CS 2750 Machine Learning
Lecture 16

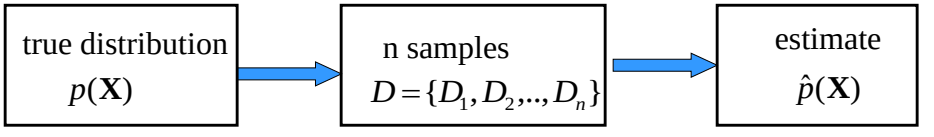
Bayesian belief networks III (Inference)

Milos Hauskrecht
milos@pitt.edu
5329 Sennott Square

Density estimation

Data: $D = \{D_1, D_2, .., D_n\}$
$D_i = \mathbf{x}_i$ a vector of attribute values

Objective: try to estimate the underlying true probability distribution over variables $\mathbf{X}$, $p(\mathbf{X})$, using examples in D



Standard (iid) assumptions: Samples
- **are independent of each other**
- **come from the same (identical) distribution (fixed $p(\mathbf{X})$)**

Bayesian belief networks (BBNs)

Bayesian belief networks (late 80s, beginning of 90s)

Key features:

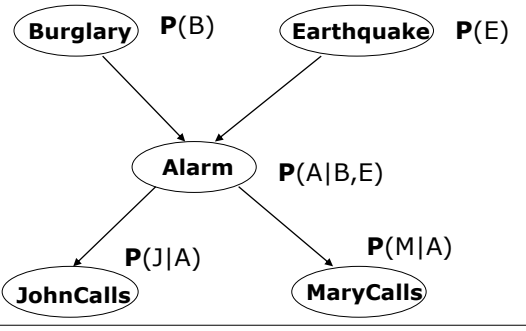
- Represent the full joint distribution over the variables more compactly with a **smaller number of parameters**.
- Take advantage of **conditional and marginal independences** among random variables
- **X and Y are independent** $P(X,Y) = P(X)P(Y)$
- **X and Y are conditionally independent given Z**

$$P(X,Y \mid Z) = P(X \mid Z)P(Y \mid Z)$$

$$P(X \mid Y,Z) = P(X \mid Z)$$

Bayesian belief network

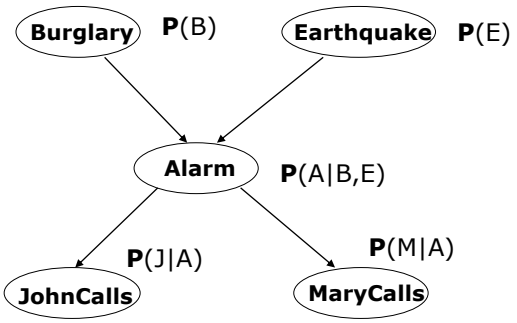
1. **Directed acyclic graph**
   - **Nodes** = random variables
     Burglary, Earthquake, Alarm, Mary calls and John calls
   - **Links** = direct (causal) dependencies between variables.
     The chance of Alarm being is influenced by Earthquake,
     The chance of John calling is affected by the Alarm



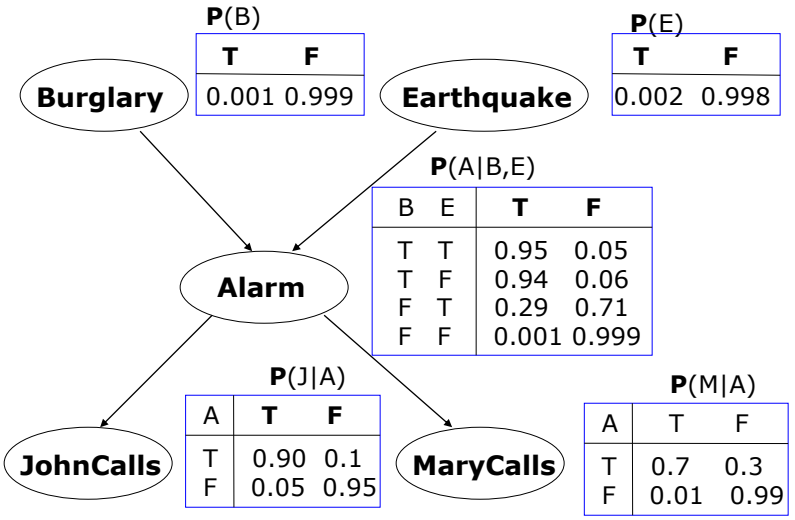
Bayesian belief network

2. Local conditional distributions

- relating variables and their parents



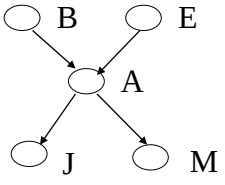
Bayesian belief network



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:

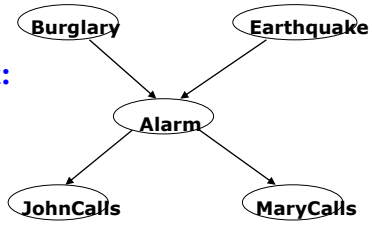
$$2^5 = 32$$

One parameter depends on the rest:

$$2^5 - 1 = 31$$

# of parameters of the BBN:

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



One parameter in every conditional depends on the rest:

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

Inference in Bayesian networks

- BBN models compactly the full joint distribution by taking advantage of existing independences between variables
- Simplifies the representation and learning of a model
- Can be used for solving various **inference tasks**:
  - **Diagnostic task. (from effect to cause)**

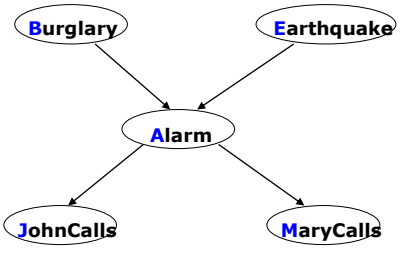
    $$\mathbf{P}(Burglary \mid JohnCalls = T)$$
  - **Prediction task. (from cause to effect)**

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

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

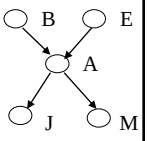
Inference in Bayesian network

- **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

$$P(J=T) =$$

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

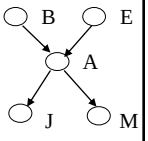
$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

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

$$P(J=T) =$$

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

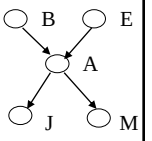
$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

Computational cost:

Number of additions: **15** (adding 16 terms)

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**

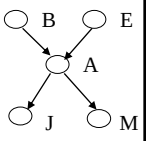
Number of products: 16*4=**64** (4 multiplications repeated 16 times)

Inference in Bayesian networks

How to compute sums and products more efficiently?

$$\sum_x af(x) = a\sum_x f(x)$$

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)

$$P(J=T) =$$

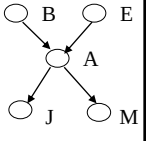
$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

- Use variable e and its sum using the rewrite

$$\sum_x af(x) = a\sum_x f(x)$$

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

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)

$$P(J=T) =$$

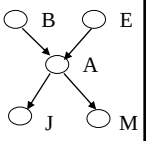
$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

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

• • •

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

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)

$$P(J=T)=$$

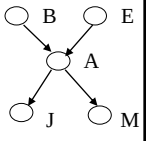
$$=\sum_{a\in T,F}P(J=T\mid A=a)\left[\sum_{m\in T,F}P(M=m\mid A=a)\right]\left[\sum_{b\in T,F}P(B=b)\left[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)\right]\right.$$

1

Computational cost:

Number of additions: ?

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)

$$P(J=T)=$$

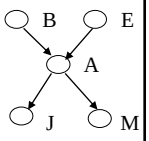
$$=\sum_{a\in T,F}P(J=T\mid A=a)\left[\sum_{m\in T,F}P(M=m\mid A=a)\right]\left[\sum_{b\in T,F}P(B=b)\left[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)\right]\right.$$

2*1

Computational cost:

Number of additions: ?

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)

$$P(J=T) =$$

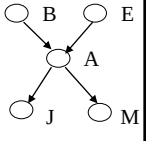
$$= \sum_{a \in T,F} P(J=T \mid A=a) \left[\sum_{m \in T,F} P(M=m \mid A=a) \right] \left[\sum_{b \in T,F} P(B=b) \left[\sum_{e \in T,F} P(A=a \mid B=b, E=e) P(E=e) \right] \right]$$

2*2*1

Computational cost:

Number of additions: ?

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)

$$P(J=T) =$$

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

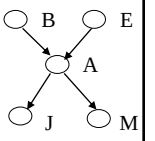
2*1

2*2*1

Computational cost:

Number of additions: ?

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)

$$P(J=T)=$$

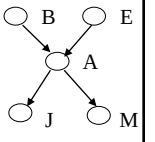
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

1 2*1 2*1 2*2*1

Computational cost:

Number of additions: **?**

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)

$$P(J=T)=$$

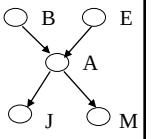
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

1 2*1 2*1 2*2*1

Computational cost:

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

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)

$$P(J=T) =$$

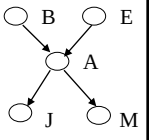
$$= \sum_{a \in T,F} P(J=T \mid A=a) \left[\sum_{m \in T,F} P(M=m \mid A=a) \right] \left[\sum_{b \in T,F} P(B=b) \left[\sum_{e \in T,F} P(A=a \mid B=b, E=e) P(E=e) \right] \right]$$

1

Computational cost:

Number of products: ?

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)

$$P(J=T) =$$

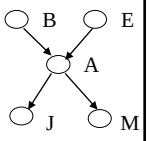
$$= \sum_{a \in T,F} P(J=T \mid A=a) \left[\sum_{m \in T,F} P(M=m \mid A=a) \right] \left[\sum_{b \in T,F} P(B=b) \left[\sum_{e \in T,F} P(A=a \mid B=b, E=e) P(E=e) \right] \right]$$

2*2 *2*1

Computational cost:

Number of products: ?

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)

$$P(J=T)=$$

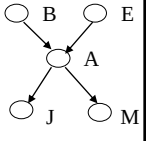
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

2*2 2*2*1 2*2 *2*1

Computational cost:

Number of products: ?

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)

$$P(J=T)=$$

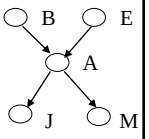
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

2*2 2*2*1 2*2 *2*1

Computational cost:

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

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)

$$P(J=T)=$$

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

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

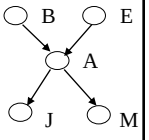
$$=\sum_{a\in T,F}P(J=T\mid A=a)\left[\sum_{m\in T,F}P(M=m\mid A=a)\right]\left[\sum_{b\in T,F}P(B=b)\left[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

Computational cost:

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

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

Variable elimination

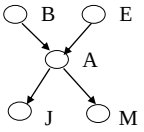


- **Variable elimination:**
  - Similar idea but interleave sum and products one variable at the time during inference
  - E.g. Query $P(J=T)$ requires to eliminate A,B,E,M and this can be done in different order

$$P(J=T)=$$

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

Variable elimination



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

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

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

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

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

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

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

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

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

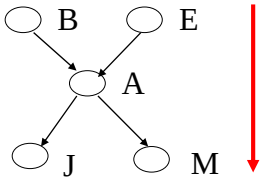
- **MC approximation**:
  - The probability is approximated using sample frequencies
  - **Example:**

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

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

N ← *total # samples*

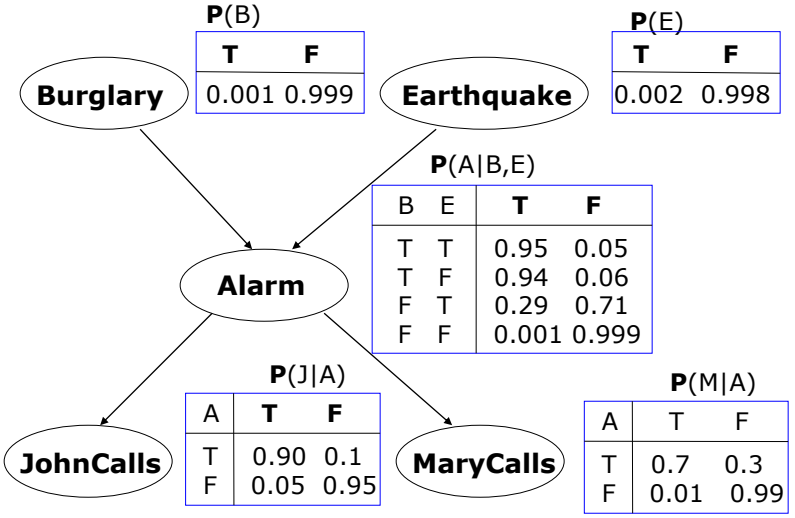
- **Sample generation: BBN sampling of the joint is easy**



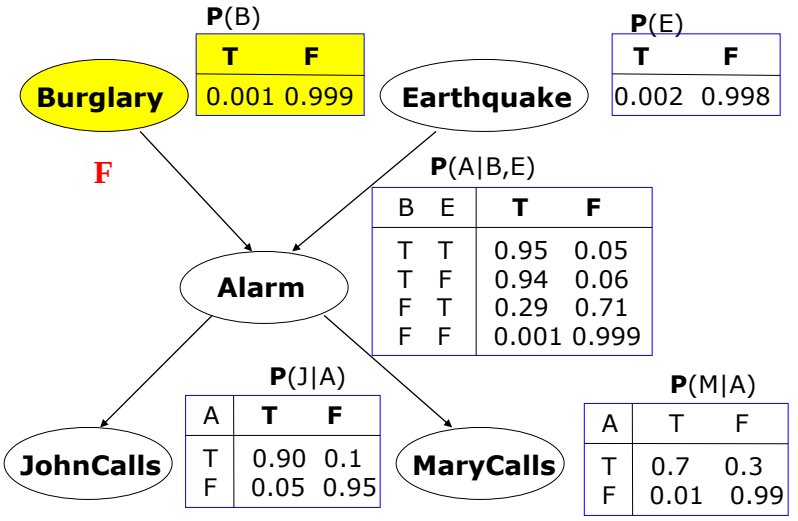
Generate sample in a top down manner, following the links

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

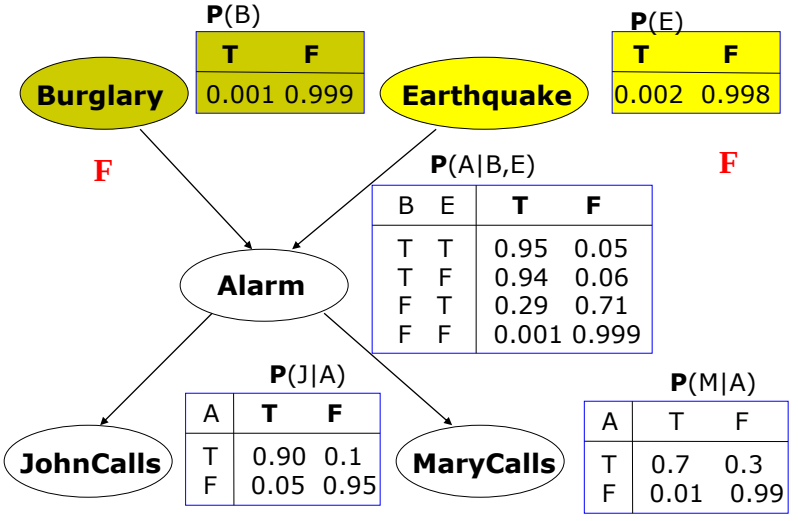
BBN sampling example



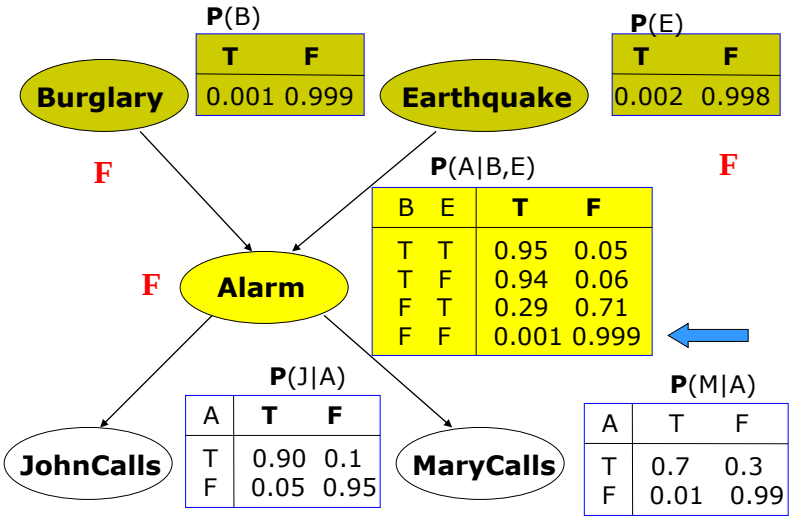
BBN sampling example



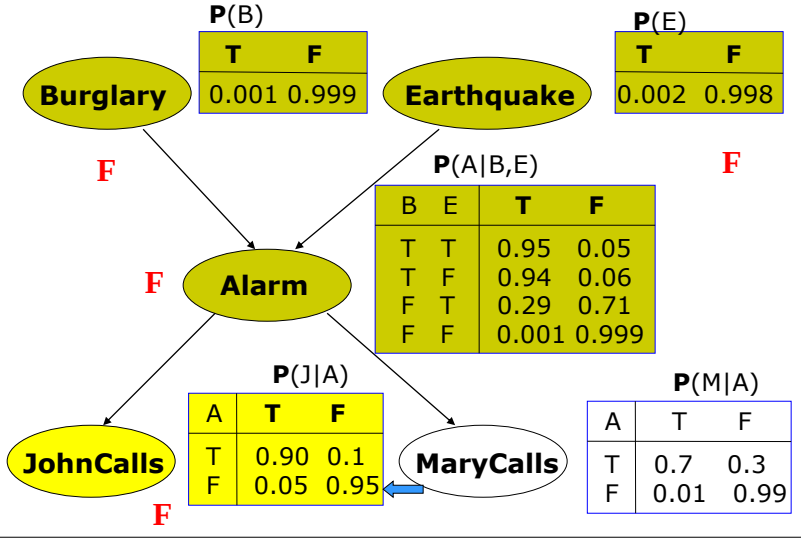
BBN sampling example



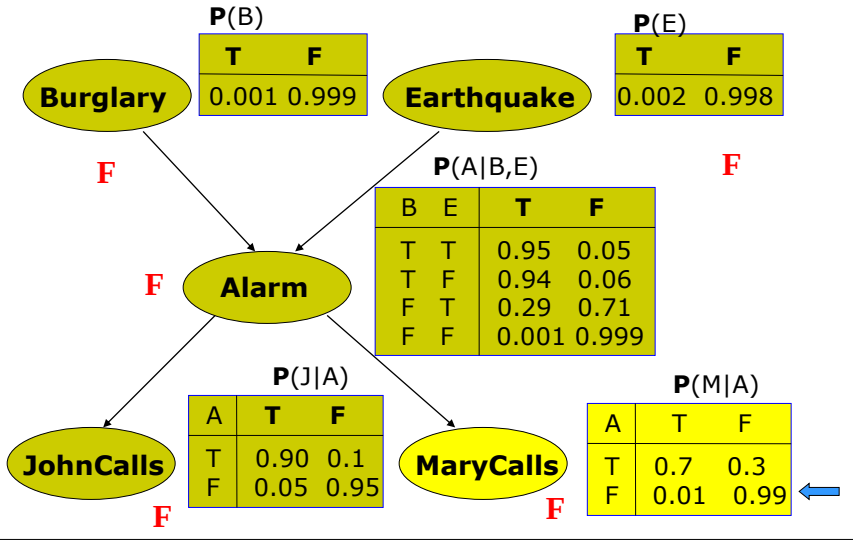
BBN sampling example



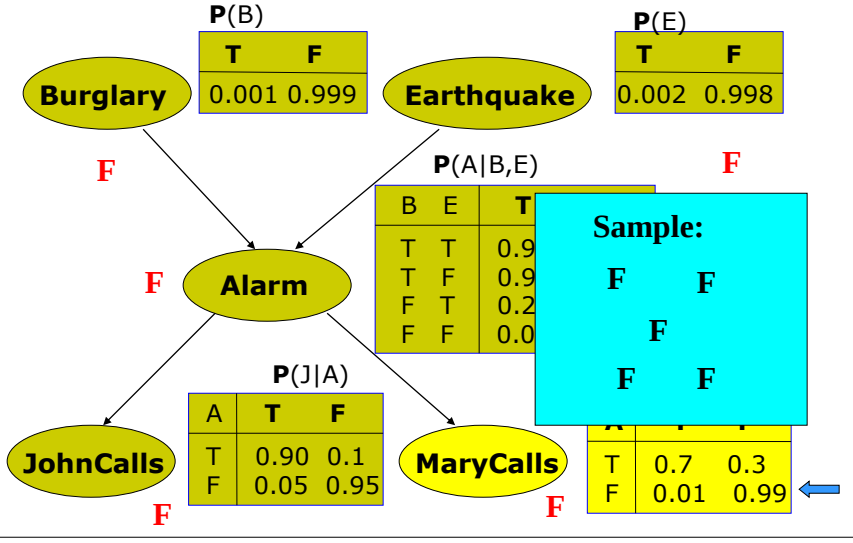
BBN sampling example



BBN sampling example



BBN sampling example



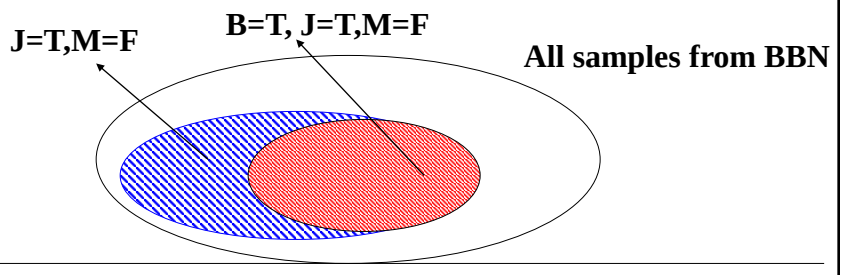
Monte Carlo approaches

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

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

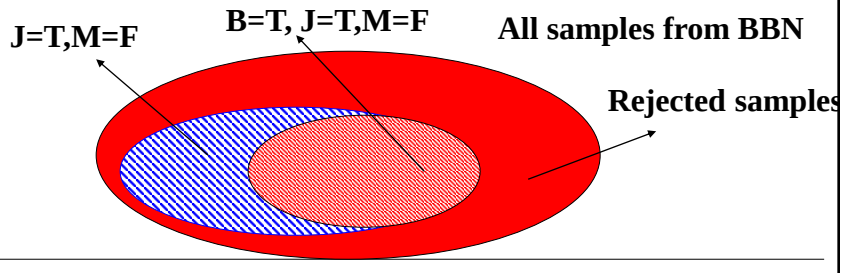
$\#\,samples\ with\ B=T, J=T, M=F$

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



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:

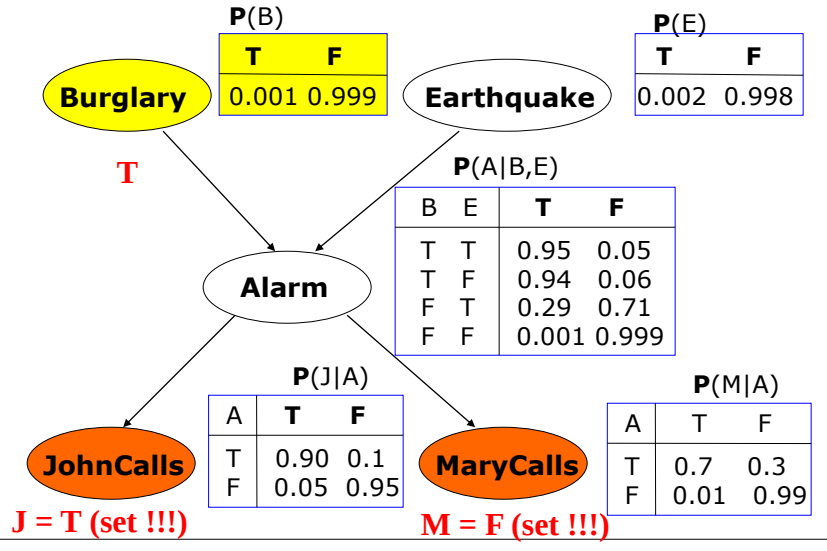
- **the distribution generated by enforcing the conditioning variables to set values is biased**
- simple counts are not sufficient to estimate the probabilities

Solution:

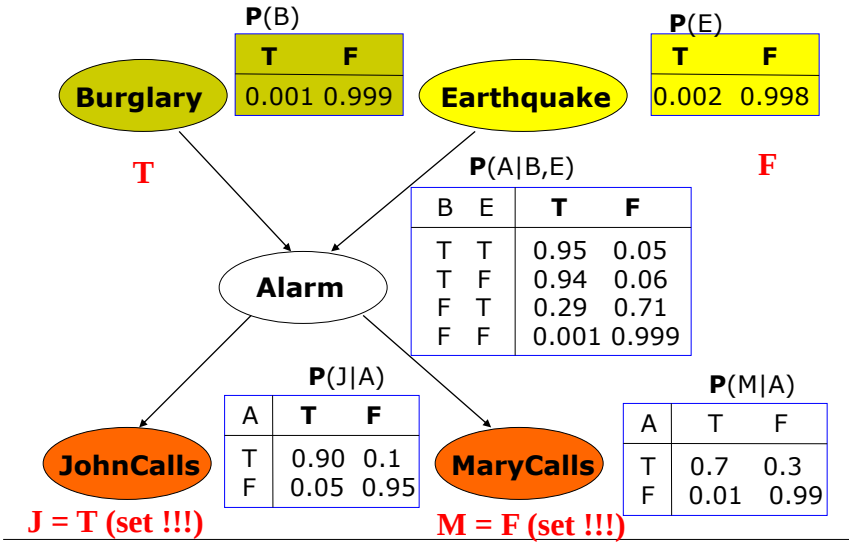
- **With every sample keep a weight with which it should count towards the estimate**

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

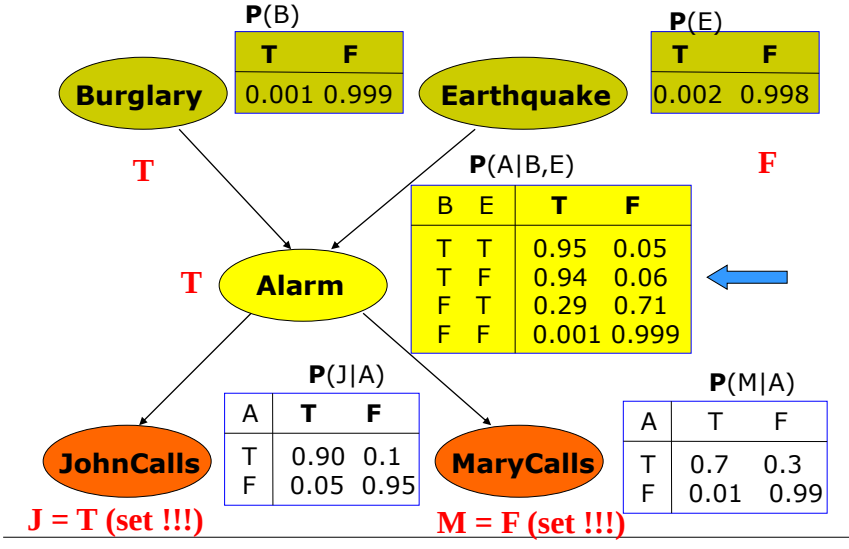
BBN likelihood weighting example



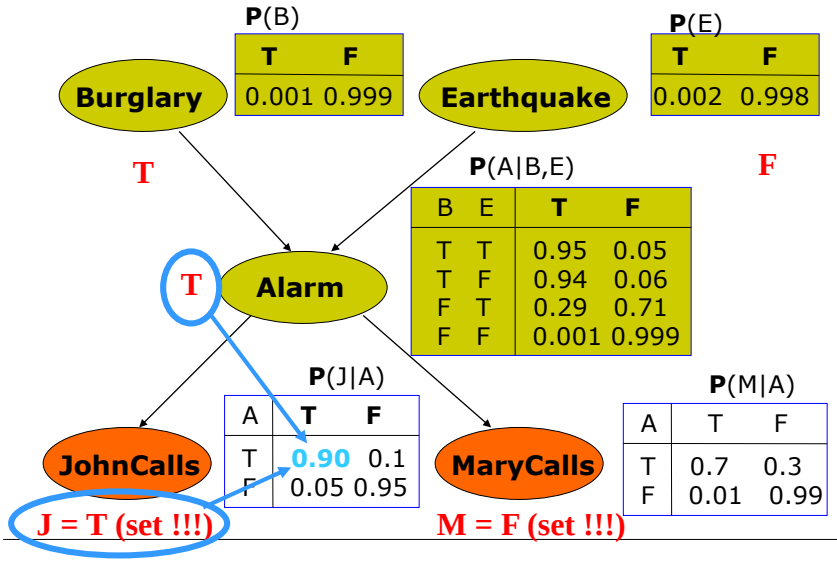
BBN likelihood weighting example



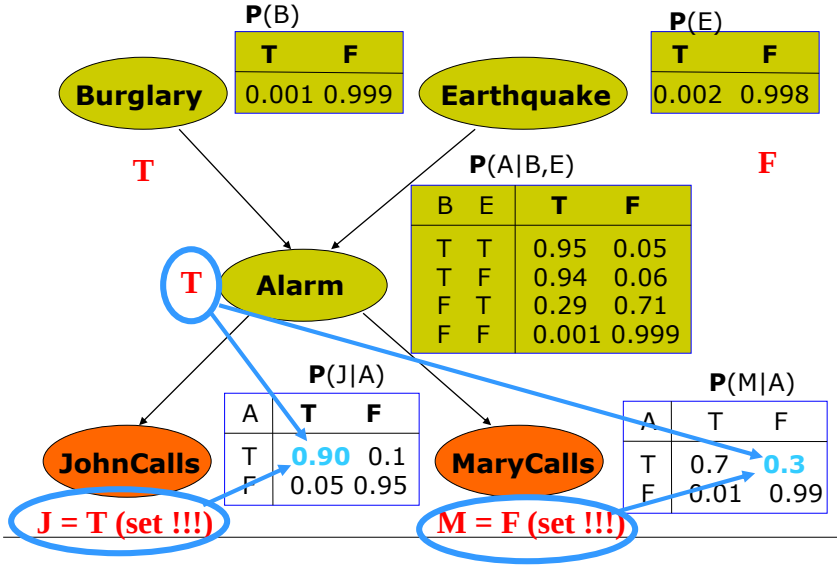
BBN likelihood weighting example



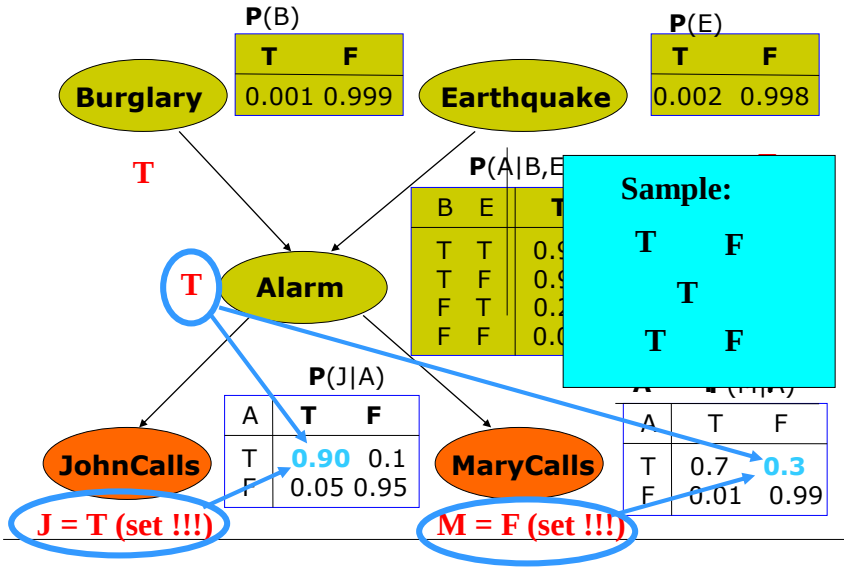
BBN likelihood weighting example



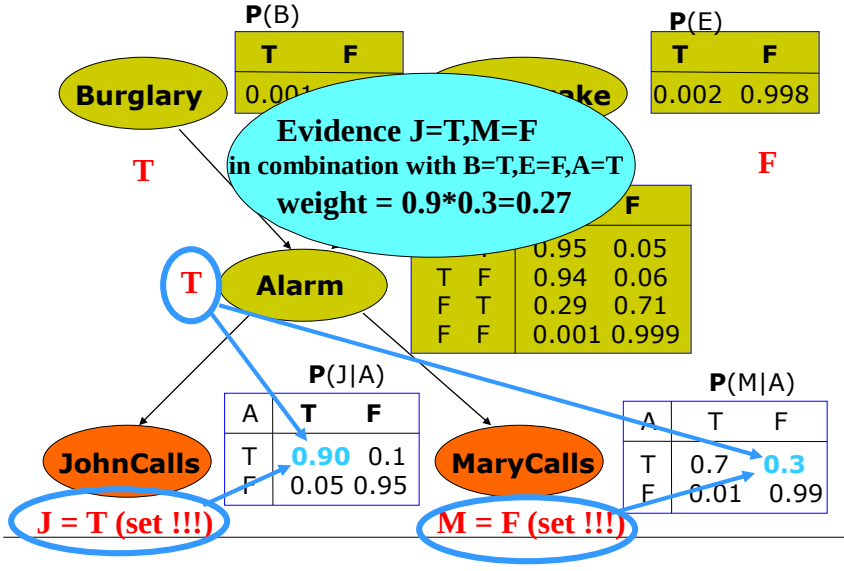
BBN likelihood weighting example



BBN likelihood weighting example

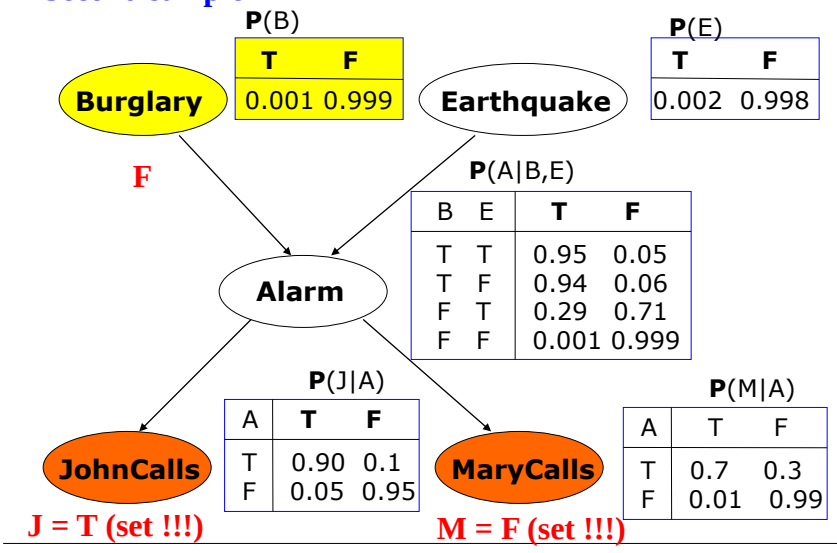


BBN likelihood weighting example



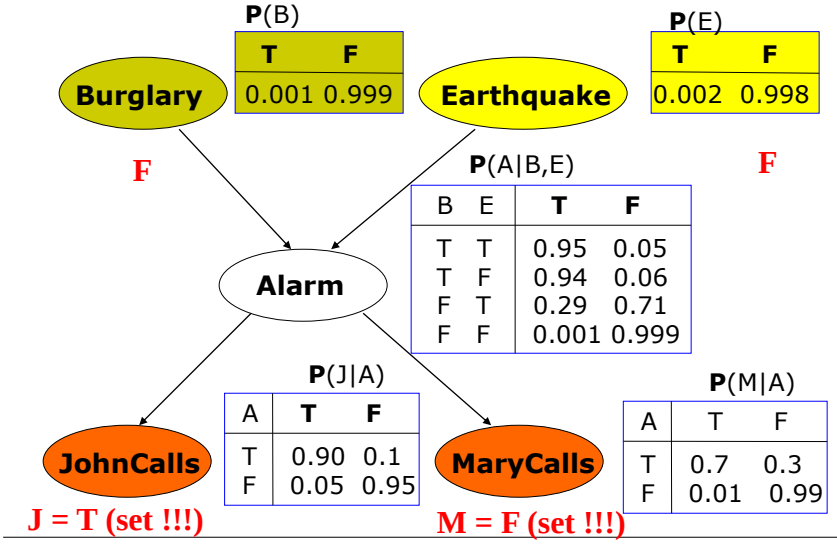
BBN likelihood weighting example

Second sample



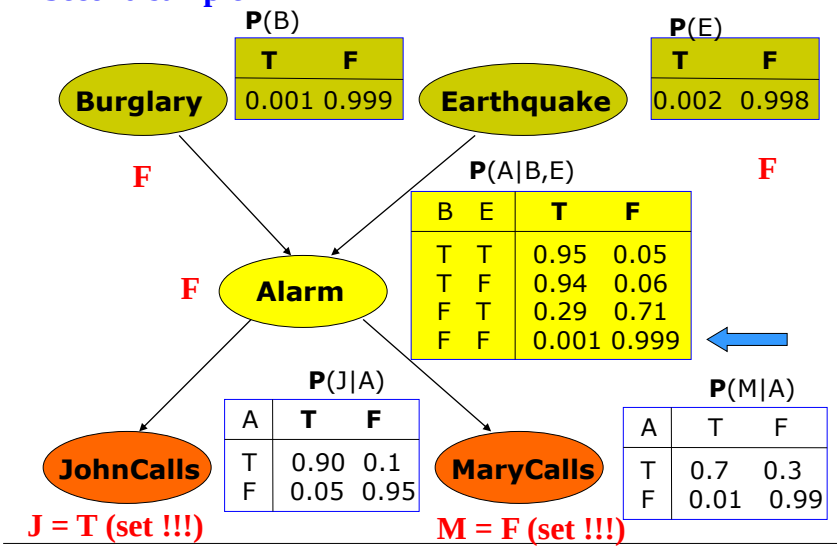
BBN likelihood weighting example

Second sample



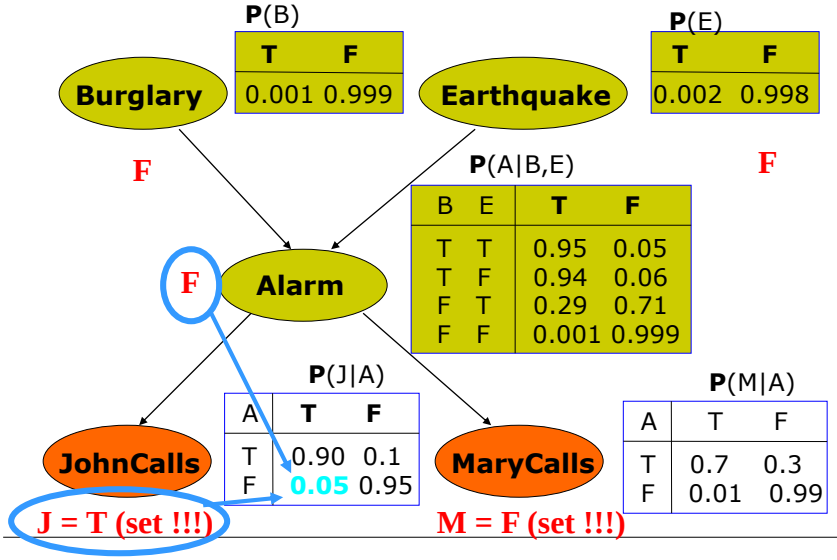
BBN likelihood weighting example

Second sample



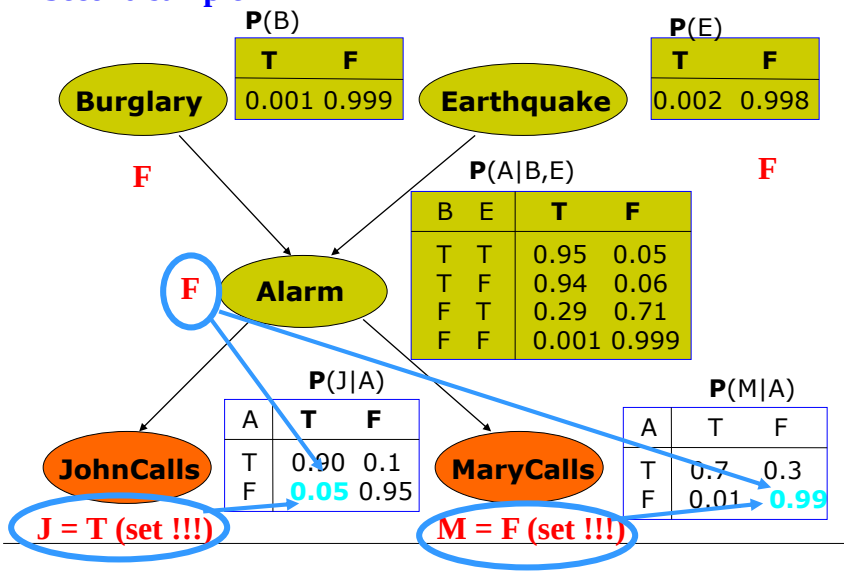
BBN likelihood weighting example

Second sample



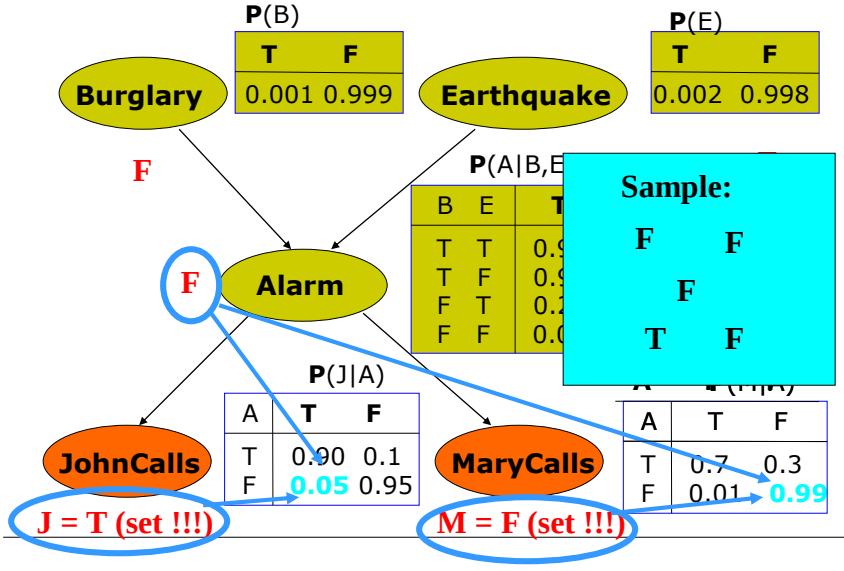
BBN likelihood weighting example

Second sample



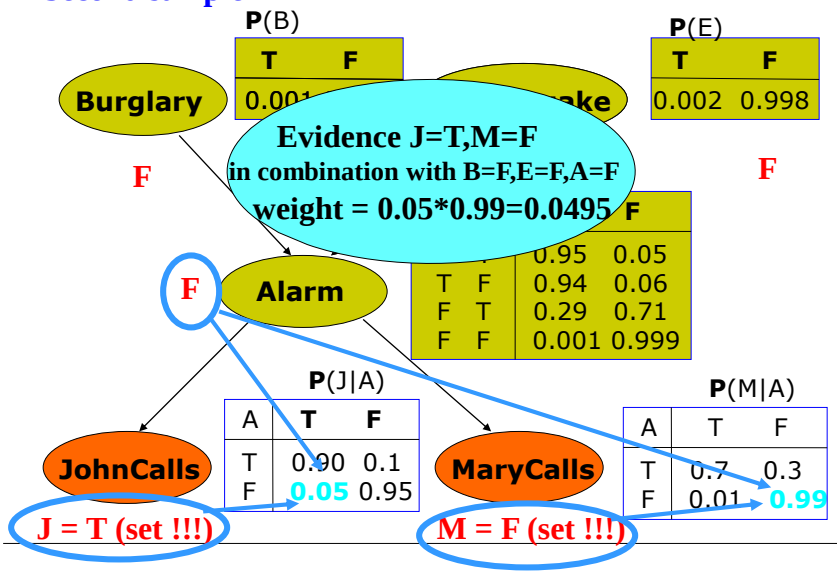
BBN likelihood weighting example

Second sample



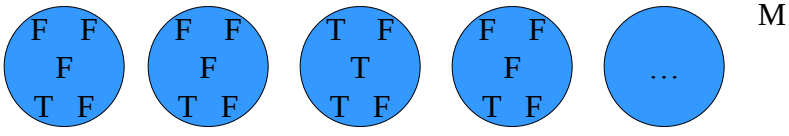
BBN likelihood weighting example

Second sample



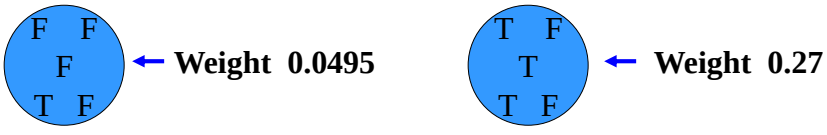
Likelihood weighting

- Assume we have generated the following M samples:



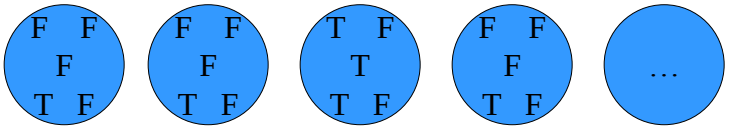
How to make the samples consistent?

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



Likelihood weighting

- Assume we have generated the following M samples:



M

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

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