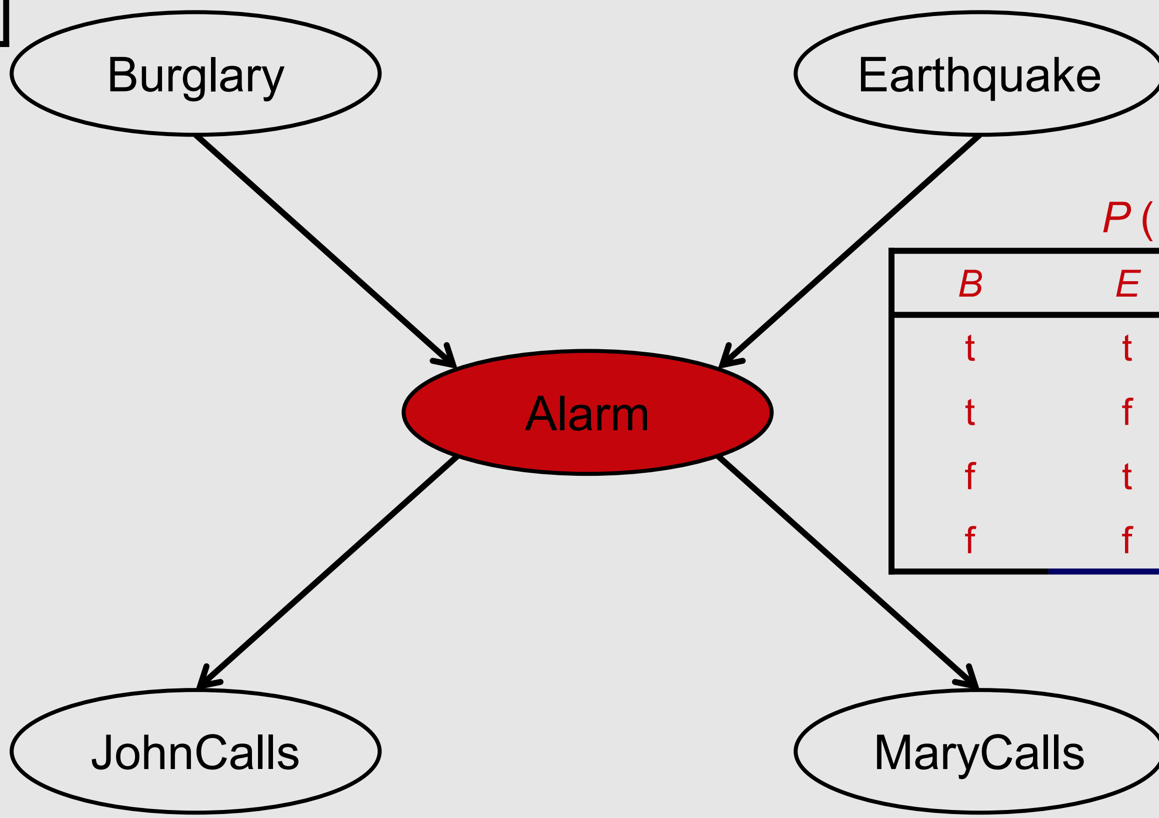
Bayesian network example

P (*B*)

t	f
0.001	0.999

P (*E*)

t	f
0.001	0.999



P (*A* | *B*, *E*)

B	*E*	t	f
t	t	0.95	0.05
t	f	0.94	0.06
f	t	0.29	0.71
f	f	0.001	0.999

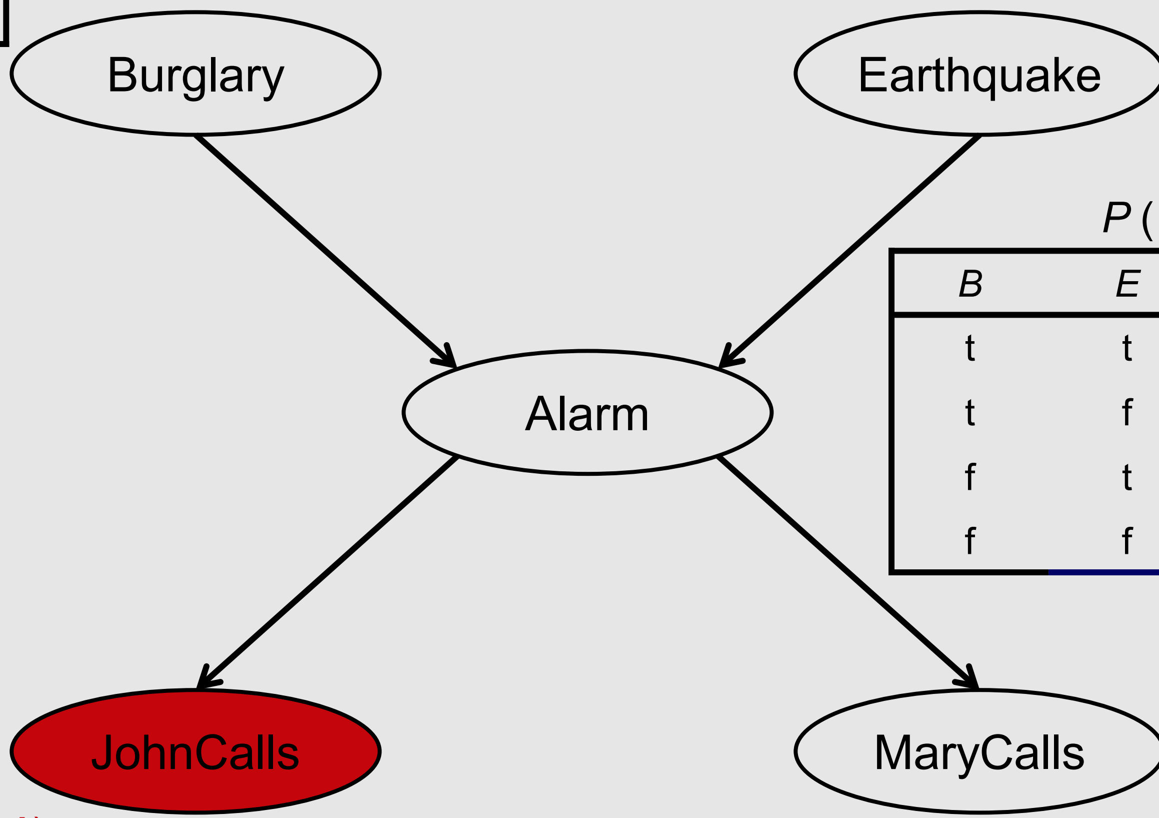
Bayesian network example

P (*B*)

t	f
0.001	0.999

P (*E*)

t	f
0.001	0.999



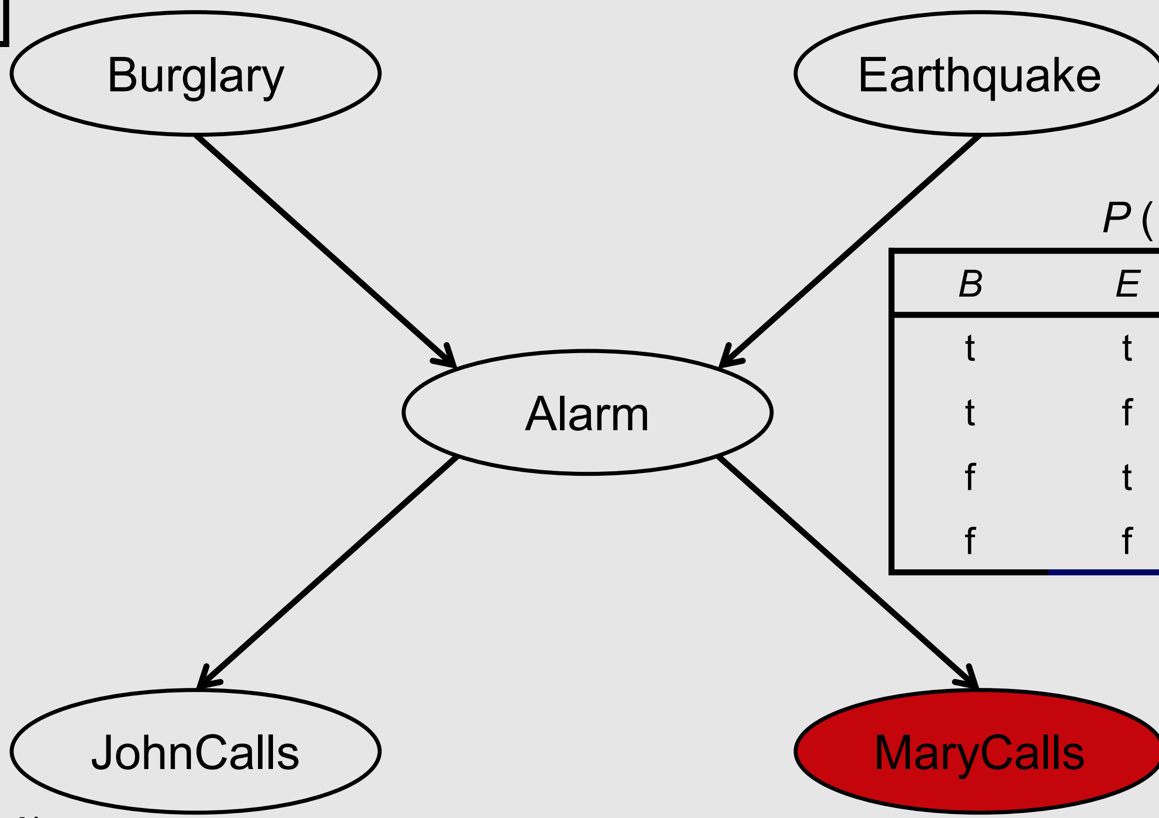
P (*A* | *B*, *E*)

B	*E*	t	f
t	t	0.95	0.05
t	f	0.94	0.06
f	t	0.29	0.71
f	f	0.001	0.999

P (*J* | *A*)

A	t	f
t	0.9	0.1
f	0.05	0.95

Bayesian network example



P (*B*)

t	f
0.001	0.999

P (*E*)

t	f
0.001	0.999

P (*A* \| *B*, *E*)

B	*E*	t	f
t	t	0.95	0.05
t	f	0.94	0.06
f	t	0.29	0.71
f	f	0.001	0.999

P (*J* \| *A*)

A	t	f
t	0.9	0.1
f	0.05	0.95

P (*M* \| *A*)

A	t	f
t	0.7	0.3
f	0.01	0.99

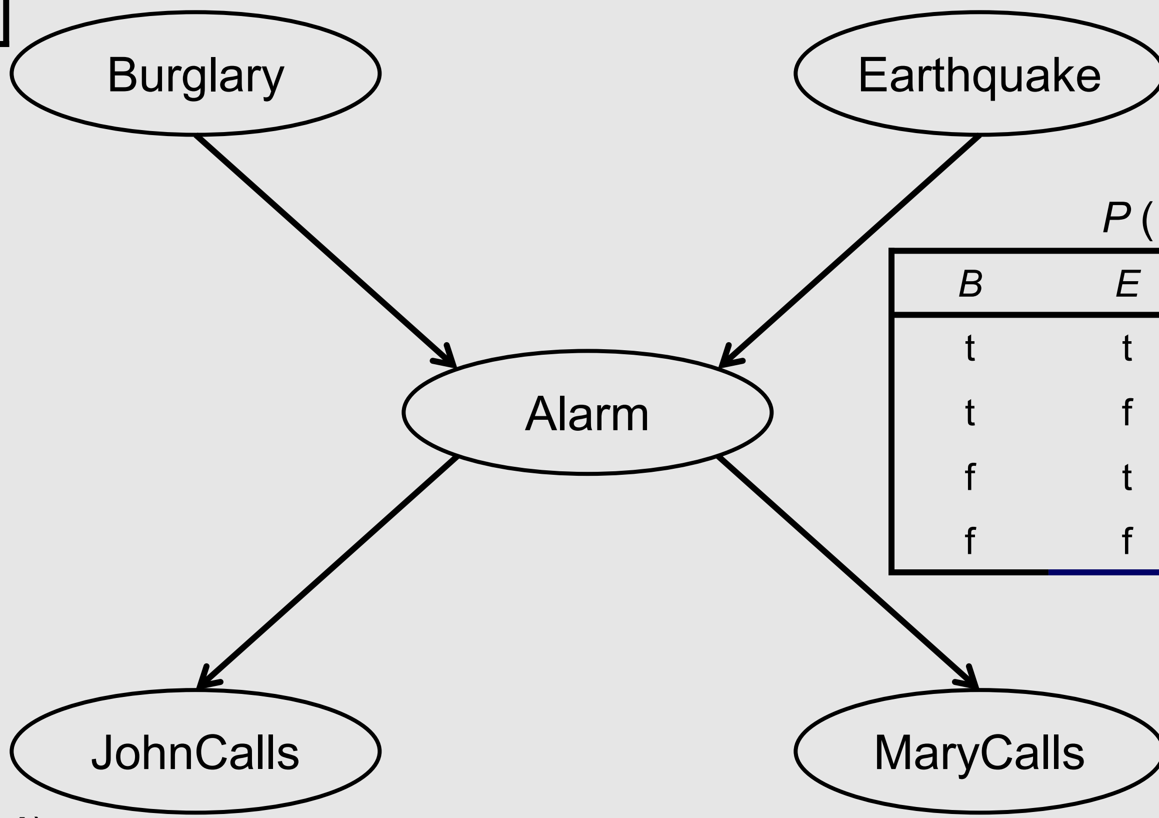
Bayesian network example

P (*B*)

t	f
0.001	0.999

P (*E*)

t	f
0.001	0.999



P (*A* | *B*, *E*)

B	*E*	t	f
t	t	0.95	0.05
t	f	0.94	0.06
f	t	0.29	0.71
f	f	0.001	0.999

P (*J* | *A*)

A	t	f
t	0.9	0.1
f	0.05	0.95

P (*M* | *A*)

A	t	f
t	0.7	0.3
f	0.01	0.99

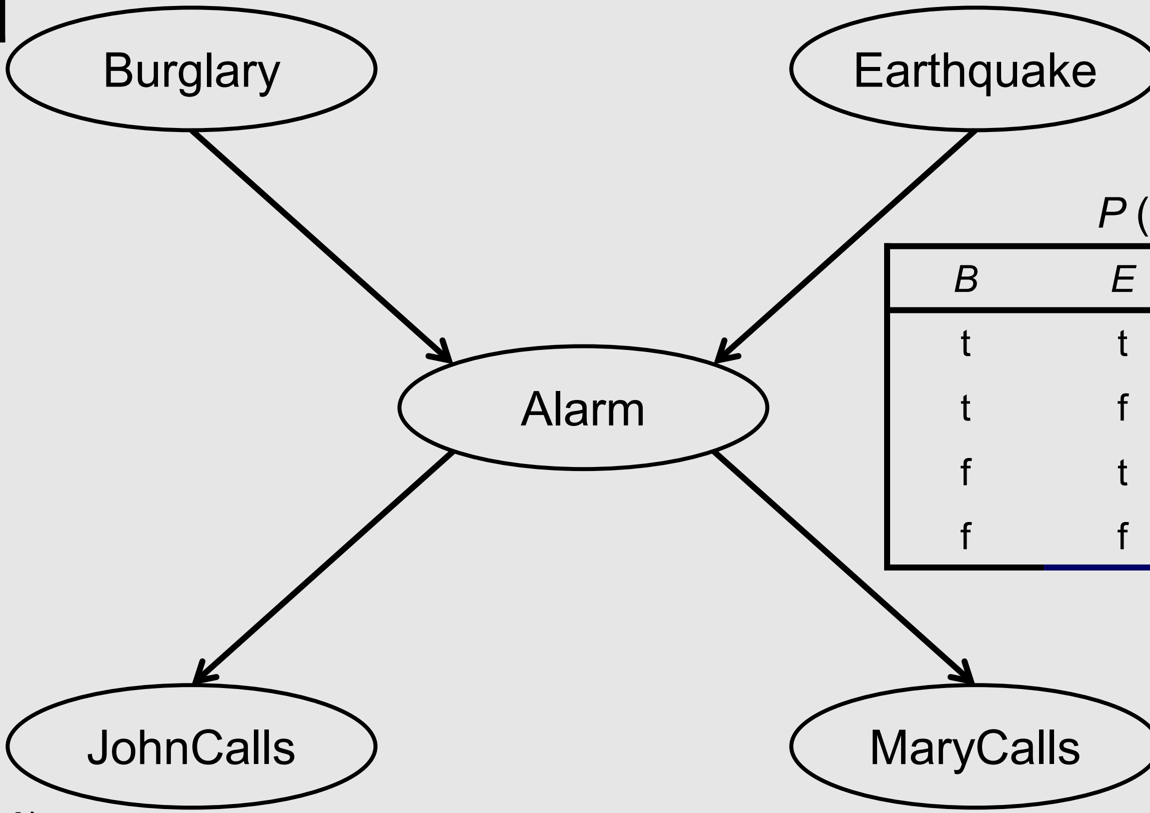
Bayesian network example (different parameters)

P(B)

t	f
0.1	0.9

P(E)

t	f
0.2	0.8



P(A | B, E)

B	*E*	t	f
t	t	0.9	0.1
t	f	0.8	0.2
f	t	0.3	0.7
f	f	0.1	0.9

P(J | A)

A	t	f
t	0.9	0.1
f	0.2	0.8

P(M | A)

A	t	f
t	0.7	0.3
f	0.1	0.9

Bayesian networks

- a BN consists of a Directed Acyclic Graph (DAG) and a set of conditional probability distributions

- in the DAG
  - each node denotes a random variable
  - each edge from X to Y represents that X *directly influences* Y
  - (formally: each variable X is independent of its non-descendants given its parents)

- each node X has a *conditional probability distribution* (CPD) representing $P(X \mid Parents(X))$

Bayesian networks

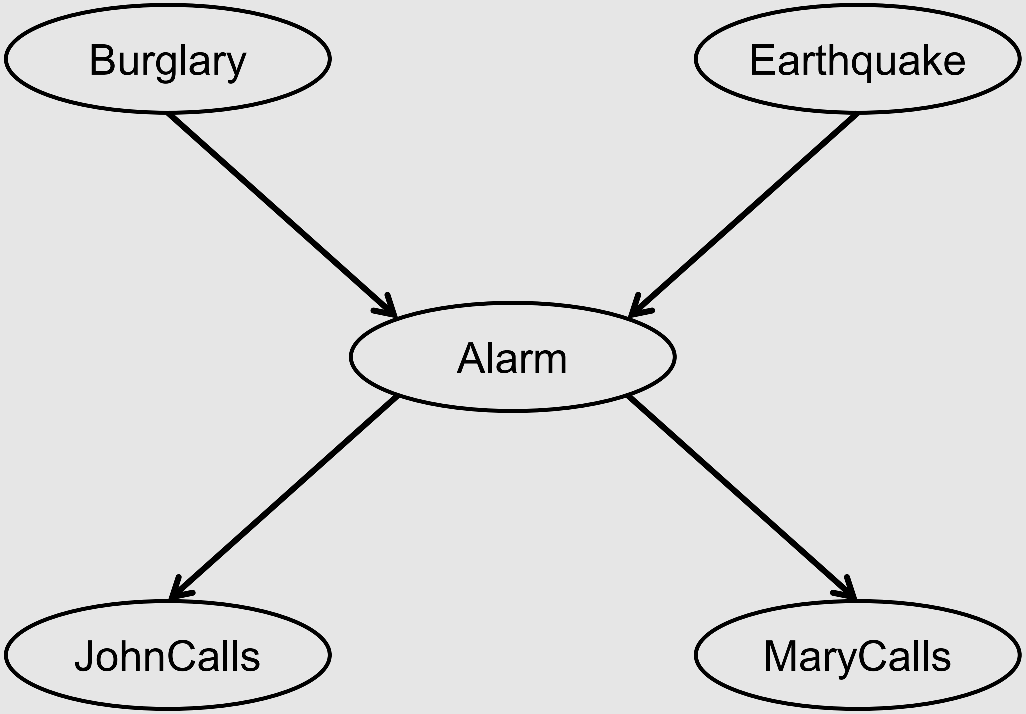
- using the chain rule, a joint probability distribution can always be expressed as

$$P(X_1,\ldots,X_n) = P(X_1)\prod_{i=2}^{n} P(X_i \mid X_1,\ldots,X_{i-1})$$

- a BN provides a compact representation of a joint probability distribution. It corresponds to the assumption:

$$P(X_1,\ldots,X_n) = P(X_1)\prod_{i=2}^{n} P(X_i \mid Parents(X_i))$$

Bayesian networks



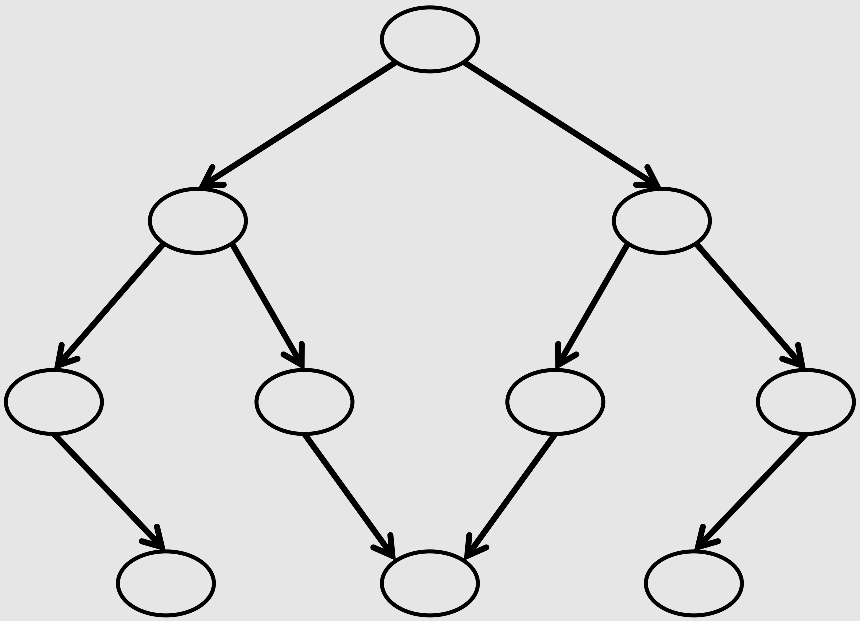
$$P(B, E, A, J, M)$$
$$= P(B)$$
$$\times P(E)$$
$$\times P(A \mid B, E)$$
$$\times P(J \mid A)$$
$$\times P(M \mid A)$$

- a standard representation of the joint distribution for the Alarm example has $2^5 = 32$ parameters
- the BN representation of this distribution has 20 parameters

Bayesian networks

- consider a case with 10 binary random variables

- How many parameters does a BN with the following graph structure have?

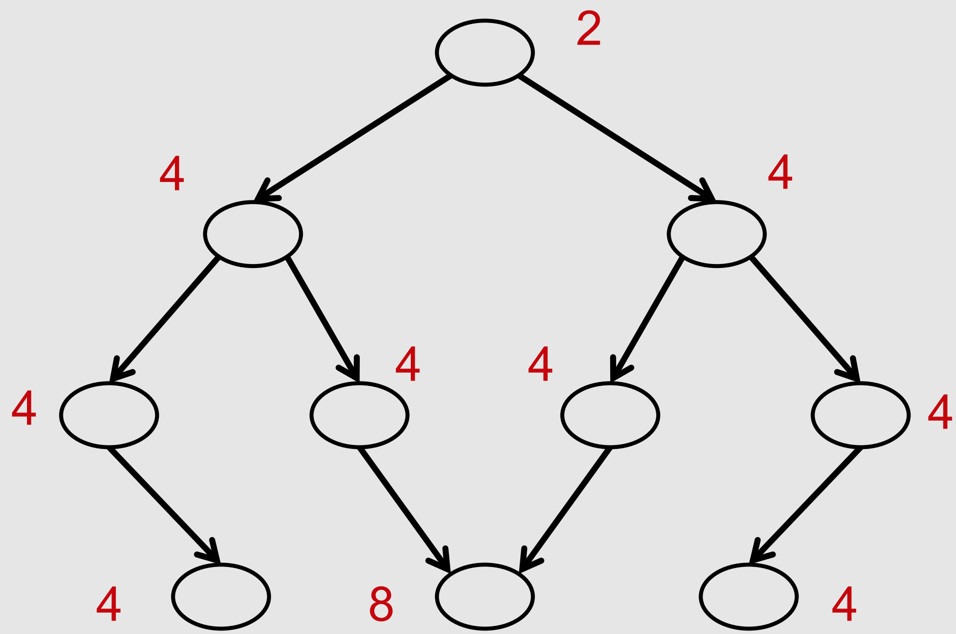


- How many parameters does the standard table representation of the joint distribution have?

Bayesian networks

- consider a case with 10 binary random variables

- How many parameters does a BN with the following graph structure have?



= 42

- How many parameters does the standard table representation of the joint distribution have?

= 1024

Advantages of Bayesian network representation

- Captures independence and conditional independence where they exist
- Encodes the relevant portion of the full joint among variables where dependencies exist
- Uses a graphical representation which lends insight into the complexity of inference

Inference

The inference task in Bayesian networks

Given: values for some variables in the network (*evidence*), and a set of *query* variables

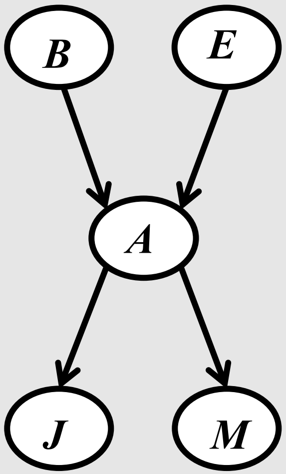
Do: compute the posterior distribution over the query variables

- variables that are neither evidence variables nor query variables are *hidden* variables
- the BN representation is flexible enough that any set can be the evidence variables and any set can be the query variables

Inference by enumeration

- let a denote A=true, and $\neg a$ denote A=false
- suppose we're given the query: $P(b \mid j, m)$

  "probability the house is being burglarized given that John and Mary both called"
- from the graph structure we can first compute:



$$P(b, j, m) = \sum_{e, \neg e} \sum_{a, \neg a} P(b) P(E) P(A \mid b, E) P(j \mid A) P(m \mid A)$$

sum over possible values for E and A variables ($e, \neg e, a, \neg a$)

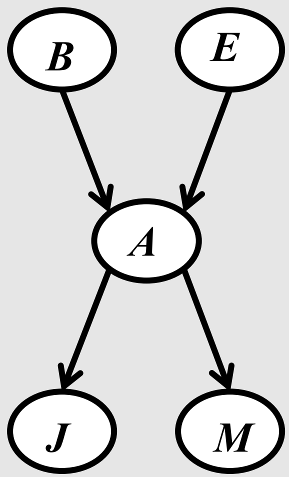
Inference by enumeration

$$P(b,j,m) = \sum_{e,\neg e}\sum_{a,\neg a} P(b)P(E)P(A\mid b,E)P(j\mid A)P(m\mid A)$$

$$= P(b)\sum_{e,\neg e}\sum_{a,\neg a} P(E)P(A\mid b,E)P(j\mid A)P(m\mid A)$$

P(B)
0.001

P(E)
0.001



B	E	P(A)
t	t	0.95
t	f	0.94
f	t	0.29
f	f	0.001

A	P(J)
t	0.9
f	0.05

A	P(M)
t	0.7
f	0.01

B   *E*   *A*   *J*   *M*

$$= 0.001 \times (0.001 \times 0.95 \times 0.9 \times 0.7 + \quad e,\ a$$

$$0.001 \times 0.05 \times 0.05 \times 0.01 + \quad e,\ \neg a$$

$$0.999 \times 0.94 \times 0.9 \times 0.7 + \quad \neg e,\ a$$

$$0.999 \times 0.06 \times 0.05 \times 0.01) \quad \neg e,\ \neg a$$

Inference by enumeration

- now do equivalent calculation for $P(\neg b,\ j,\ m)$
- and determine $P(b \mid j,\ m)$

$$P(b \mid j, m) = \frac{P(b, j, m)}{P(j, m)} = \frac{P(b, j, m)}{P(b, j, m) + P(\neg b, j, m)}$$

Comments on BN inference

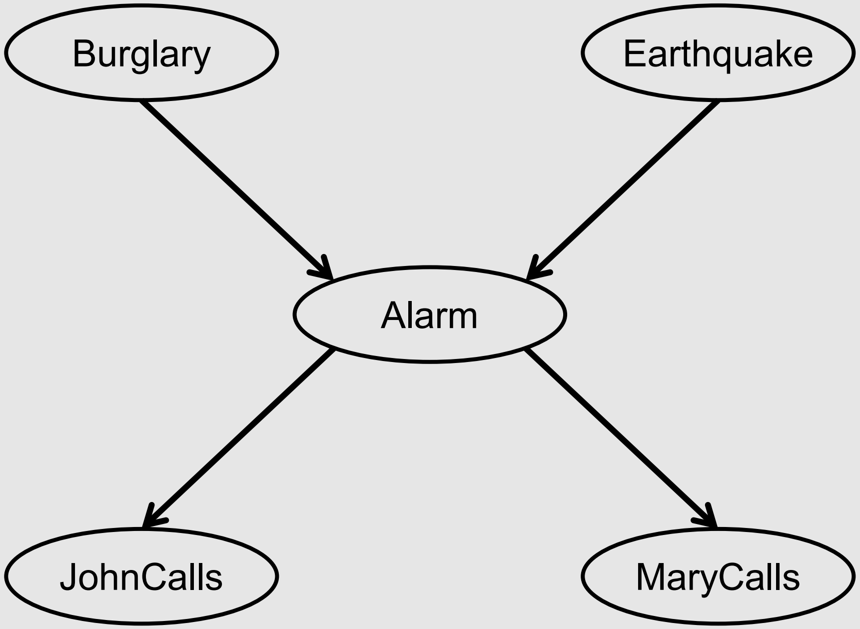
- *inference by enumeration* is an *exact* method (i.e. it computes the exact answer to a given query)
- it requires summing over a joint distribution whose size is exponential in the number of variables
- in many cases we can do exact inference efficiently in large networks
  - key insight: save computation by pushing sums inward
- in general, the Bayes net inference problem is NP-hard
- there are also methods for approximate inference – these get an answer which is "close"
- in general, the approximate inference problem is NP-hard also, but approximate methods work well for many real-world problems

Learning

The parameter learning task

- Given: a set of training instances, the graph structure of a BN

B	E	A	J	M
f	f	f	t	f
f	t	f	f	f
f	f	t	f	t
		…		



- Do: infer the parameters of the CPDs

The structure learning task

- Given: a set of training instances

B	E	A	J	M
f	f	f	t	f
f	t	f	f	f
f	f	t	f	t
		…		

- Do: infer the graph structure (and perhaps the parameters of the CPDs too)

Parameter learning and MLE

- *maximum likelihood estimation* (MLE)
  - given a model structure (e.g. a Bayes net graph) G and a set of data D
  - set the model parameters θ to maximize $P(D \mid G, \theta)$

- i.e. make the data D look as likely as possible under the model $P(D \mid G, \theta)$

Maximum likelihood estimation review

consider trying to estimate the parameter θ (probability of heads) of a biased coin from a sequence of flips (1 stands for head)

$$\boldsymbol{x} = \{1,1,1,0,1,0,0,1,0,1\}$$

the likelihood function for θ is given by:

$$L(\theta : x_1,\ldots,x_n) = \theta^{x_1}(1-\theta)^{1-x_1}\cdots\theta^{x_n}(1-\theta)^{1-x_n}$$

$$= \theta^{\sum x_i}(1-\theta)^{n-\sum x_i}$$

What's MLE of the parameter?

MLE in a Bayes net

$$L(\theta : D, G) = P(D \mid G, \theta) = \prod_{d \in D} P(x_1^{(d)}, x_2^{(d)}, \ldots, x_n^{(d)})$$

$$= \prod_{d \in D} \prod_i P(x_i^{(d)} \mid Parents(x_i^{(d)}))$$

$$= \prod_i \left(\prod_{d \in D} P(x_i^{(d)} \mid Parents(x_i^{(d)})) \right)$$

MLE in a Bayes net

$$L(\theta : D, G) = P(D \mid G, \theta) = \prod_{d \in D} P(x_1^{(d)}, x_2^{(d)}, \ldots, x_n^{(d)})$$

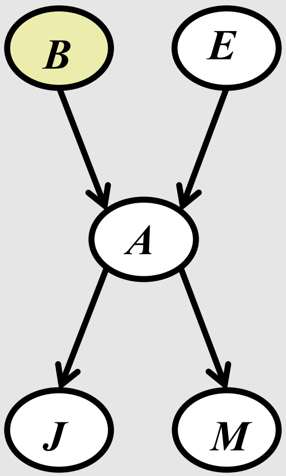
$$= \prod_{d \in D} \prod_i P(x_i^{(d)} \mid Parents(x_i^{(d)}))$$

$$= \prod_i \left(\prod_{d \in D} P(x_i^{(d)} \mid Parents(x_i^{(d)})) \right)$$

independent parameter learning problem for each CPD

Maximum likelihood estimation

now consider estimating the CPD parameters for B and J in the alarm network given the following data set



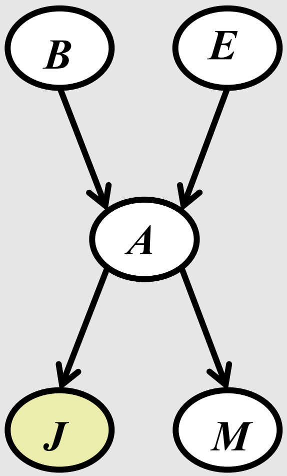
B	E	A	J	M
f	f	f	t	f
f	t	f	f	f
f	f	f	t	t
t	f	f	f	t
f	f	t	t	f
f	f	t	f	t
f	f	t	t	t
f	f	t	t	t

$$P(b) = \frac{1}{8} = 0.125$$

$$P(\neg b) = \frac{7}{8} = 0.875$$

Maximum likelihood estimation

now consider estimating the CPD parameters for B and J in the alarm network given the following data set



B	E	A	J	M
f	f	f	t	f
f	t	f	f	f
f	f	f	t	t
t	f	f	f	t
f	f	t	t	f
f	f	t	f	t
f	f	t	t	t
f	f	t	t	t

$$P(b) = \frac{1}{8} = 0.125$$

$$P(\neg b) = \frac{7}{8} = 0.875$$

$$P(j \mid a) = \frac{3}{4} = 0.75$$

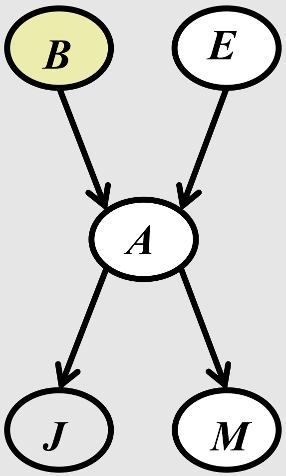
$$P(\neg j \mid a) = \frac{1}{4} = 0.25$$

$$P(j \mid \neg a) = \frac{2}{4} = 0.5$$

$$P(\neg j \mid \neg a) = \frac{2}{4} = 0.5$$

Maximum likelihood estimation

suppose instead, our data set was this…



B	E	A	J	M
f	f	f	t	f
f	t	f	f	f
f	f	f	t	t
f	f	f	f	t
f	f	t	t	f
f	f	t	f	t
f	f	t	t	t
f	f	t	t	t

$$P(b) = \frac{0}{8} = 0$$

$$P(\neg b) = \frac{8}{8} = 1$$

do we really want to set this to 0?

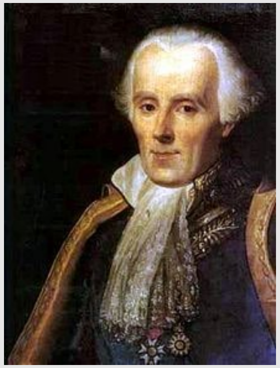
Laplace estimates

- instead of estimating parameters strictly from the data, we could start with some prior belief for each

- for example, we could use *Laplace estimates*

$$P(X = x) = \frac{n_x + 1}{\sum_{v \in \text{Values}(X)} (n_v + 1)}$$

pseudocounts



- where n_v represents the number of occurrences of value v

M-estimates

a more general form: *m-estimates*

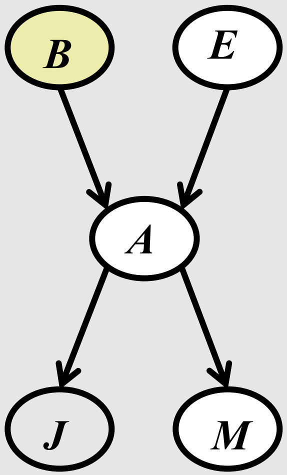
$$P(X=x)=\frac{n_x+p_x m}{\left(\sum_{v\in \text{Values}(X)} n_v\right)+m}$$

prior probability of value x

number of "virtual" instances

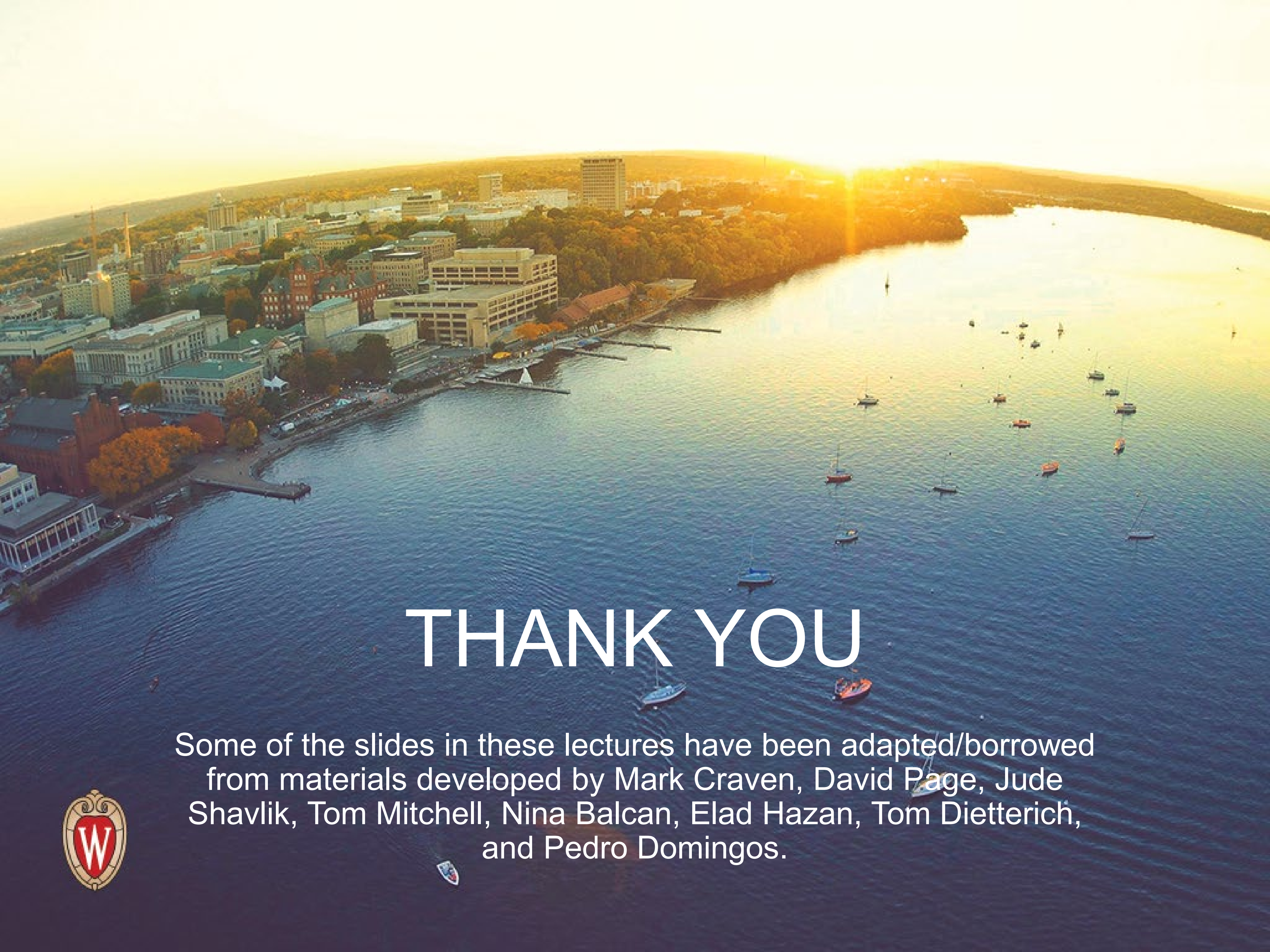
M-estimates example

now let's estimate parameters for B using $m=4$ and $p_b=0.25$



B	*E*	*A*	*J*	*M*
f	f	f	t	f
f	t	f	f	f
f	f	f	t	t
f	f	f	f	t
f	f	t	t	f
f	f	t	f	t
f	f	t	t	t
f	f	t	t	t

$$P(b) = \frac{0 + 0.25 \times 4}{8 + 4} = \frac{1}{12} = 0.08 \qquad P(\neg b) = \frac{8 + 0.75 \times 4}{8 + 4} = \frac{11}{12} = 0.92$$



THANK YOU

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, and Pedro Domingos.