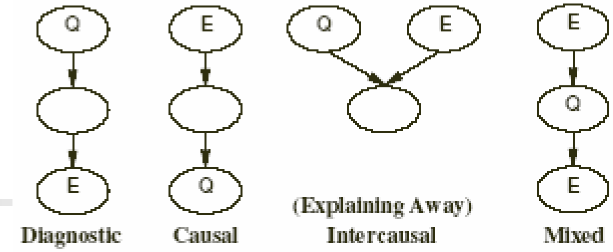
RN, Chapter 14.4

Bayesian Belief Network Inference

Decision Theoretic Agents

- Introduction to Probability [Ch13]
- Belief networks [Ch14]
  - Introduction [Ch14.1-14.2]
  - Bayesian Net Inference [Ch14.4] (Bucket Elimination)
- Dynamic Belief Networks [Ch15]
- Single Decision [Ch16]
- Sequential Decisions [Ch17]
- Game Theory [Ch17.6 – 17.7]

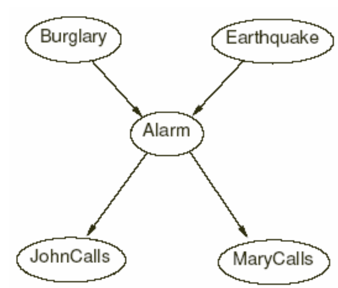
Types of Reasoning



- **Typical case:** P(QueryVar | EvidenceVars = vals)
  - Eg: P(+Burglary | +JohnCalls, ¬MaryCalls)
- **Diagnostic**: from effect to (possible) causes
  - P(+Burglary | +JohnCalls) = 0.016
- **Causal**: from cause to effects
  - P(+JohnCalls | +Burglary) = 0.86
- **InterCausal**: between causes of common effect
  - P(+Burglary | + Alarm) = 0.376
  - P(+Burglary | +Alarm, +Earthquake) = 0.003

Earthquake EXPLAINS alarms, and so

Earthquake **EXPLAINS AWAY** burglary

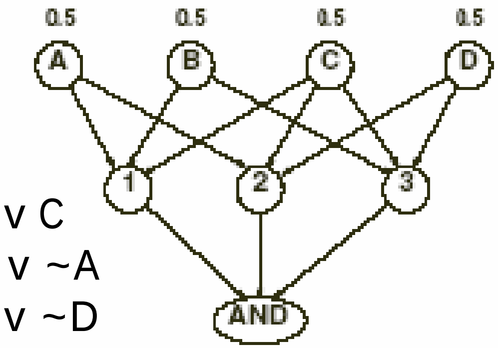


- **Mixed**: combinations of . . .
  - P(Alarm | JohnCall, ¬Earthquake) = 0.03

Approaches to Belief Assessment

- Exact, Guaranteed
  - PolyTree Algorithm
  - Inherent complexity. . .
  - Clustering Approach
  - Bucket Elimination
  - CutSet Approach
- Approximate, Guaranteed
  - Algorithm Modification
  - Value Merging
  - Node Merging
  - Arc Removal
- Approximate, Probabilistic
  - Logic Sampling
  - Likelihood Sampling

Inherent Complexity

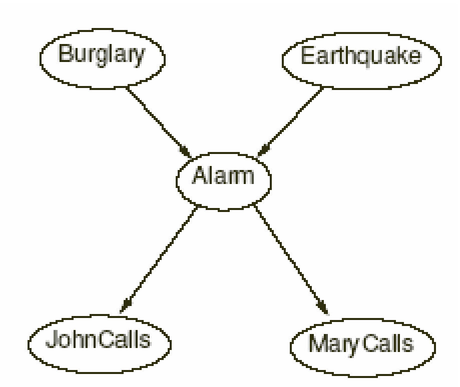


1. A v B v C
2. C v D v ~A
3. B v C v ~D

- Worst case:
  - NP-hard to get exact answer (#P-complete)
  - NP-hard to get answer within 0.5
  - Cannot get relative error within $2^{n^{1-\varepsilon}}$ unless P = NP
  - Cannot stochastically approximate 1-bit, unless P=RP
- Efficient algorithm . . .
  - for "PolyTree": Poly time
    - $\leq$ 1 path between any two nodes
  - if CPtable "bounded" (sub-exp time) wrt λ = M/m
    M = largest CPtable entry; m = smallest

Exact Inference: Re-arrange Sums

$P(A = a) = \sum_b P(A = a, B = b)$



$P(+b, +j, +m)$

$= \sum_e \sum_a P(+b, E=e, A=a, +j, +m)$

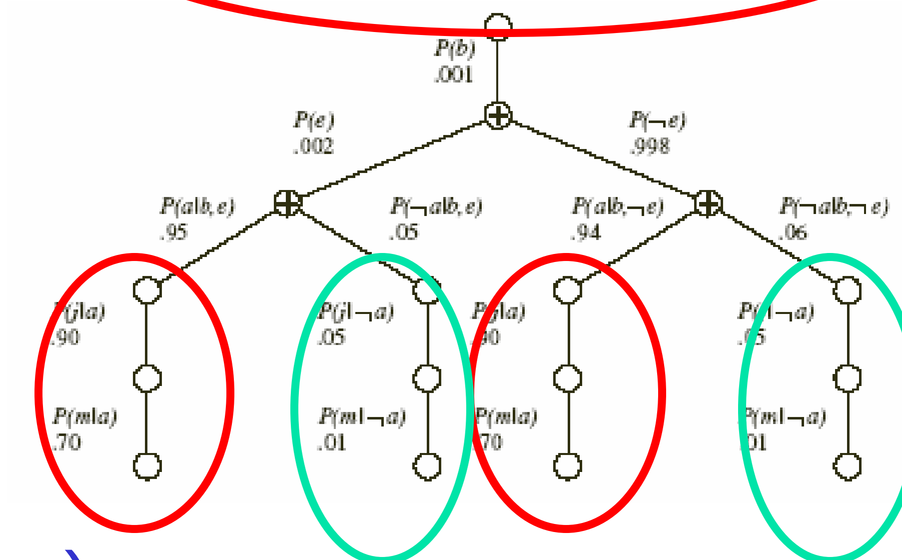
$= \sum_e \sum_a P(+b) P(e) P(a|+b,e) P(+j|a) P(+m|a)$

$= P(+b) \sum_e P(e) \sum_a P(a|+b,e) P(+j|a) P(+m|a)$

Still Duplicated Computation!

$P(+b, +j, +m) =$

$P(+b) \sum_e P(e) \sum_a P(a \mid +b, e)\ P(+j \mid a)\ P(+m \mid a)$



- Enumeration is inefficient:

  … as repeated computation

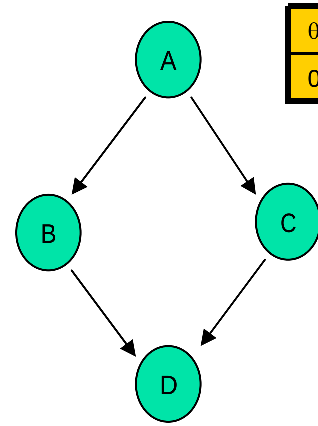
  Computes $P(+j \mid a) P(+m \mid a)$

  for each value of E: $\{ +e, -e \}$

- Better to have DAG…

  re-use COMMON SUBEXPRESSION !

Bucket-Elimination: Set-up



$\theta_{A=1}$	$\theta_{A=0}$
0.4	0.6

a	$\theta_{B=1\|A=a}$	$\theta_{B=0\|A=a}$
1	0.325	0.675
0	0.440	0.550

a	$\theta_{C=1\|A=a}$	$\theta_{C=0\|A=a}$
1	0.200	0.800
0	0.367	0.633

b	c	$\theta_{D=1\|B=b,C=c}$	$\theta_{D=0\|B=b,C=c}$
1	1	0.300	0.700
1	0	0.333	0.667
0	1	0.250	0.750
0	0	0.450	0.550

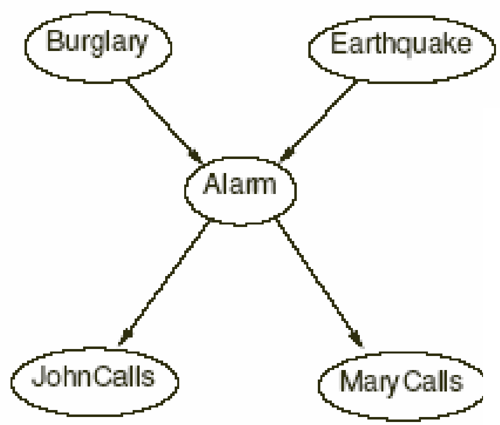
- Given
  - specific structure
  - specific CPtable entries
  - Fixed ordering over variables: $\pi_0 = \langle A,B,C,D \rangle$
- Create |Vars|+1 buckets
  - $b_{\{\}}$, b_A, b_B, b_C, b_D

$f_B(b) = \lambda \langle b \rangle.$

b	f(b)
0	0.999
1	0.001

$f_E(e) = \lambda \langle e \rangle.$

e	f(e)
0	0.998
1	0.002



$f_A(a,e,b) = \lambda \langle A,E,B \rangle.$

a	e	b	f(a, e, b)
1	1	1	0.95
1	1	0	0.29
:	:	:	:
0	0	1	0.06
0	0	0	0.999

$f_J(j,a) = \lambda \langle \mathbf{J,}A \rangle.$

j	a	f(j,a)
1	1	0.90
1	0	0.05
0	1	0.10
0	0	0.95

$f_M(m,a) = \lambda \langle \mathbf{M,}A \rangle.$

m	a	f(m,a)
1	1	0.70
1	0	0.01
0	1	0.30
0	0	0.99

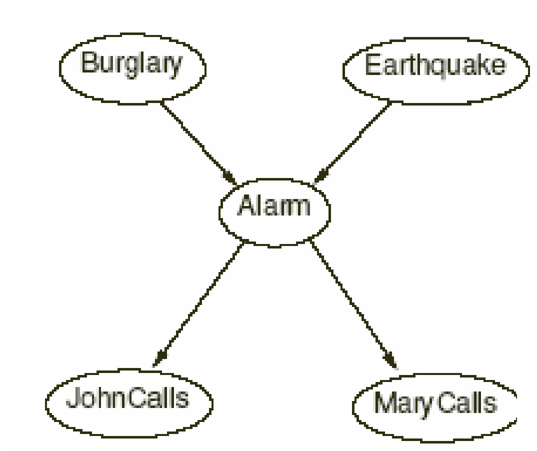
–b, +j, +m

$f_{-b}() = \lambda\, \langle\rangle.$

b	f(b)
0	0.999
~~1~~	~~0.001~~

$f_E(e) = \lambda\, \langle e \rangle.$

e	f(e)
0	0.998
1	0.002



$f_{A,-b}(a,e) = \lambda\, \langle A,E \rangle.$

a	e	b	f(a, e, b)
~~1~~	~~1~~	~~1~~	~~0.95~~
1	1	0	0.29
:	:	:	:
~~0~~	~~0~~	~~1~~	~~0.06~~
0	0	0	0.999

$f_{+j}(a) = \lambda\, \langle A \rangle.$

j	a	f(j,a)
1	1	0.90
1	0	0. 05
~~0~~	~~1~~	~~0.10~~
~~0~~	~~0~~	~~0.95~~

$f_{+m}(a) = \lambda\, \langle A \rangle.$

m	a	f(m,a)
1	1	0.70
1	0	0. 01
~~0~~	~~1~~	~~0.30~~
~~0~~	~~0~~	~~0.99~~

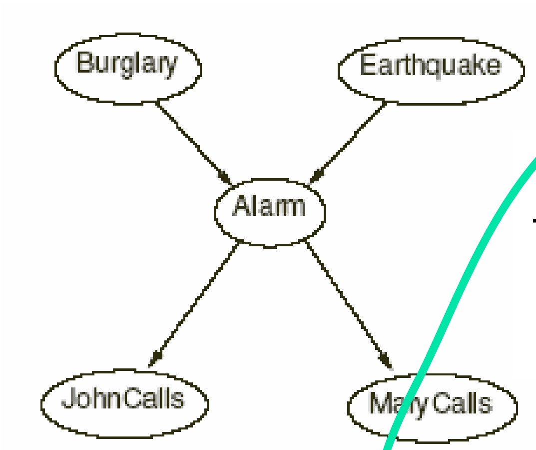
–b, +j, +m

$f_{-b}() = \lambda \langle\rangle.$

b	f(-b)
0	0.999

$f_E(e) = \lambda \langle e \rangle.$

e	f(e)
0	0.998
1	0.002



$f_{A,-b}(a,e) = \lambda \langle A,E \rangle.$

a	e	f(a, e, -b)
1	1	0.29
:	:	:
0	0	0.999

$f_{+j}(a) = \lambda \langle A \rangle.$

a	f(+j,a)
1	0.90
0	0.05

$f_{+m}(a) = \lambda \langle A \rangle.$

a	f(+m,a)
1	0.70
0	0.01

$b_{\{\}}$	b_B	b_E	b_A	b_J	b_M
$f_{\{\},1}() = \theta_{-b}$	-	$f_{E,1}(e) = \theta_e$	$f_{A,1}(a,e) = \theta_{a\|-b,e}$ $f_{A,2}(a) = \theta_{+j\|a}$ $f_{A,3}(a) = \theta_{+m\|a}$	-	-

"Variable Elimination": Factors

$$P(-b, +j, +m) = \underbrace{P(-b)}_{B} \underbrace{\sum_e P(e)}_{E} \underbrace{\sum_a P(a \mid -b, e)}_{A} \underbrace{P(+j \mid a)}_{J} \underbrace{P(+m \mid a)}_{M}$$

- Store intermediate results (factors) to avoid recomputation
  - Factor for M:
    2-element vector

$$\lambda_M(A) = \begin{pmatrix} P(+M \mid +A) \\ P(+M \mid -A) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$$

$$\lambda_M(+A) = 0.70; \quad \lambda_M(-A) = 0.01$$

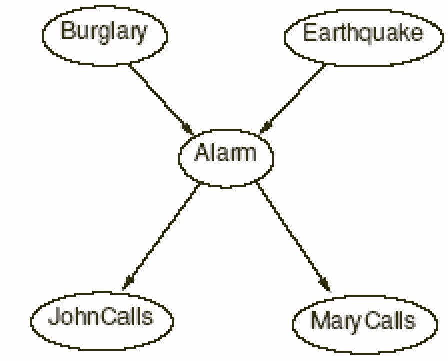
  - Factor for J:

$$\lambda_J(A) = \begin{pmatrix} P(+J \mid +A) \\ P(+J \mid -A) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix}$$

  - Factor for A:
    ≡ 4-element vector

$$\lambda_A(A, E) = \begin{pmatrix} P(+A \mid -B, +E) \\ P(+A \mid -B, -E) \\ P(-A \mid -B, +E) \\ P(-A \mid -B, -E) \end{pmatrix} = \begin{pmatrix} 0.29 \\ 0.001 \\ 0.71 \\ 0.999 \end{pmatrix}$$

BE Alg, con't



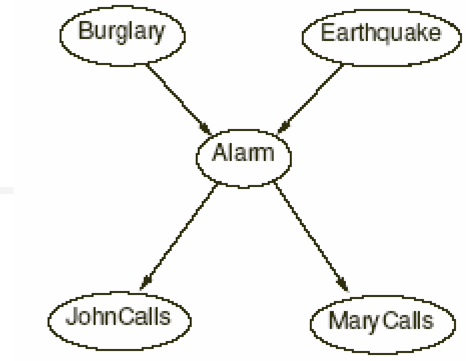
- Process buckets, from highest to lowest
  - $g_X := \text{elim}_X[\ f_{X,1} \bowtie f_{X,2} \bowtie \dots \bowtie f_{X,k}\ ]$
  - g_X is function of $\bigcup_i \text{Vars}(\ f_{X,I}\ ) - \{X\}$
    Let highest index by "Y"
  Store g_X into b_Y

- Process b_A
  - $g_A(e) = \text{elim}_A[\ f_{A,1} \bowtie f_{A,2} \bowtie f_{A,3}]$
  - add to b_E ...

$b_{\{\}}$	b_B	b_E	b_A	b_J	b_M
$f_{\{\},1}() = \theta_{-b}$	-	$f_{E,1}(e) = \theta_e$	$f_{A,1}(a,e) = \theta_{a\|-b,e}$ $f_{A,2}(a) = \theta_{+j\|a}$ $f_{A,3}(a) = \theta_{+m\|a}$	-	-

$$f_{E,2}(e) = \text{elim}_A[\ f_{A,1} \bowtie f_{A,2} \bowtie f_{A,3}]$$

BE Alg, con't



- Process buckets, from highest to lowest
  - $g_X := \text{elim}_X[\ f_{X,1} \bowtie f_{X,2} \bowtie \ldots \bowtie f_{X,k}\ ]$
  - g_X is function of $\bigcup_i \text{Vars}(\ f_{X,I}\ ) - \{X\}$
    Let highest index by "Y"

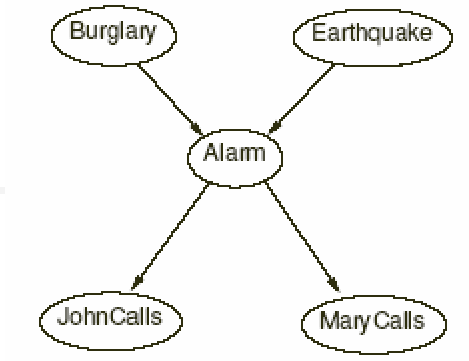
  Store g_X into b_Y

- Process b_E
  - $g_E() = \text{elim}_E[\ f_{E,1} \bowtie f_{E,2}]$
  - add to b_{nill} …

b_{nil}	b_B	b_E	b_A	b_J	b_M
$f_{\{\},1}() = \theta_{-b}$	-	~~$f_{E,1}(e) = \theta_e$~~ ~~$f_{E,2}(e) = \ldots$~~	$f_{A,1}(a,e) = \theta_{a\|-b,e}$ $f_{A,2}(a) = \theta_{+j\|a}$ $f_{A,3}(a) = \theta_{+m\|a}$	-	-

$f_{\{\},2}() = \text{elim}_E[\ f_{E,1} \bowtie f_{E,2}]$

BE Alg, con't



- Process buckets, from highest to lowest
  - $g_X := \text{elim}_X[\ f_{X,1} \bowtie f_{X,2} \bowtie \ldots \bowtie f_{X,k}\ ]$
  - g_X is function of $\cup_i \text{Vars}(\ f_{X,I}\ ) - \{X\}$
    Let highest index by "Y"

  Store g_X into b_Y

- Process $b_{\{\}}$
  - $g_{\{\}}() = [\ f_{\{\},1} \bowtie f_{\{\},2}]$
  - Return $g_{\{\}|}$ …

$b_{\{\}}$	b_B	b_E	b_A	b_J	b_M
~~$f_{\{\},1}() = \theta_{-b}$~~ ~~$f_{\{\},2}() = \ldots$~~	-	$f_{E,1}(e) = \theta_e$ $f_{E,2}(e) = \ldots$	$f_{A,1}(a,e) = \theta_{a\|-b,e}$ $f_{A,2}(a) = \theta_{+j\|a}$ $f_{A,3}(a) = \theta_{+m\|a}$	-	-

Return $f_{\{\},1} \bowtie f_{\{\},2}$

Bucket Elimination Algorithm

Given:

- Belief Net $BN = \langle N, A, C \rangle$
- Order of nodes $\pi = \langle X_1, \ldots, X_{|N|} \rangle$
- Evidence (nodes $\{E_i\} \subset N$, values $\{e_i\}$)
- (Single) Query node $X \in N$

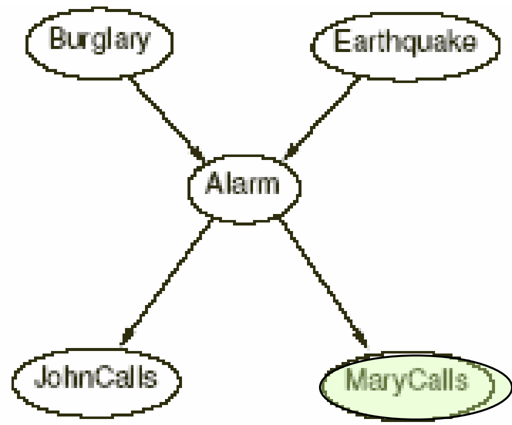
Compute: $P(X \mid E_1 = e_1, \ldots)$

by computing

$P(X = x, E_1 = e_1, \ldots) \; \forall x$

- Step#1: Initialize |N| +1 "buckets"
  - . . . bucket b_i for variable X_i
  - Each "instantiated form of CPtables" is function of variables
  - Store in bucket with highest index
- Step#2: Process each bucket
  - . . . from highest index down
  - to eliminate associated variable
- Step#3: Read off answer
  - . . . in "top" bucket, $b_{\{\}}$

Remove "Dead Variables"



$P(+b, +j) =$

$= \sum_e \sum_a \sum_m P(+b, E=e, A=a, +j, M=m)$

$= \sum_e \sum_a \sum_m P(+b)\, P(E=e)\, P(a|+b,e)\, P(+j|a)\, P(m|a)$

$= P(+b) \sum_e P(e) \sum_a P(a|+b,e)\, P(+j|a) \sum_m P(m|a)$

- Note for any A=a, $\sum_m P(M=m \mid a) = 1$

  $\Rightarrow$ can remove this node!
- In general: need to keep only nodes ABOVE query, evidence notes
  (Remove any nodes below)

Approaches to Belief Assessment

- Exact, Guaranteed
  - PolyTree Algorithm
  - Inherent complexity. . .
  - Clustering Approach
  - Bucket Elimination
  - CutSet Approach
- Approximate, Guaranteed
  - Algorithm Modification
  - Value Merging
  - Node Merging
  - Arc Removal
- Approximate, Probabilistic
  - Logic Sampling
  - Likelihood Sampling

Logic Sampling

What is P(WG = +) ?

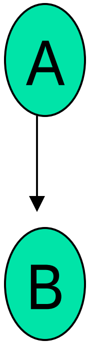
C	S	R	WG
+	0	0	+
+	+	+	0
+	+	0	0
0	+	0	+
0	0	+	0

5 tuples

- Get DataSample

- Of *5* tuples, *2* have WG = +
  Set P(WG=+) = 2/5

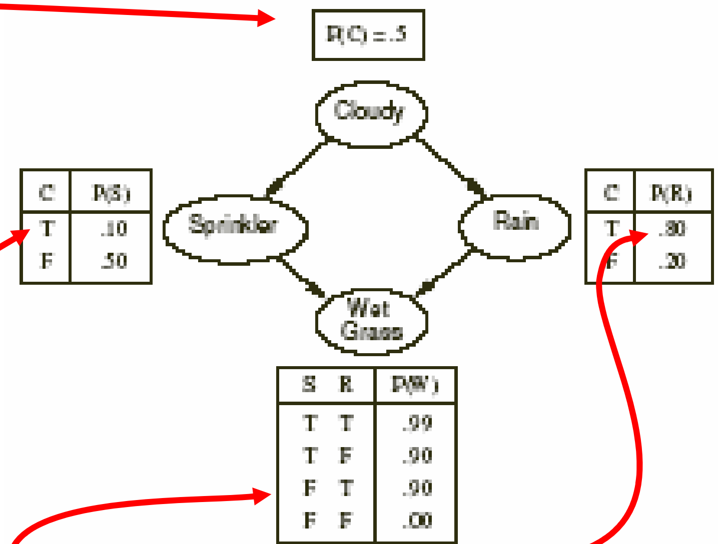
- But ... how to generate examples?
  - Uniform?? No!
    - What is P(+a, -b) ?
  - Based on distribution!!



a	P(+b|a)
+	1.0
-	0.0

Example of Logic Sampling

- To get value of "Cloudy": *Flip 0.5-coin*
  Assume "Cloudy = True"
- To get value of "Sprinkler": *Flip 0.1-coin*
  (as Cloudy = True, P(+s | +c) = 0.10)
  Assume "Sprinkler = False"
- To get value of "Rain": *Flip 0.8-coin*
  (as Cloudy = True, P(+r | +c) = 0.8)
  Assume "Rain = True"
- To get value of "WetGrass": *Flip 0.9-coin*
  (as Sprinkler = F, Rain = T, P(+w | ¬s, +r) = 0.9)
  Assume "WetGrass = True"



C	S	R	W
+	0	+	+
+	+	0	+
0	0	+	0
+	+	0	+

- On other trials, get other results, as different results of coin-flips

Stochastic Approximation 1: Logic Sampling

- To estimate P(X | E = e) :

> M := 0; m_i := 0 for each $x_i \in Dom(X)$
> Do until done
>   Generate random instance from BN
>   If $E = e$, M += 1
>   If $X = x_i$ and $E = e$, m_i += 1
> Return $\hat{P}(X = x_i \,|\, E = e) \approx m_i / M$

- To produce random instance from BN:

> For each root node $R \in \mathcal{N}$
>   draw $R = r_i$ with prob $P(\, R = r_i \,)$
> After finding values for all parents $U_1 = u_1, \ldots, U_k = u_k$,
>   of non-root W
>   draw $W = w_j$ with prob $P(W = w_j \,|\, U = u\,)$

PriorSample

- Note: if E ≠ e, just ignore instance

Aside: Flipping A Coin

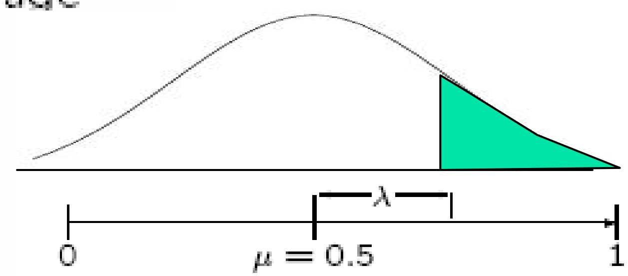
- Consider flipping a (fair) coin *m* times.
  ... expect to observe $\approx 0.5\, m$ heads
- Could have "bad run"
  ... suggesting coin is not fair.
- How (un)likely to observe $\geq$ **55%** heads?
  (10% more than expected)
- ... as function of *m*:
  What's probability of
  - (1) *m* = 100, $\geq$ 55 heads
  - (2) *m* = 500, $\geq$ 275 heads
  - (3) *m* = 1000, $\geq$ 550 heads
  - (4) *m* = 10,000, $\geq$ 5,500 heads ?

Using Chernoff Bounds

Let $X_i = \begin{cases} 1 & \text{if flip is heads} \\ 0 & \text{otherwise} \end{cases}$

- X_i's are iid... for now, with $\mu = 0.5$

Def'n: $S_m = \frac{1}{m}\sum_{i=1}^{m} X_i$ be observed average



- $Pr[S_m > \mu + \lambda] < e^{-2m\lambda^2}$

- Prob of $S_m > 0.55$ is $< e^{-2\ m\ 0.05\hat{}2}$

  m = 100 $\Rightarrow$ < 0.6

  m = 500 $\Rightarrow$ < 0.08

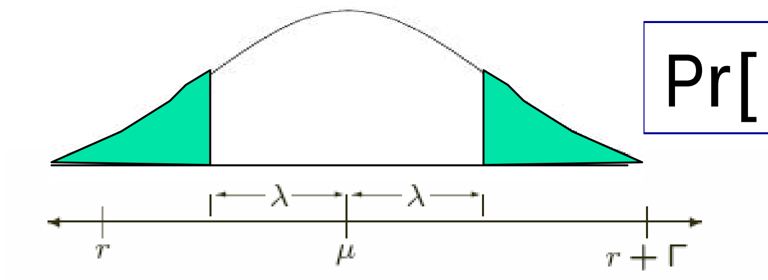
  m = 1,000 $\Rightarrow$ < 0.007

  m = 10,000 $\Rightarrow$ $< 10^{-22}$

Bad Runs are Rare

- $\Pr[\ S_m > \mu + \lambda\ ] < e^{-2m\,(\lambda/\Gamma)^2}$

  $\Pr[\ S_m < \mu - \lambda\ ] < e^{-2m\,(\lambda/\Gamma)^2}$



$$\Pr[\ |S_m - \mu\ | < \lambda\ ] \geq 1 - 2e^{-2m\,(\lambda/\Gamma)^2}$$

- Holds 8 (bounded) distributions!!!

  .. not just μ= 0.5… not just Bernoulli…
- Unrepresentative runs are **exponentially unlikely** in large samples!

⇒ Can get good results w/small ("polynomial") number of examples

- Aside: Secret behind randomized algs:

  Eg, estimating integrals, MonteCarlo simulation, . . .

  Can almost get "certainty" from probabilistic phenomenon

Use of DataSample (Logic Sampling)

- DataSample seen:

C	S	R	WG
+	0	0	+
+	+	+	0
+	+	0	0
0	+	0	+
0	0	+	0

5 tuples

- 5 tuples, including 2 with WG = +

$$\Rightarrow \text{Set} \quad \hat{P}(\text{WG} = +) = \frac{2}{5}$$

Consistent!
In the limit, produces correct answer.

- What about P(+c | +wg) ?
  Tuple is IRRELEVANT unless +wg
  so only 2 tuples relevant
  Of these: 1 has +c
  $\Rightarrow$ P(+c | +wg) = ½
- P(+r | +wg, +c) = 0/1 ??
  P(+c | +r, +wg) = ? 0/0

With *k* conditioning var's, expect ~ $(½)^k$ prob...

Stochastic Approximation 2: Likelihood Weighted Sampling

- Logic sampling is VERY SLOW if P(E) is low. . .
  as it ignores most tuples!
- INSTEAD... When generating tuples:
  Insist that each $E_i = e_i$
  ... but give it "weight" of $P(E_i = e_i \mid \mathbf{U} = \mathbf{u})$
  where $\mathbf{U}$ are E_i's parents, and $\mathbf{u}$ is current assignment to $\mathbf{U}$

Importance Sampling

Let W := 0; w_i := 0 for all x_i in $Dom(X)$
Do until done:
  Generate "random" instance from BN
    s.t. $E = e$
  Let p := $\prod_{E_i} P(E_i = e_i \mid U = u)$
  W += p
  If $X = x_i$, then w_i += p
Return $\hat{P}(X = x_i \mid E = e) \approx w_i / W$

Note:
$p \neq 1$!

Example of Likelihood Weighted Sampling

Want P(WetGrass |+Rain) :

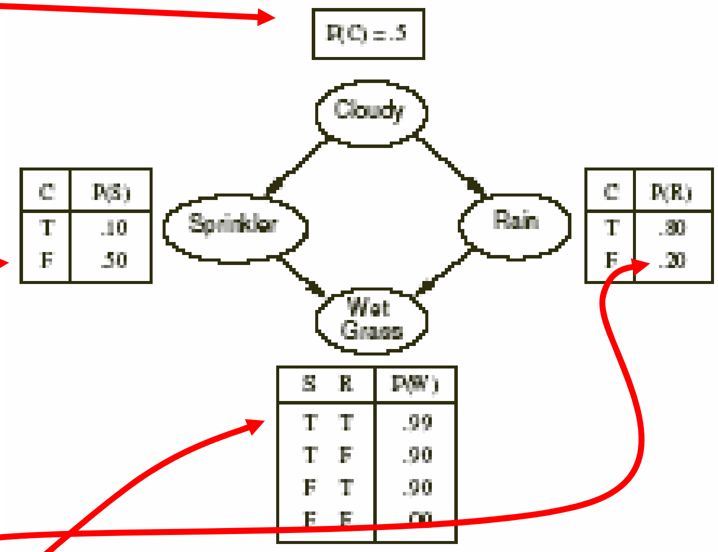
- To get value of Cloudy: Flip 0.5-coin
  Assume Cloudy = False

- To get value of Sprinkler: Flip 0.5-coin
  Assume Sprinker = True

- Now for "+Rain"
  ! evidence variable, so set to True !
  As Cloudy = False, P(+r | -c) = 0.2
  So this run counts as **0.2**

- To get value of WetGrass: Flip 0.99-coin
  Assume WetGrass = True

- So increment W by 0.2
  increment w_{+WG} by 0.2



Use of DataSample
(Logic Sampling, revisited)

- DataSample seen ... for Logic Sampling:

C	S	R	WG	(weight)	(weight when $\boxed{WG = +}$)
+	0	+	+	1	$\surd$
0	+	0	0	0	
0	+	0	0	0	
0	+	+	+	1	$\surd$
⋮	⋮	⋮	⋮		
0	0	+	0	1	
				7	2

- Out of 100 tuples, only 5 relevant ... with +r
- Of these 5, only 3 also have +wg

$\Rightarrow$ P(+wg | +r) = 3/5

Use of DataSample
(Likelihood Weighted Sampling)

- DataSample seen:

C	S	R	WG	(weight)	(weight when WG = +)
+	0	+	+	0.8	√
0	+	+	0	0.2	
0	+	+	0	0.2	
0	+	+	+	0.2	√
⋮	⋮	⋮	⋮		
0	0	+	0	0.2	
				1.6	1.0

- All 5 tuples now have +r
- Total "weight" – summing over ALL tuples: 1.6

  Weight, summing only when +wg : 1.0

$\Rightarrow$ P(+wg | +r) = 1.0/1.6

Other Techniques

- MCMC [Markov Chain Monte Carlo]
  - Move about in space of instances:
  - Fix evidence variables;
    guess at values of other variables
  - Guess new values of each non-evidence var, based on its distribution (markov blanket)
  - Collect instances... then take average
- Variational Methods

Other BN Tasks

- **MPE (Most Probable Explanation):**
  Given evidence $\mathbf{E = e}$ $(E_1 = e_1, \ldots, E_m = e_m)$
  - find assignment $\mathbf{x}$ that maximizes $P(\mathbf{x} \mid \mathbf{E = e})$

    $= \arg\max_x \prod_{i=1..m} P(x_i \mid e, pa_i)$
  - Alg $\approx$ like BucketElim for BeliefAssessment
    but replace $\sum$ with max
- **MAP (Maximum a Posteriori):**
  Given evidence $\mathbf{E = e}$

  and set of hypothesis $H_1, \ldots, H_k$
  - find assignment to HYPOTHESIS h that maximizes
    $P(h \mid \mathbf{E = e}) = \arg\max_h \prod_{i=1..m} P(x_i \mid \mathbf{e}, pa_i)$

Probabilistic Inference Tasks, in Gen'l

- **Simple queries**: compute posterior marginal P(X | E = e)
  - P(NoGas | Gauge = empty, Lights = on, Starts = false)
- **Conjunctive queries**:
  P(X, Y | E = e) = P(X | E = e) P(Y | X, E = e)
- **Optimal decisions**:
  - decision networks include utility information.
  - Probabilistic inference required for P(outcome | action, evid)
- **Value of information**: Which evidence to seek next?
- **Sensitivity analysis**:
  Which probability values are most critical?
- **Explanation**: Why do I need a new starter motor?

Summary

- Belief Net Inference is Intractable
  - In theory, and in Practice
- ... unless TREE-Structured
  - Fast O(n) algorithms
- Exact algorithms:
  - Many “reduce” to tree algorithm (cut-set, clustering)
  - Others “common out” redundancies
- Stochastic algorithms are effective
  - Need to worry about rare conditioning events