CS 2750 Machine Learning
Lecture 14

Bayesian belief networks

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Midterm exam

When: Monday, March 2 , 2015

Midterm is:

- **In-class (75 minutes)**
- **closed book**
- **material covered during the semester including lecture today**

Project proposals

Due: Monday, March 16, 2015

- **1 page long**

Proposal

- **Written proposal:**
  1. Outline of a learning problem, type of data you have available. Why is the problem important?
  2. Learning methods you plan to try and implement for the problem. References to previous work.
  3. How do you plan to test, compare learning approaches
  4. Schedule of work (approximate timeline of work)

Bayesian belief networks (BBNs)

Bayesian belief networks.

- Represent the full joint distribution over the variables more compactly with a **smaller number of parameters**.
- Take advantage of **conditional and marginal independences** among random variables

- **A and B are independent**

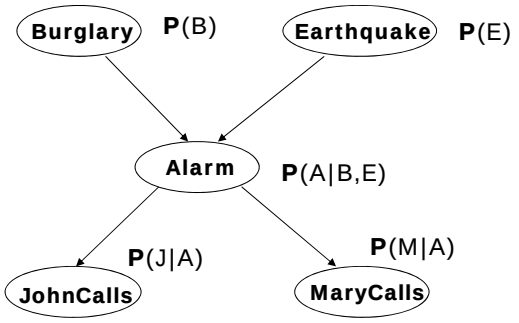
$$P(A,B) = P(A)P(B)$$

- **A and B are conditionally independent given C**

$$P(A,B \mid C) = P(A \mid C)P(B \mid C)$$
$$P(A \mid C,B) = P(A \mid C)$$

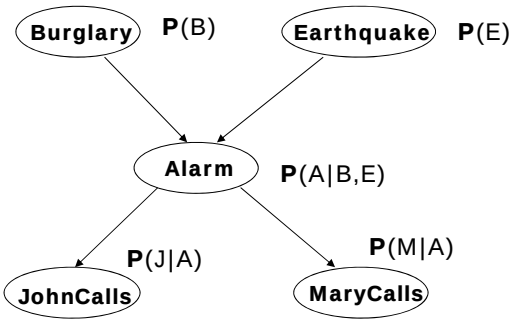
Bayesian belief network

1. **Directed acyclic graph**
   - **Nodes** = random variables
     Burglary, Earthquake, Alarm, Mary calls and John calls
   - **Links** = direct (causal) dependencies between variables.
     The chance of Alarm being is influenced by Earthquake,
     The chance of John calling is affected by the Alarm

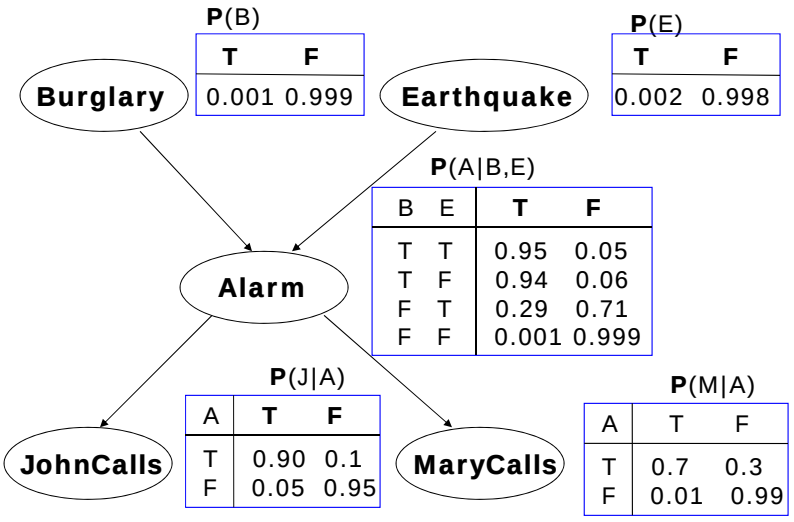


Bayesian belief network

2. **Local conditional distributions**
   - relate variables and their parents



Bayesian belief network

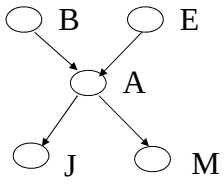


Full joint distribution in BBNs

Full joint distribution is defined in terms of local conditional distributions (obtained via the chain rule):

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

Example:



Assume the following assignment of values to random variables

$B = T, E = T, A = T, J = T, M = F$

Then its probability is:

$$P(B = T, E = T, A = T, J = T, M = F) =$$

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

Bayesian belief networks (BBNs)

Bayesian belief networks

- Represent the full joint distribution over the variables more compactly using the product of local conditionals.
- **But how did we get to local parameterizations?**

Answer:

- **Graphical structure** encodes **conditional and marginal independences** among random variables
- **A and B are independent** $P(A,B) = P(A)P(B)$
- **A and B are conditionally independent given C**

$$P(A \mid C, B) = P(A \mid C)$$

$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$

- **The graph structure implies the decomposition !!!**

Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

- **What did we save?**

Alarm example: 5 binary (True, False) variables

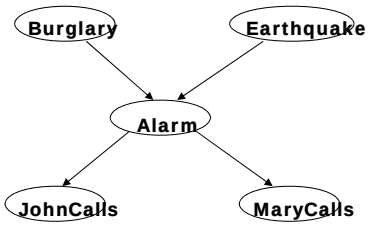
# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

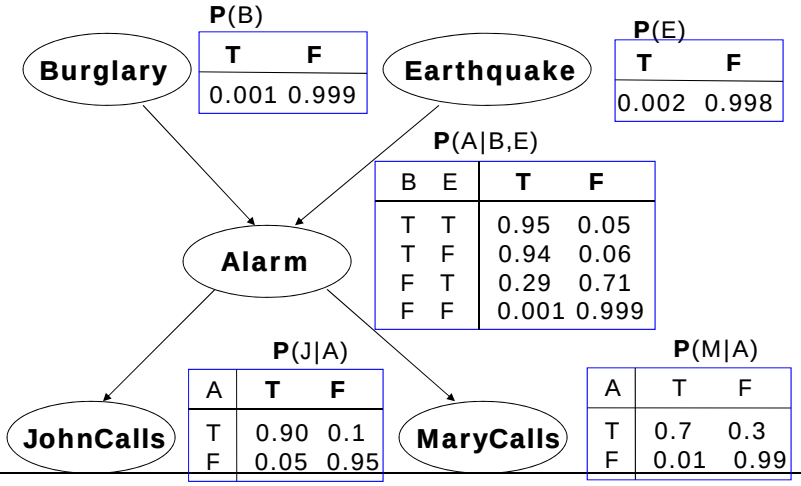
$$2^5 - 1 = 31$$

# of parameters of the BBN: ?



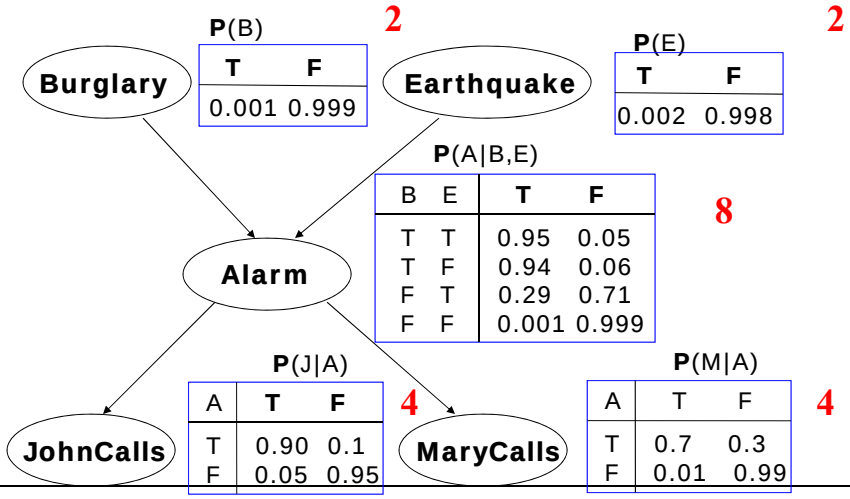
Bayesian belief network.

- In the BBN the **full joint distribution** is expressed using a set of local conditional distributions



Bayesian belief network.

- In the BBN the **full joint distribution** is expressed using a set of local conditional distributions



Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, ..., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

- **What did we save?**

Alarm example: 5 binary (True, False) variables

# of parameters of the full joint:

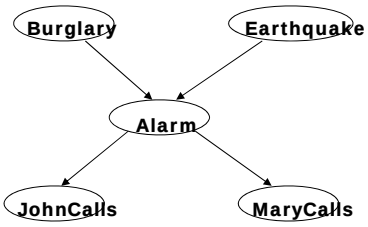
$$2^5 = 32$$

One parameter is for free:

$$2^5 - 1 = 31$$

# of parameters of the BBN:

$$2^3 + 2(2^2) + 2(2) = 20$$



One parameter in every conditional is for free:

?

Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, ..., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

- **What did we save?**

Alarm example: 5 binary (True, False) variables

# of parameters of the full joint:

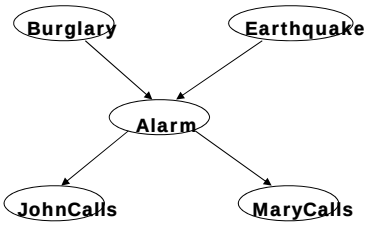
$$2^5 = 32$$

One parameter is for free:

$$2^5 - 1 = 31$$

# of parameters of the BBN:

$$2^3 + 2(2^2) + 2(2) = 20$$



One parameter in every conditional is for free:

$$2^2 + 2(2) + 2(1) = 10$$

Inference in Bayesian networks

- BBN models compactly the full joint distribution by taking advantage of existing independences between variables
- Simplifies the acquisition of a probabilistic model
- But we are interested in solving various **inference tasks**:
  - **Diagnostic task. (from effect to cause)**

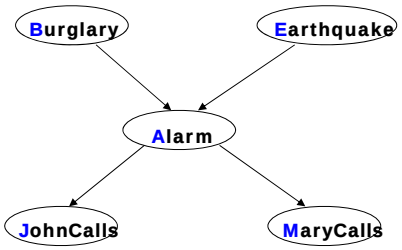
    $$\mathbf{P}(Burglary \mid JohnCalls = T)$$
  - **Prediction task. (from cause to effect)**

    $$\mathbf{P}(JohnCalls \mid Burglary = T)$$
  - **Other probabilistic queries** (queries on joint distributions).

    $$\mathbf{P}(Alarm)$$
- **Main issue:** Can we take advantage of independences to construct special algorithms and speeding up the inference?

Inference in Bayesian network

- **Bad news:**
  - Exact inference problem in BBNs is NP-hard (Cooper)
  - Approximate inference is NP-hard (Dagum, Luby)
- **But** very often we can achieve significant improvements
- Assume our Alarm network



- Assume we want to compute: $P(J = T)$

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J = T) =$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(B = b, E = e, A = a, J = T, M = m)$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)$$

Computational cost:

Number of additions: ?

Number of products: ?

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J = T) =$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(B = b, E = e, A = a, J = T, M = m)$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)$$

Computational cost:

Number of additions: 15

Number of products: ?

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J = T) =$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(B = b, E = e, A = a, J = T, M = m)$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)$$

Computational cost:

Number of additions: 15

Number of products: 16*4=64

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$P(J = T) =$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)$$

$$= \sum_{b \in T,F} \sum_{a \in T.F} \sum_{m \in T,F} P(J = T \mid A = a) P(M = m \mid A = a) P(B = b) [\sum_{e \in T,F} P(A = a \mid B = b, E = e) P(E = e)]$$

$$= \sum_{a \in T,F} P(J = T \mid A = a) [\sum_{m \in T,F} P(M = m \mid A = a)] [\sum_{b \in T,F} P(B = b) [\sum_{e \in T,F} P(A = a \mid B = b, E = e) P(E = e)]]$$

Computational cost:

Number of additions: 1+2*[1+1+2*1]=**?**

Number of products: 2*[2+2*(1+2*1)]=?

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$=\sum_{b\in T,F}\sum_{a\in T.F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]$$

$$=\sum_{a\in T,F}P(J=T\mid A=a)[\sum_{m\in T,F}P(M=m\mid A=a)][\sum_{b\in T,F}P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]]$$

Computational cost:

Number of additions: 1+2*[1+1+2*1]=**9**

Number of products: 2*[2+2*(1+2*1)]=?

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$=\sum_{b\in T,F}\sum_{a\in T.F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]$$

$$=\sum_{a\in T,F}P(J=T\mid A=a)[\sum_{m\in T,F}P(M=m\mid A=a)][\sum_{b\in T,F}P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]]$$

Computational cost:

Number of additions: 1+2*[1+1+2*1]=**9**

Number of products: 2*[2+2*(1+2*1)]=16

CS 2750 Machine Learning

Variable elimination

- **Variable elimination:**
  - Similar idea but interleave sum and products one variable at the time during inference
  - E.g. Query $P(J=T)$ requires to eliminate A,B,E,M and this can be done in different order

$$P(J=T)=$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

Variable elimination

Assume order: M, E, B,A to calculate $P(J=T)$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}P(J=T\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)\left[\sum_{m\in T,F}P(M=m\mid A=a)\right]$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}P(J=T\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)\ \ 1$$

$$=\sum_{a\in T,F}\sum_{b\in T,F}P(J=T\mid A=a)P(B=b)\left[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)\right]$$

$$=\sum_{a\in T,F}\sum_{b\in T,F}P(J=T\mid A=a)P(B=b)\,\tau_1(A=a,B=b)$$

$$=\sum_{a\in T,F}P(J=T\mid A=a)\left[\sum_{e\in T,F}P(B=b)\,\tau_1(A=a,B=b)\right]$$

$$=\sum_{a\in T,F}P(J=T\mid A=a)\ \ \tau_2(A=a)\quad=\boxed{P(J=T)}$$

Inference in Bayesian network

- **Exact inference algorithms:**
  - **Variable elimination**
  - Recursive decomposition (Cooper, Darwiche)
  - Symbolic inference (D'Ambrosio)
  - Belief propagation algorithm (Pearl)
  - Clustering and joint tree approach (Lauritzen, Spiegelhalter)
  - Arc reversal (Olmsted, Schachter)

- **Approximate inference algorithms:**
  - **Monte Carlo methods:**
    - Forward sampling, Likelihood sampling
  - Variational methods

Monte Carlo approaches

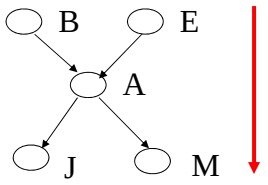
- **MC approximation**:
  - The probability is approximated using sample frequencies
  - **Example:**

$$\tilde{P}(B=T, J=T) = \frac{N_{B=T,J=T}}{N}$$

$N_{B=T,J=T}$ ← *# samples with* $B=T, J=T$

N ← *total # samples*

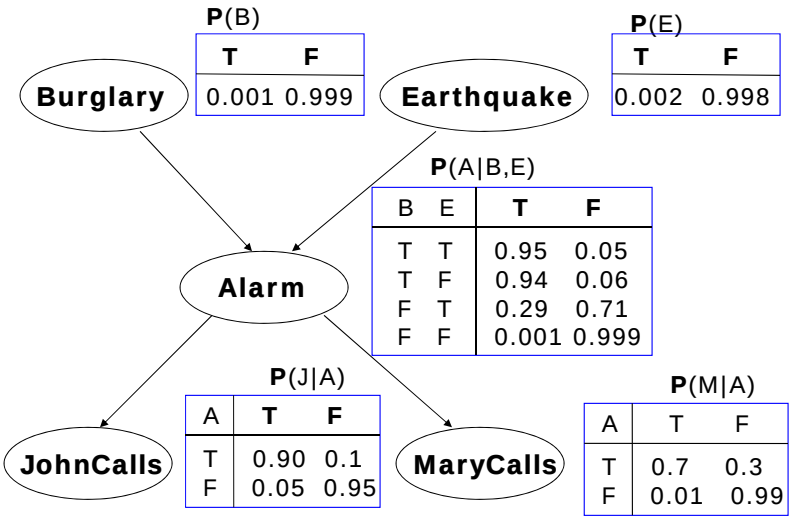
- **Sample generation: BBN sampling of the joint is easy**



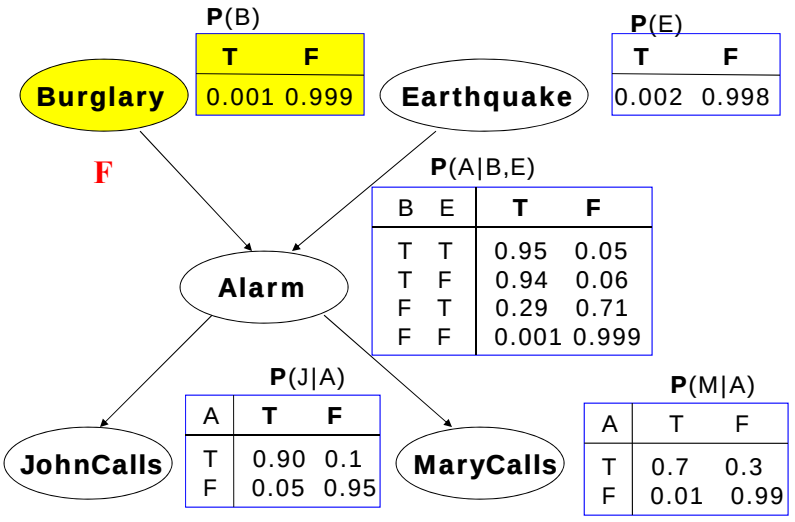
Generate sample in a top down manner, following the links

- **One sample gives one assignment of values to all variables**

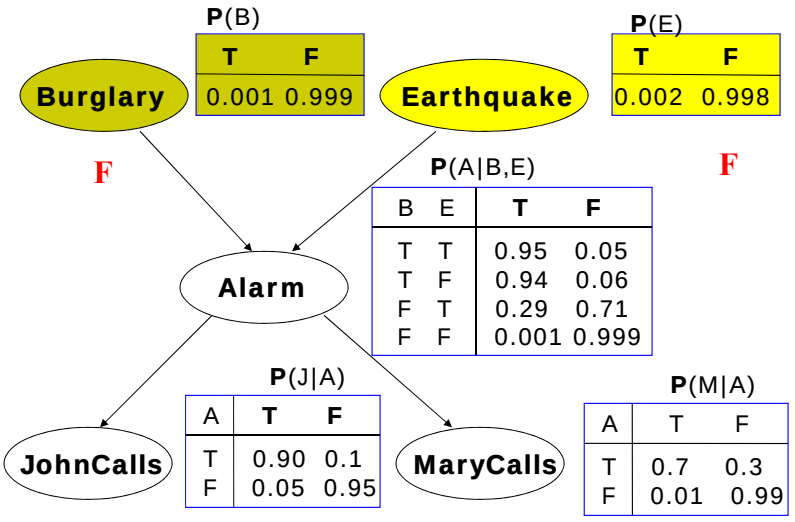
BBN sampling example



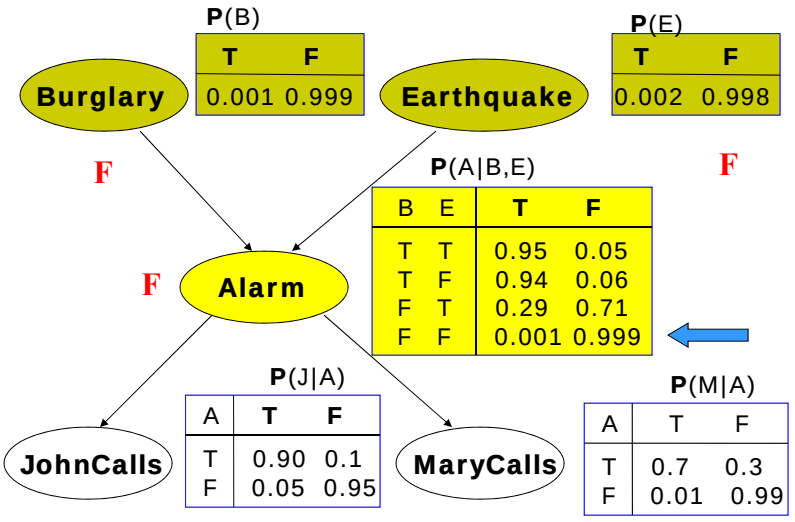
BBN sampling example



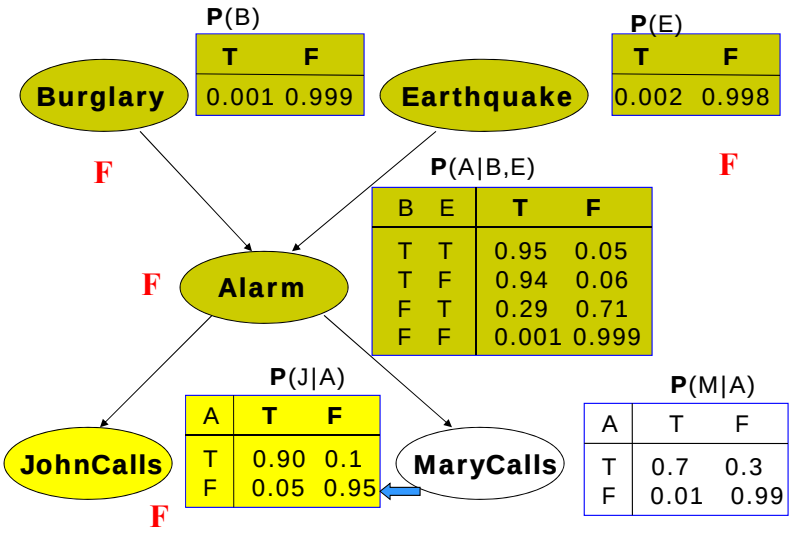
BBN sampling example



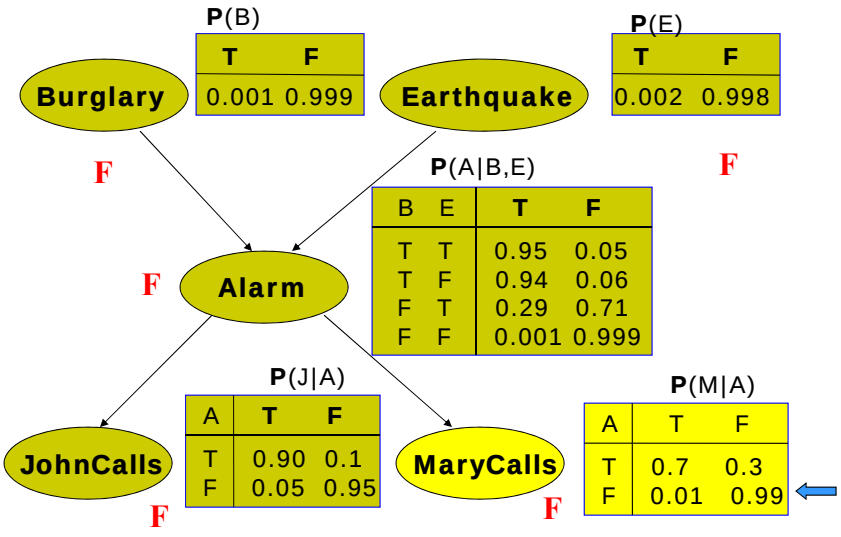
BBN sampling example



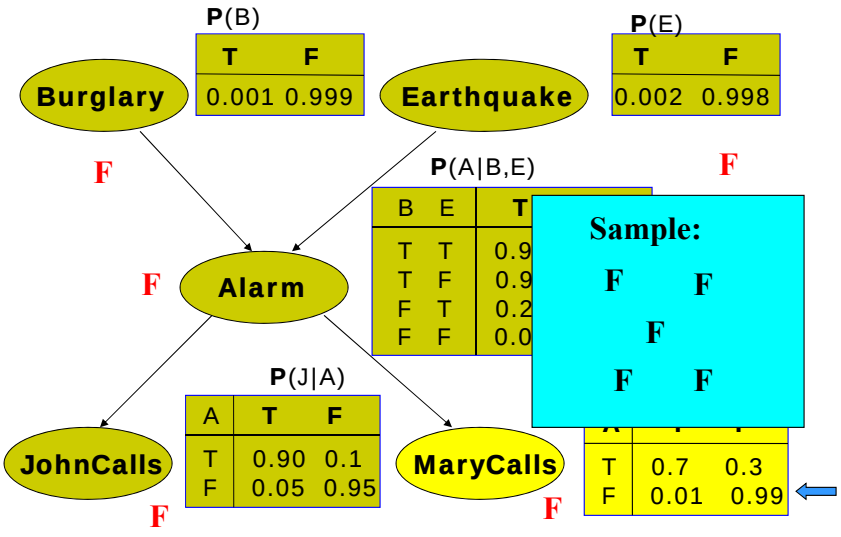
BBN sampling example



BBN sampling example



BBN sampling example



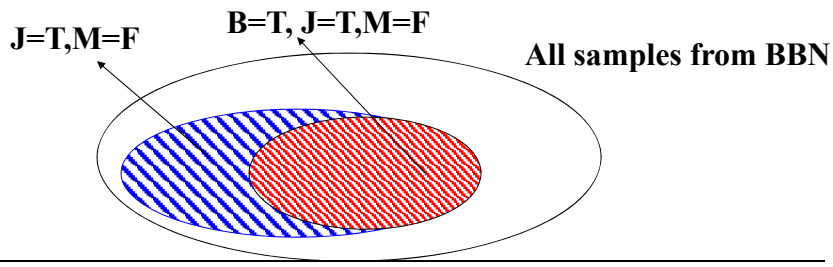
Monte Carlo approaches

- **MC approximation of conditional probabilities**:
  - The probability is approximated using sample frequencies
  - **Example:**

$$\tilde{P}(B=T \mid J=T, M=F) = \frac{N_{B=T,J=T,M=F}}{N_{J=T,M=F}}$$

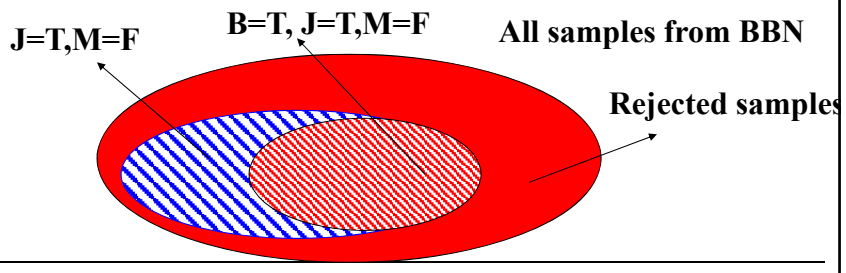
$\#\ samples\ \ with\ \ B=T, J=T, M=F$

$\#\ samples\ \ with\ \ J=T, M=F$



Monte Carlo approaches

- **Rejection sampling**
  - Generate samples from the full joint by sampling BBN
  - Use only samples that agree with the condition, the remaining samples are rejected
- **Problem:** many samples can be rejected



Likelihood weighting

Idea: generate only samples consistent with an evidence (or conditioning event)

- Benefit: Avoids inefficiencies of rejection sampling

Problem:

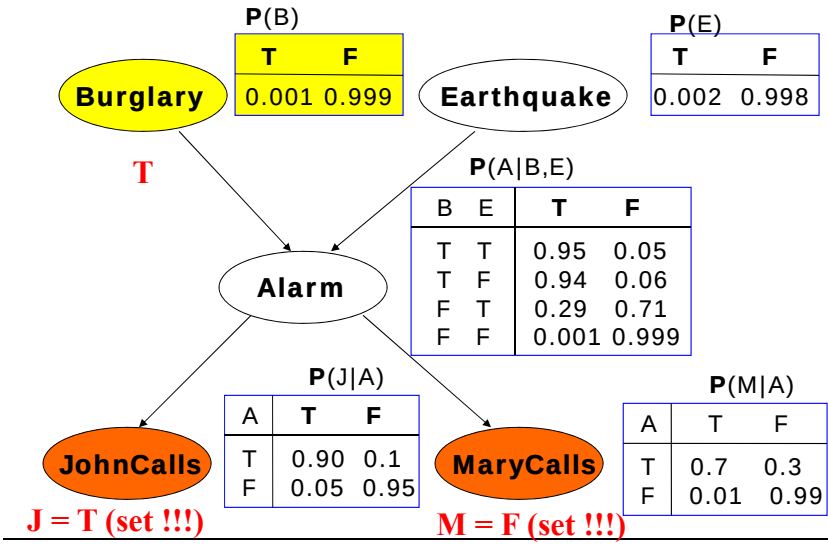
- **the distribution generated by enforcing the conditioning variables to set values is biased**
- simple counts are not sufficient to estimate the probabilities

Solution:

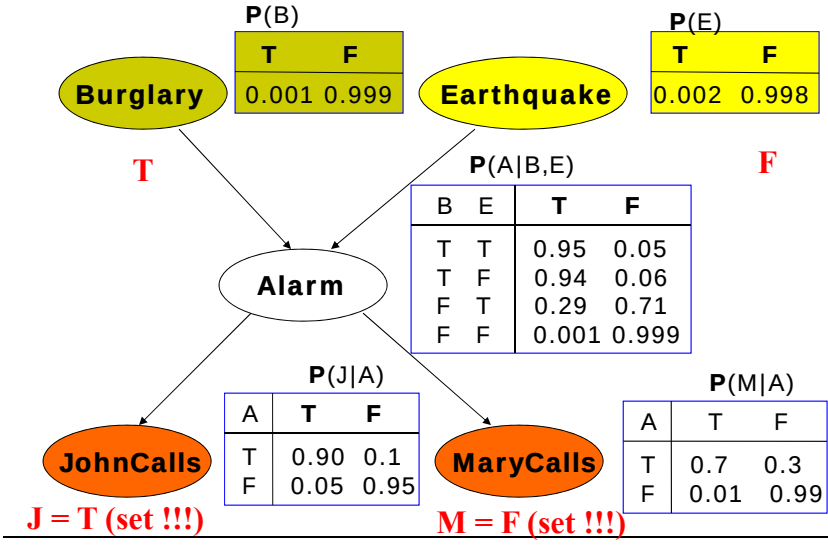
- **With every sample keep a weight with which it should count towards the estimate**

$$\tilde{P}(B=T \mid J=T, M=F) = \frac{\sum_{\text{samples with } B=T, M=F \text{ and } J=T} w_{B=T \mid J=T, M=F}}{\sum_{\text{samples with any value of } B \text{ and } J=T, M=F} w_{B=x \mid J=T, M=F}}$$

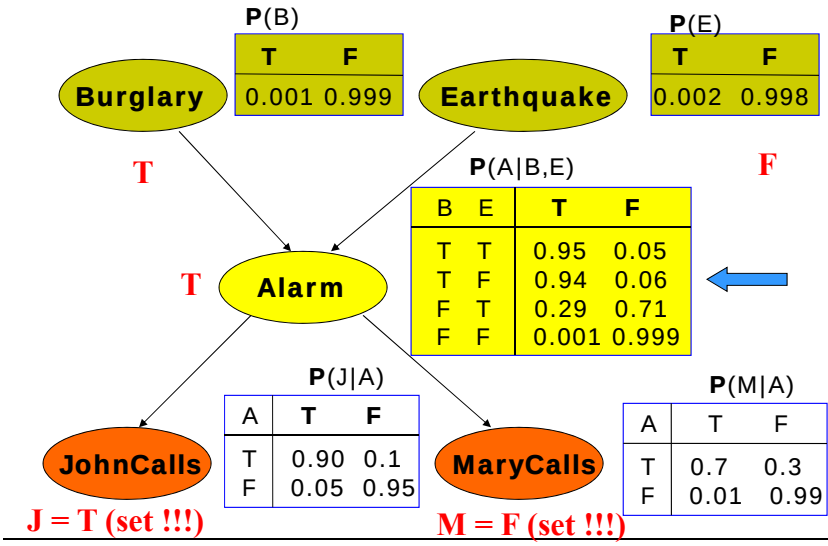
BBN likelihood weighting example



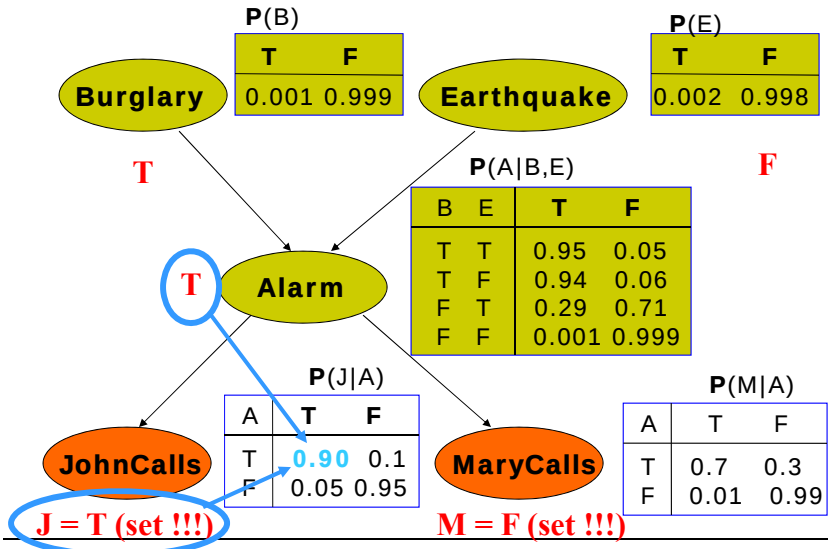
BBN likelihood weighting example



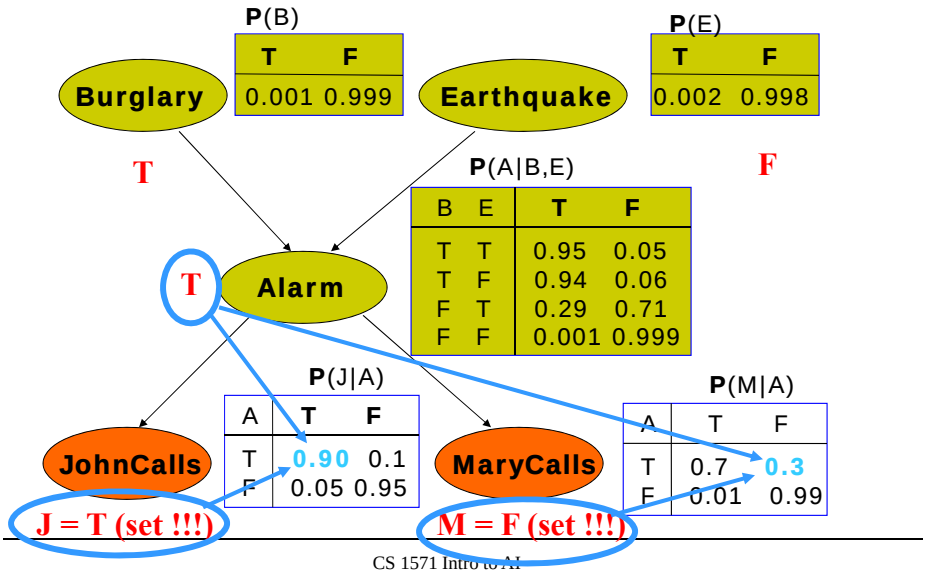
BBN likelihood weighting example



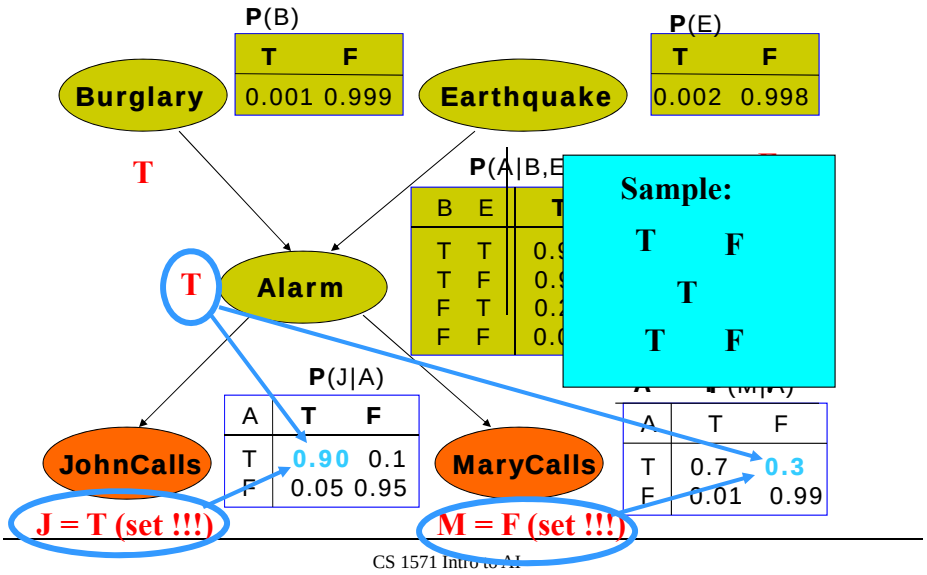
BBN likelihood weighting example



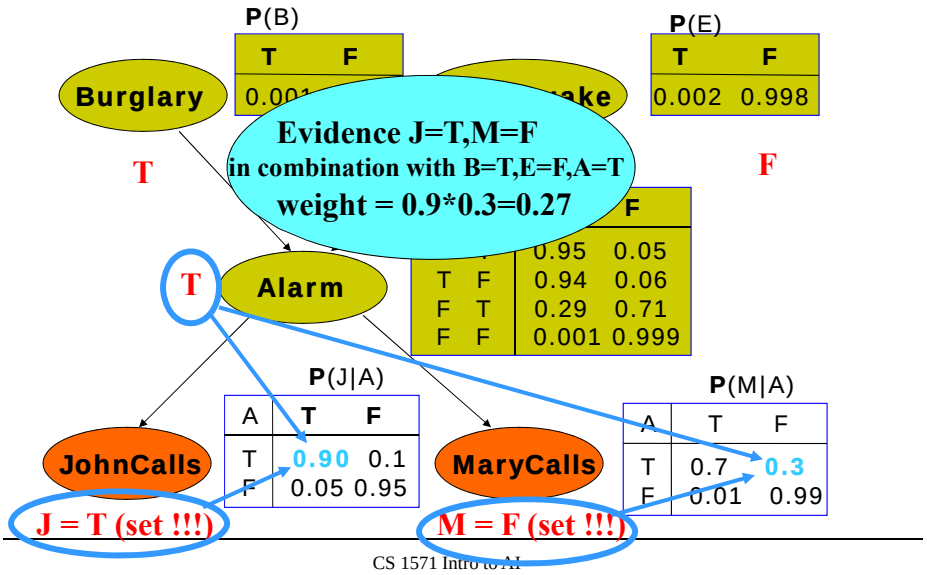
BBN likelihood weighting example



BBN likelihood weighting example

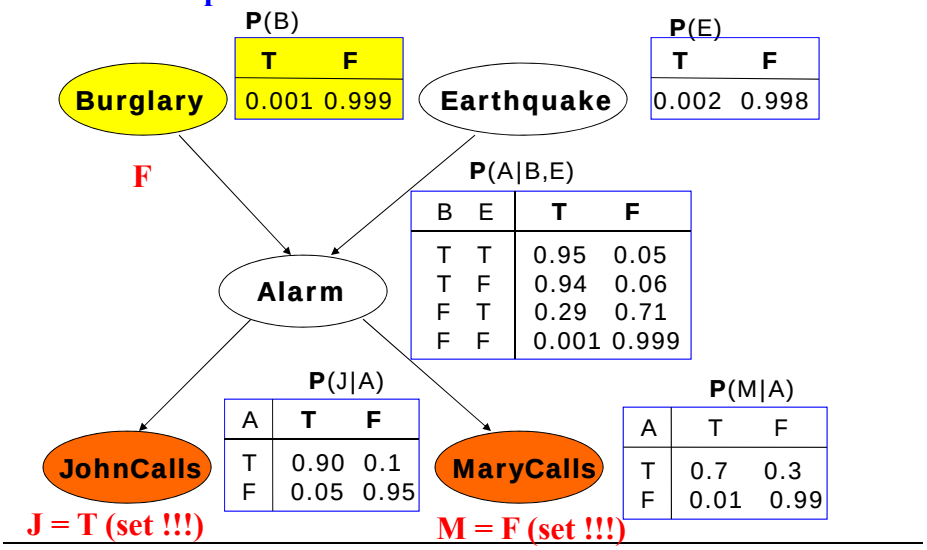


BBN likelihood weighting example



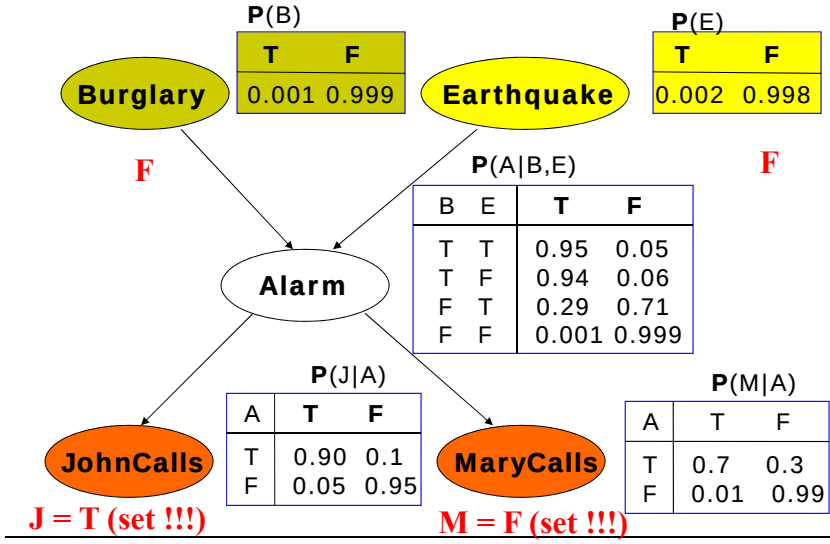
BBN likelihood weighting example

Second sample



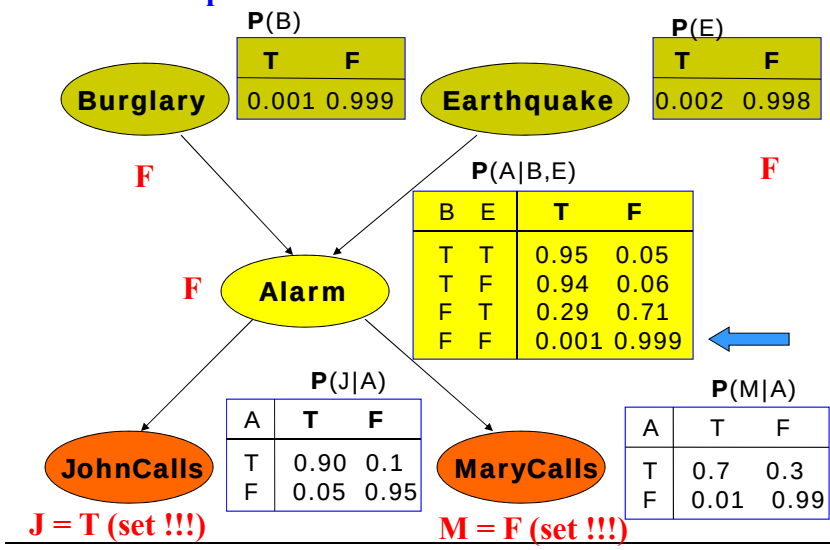
BBN likelihood weighting example

Second sample



BBN likelihood weighting example

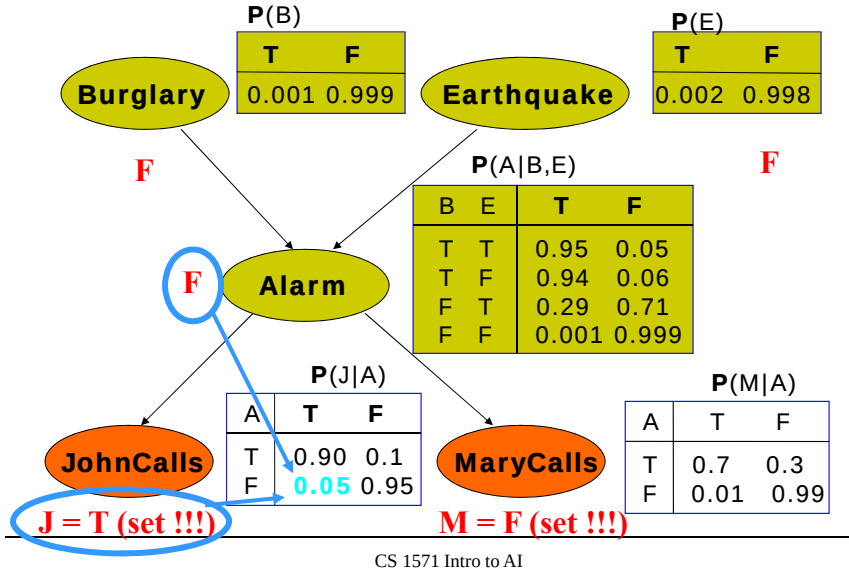
Second sample



CS 1571 Intro to AI

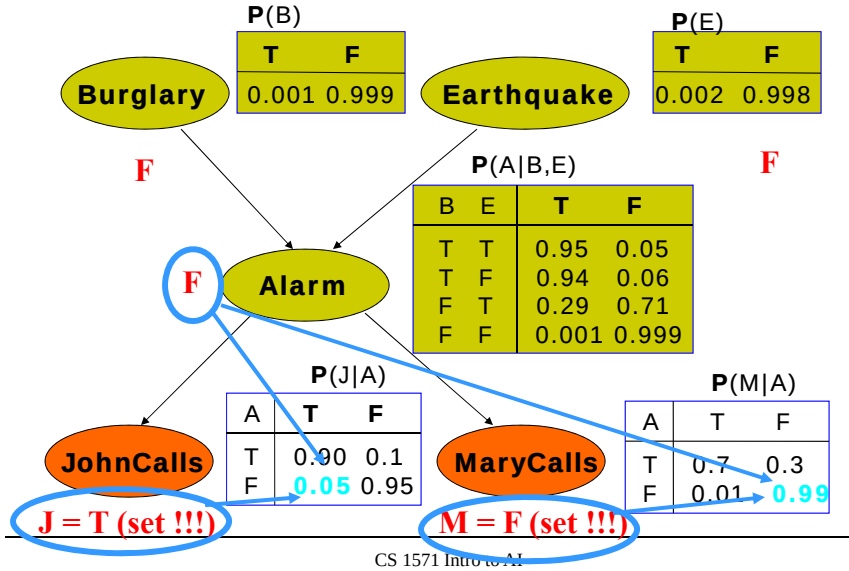
BBN likelihood weighting example

Second sample



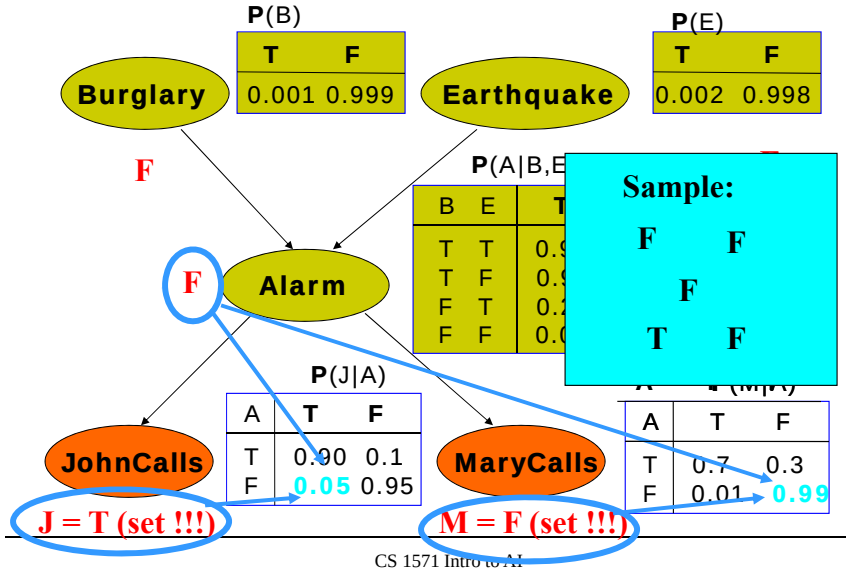
BBN likelihood weighting example

Second sample



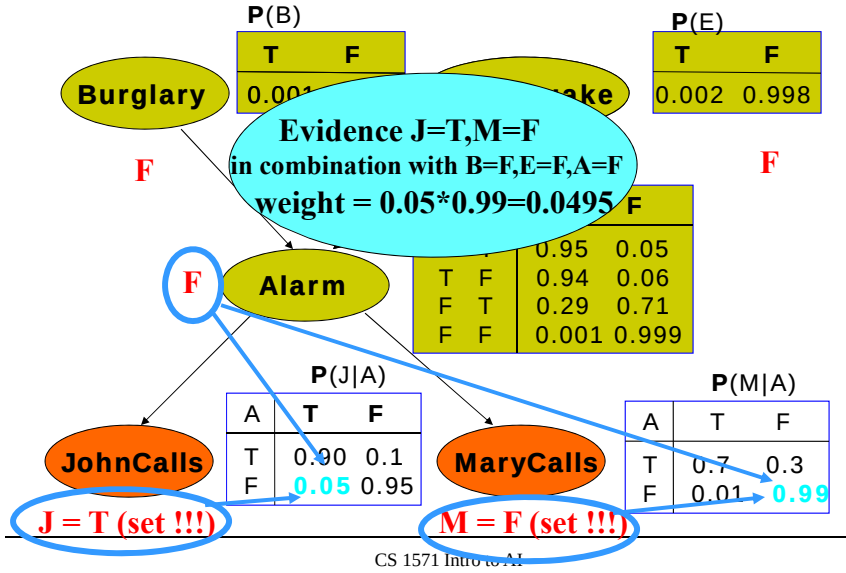
BBN likelihood weighting example

Second sample



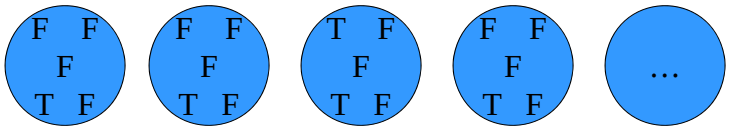
BBN likelihood weighting example

Second sample



Likelihood weighting

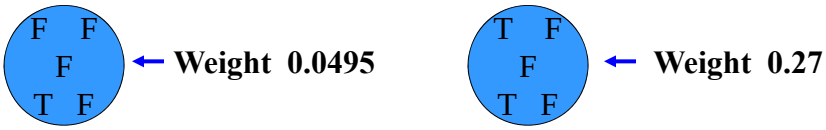
- Assume we have generated the following M samples:



M

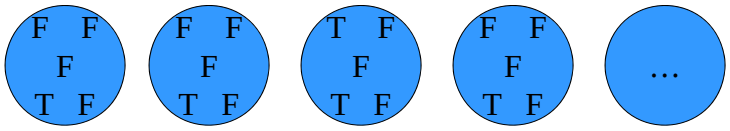
How to make the samples consistent?

Weight each sample by probability with which it agrees with the conditioning evidence P(e).



Likelihood weighting

- Assume we have generated the following M samples:



M

0.0495 **0.0495** **0.27** **0.0495**

$$\tilde{P}(B=T \mid J=T, M=F) = \frac{\sum_{\text{samples with } B=T, M=F \text{ and } J=T} w_{B=T \mid J=T, M=F}}{\sum_{\text{samples with any value of } B \text{ and } J=T, M=F} w_{B=x \mid J=T, M=F}}$$

Learning of BBN

Learning.

- **Learning of parameters of conditional probabilities**
- **Learning of the network structure**

Variables:

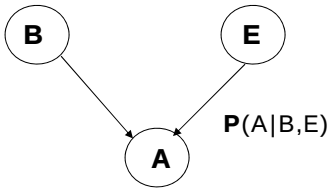
- **Observable** – values present in every data sample
- **Hidden** – they values are never observed in data
- **Missing values** – values sometimes present, sometimes not

Next:

- Learning of parameters of BBN
- All variables are observable

Estimation of parameters of BBN

- **Idea:** decompose the estimation problem for the full joint over a large number of variables to a set of smaller estimation problems corresponding to local parent-variable conditionals.
- **Example:** Assume A,E,B are binary with *True, False* values



4 estimation problems

$\mathbf{P}(A|B=T,E=T)$
$\mathbf{P}(A|B=T,E=F)$
$\mathbf{P}(A|B=F,E=T)$
$\mathbf{P}(A|B=F,E=F)$

- **Assumption that enables the decomposition:** parameters of conditional distributions are independent

Estimates of parameters of BBN

- Two assumptions that permit the decomposition:
  - **Sample independence**

$$P(D \mid \Theta, \xi) = \prod_{u=1}^{N} P(D_u \mid \Theta, \xi)$$

  - **Parameter independence**

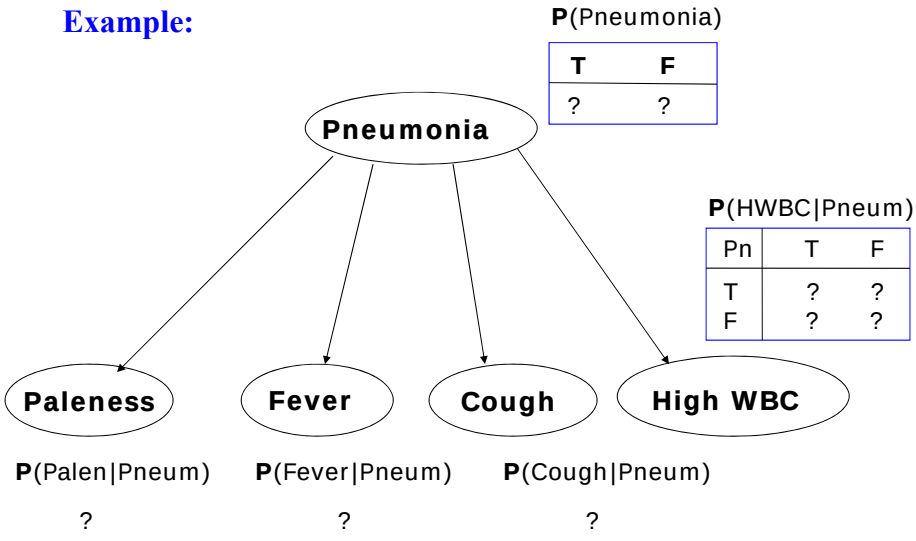
# of nodes
# of parents values

$$p(\Theta \mid D, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\theta_{ij} \mid D, \xi)$$

Parameters of **each conditional** (one for every assignment of values to parent variables) can be learned independently

Learning of BBN parameters. Example.

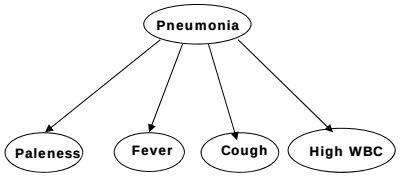
Example:



Learning of BBN parameters. Example.

Data D (different patient cases):

Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F



Estimates of parameters of BBN

- Much like multiple **coin toss or roll of a dice** problems.
- A "smaller" learning problem corresponds to the learning of exactly one conditional distribution
- **Example:**

$$\mathbf{P}(Fever \mid Pneumonia = T)$$

- **Problem:** How to pick the data to learn?

Estimates of parameters of BBN

Much like multiple **coin toss or roll of a dice** problems.

- A "smaller" learning problem corresponds to the learning of exactly one conditional distribution

Example:

$$\mathbf{P}(Fever \mid Pneumonia = T)$$

Problem: How to pick the data to learn?

Answer:

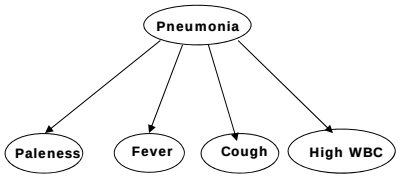
1. Select data points with Pneumonia=T
   (ignore the rest)
2. Focus on (select) only values of the random variable defining the distribution (Fever)
3. Learn the parameters of the conditional the same way as we learned the parameters for a coin or a dice

Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 1: Select data points with Pneumonia=T

Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F

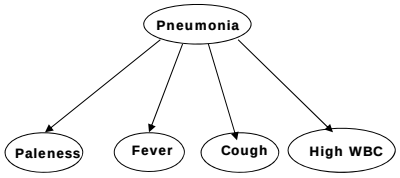


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 1: Ignore the rest

Pal	Fev	Cou	HWB	Pneu
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	T	T	T	T
F	T	F	T	T

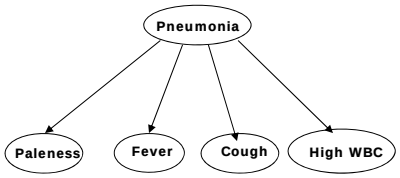


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 2: Select values of the random variable defining the distribution of Fever

Pal	Fev	Cou	HWB	Pneu
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	T	T	T	T
F	T	F	T	T

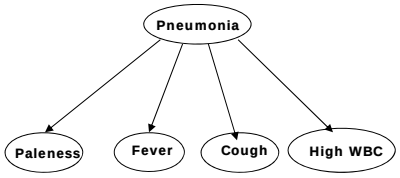


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 2: Ignore the rest

Fev
F
F
T
T
T

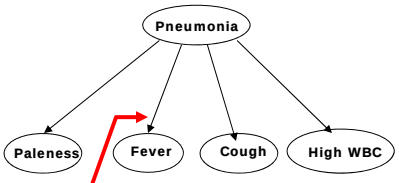


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 3a: **Learning the ML estimate**

Fev
F
F
T
T
T



$\mathbf{P}(Fever \mid Pneumonia = T)$

T	F
0.6	0.4

Learning of BBN parameters. Bayesian learning.

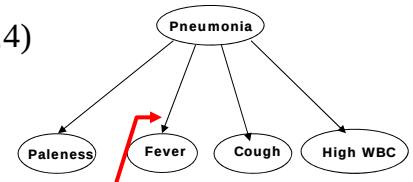
Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 3b: Learning the Bayesian estimate

Assume the prior

$$\theta_{Fever|Pneumonia=T} \sim Beta(3,4)$$

Fev
F
F
T
T
T



Posterior:

$$\theta_{Fever|Pneumonia=T} \sim Beta(6,6)$$