CS 1571 Introduction to AI
Lecture 25

Bayesian belief networks
Inference

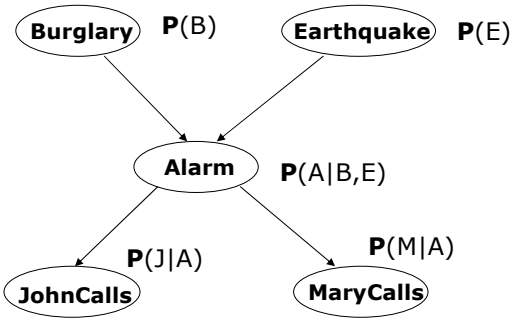
Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Administration

- **Homework assignment 10 is out and due on Wednesday**
- **Final exam:**
  - December 11, 2006
  - 12:00-1:50pm, 5129 Sennott Square

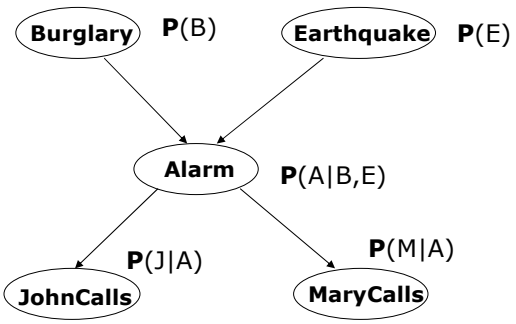
Bayesian belief network.

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 is influenced by Earthquake, The chance of John calling is affected by the Alarm

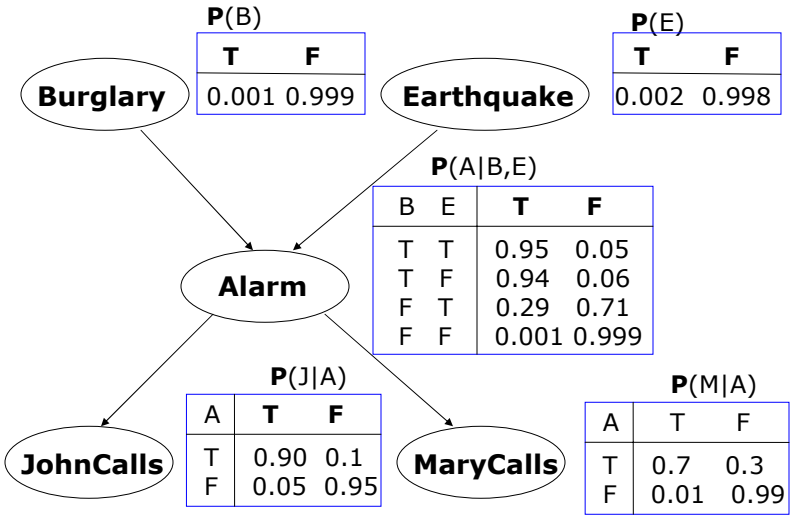


Bayesian belief network.

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



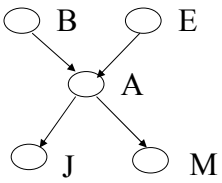
Bayesian belief network.



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:

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

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?**

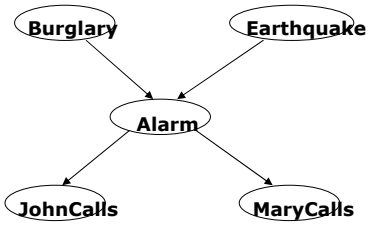
Alarm example: 5 binary (True, False) variables

# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

$$2^5 - 1 = 31$$



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

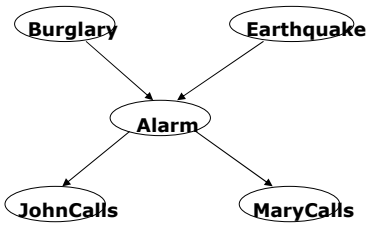
# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

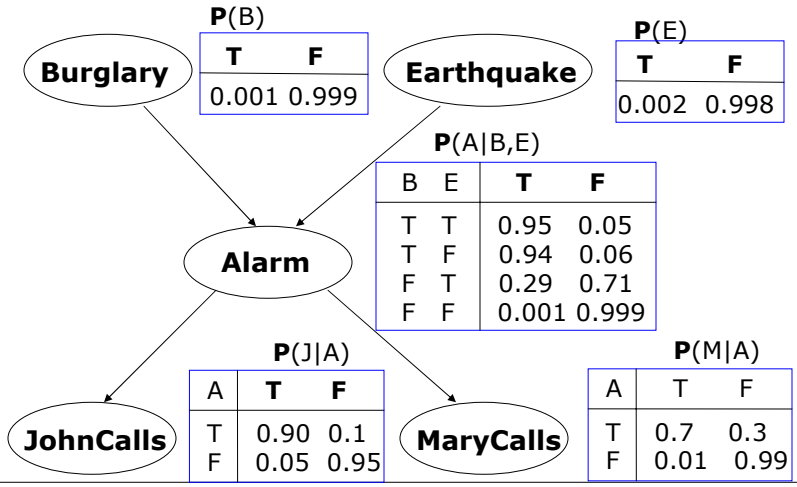
$$2^5 - 1 = 31$$

# of parameters of the BBN: ?



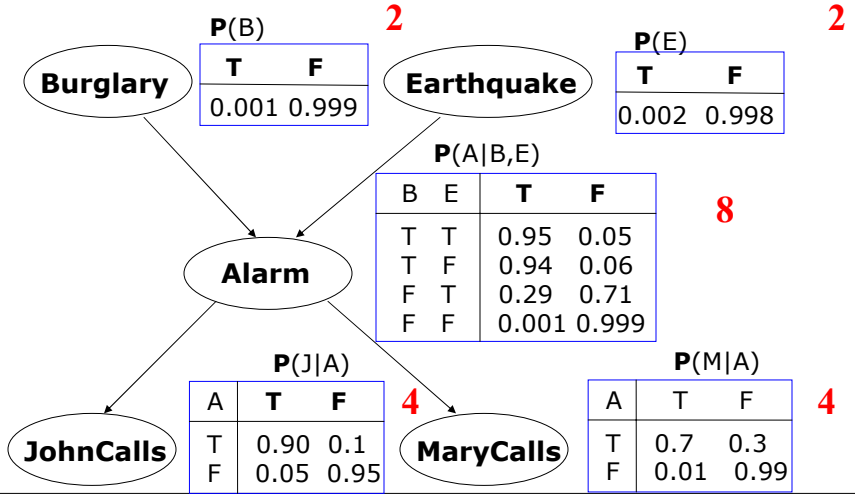
Bayesian belief network.

- 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:

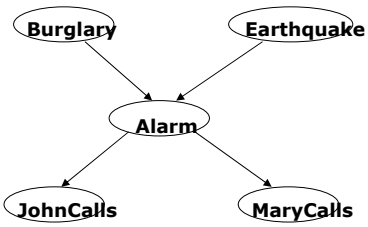
$$2^5 = 32$$

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:

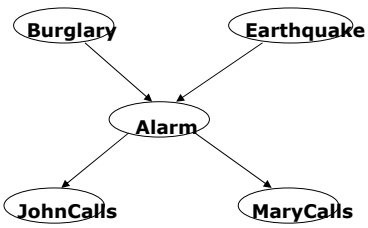
$$2^5 = 32$$

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)**

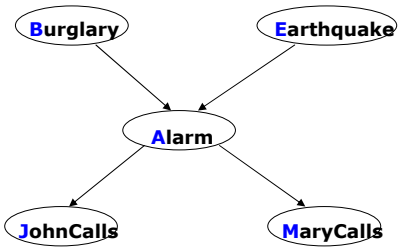
    $$\mathbf{P}(Burglary \mid JohnCalls = T)$$
  - **Prediction task. (from cause to effect)**

    $$\mathbf{P}(JohnCalls \mid Burglary = T)$$
  - **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 1571 Intro to AI M. Hauskrecht

Inference in Bayesian networks

- The smart interleaving of sums and products can help us to speed up the computation of joint probability queries
- What if we want to compute: $P(B = T, J = T)$

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

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

- A lot of shared computation
  - Smart cashing of results can save the time for more queries

Inference in Bayesian networks

- The smart interleaving of sums and products can help us to speed up the computation of joint probability queries
- What if we want to compute: $P(B = T, J = T)$

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

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

- A lot of shared computation
  - Smart cashing of results can save the time if more queries

CS 1571 Intro to AI M. Hauskrecht

Inference in Bayesian networks

- When cashing of results becomes handy?
- What if we want to compute a diagnostic query:

$$P(B = T \mid J = T) = \frac{P(B = T, J = T)}{P(J = T)}$$

- Exactly probabilities we have just compared !!
- There are other queries when cashing and ordering of sums and products can be shared and saves computation

$$\mathbf{P}(B \mid J = T) = \frac{\mathbf{P}(B, J = T)}{P(J = T)} = \alpha \mathbf{P}(B, J = T)$$

- General technique: **Recursive decomposition**

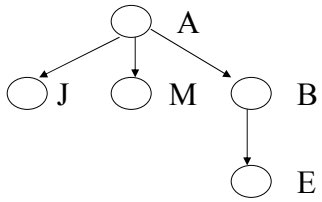
Inference in Bayesian networks

General idea:

$$P(True) = 1 =$$

$$= \sum_{a \in T,F} [\underbrace{\sum_{j \in T,F} P(J=j \mid A=a)}_{f_J(a)}][\underbrace{\sum_{m \in T,F} P(M=m \mid A=a)}_{f_M(a)}][\underbrace{\sum_{b \in T,F} P(B=b)[\underbrace{\sum_{e \in T,F} P(A=a \mid B=b, E=e)P(E=e)}_{f_E(a,b)}]}_{f_B(a)}]$$

Recursive decomposition:



Results cashed in the tree structure

Variable elimination

- **Recursive decomposition:**
  - Interleave sum and products before inference
- **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 $P(J=T)$

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

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

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

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

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

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

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

Inference in Bayesian network

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

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

Monte Carlo approaches

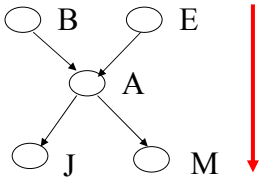
- **MC approximation**:
  - The probability is approximated using sample frequencies
  - **Example:**

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

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

N ← *total # samples*

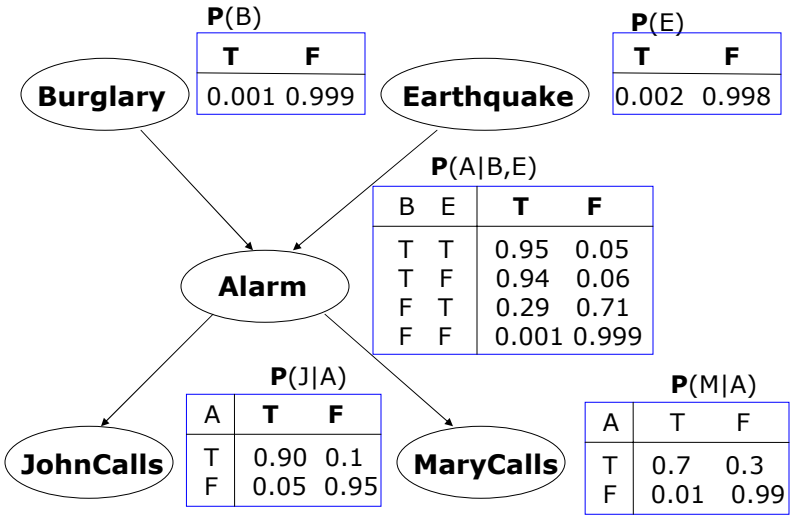
- **BBN sampling:**



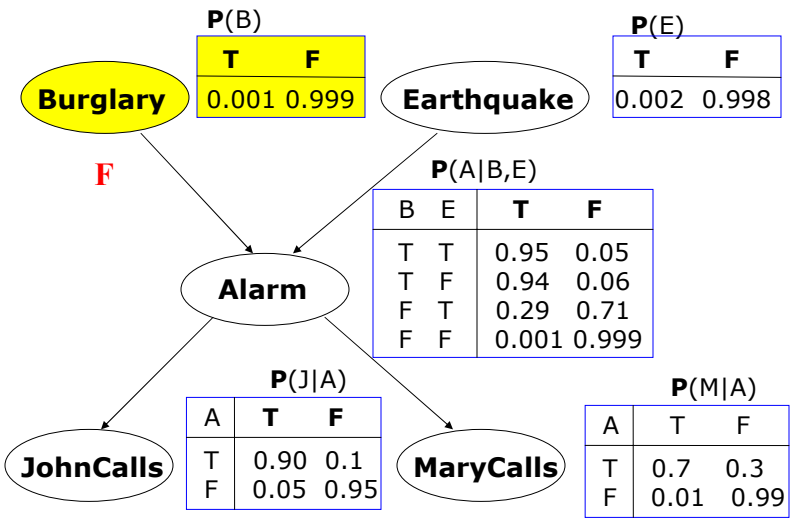
Generate sample in a top down manner, following the links

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

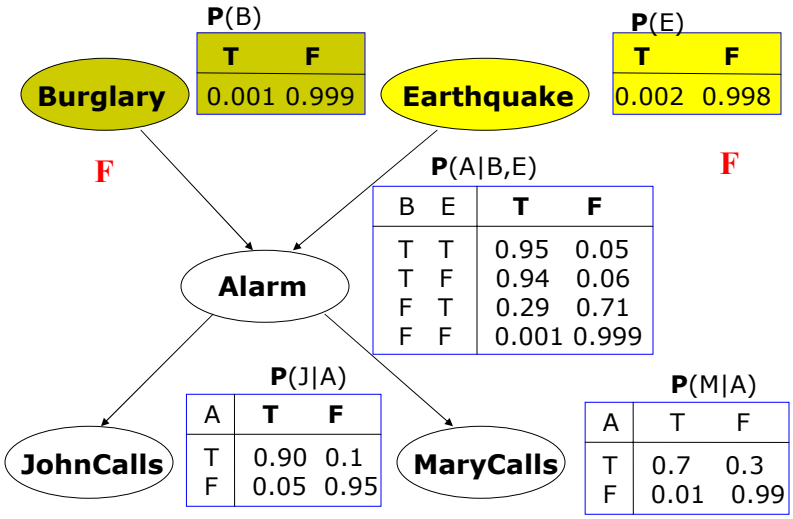
BBN sampling example



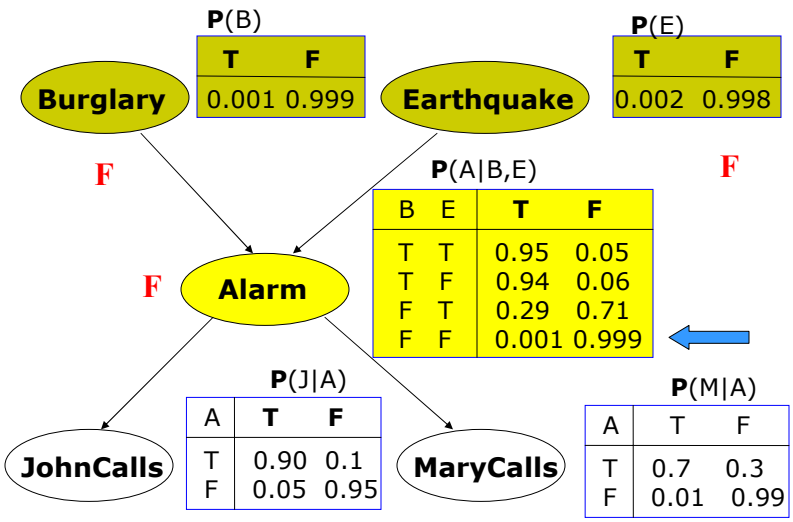
BBN sampling example



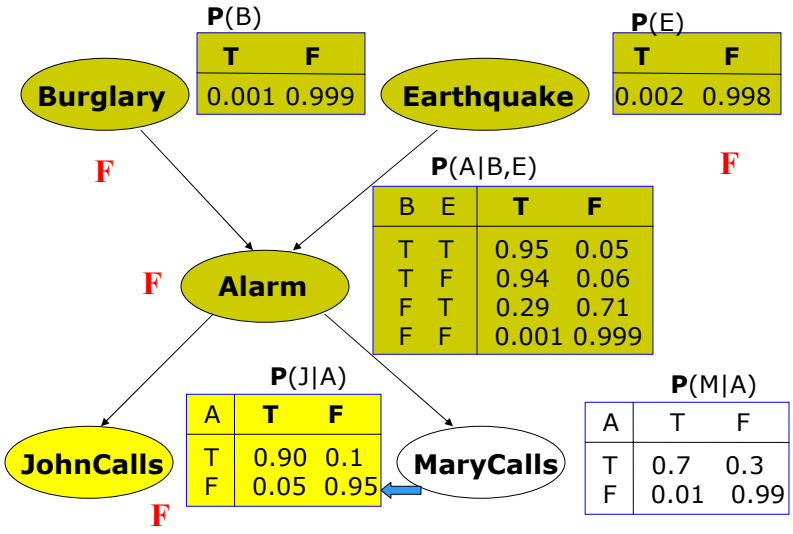
BBN sampling example



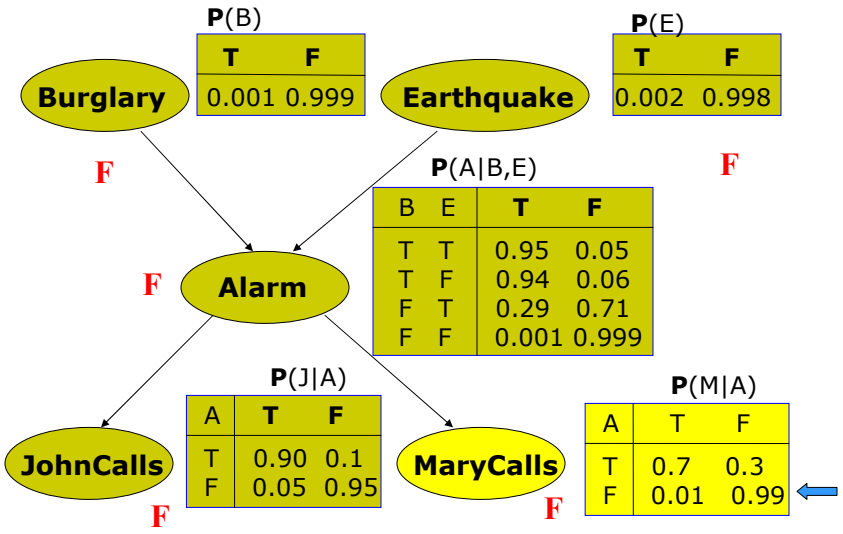
BBN sampling example



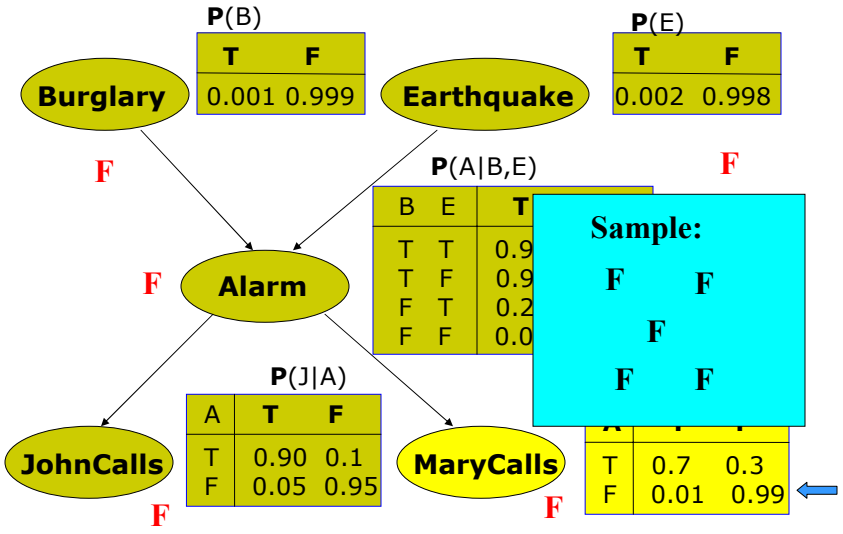
BBN sampling example



BBN sampling example



BBN sampling example



Monte Carlo approaches

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

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

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

$N_{J=T}$: # *samples with* $J=T$

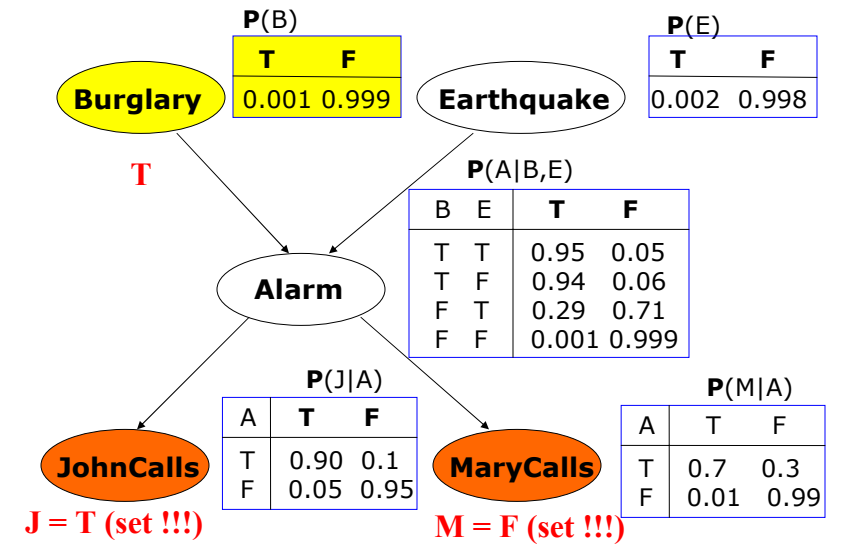
- **Rejection sampling:**
  - Generate samples from the full joint by sampling BBN
  - Use only samples that agree with the condition, the remaining samples are rejected
- **Problem:** many samples can be rejected

Likelihood weighting

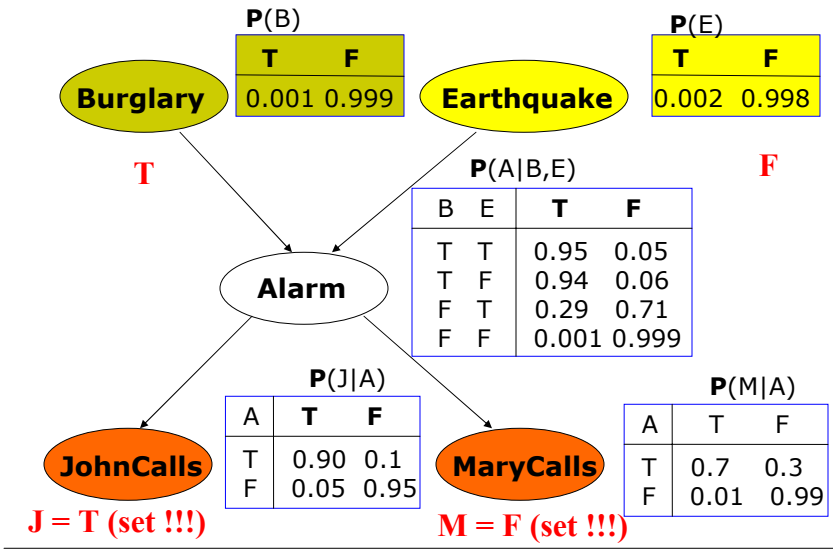
- **Avoids inefficiencies of rejection sampling**
  - **Idea:** generate only samples consistent with an evidence (or conditioning event)
  - **If the value is set no sampling**
- **Problem:** using simple counts is not enough since these may occur with different probabilities
- Likelihood weighting:
  - **With every sample keep a weight with which it should count towards the estimate**

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

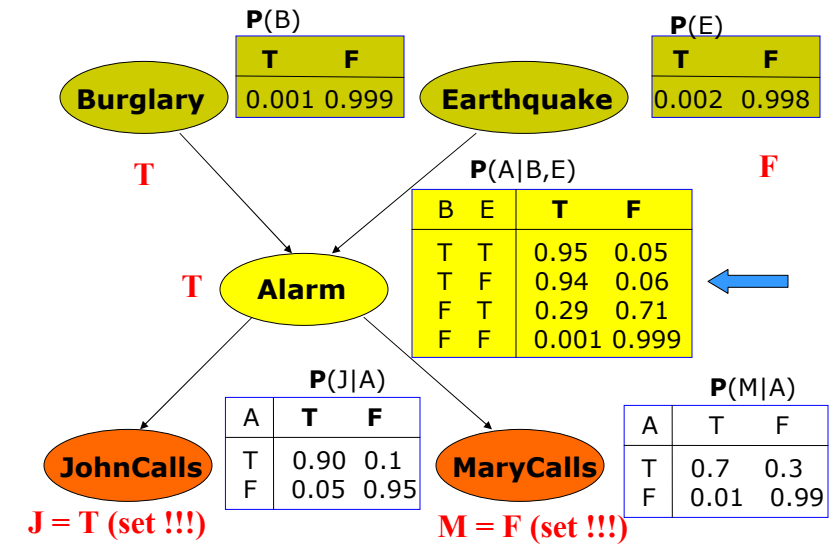
BBN likelihood weighting example



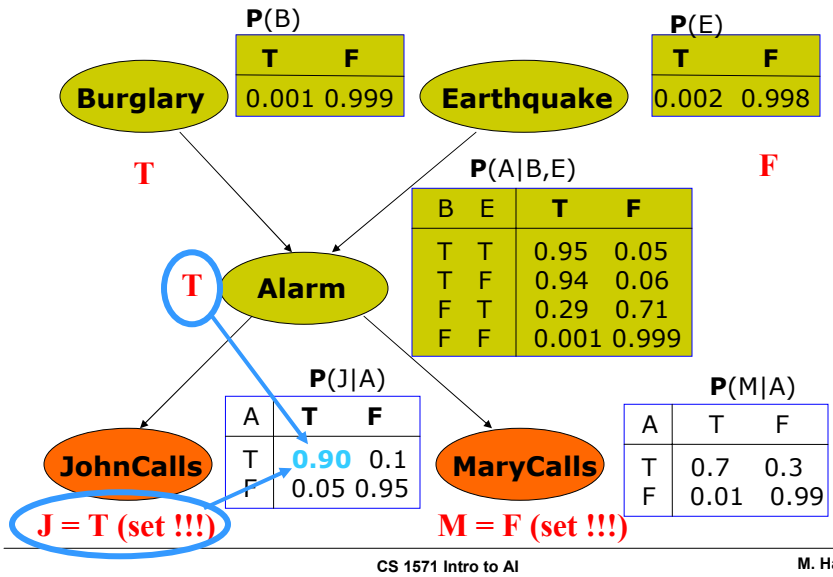
BBN likelihood weighting example



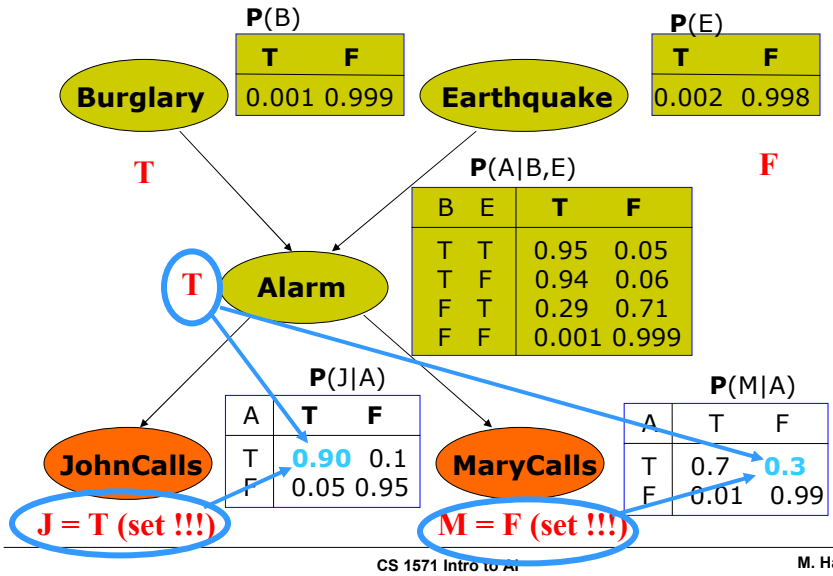
BBN likelihood weighting example



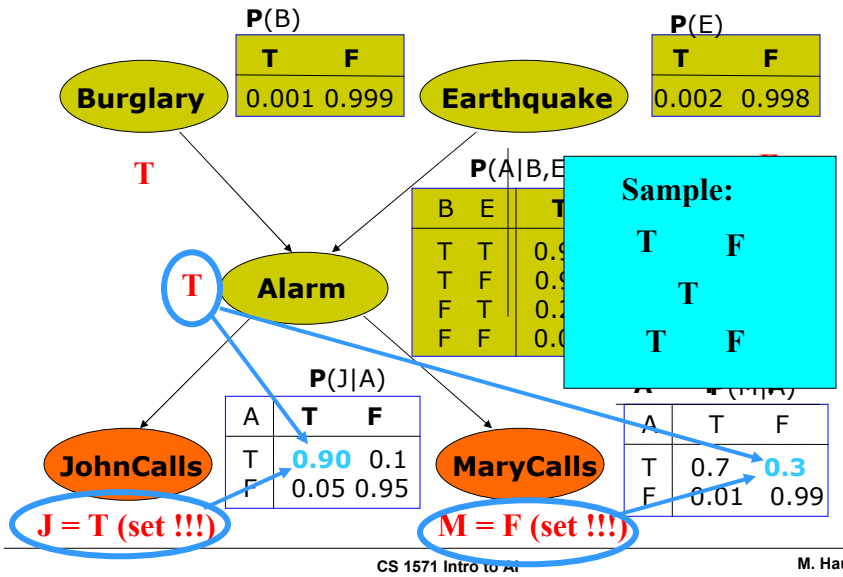
BBN likelihood weighting example



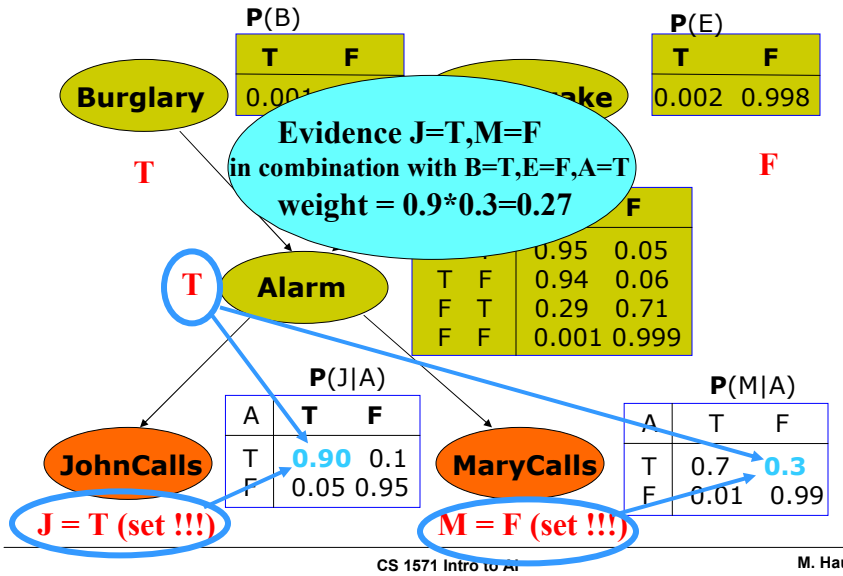
BBN likelihood weighting example



BBN likelihood weighting example

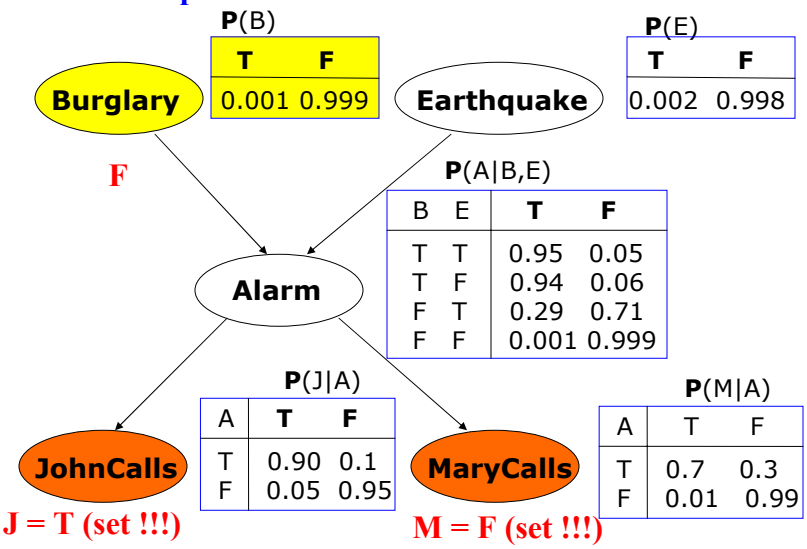


BBN likelihood weighting example



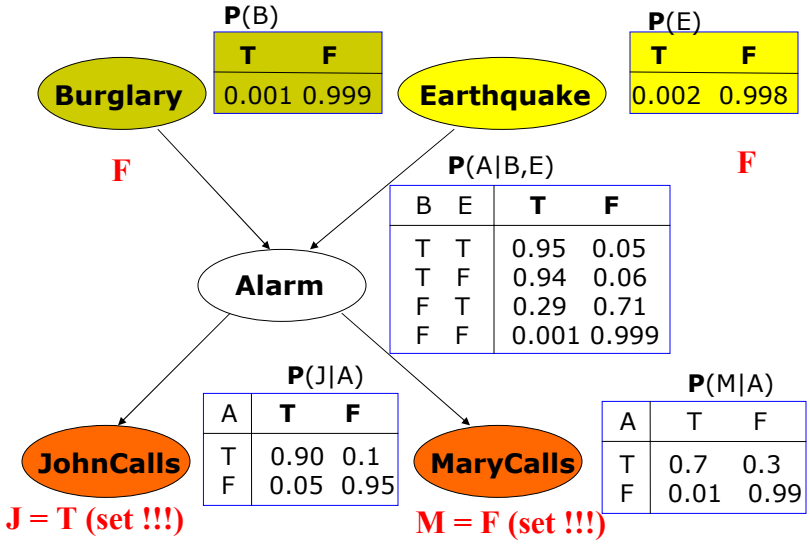
BBN likelihood weighting example

Second sample



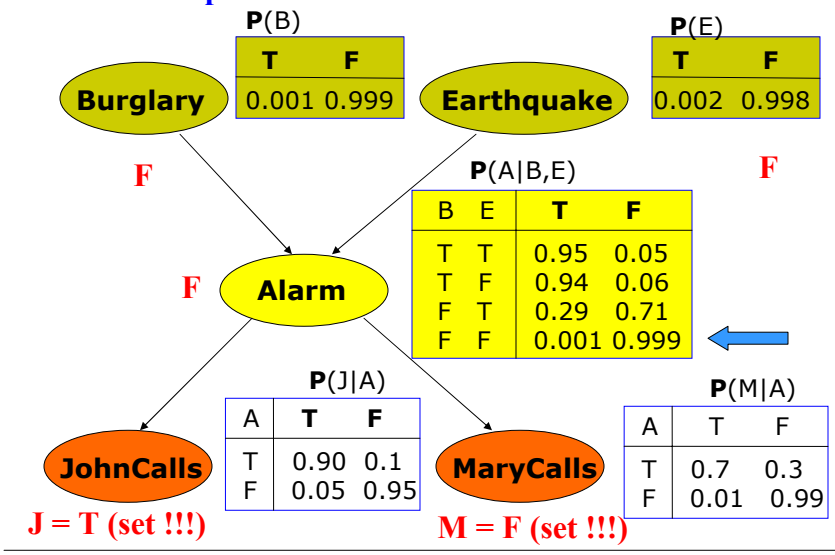
BBN likelihood weighting example

Second sample



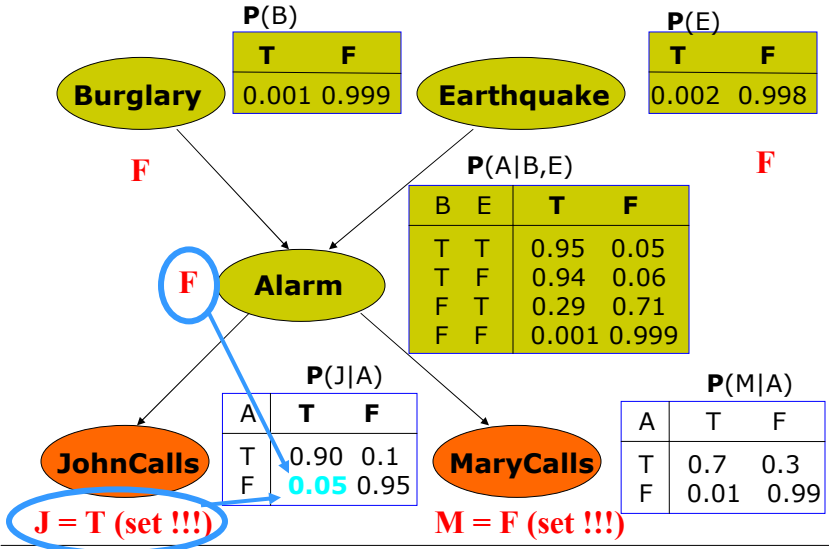
BBN likelihood weighting example

Second sample



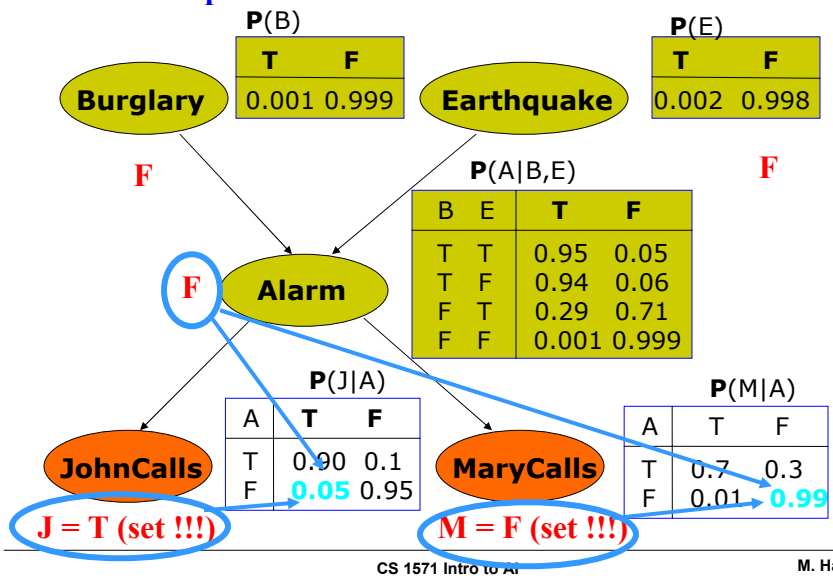
BBN likelihood weighting example

Second sample



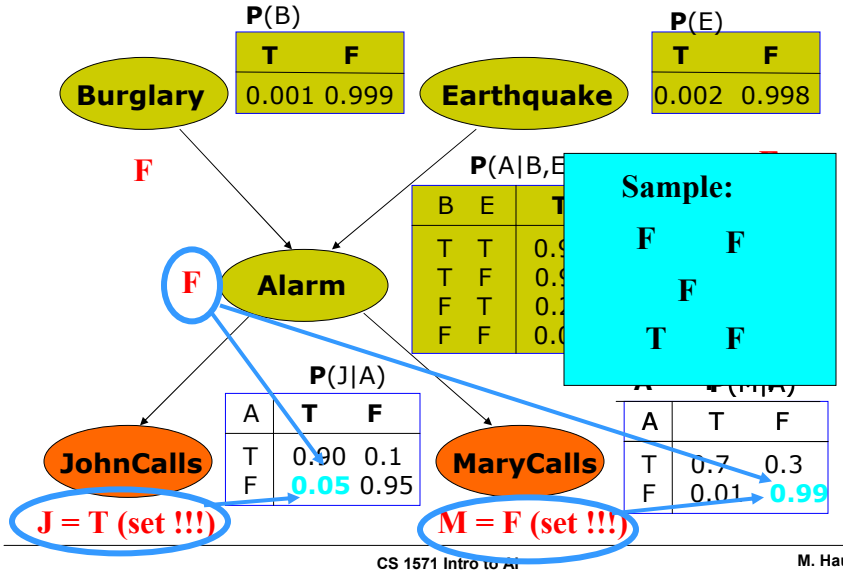
BBN likelihood weighting example

Second sample



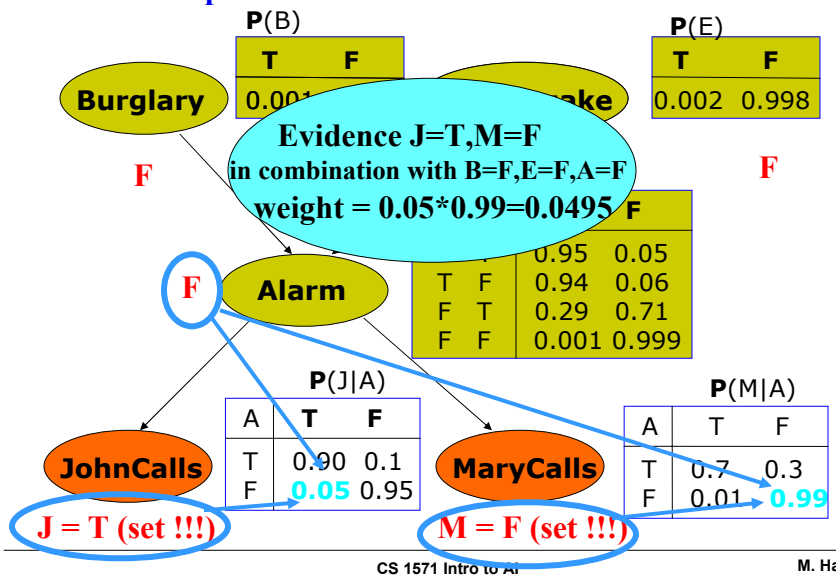
BBN likelihood weighting example

Second sample



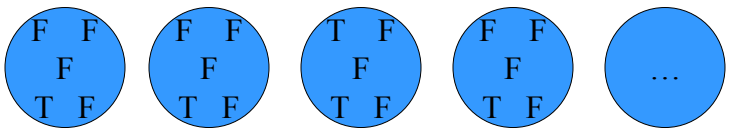
BBN likelihood weighting example

Second sample



Likelihood weighting

- Assume we have generated the following M samples:



How to make the samples consistent?

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