CS 3750 Machine Learning
Lecture 4

Monte Carlo methods

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

Monte Carlo inference

- Let us assume we have a probability distribution *P*(X) represented e.g. using BBN or MRF, and we want calculate *P*(X=x) (*P*(x) in short)
- We can use exact probabilistic inference, but it may be hard to calculate
- **Monte Carlo approximation**:
  - **Idea:** The probability *P*(x) is approximated using sample frequencies
- **Idea (first method):**
  - Generate a random sample *D* of size *M* from *P*(X)
  - Estimate P(x) as:

$$\hat{P}_D(X = x) = \frac{M_{X=x}}{M}$$

Absolute Error Bound

- **Hoeffding's bound** lets us bound the probability with which the estimate $\hat{P}_D(x)$ differs from $P(x)$ by more than ε

$$P\,(\hat{P}_D(x) \notin [P(x)-\varepsilon, P(x)+\varepsilon]) \leq 2e^{-2M\varepsilon^2} \leq \delta$$

The bound can be used to decide on how many samples are required to achieve a desired accuracy:

$$M \geq \frac{\ln(2/\delta)}{2\varepsilon^2}$$

Relative Error Bound

- **Chernoff's bound** lets us bound the probability of the estimate $\hat{P}_D(x)$ exceeding a relative error ε of the true value $P(x)$

$$P\,(\hat{P}_D(x) \notin P(x)(1+\in)) \leq 2e^{-MP(x)\varepsilon^2/3} \leq \delta$$

- This leads to the following sample complexity bound:

$$M \geq 3\frac{\ln(2/\delta)}{P(x)\varepsilon^2}$$

Monte Carlo inference challenges

Challenge 1: How to generate M (unbiased) examples from the target distribution P(X) ?

- Generating (unbiased) examples from P(X) may be hard, or very inefficient

Example:

- Assume I have a distribution over 100 binary variables
  - There are 2^{100} possible configurations of variable values
- **Trivial sampling solution:**
  - calculate and store the probability of each configuration
  - Pick randomly a configuration based on its probability
- **Problem:** terribly inefficient in time and memory

Monte Carlo inference challenges

Challenge 2: How to estimate the expected value of f(x) for P(x):

- Generally, we can estimate this expectation by generating samples x[1], …, x[M] from P, and then estimating it as:

$$E_P[f] = \sum_x P(x) f(x) \qquad E_P[f] = \int_x p(x) f(x) dx$$

$$\boxed{\hat{\Phi} = \hat{E}_P[f] = \frac{1}{M} \sum_{m=1}^{M} f(x[m])}$$

- Using the central limit theorem, the estimate $\hat{\Phi}$ follows $N\left(0, \frac{\sigma^2}{M}\right)$
  - Where is the variance for f(x) is

$$\sigma^2 = \int_x p(x)[f(x) - E_P(f(x))]^2 dx$$

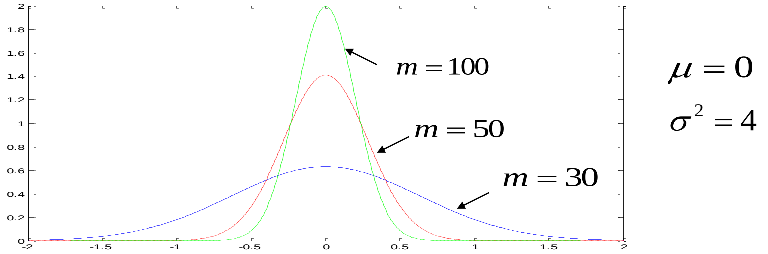
- **Problem:** we are unable to efficiently sample P(x). What to do?

Central limit theorem

- **Central limit theorem:**
  Let random variables $X_1, X_2, \cdots X_m$ form a random sample from a distribution with mean μ and variance σ^2, then if the sample n is large, the distribution

$$\sum_{i=1}^{m} X_i \approx N(m\mu, m\sigma^2) \qquad \text{or} \qquad \frac{1}{m}\sum_{i=1}^{m} X_i \approx N(\mu, \sigma^2 / m)$$

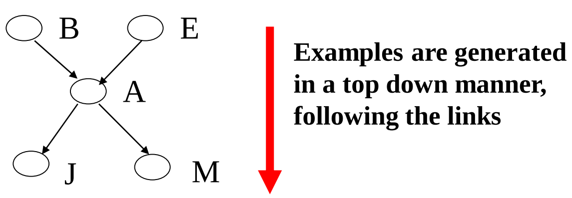
Effect of increasing the sample size *m* on the sample mean:



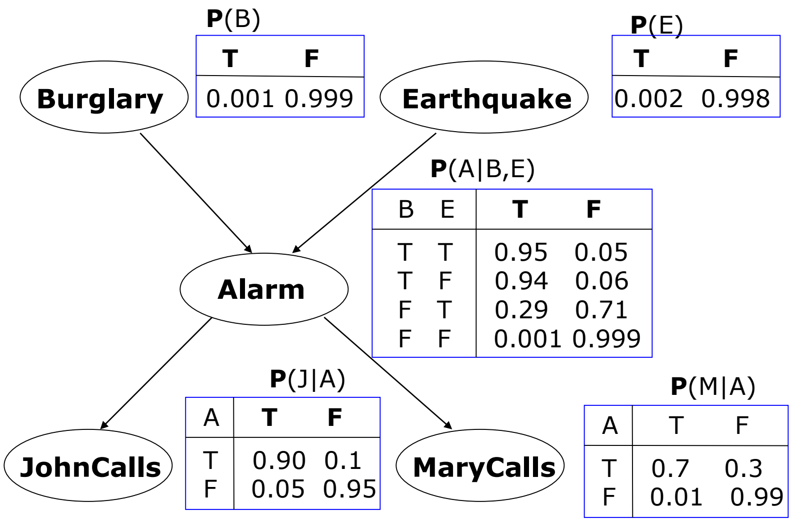
Monte Carlo inference: BBNs

Challenge 1: How to generate M (unbiased) examples from the target distribution P(X) defined by a BBN?

- **Good news: Sample generation for the full joint defined by the BBN is easy**
  - One top down sweep through the network lets us generate one example according to P(X)
  - **Example:**



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

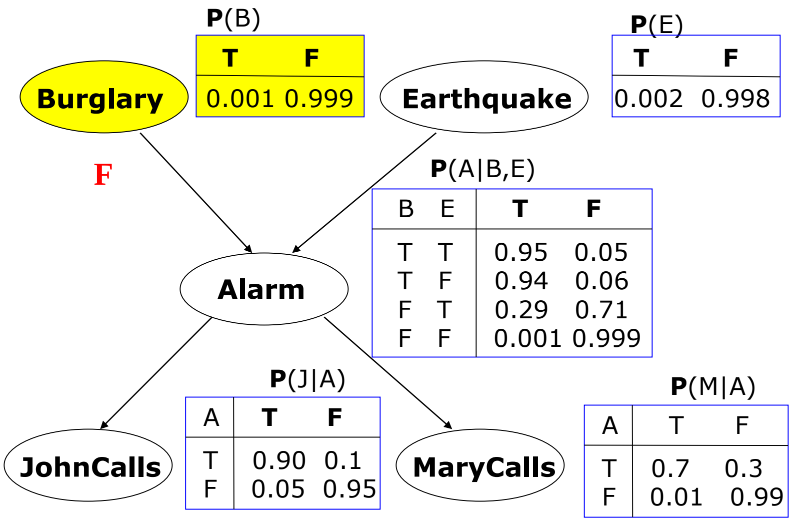
P(J|A)

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

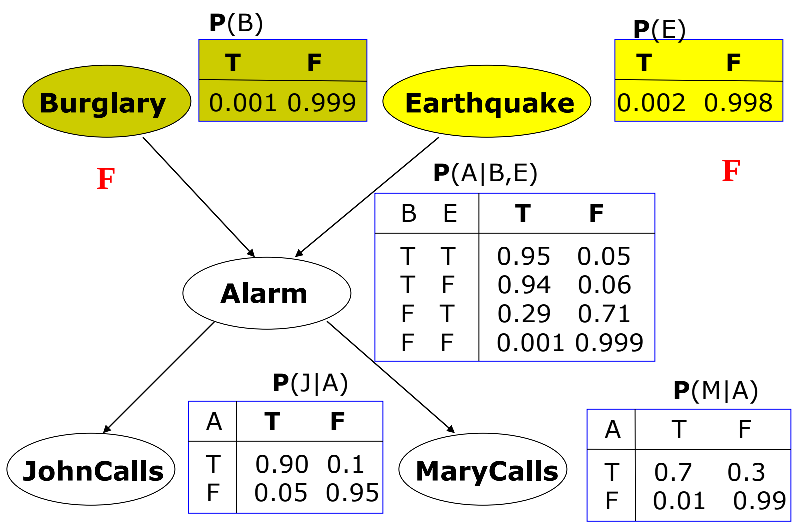
P(J|A)

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

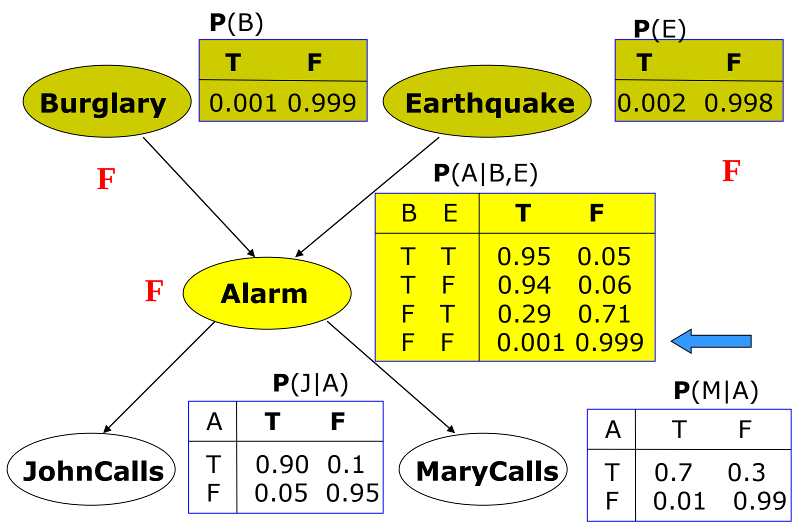
P(J|A)

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

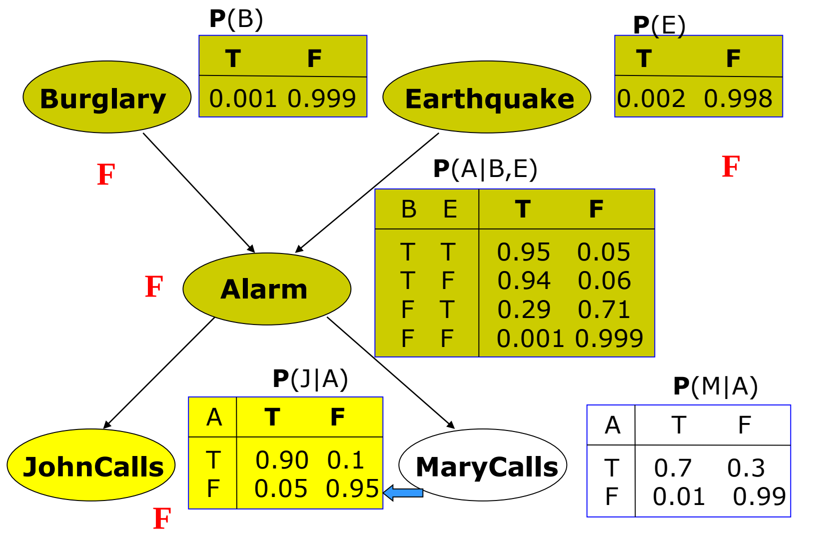
P(J|A)

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

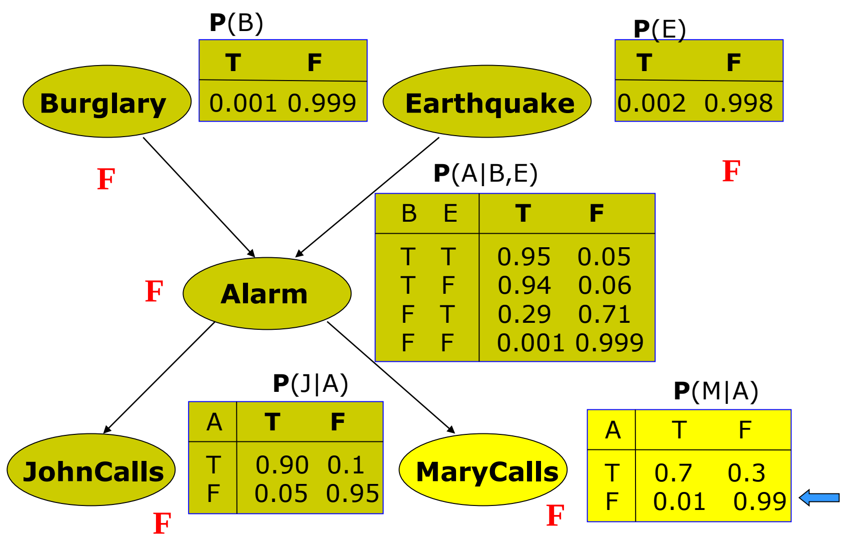
P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

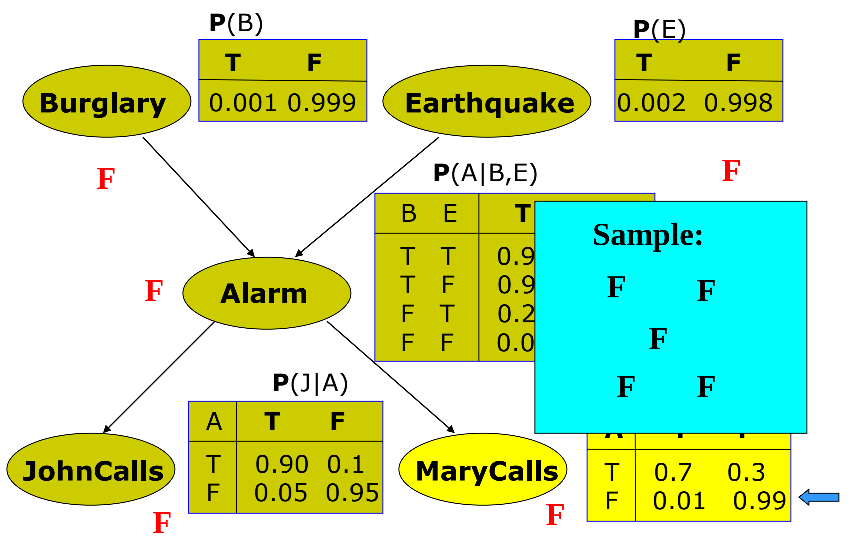
BBN sampling example



BBN sampling example



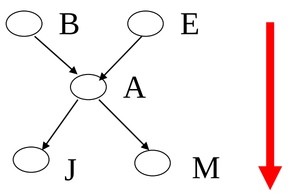
BBN sampling example



Monte Carlo inference: BBNs

Challenge 1: How to generate M (unbiased) examples from the target distribution P(X) defined by BBN?

- **Good news: Sample generation for the full joint defined by the BBN is easy**
  - One top down sweep through the network lets us generate one example according to P(X)
  - **Example:**



Examples are generated in a top down manner, following the links

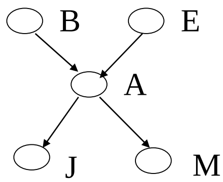
  - **Repeat many times to get enough of examples**

Monte Carlo inference: BBNs

Knowing how to generate efficiently examples from the full joint lets us efficiently estimate:

- Joint probabilities over a subset variables
- Marginals on variables

- **Example:**



The probability is approximated using sample frequency

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

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

N: $total\ \#\ samples$

Monte Carlo inference: BBNs

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

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

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

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

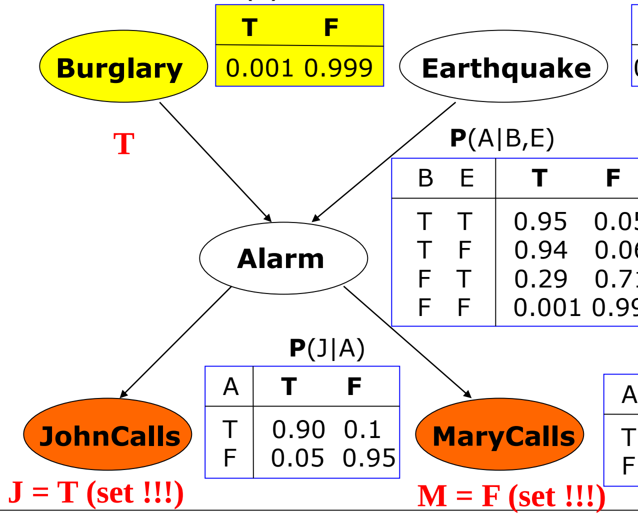
- **Solution 1 (rejection sampling):**
  - Generate examples from $P(X)$ which we know how to do efficiently
  - Use only samples that agree with the condition (J=T), the remaining samples are rejected
- **Problem:** many examples are rejected. What if $P(J=T)$ is very small?

Monte Carlo inference: BBNs

- **MC approximation of conditional probabilities**
- **Solution 2 (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



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

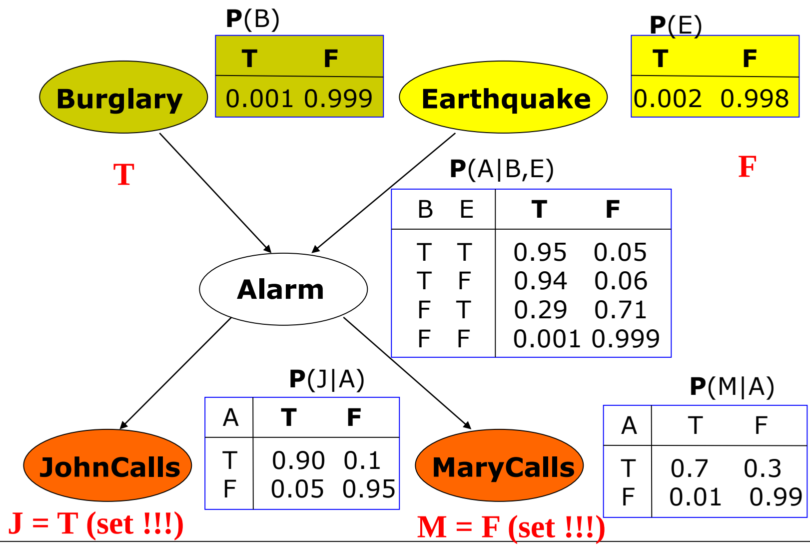
P(J|A)

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

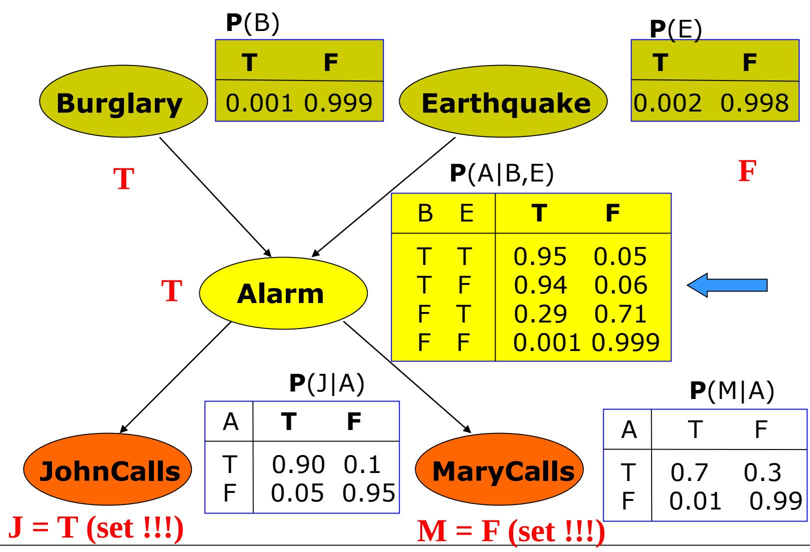
P(J|A)

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

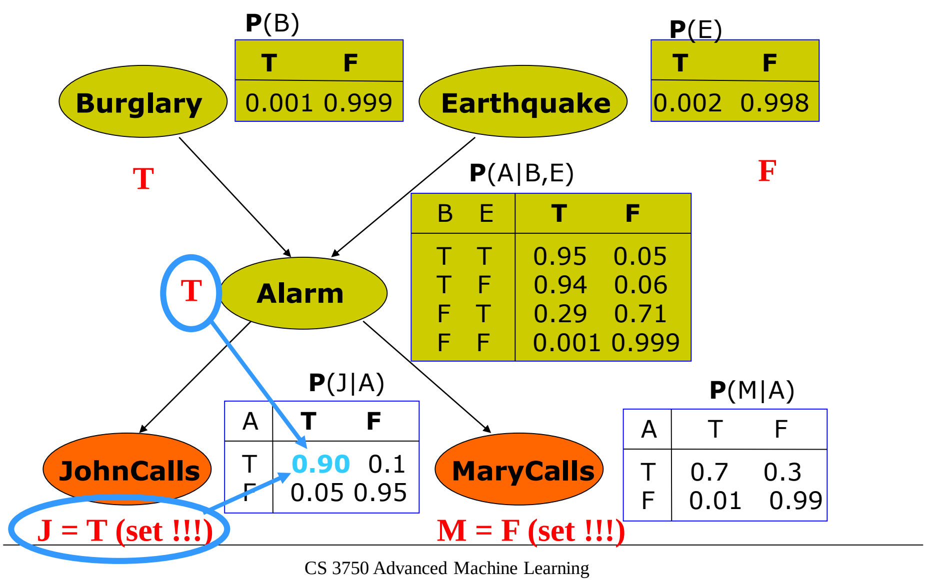
P(J|A)

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

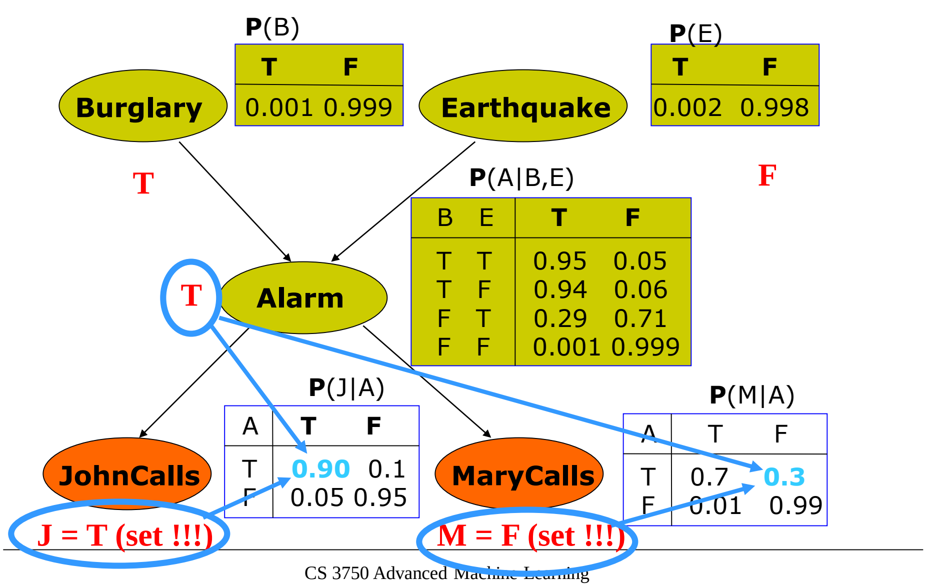
P(J|A)

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)

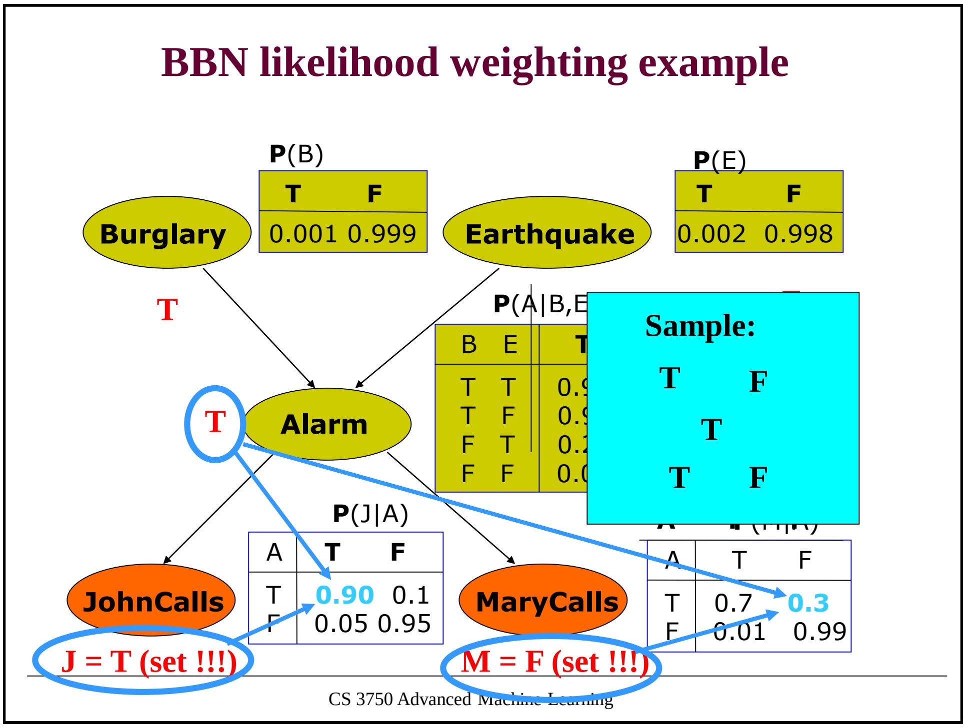
B	E	T	F
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

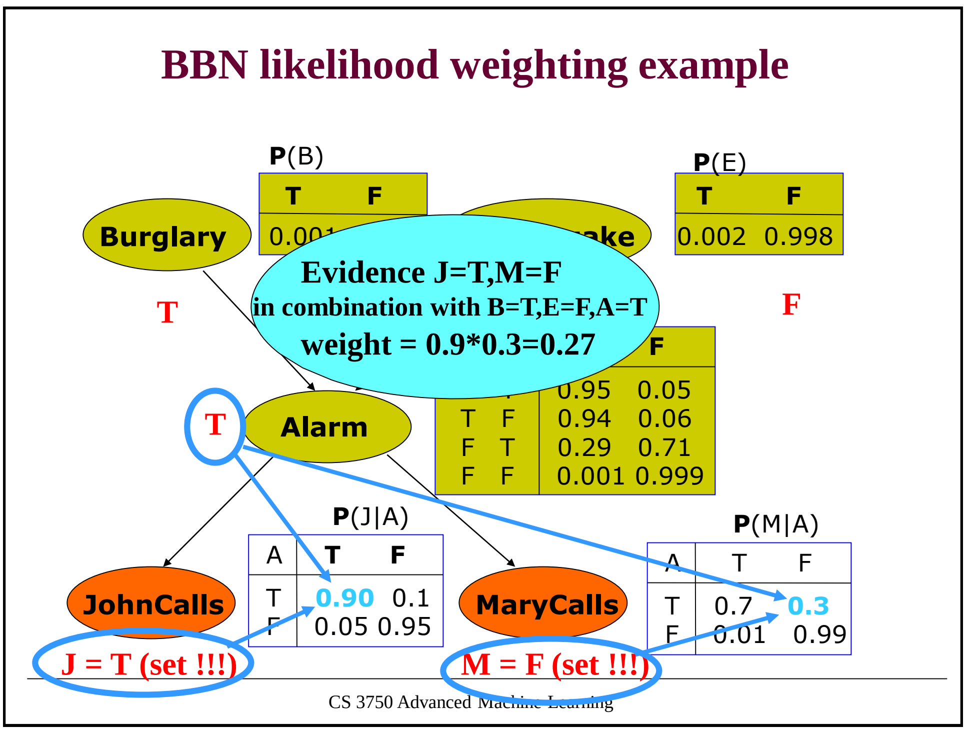
P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

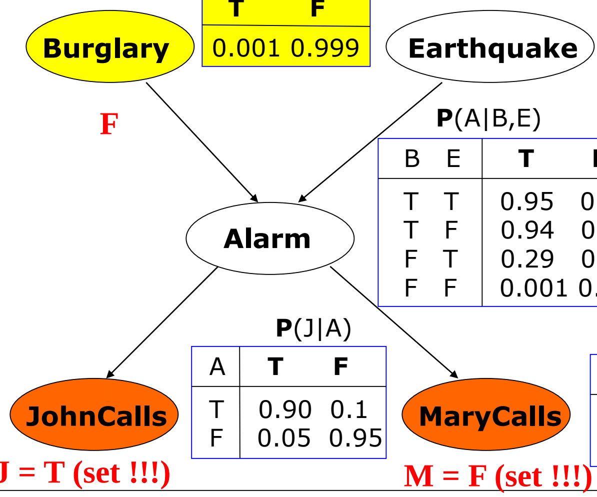
A	T	F
T	0.7	0.3
F	0.01	0.99


BBN likelihood weighting example
P(B)
T F
0.001 0.999
P(E)
T F
0.002 0.998
Burglary
Earthquake
T
P(A|B,E
B E T
T T 0.9
T F 0.9
F T 0.2
F F 0.0
Sample:
T F
T
T F
T
Alarm
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
JohnCalls
MaryCalls
J = T (set !!!)
M = F (set !!!)
CS 3750 Advanced Machine Learning


BBN likelihood weighting example
P(B)
T F
0.001
P(E)
T F
0.002 0.998
Burglary
Evidence J=T,M=F
in combination with B=T,E=F,A=T
weight = 0.9*0.3=0.27
T
F
F
0.95 0.05
T F 0.94 0.06
F T 0.29 0.71
F F 0.001 0.999
T
Alarm
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
JohnCalls
MaryCalls
J = T (set !!!)
M = F (set !!!)
CS 3750 Advanced Machine Learning

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)

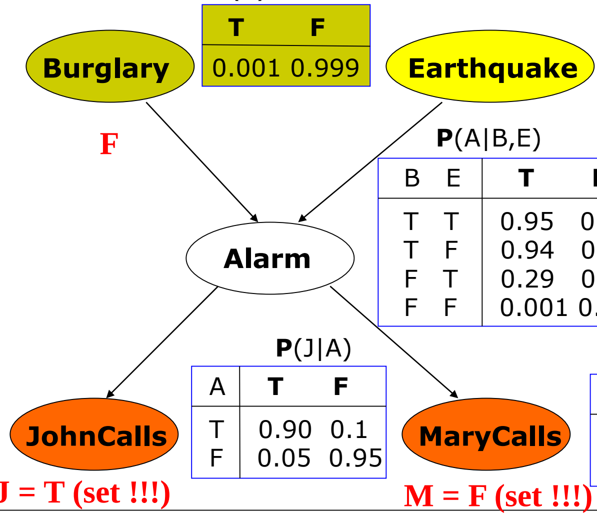
A	**T**	**F**
T	0.90	0.1
F	0.05	0.95

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

F

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)

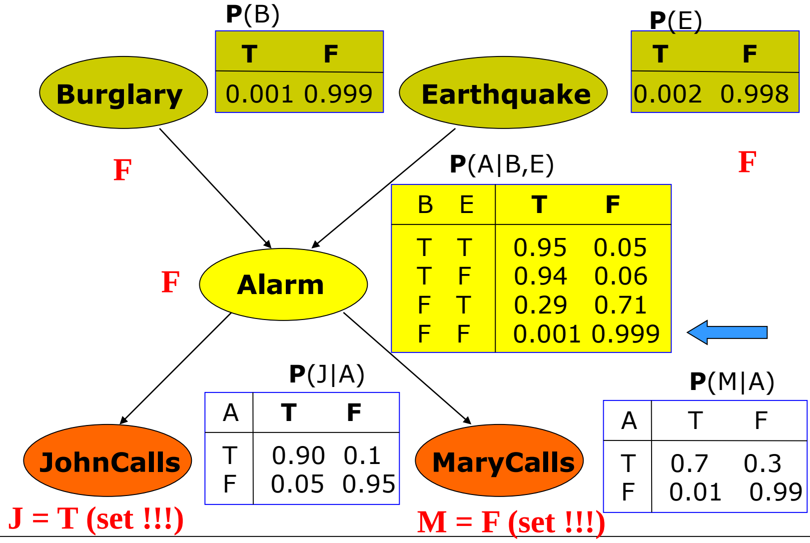
A	**T**	**F**
T	0.90	0.1
F	0.05	0.95

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)

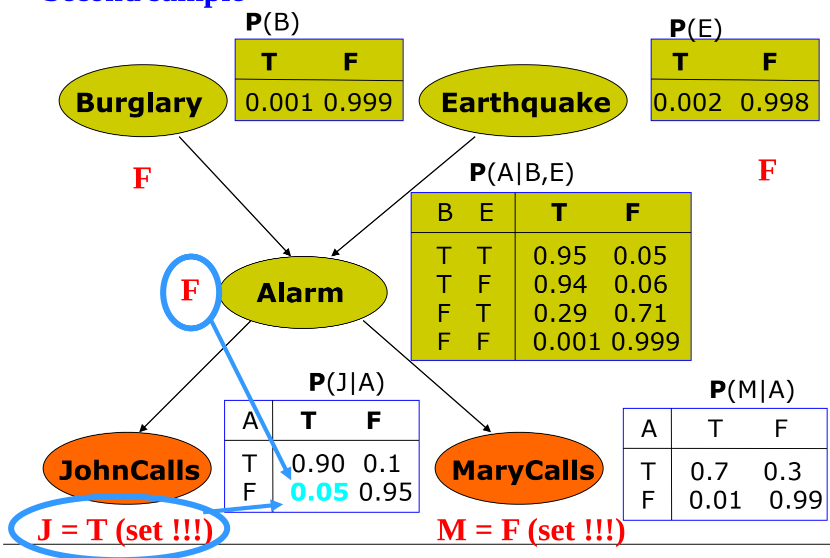
A	T	F
T	0.90	0.1
F	0.05	0.95

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

Burglary F

Earthquake F

Alarm F

JohnCalls

J = T (set !!!)

MaryCalls

M = F (set !!!)

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
T	T	0.9
T	F	0.9
F	T	0.2
F	F	0.0

Sample:

F F

F

T F

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

Burglary F

Earthquake

Alarm F

JohnCalls

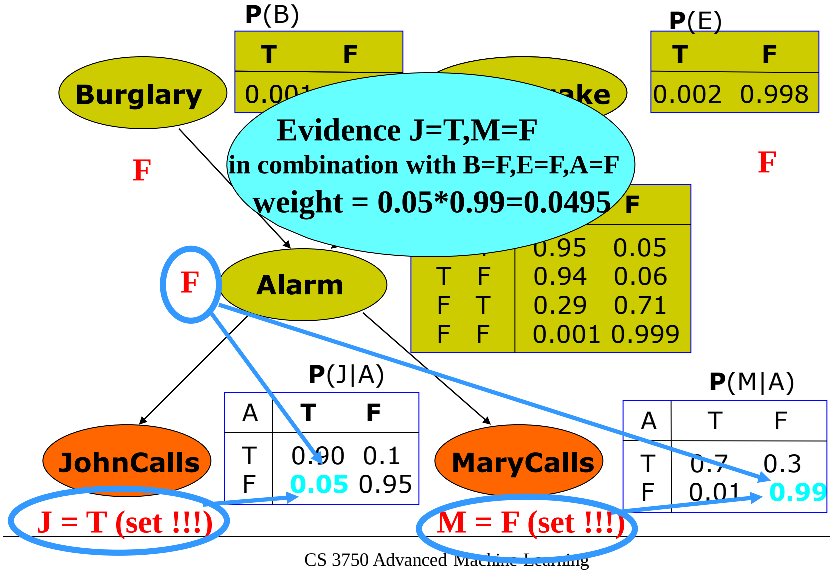
J = T (set !!!)

MaryCalls

M = F (set !!!)

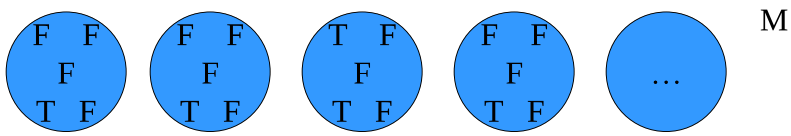
BBN likelihood weighting example

Second sample



Likelihood weighting

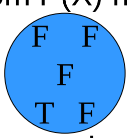
- Assume we have generated the following M samples:



- If we calculate the estimate:

$$P(B=T \mid J=T, M=F) = \frac{\#sample\_with(B=T)}{\#total\_sample}$$

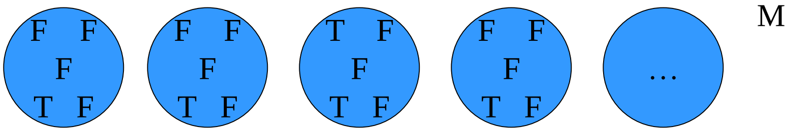
a less likely sample from P(X) may be generated more often.

- For example, sample  is generated more often than in P(X)

- So the samples are not consistent with P(X).

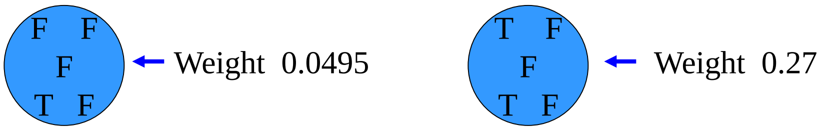
Likelihood weighting

- 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

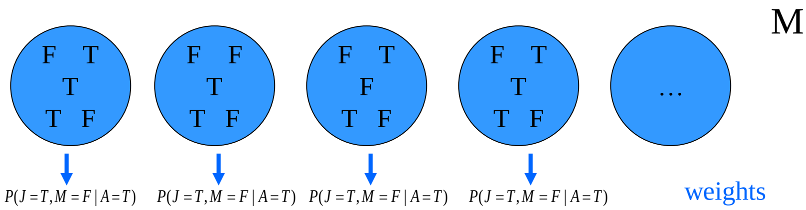
- How to compute weights for the sample?
- Assume the query $P(B=T \mid J=T, M=F)$
- Likelihood weighting:
  - **With every sample keep a weight with which it should count towards the estimate**

$$\tilde{P}(B=T \mid J=T, M=F) = \frac{\sum_{i=1}^{M} 1\{B^{(i)}=T\} w^{(i)}}{\sum_{i=1}^{M} w^{(i)}}$$

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

Likelihood weighting

- Assume M samples where evidence is enforced:



- We can use P(e) to weight each sample and correct the bias.
- The correct estimate is then:

$$\tilde{P}(A=T \mid J=T, M=F) = \frac{\sum_{i=1}^{M} 1\{A^{(i)}=T\} w^{(i)}}{\sum_{i=1}^{M} w^{(i)}}$$

Monte Carlo inference: MRFs

Challenge: How to generate M (unbiased) examples from the target distribution P(X) defined by an MRF?

- **Trivial solution:**
  - calculate and store the probability of each configuration
  - Pick randomly a configuration based on its probability
- **Problem:** terribly inefficient for a large number of variables

- Can we do better, similarly to BBN?
- In general, sampling P(X) or P(X'|Evidence) can be hard?

Next: avoid sampling P(X) by sampling Q(X)

Importance Sampling

- An approach for estimating the expectation of a function f(x) relative to some distribution P(X) (**target distribution**)
- generally, we can estimate this expectation by generating samples x[1], …, x[M] from P, and then estimating

$$E_P[f] = \frac{1}{M}\sum_{m=1}^{M} f(x[m])$$

- However, we might prefer to generate samples from a different distribution Q (**proposal or sampling distribution**) instead, since it might be impossible or computationally very expensive to generate samples directly from P(X).
- Q can be arbitrary, but it should dominate P, i.e.
  Q(x)>0 whenever P(x)>0

Unnormalized Importance Sampling

- Since we generate samples from Q instead of P,
- we need to adjust our estimator to compensate for the incorrect sampling distribution.

$$E_{p(X)}[f(X)] = E_{Q(x)}[f(x)\frac{P(x)}{Q(x)}]$$

- So we can use standard estimator for expectations relative to Q.
- **Method:** We generate a set of M samples D={x[1],…,x[M]} from Q, and estimate:

$$\hat{E}_D(f) = \frac{1}{M}\sum_{m=1}^{M} f(x[m])\frac{P(x[m])}{Q(x[m])}$$

Importance sampling

- This is an unbiased estimator: its mean for any data set is precisely the desired value

$w(x) = P(x)/Q(x)$ - a weighting function, or a correction weight

- We can estimate the distribution of the estimator around its mean: as M $\rightarrow \infty$

$$E_{Q(X)}[f(X)w(X)] - E_{P(X)}[f(X)] \propto N(0; \sigma_Q^{\ 2}/M)$$

where $\sigma_Q^{\ 2} = [E_{Q(X)}[(f(X)w(X))^2]] - (E_{Q(X)}[f(X)w(X)])^2$

$$\sigma_Q^{\ 2} = [E_{Q(X)}[(f(X)w(X))^2]] - (E_{P(X)}[f(X)])^2$$

Importance sampling

- When f(X)=1, the variance is simply the variance of the weighting function P(X)/Q(X). Thus, the more different Q is from P, the higher is the variance of the estimator.
- In general, the lowest variance is achieved when

$$Q(X) \propto |f(X)| P(X)$$

- We should avoid cases where our sampling probability Q(X)<<P(X)f(X) in any part of the space, as these cases can lead to very large or even infinite variance.

- Problem with un-normalized IS: P is assumed to be known

Normalized Importance Sampling

- When P is only known up to a normalizing constant α
- We have access to a function $P'(X)$, such that P' is not a normalized distribution, but $P'(X) = \alpha P(X)$

- In this context, we cannot define the weights relative to *P*, so we define:

$$w(X) = \frac{P'(X)}{Q(X)}$$

$$E_{P(X)}[f(X)] = \sum_x P(x) f(x) = \sum_x Q(x) f(x) \frac{P(X)}{Q(x)} = \frac{1}{\alpha} \sum_x Q(x) f(x) \frac{P'(x)}{Q(x)}$$

$$= \frac{1}{\alpha} E_{Q(x)}[f(X) w(X)] = \frac{E_{Q(X)}[f(X) w(X)]}{E_{Q(X)}[w(X)]}$$

Why? $$E_{Q(X)}[w(X)] = \sum_x Q(x) \frac{P'(x)}{Q(x)} = \sum_x P'(x) = \alpha$$

Importance sampling

- Using an empirical estimator for both the numerator and denominator, we can estimate:

$$\hat{E}_D(f) = \frac{\sum_{m=1}^{M} f(x[m]) w(x[m])}{\sum_{m=1}^{M} w(x[m])}$$

- Although the normalized estimator is biased, its variance is typically lower than that of the unnormalized estimator. This reduction in variance often outweighs the bias term.

- So normalized estimator is often used in place of the unnormalized estimator, even in cases where P is known and we can sample from it effectively.

Importance sampling for estimating conditional probabilities in BBNs

Assume a Bayesian Network

- We want to calculate P(x'|evidence)
- This is hard if we need to go opposite the links and account for the effect of evidence on non-descendants

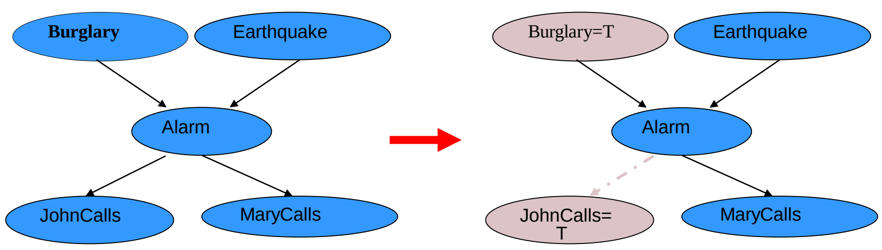
Objective: generate samples efficiently using a simpler proposal distribution Q(x)

Solution: a mutilated belief network (Koller, Friedman 2009)

- **Idea:**
  - Avoid propagation of evidence effects to nondescendants;
  - Disconnect all variables in the evidence from their parents

Mutilated Belief network

- Assume we want to calculate P(x | B=T, J=T) in the Alarm network
- Use B=T and J=T to build a mutilated network



Original network

Mutilated network

Mutilated Belief network

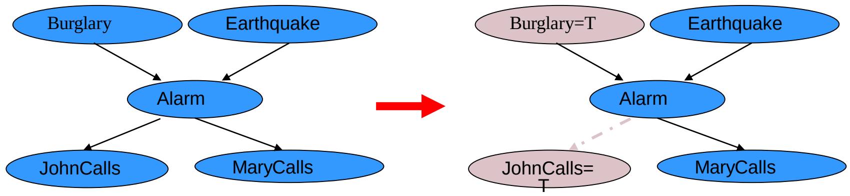
- Assume the evidence is J=j* and B=b*
- Original network:

$$P(E=e, A=a, M=m, J=j^*, B=b^*) = P(b^*)P(e)P(a\,|\,b^*,e)P(j^*\,|\,a)P(m\,|\,a)$$

- Mutilated network:

$$Q(E=e, A=a, M=m, J=j^*, B=b^*) = P(e)P(a\,|\,b^*,e)P(m\,|\,a)$$

- Note that $w(x) = \dfrac{P(x)}{Q(x)} = P(b^*)P(j^*\,|\,a)$



Original network

Mutilated network

Mutilated Belief network

- Assume the evidence is J=j* and B=b*
- Original network:

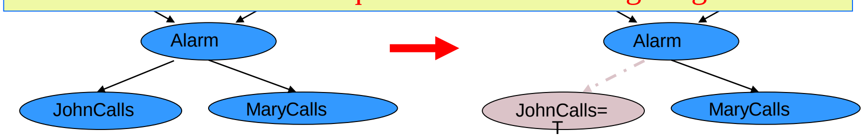
$$P(E=e, A=a, M=m, J=j^*, B=b^*) = P(b^*)P(e)P(a\,|\,b^*,e)P(j^*\,|\,a)P(m\,|\,a)$$

- Mutilated network:

$$Q(E=e, A=a, M=m, J=j^*, B=b^*) = P(e)P(a\,|\,b^*,e)P(m\,|\,a)$$

- Note that $w(x) = \dfrac{P(x)}{Q(x)} = P(b^*)P(j^*\,|\,a)$

So importance sampling with a proposal distribution based on mutilated network is equal to likelihood weighting



Original network

Mutilated network

Likelihood Weighting

- **Question:** When to stop? How many samples do we need to see?
- **Intuition:** not every sample contribute equally to the quality of the estimate. A sample with a high weight is more compatible with the evidence e, and may provide us with more information.
- **Solution:** We stop sampling when the total weight of the generated samples reaches a pre-defined value.

- **Benefits:** It allows early stopping in cases where we were lucky in our random choice of samples.

Markov chain Monte Carlo

- **Likelihood weighting:** samples are generated according to Q and every sample from Q is reweighted according to its likelihood, but the Q distribution may be very far from the target
- **MCMC** is a strategy for generating samples from the target distribution, including conditional distributions
- **MCMC:**
  - Markov chain defines a sampling process that
  - initially generates samples very different from the target distribution (e.g. posterior)
  - but gradually refines the samples so that they are closer and closer to the posterior.

MCMC

- The construction of a Markov chain requires two basic ingredients
  - a transition matrix P
  - an initial distribution π_0
- Assume a finite set S={1,…m} of states, then **a transition matrix** is

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{pmatrix}$$

Where $p_{ij} \geq 0 \quad \forall (i,j) \in S^2$ and $\sum_{j \in S} p_{ij} = 1 \quad \forall i \in S$

Markov Chain

- Markov chain defines a random process of selecting states

$x^{(0)}, x^{(1)}, \ldots x^{(m)}, \ldots$

Initial state selected based on π_0

Subsequent states selected based on the previous state and the transition matrix

$$\boxed{x} \longrightarrow \boxed{x'}$$

t **t+1**

- **Chain Dynamics**

$$P^{(t+1)}(X^{(t+1)} = x') = \sum_{x \in Dom(X)} P^{(t)}(X^{(t)} = x) T(x \to x')$$

Probability of a state x' being selected at time t+1

transition matrix

MCMC

- **Markov chain** satisfies

$$P(X_{n+1} = j \mid X_0 = i_0, X_1 = i_1, \ldots X_n = i_n) = P(X_{n+1} = j \mid X_n = i_n)$$

- **Irreducibility:** A MC is called irreducible (or un-decomposable) if there is a positive transition probability for all pairs of states within a limited number of steps
- In irreducible chains there may still exist a periodic structure such that for each state $i \in S$, the set of possible return times to i when starting in i is a subset of the set $p\mathrm{N} = \{p, 2p, 3p, \ldots\}$ containing all but a finite set of these elements. The smallest number p with this property is the so-called **period of the chain**

$$p = \gcd\{n \in N : p_{ii}^{(n)} > 0\}$$

MCMC

- **Aperiodicity:** An irreducible chain is called aperiodic (or acyclic) if the period p equals 1 or, equivalently, if for all pairs of states there is an integer n_{ij} such that for all $n \geq n_{ij}$, the probability $p^{(n)}_{ij}>0$.
- If a Markov chain satisfy both **irreducibility and aperiodicity**, then it **converges to an invariant distribution *q(x)***
- A Markov chain with transition matrix P will have an equilibrium distribution q **iff** $q = qP$.
- A sufficient, but not necessary, condition to ensure a particular q(x) is the invariant distribution of transition matrix P is the following **reversibility (detailed balance) condition**

$$q(x^i)P(x^{i-1} \mid x^i) = q(x^{i-1})P(x^i \mid x^{i-1})$$

Markov Chain Monte Carlo

Objective: generate samples from the target distribution (e.g. posterior)

- **Idea:**

  Markov chain defines a sampling process that:
  - initially generates samples very different from the target posterior
  - but gradually refines the samples so that they are closer and closer to the target distribution