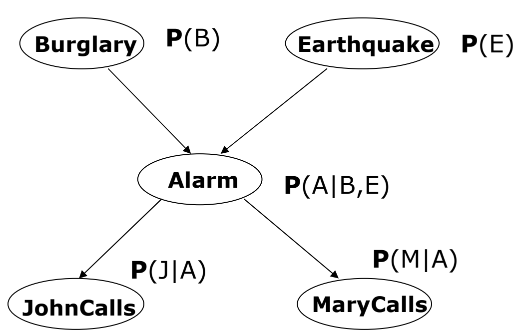
CS 2750 Machine Learning
Lecture 18

Bayesian belief networks. Inference and Learning

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

Bayesian belief network.

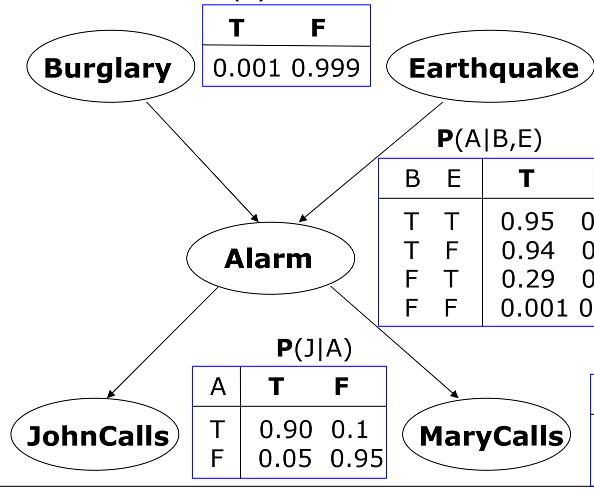
1. **Directed acyclic graph**
   - **Nodes** = random variables
   - **Links** = missing links encode independences.



Bayesian belief network

2. Local conditional distributions

- relate variables and their parents



P(B)

T	**F**
0.001	0.999

P(E)

T	**F**
0.002	0.998

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	0.90	0.1
F	0.05	0.95

P(M|A)

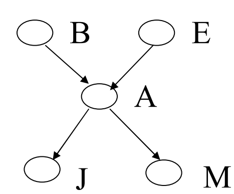
A	T	F
T	0.7	0.3
F	0.01	0.99

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

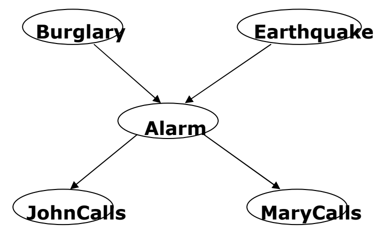
# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

$$2^5 - 1 = 31$$

# of parameters of the BBN: ?



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:

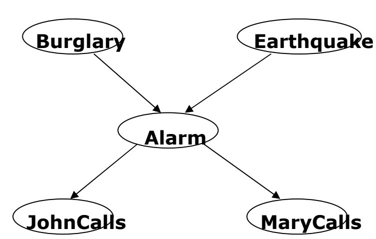
$$2^5 = 32$$

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:

?

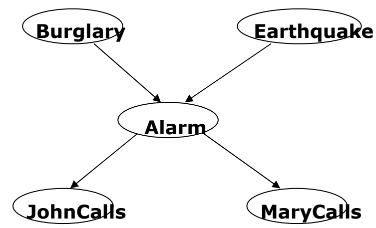
Parameter complexity problem

- 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
  – Smaller number of parameters
- 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)**

$$\mathbf{P}(JohnCalls \mid Burglary = T)$$

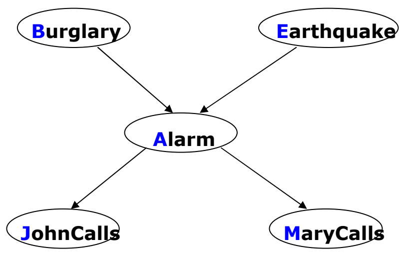
  – **Other probabilistic queries** (queries on joint distributions).

$$\mathbf{P}(Alarm)$$

- **Question:** Can we take advantage of independences to construct special algorithms and speedup 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

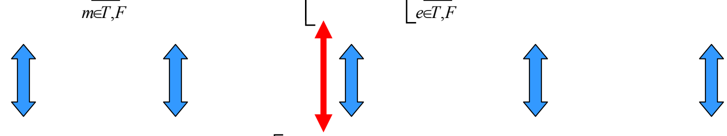
Inference in Bayesian networks

- 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: $P(B=T, J=T)$

$$P(B=T,J=T)=$$

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

$$P(J=T)=$$



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

- 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

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**

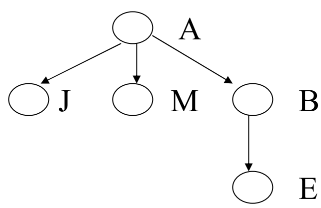
Inference in Bayesian networks

General idea:

$$P(True) = 1 =$$

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

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 the inference
  - Typically relies on a special structure (called **joint tree**) that groups together multiple variables
  - E.g. Query $P(J=T)$ requires to eliminate A,B,E,M and this can be done in different order

$$P(J=T) =$$

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

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**
  - **Recursive decomposition** (Cooper, Darwiche)
  - Symbolic inference (D'Ambrosio)
  - Belief propagation algorithm (Pearl)
  - Arc reversal (Olmsted, Schachter)
- **Approximate inference algorithms:**
  - **Monte Carlo methods:**
    - Forward sampling, Likelihood sampling
  - Variational methods

Monte Carlo approaches

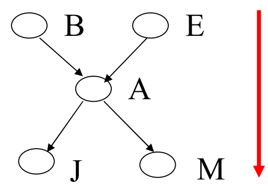
- **MC approximation:**
  - The probability is approximated using sample frequencies
  - **Example:**

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

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

N ← *total # examples*

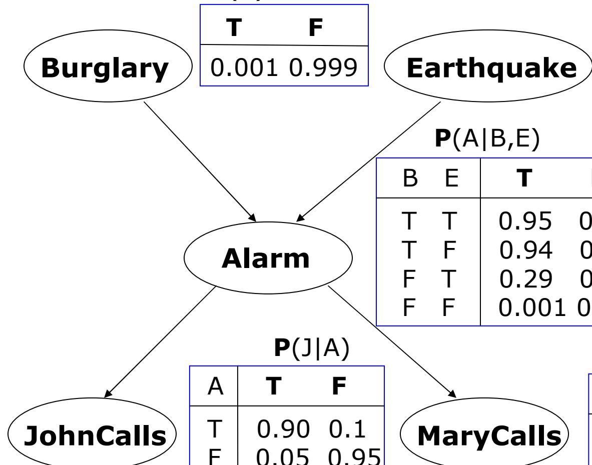
- **BBN sampling:**



Generate sample in the top down manner, following the links

- <u>**One example gives one assignment of values to all variables**</u>

BBN sampling example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

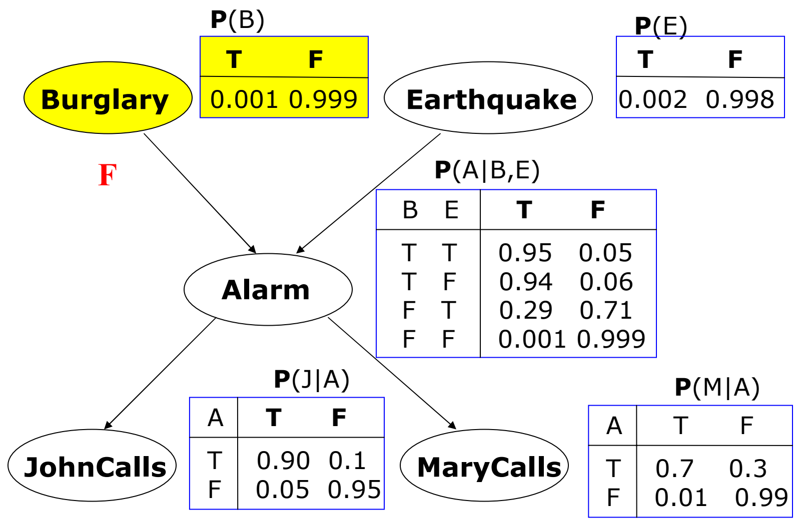
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN sampling example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

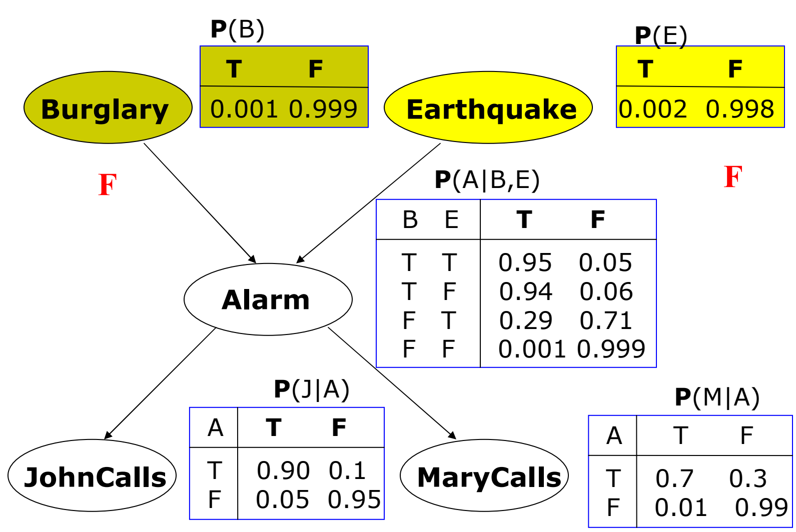
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN sampling example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

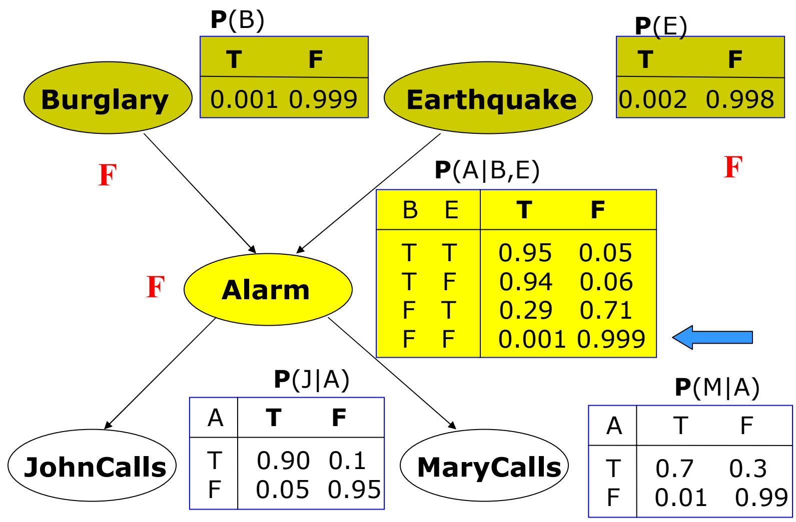
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN sampling example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

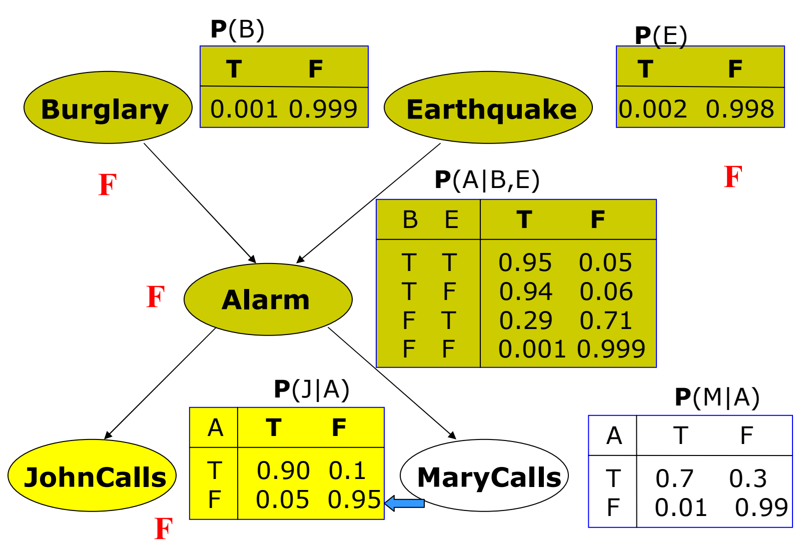
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN sampling example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

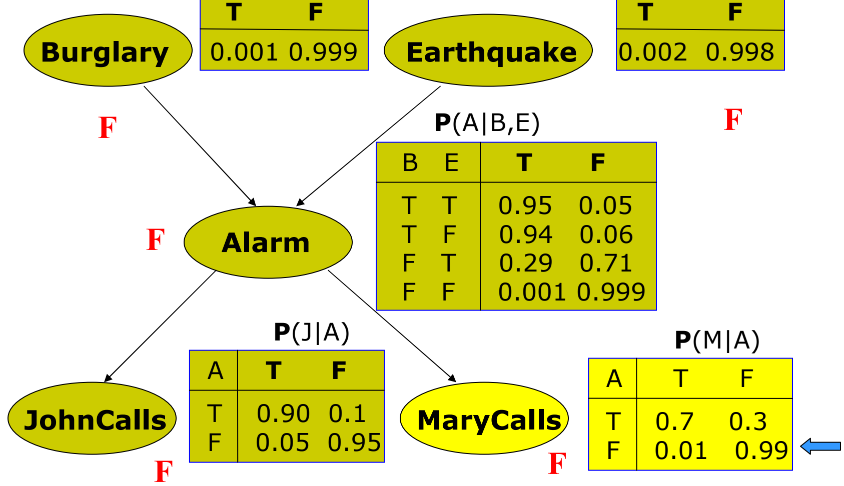
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN sampling example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

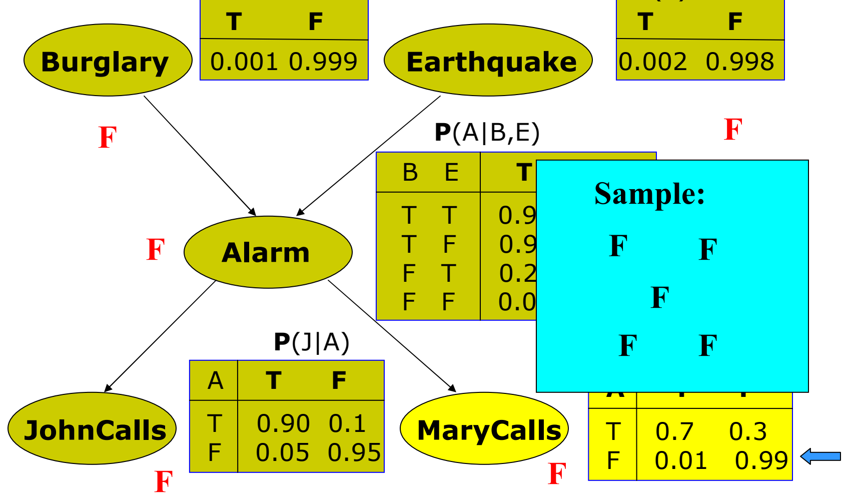
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN sampling example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

P(A|B,E)

B	E	T
T	T	0.9
T	F	0.9
F	T	0.2
F	F	0.0

P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

T	0.7	0.3
F	0.01	0.99

Monte Carlo approaches

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

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

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

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

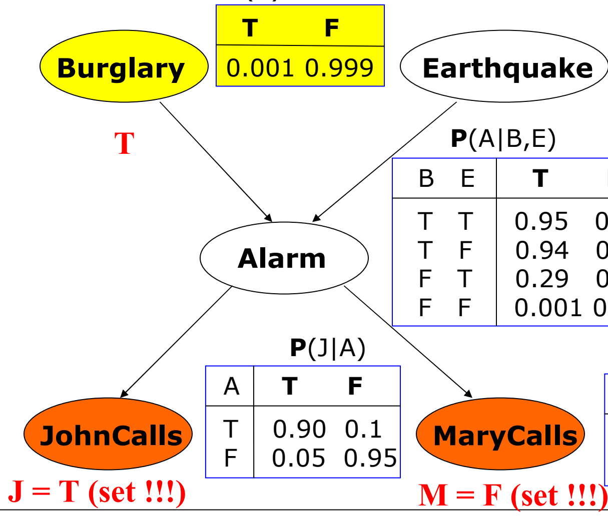
- **Rejection sampling:**
  - Generate samples from the full joint by sampling BBN
  - Use only samples that agree with the condition, the remaining samples are rejected
- **Problem:** many samples can be rejected

Likelihood weighting

- **Avoids inefficiencies of rejection sampling**
  - **Idea:** generate only samples **consistent with the evidence** (or conditioning event)
    - **the value of evidence nodes is not sampled**
- **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**

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

BBN likelihood weighting example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

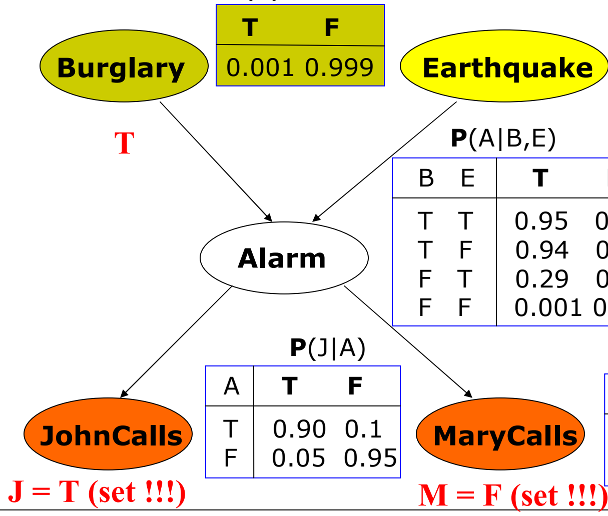
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

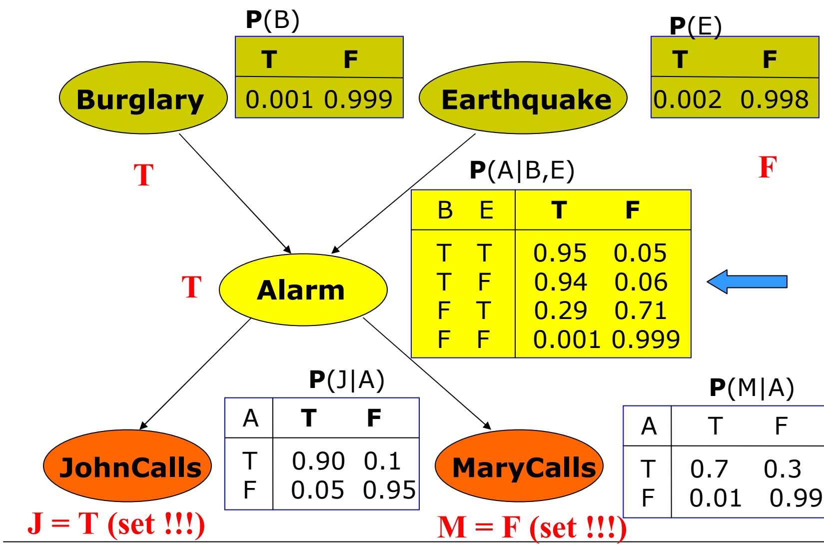
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

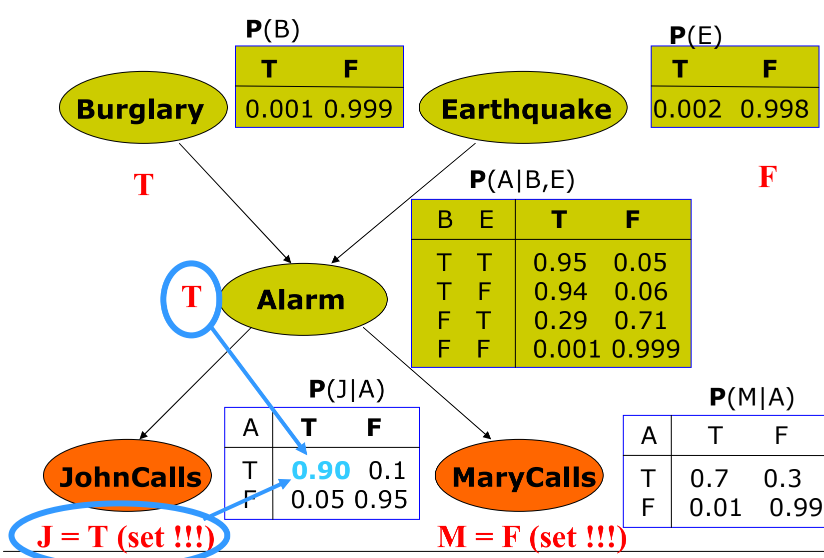
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

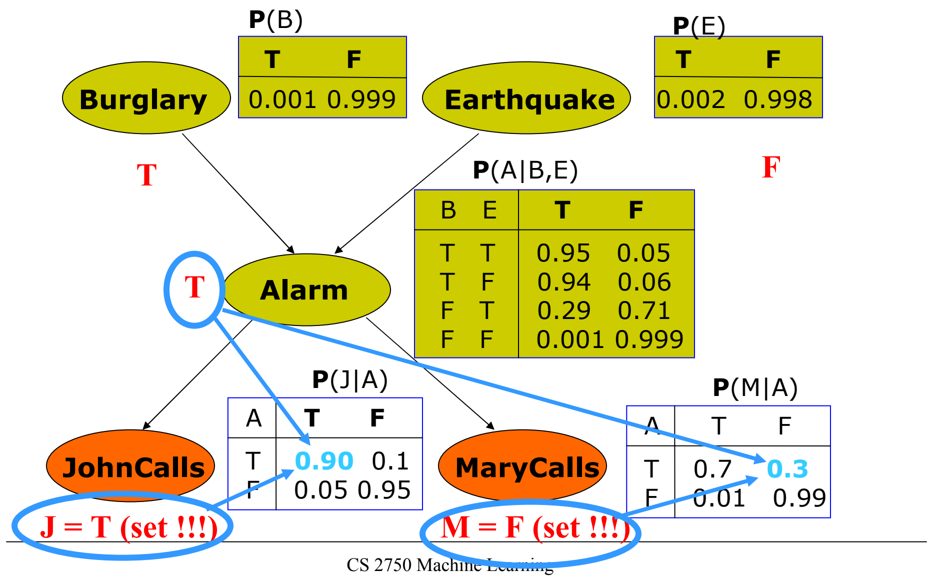
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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

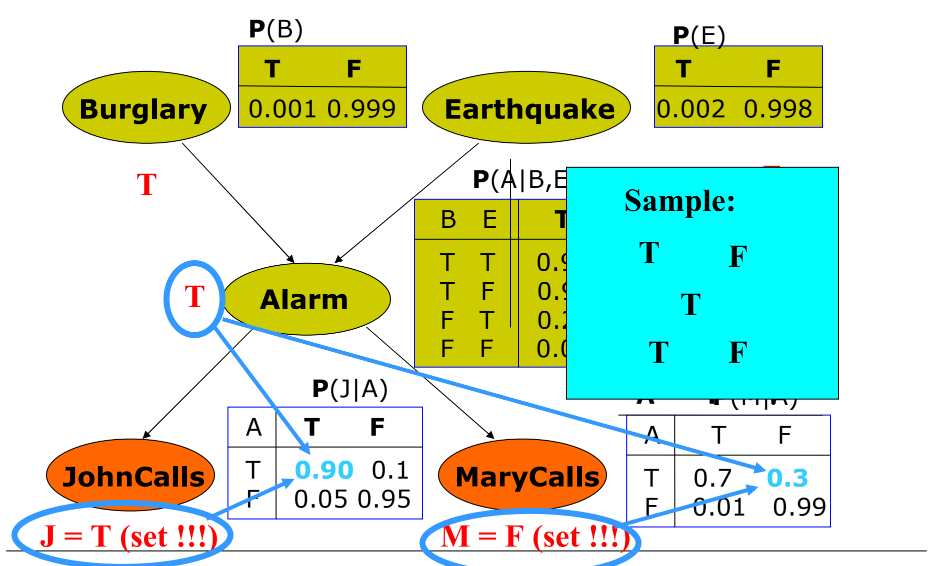
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example



P(B)

T	F
0.001	0.999

P(E)

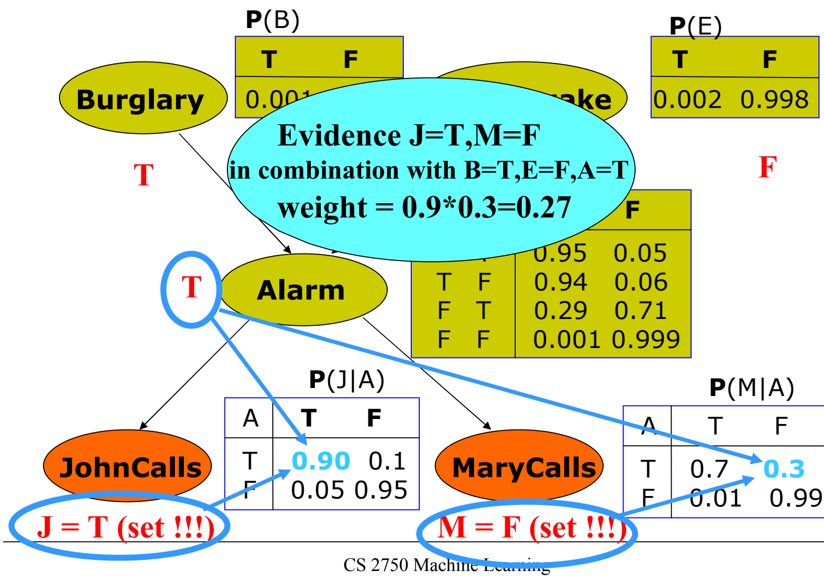
T	F
0.002	0.998

P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example



P(B)

T	**F**
0.001	

P(E)

T	**F**
0.002	0.998

B	E	T	**F**
		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)

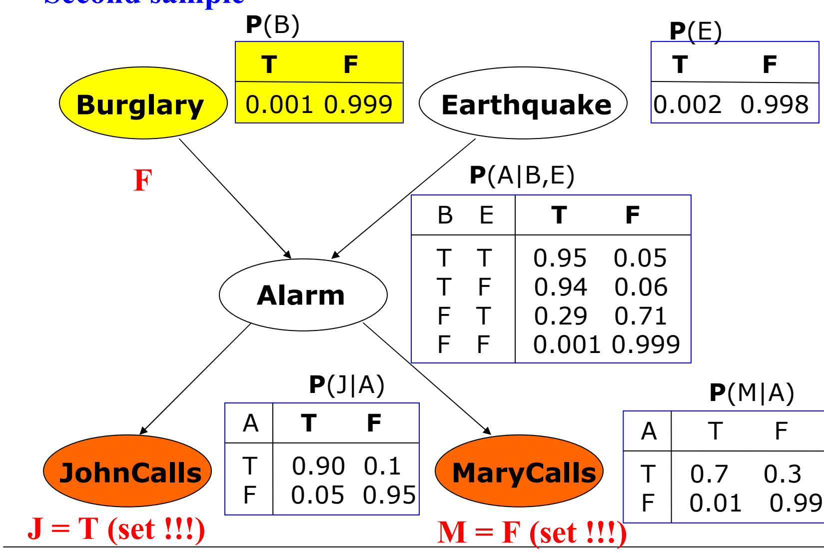
A	**T**	**F**
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example

Second sample



P(B)

T	**F**
0.001	0.999

P(E)

T	**F**
0.002	0.998

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)

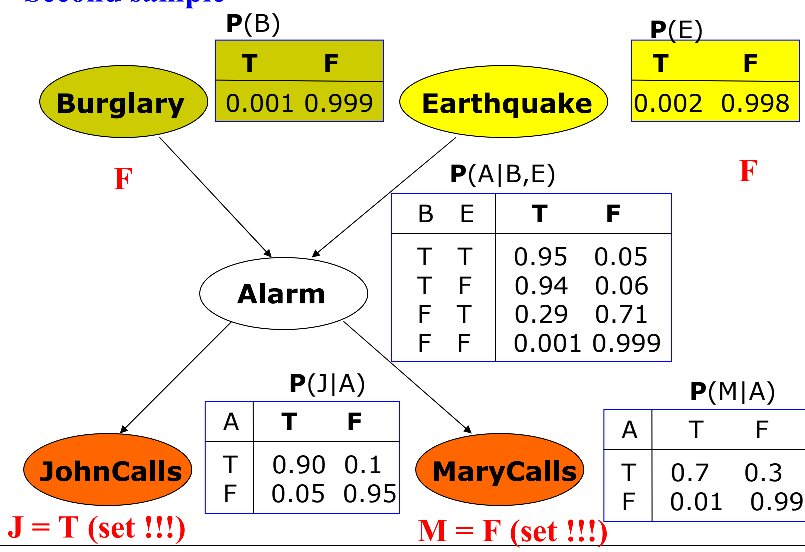
A	**T**	**F**
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example

Second sample



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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)

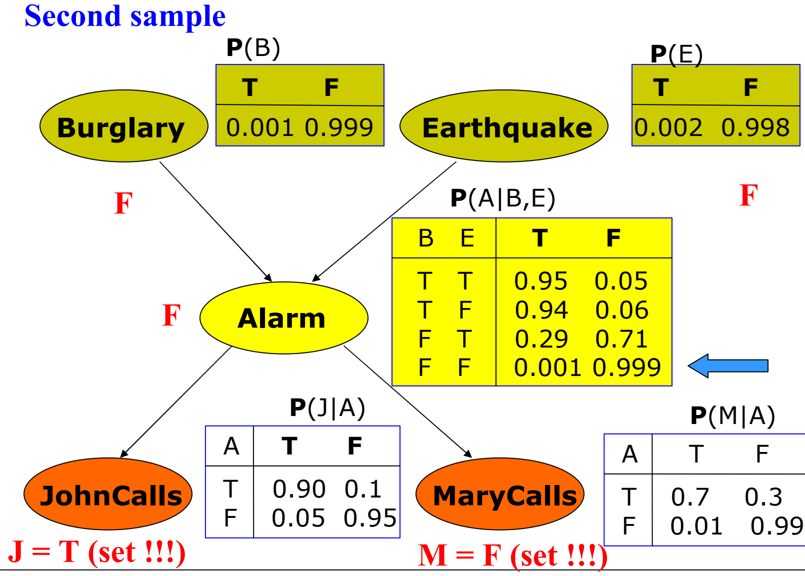
A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example

Second sample



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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)

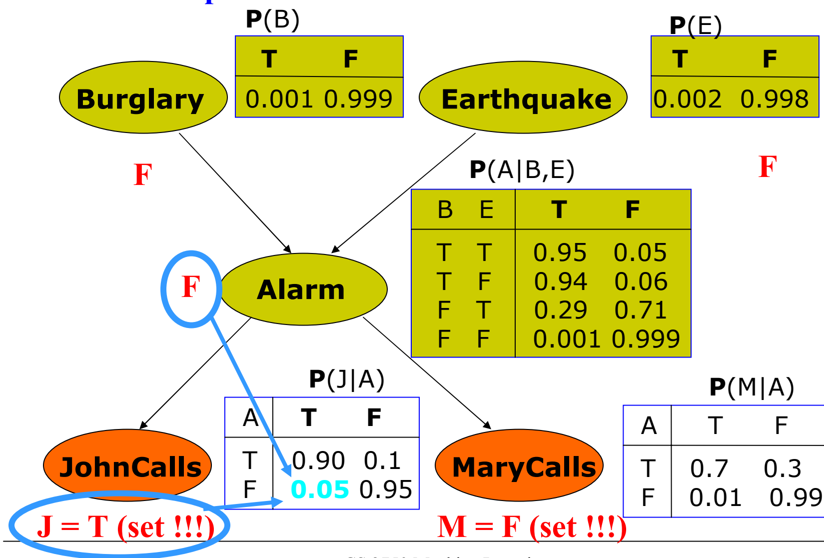
A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example

Second sample



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

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)

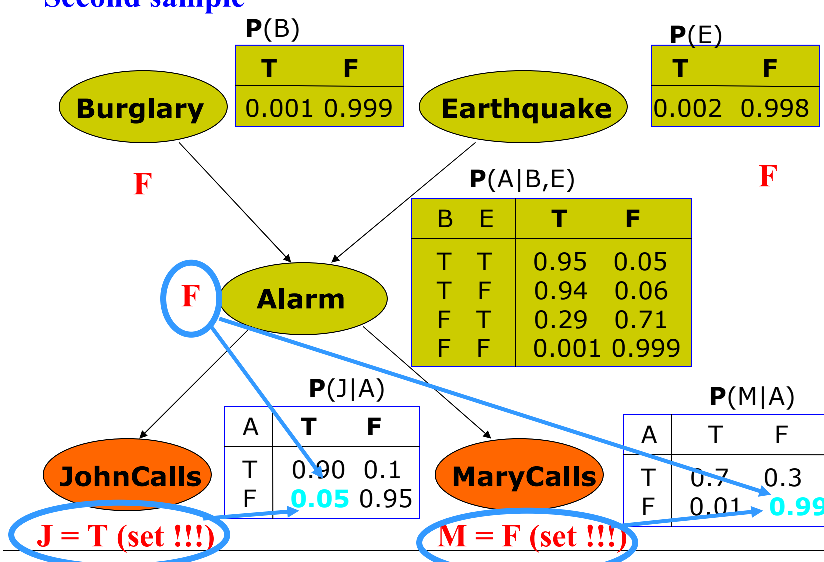
A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

BBN likelihood weighting example

Second sample



P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

P(A|B,E)

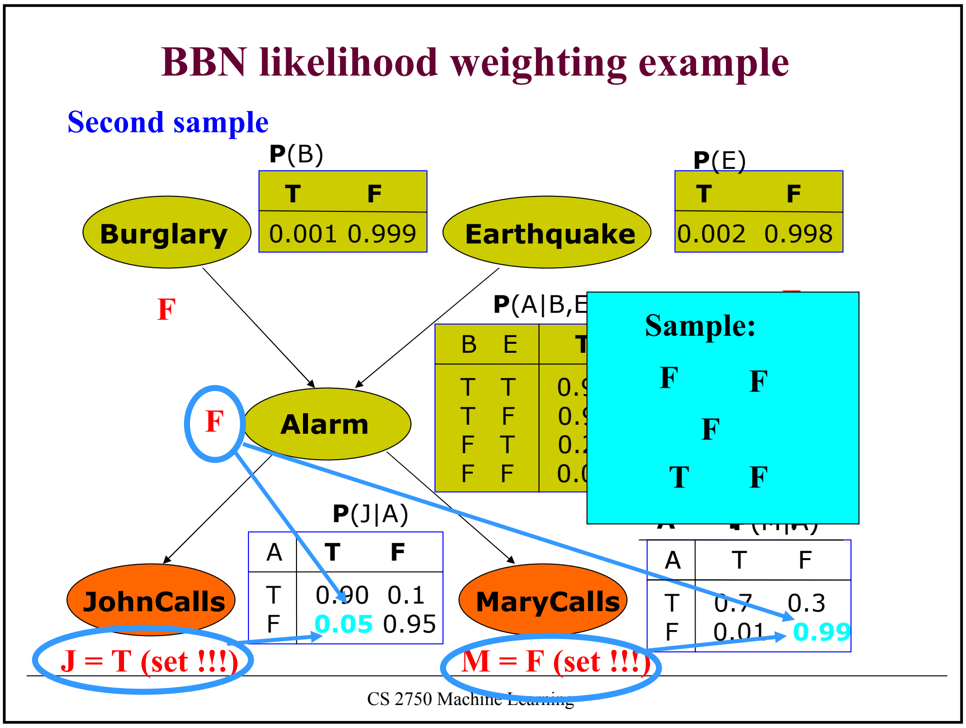
B	E	T	F
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

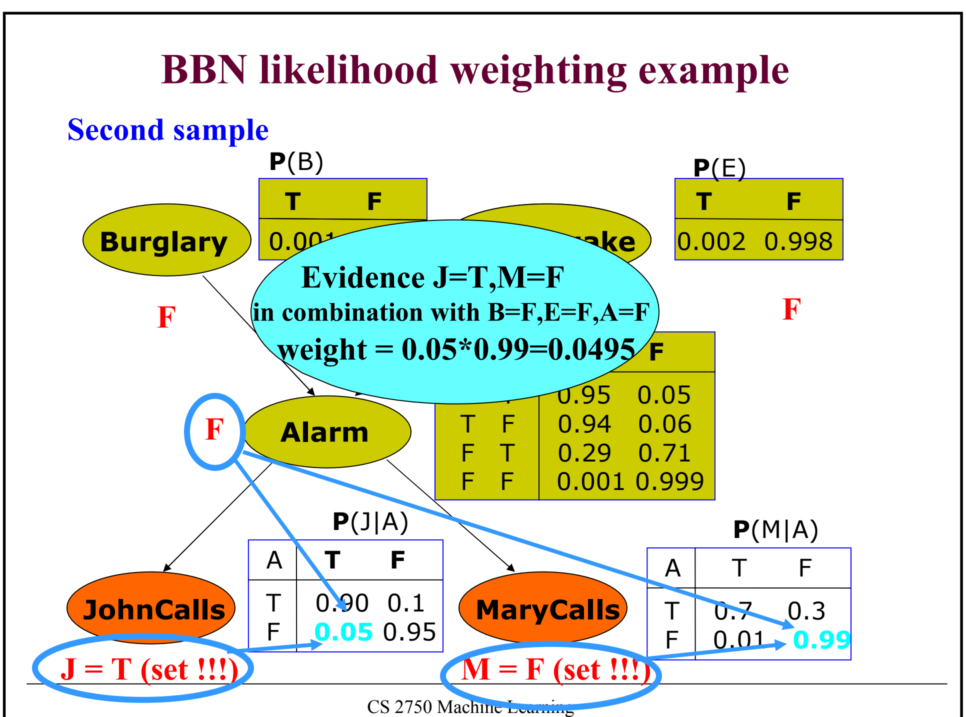
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

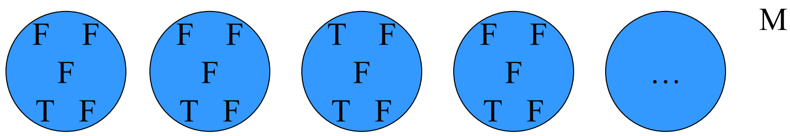
A	T	F
T	0.7	0.3
F	0.01	0.99


BBN likelihood weighting example
Second sample
P(B)
T F
0.001 0.999
Burglary
Earthquake
P(E)
T F
0.002 0.998
F
P(A|B,E
B E
T T
T F
F T
F F
Sample:
F F
F
T F
F
Alarm
P(J|A)
A T F
T 0.90 0.1
F 0.05 0.95
JohnCalls
MaryCalls
A T F
T 0.7 0.3
F 0.01 0.99
J = T (set !!!)
M = F (set !!!)
CS 2750 Machine Learning


BBN likelihood weighting example
Second sample
P(B)
T F
Burglary
P(E)
T F
0.002 0.998
Evidence J=T,M=F
in combination with B=F,E=F,A=F
weight = 0.05*0.99=0.0495
F
F
F
0.95 0.05
T F 0.94 0.06
F T 0.29 0.71
F F 0.001 0.999
F
Alarm
P(J|A)
A T F
T 0.90 0.1
F 0.05 0.95
JohnCalls
MaryCalls
P(M|A)
A T F
T 0.7 0.3
F 0.01 0.99
J = T (set !!!)
M = F (set !!!)
CS 2750 Machine Learning

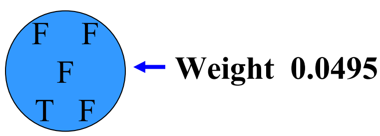
Likelihood weighting

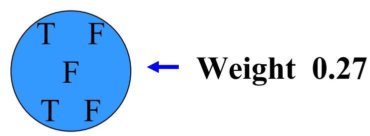
- Assume we have generated the following M samples:



How to make examples consistent with the original distribution?

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





Learning of BBN

Learning.

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

Variables:

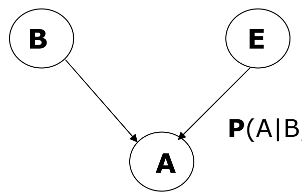
- **Observable** – values present in every data sample
- **Hidden** – they values are never observed in data
- **Missing values** – values sometimes present, sometimes not

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

P(A|B,E)

- **P**(A|B=T,E=T)
- **P**(A|B=T,E=F)
- **P**(A|B=F,E=T)
- **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**

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

  - **Parameter independence**

\# of nodes

\# of parents values

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

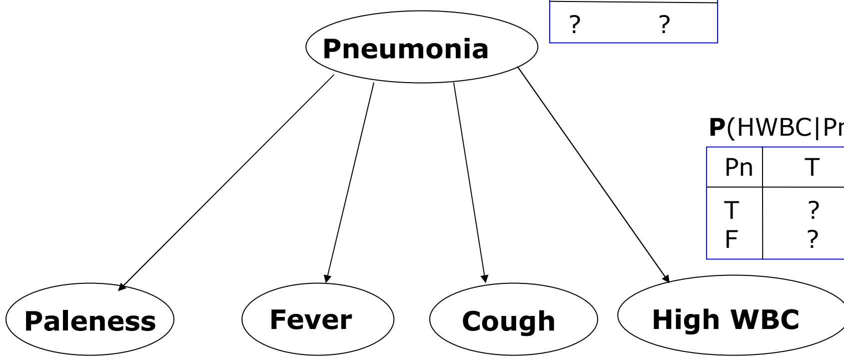
Parameters of **each conditional** (one for every assignment of values to parent variables) can be learned independently

Learning of BBN parameters. Example.

Example:

P(Pneumonia)

T	**F**
?	?



P(HWBC|Pneum)

Pn	T	F
T	?	?
F	?	?

P(Palen|Pneum) ?

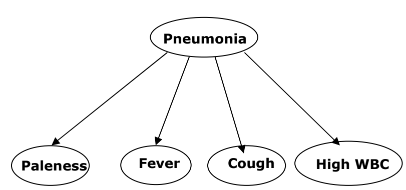
P(Fever|Pneum) ?

P(Cough|Pneum) ?

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:

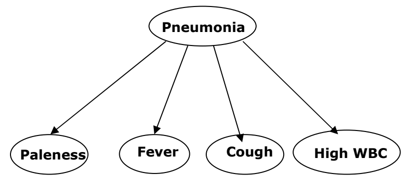
1. Select data points with Pneumonia=T (ignore the rest)
2. Focus on (select) only values of the random variable defining the distribution (Fever)
3. Learn the parameters of the conditional the same way as we learned the parameters of the biased coin or dice

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

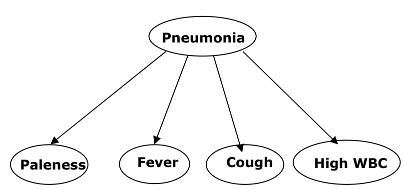


Learning of BBN parameters. Example.

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

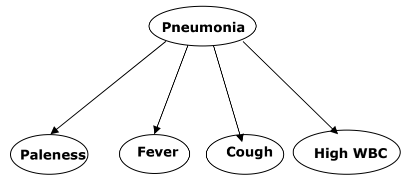


Learning of BBN parameters. Example.

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

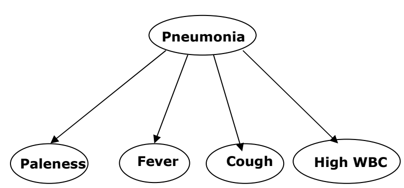


Learning of BBN parameters. Example.

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

Step 2: Ignore the rest

Fev
F
F
T
T
T

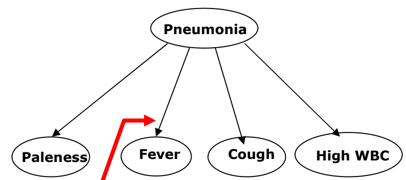


Learning of BBN parameters. Example.

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.

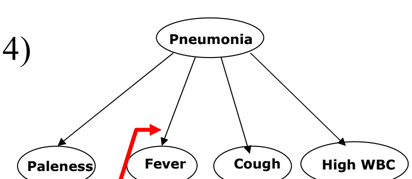
Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 3b: Learning the Bayesian estimate

Assume the prior

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

Fev
F
F
T
T
T



Posterior:

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