

Artificial Intelligence

Probabilistic Inference

Christopher Simpkins

Kennesaw State University

Exact Inference in Bayesian Networks

AIMA

Enumeration Algorithm

function ENUMERATION-ASK(X, \mathbf{e}, bn) **returns** a distribution over X

inputs: X , the query variable

\mathbf{e} , observed values for variables \mathbf{E}

bn , a Bayes net with variables $vars$

$Q(X) \leftarrow$ a distribution over X , initially empty

for each value x_i of X **do**

$Q(x_i) \leftarrow$ ENUMERATE-ALL($vars, \mathbf{e}_{x_i}$)

where \mathbf{e}_{x_i} is \mathbf{e} extended with $X = x_i$

return NORMALIZE($Q(X)$)

function ENUMERATE-ALL($vars, \mathbf{e}$) **returns** a real number

if EMPTY?($vars$) **then return** 1.0

$V \leftarrow$ FIRST($vars$)

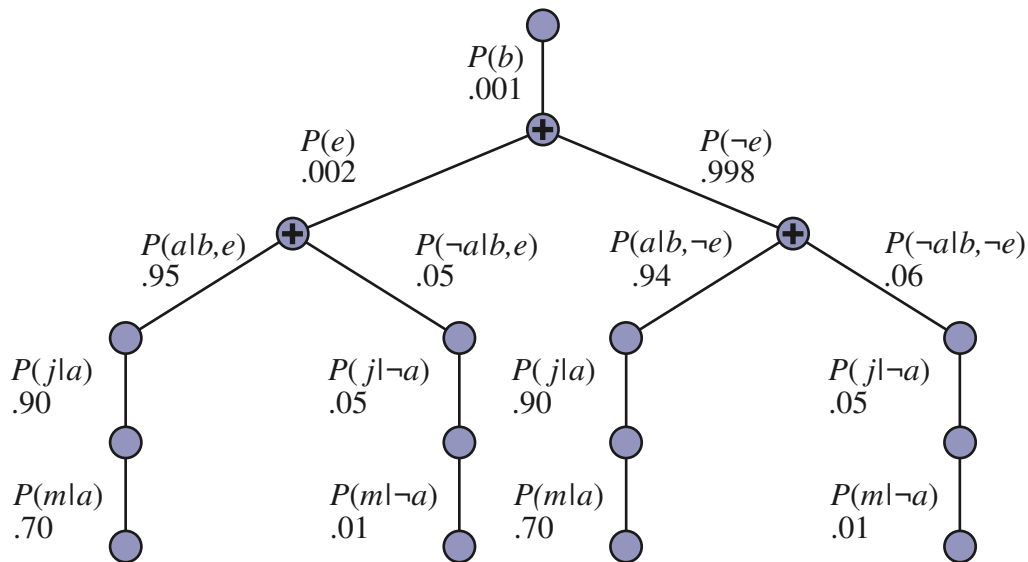
if V is an evidence variable with value v in \mathbf{e}

then return $P(v | parents(V)) \times$ ENUMERATE-ALL(REST($vars$), \mathbf{e})

else return $\sum_v P(v | parents(V)) \times$ ENUMERATE-ALL(REST($vars$), \mathbf{e}_v)

where \mathbf{e}_v is \mathbf{e} extended with $V = v$

Repeated Calculations



Pointwise Products

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

Variable Elimination Algorithm

function ELIMINATION-ASK(X, \mathbf{e}, bn) **returns** a distribution over X

inputs: X , the query variable

\mathbf{e} , observed values for variables \mathbf{E}

bn , a Bayesian network with variables $vars$

$factors \leftarrow []$

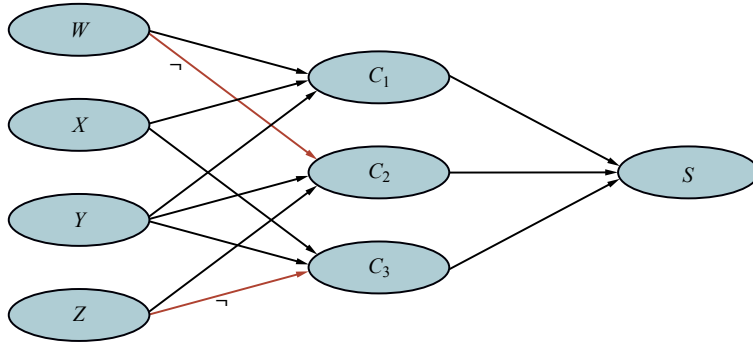
for each V **in** ORDER($vars$) **do**

$factors \leftarrow [\text{MAKE-FACTOR}(V, \mathbf{e})] + factors$

if V is a hidden variable **then** $factors \leftarrow \text{SUM-OUT}(V, factors)$

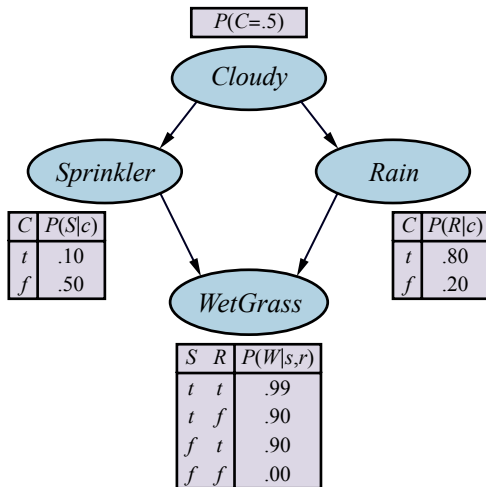
return NORMALIZE(POINTWISE-PRODUCT($factors$))

Complexity of Exact Inference

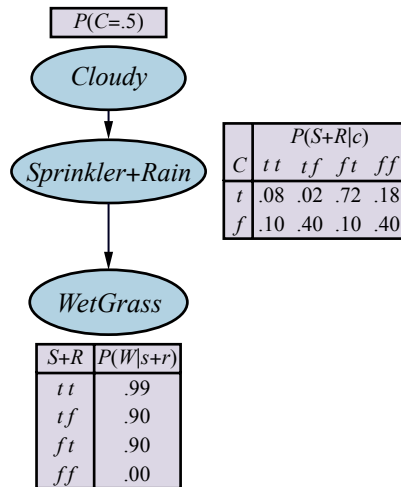


Clustering Algorithms

aka joint trees.



(a)



(b)

Direct Sampling Methods

Prior Sampling

function PRIOR-SAMPLE(bn) **returns** an event sampled from the prior specified by bn
inputs: bn , a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \dots, X_n)$

$\mathbf{x} \leftarrow$ an event with n elements
for each variable X_i **in** X_1, \dots, X_n **do**
 $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid \text{parents}(X_i))$
return \mathbf{x}

Rejection Sampling

function REJECTION-SAMPLING(X, \mathbf{e}, bn, N) **returns** an estimate of $\mathbf{P}(X | \mathbf{e})$

inputs: X , the query variable

\mathbf{e} , observed values for variables \mathbf{E}

bn , a Bayesian network

N , the total number of samples to be generated

local variables: \mathbf{C} , a vector of counts for each value of X , initially zero

for $j = 1$ **to** N **do**

$\mathbf{x} \leftarrow \text{PRIOR-SAMPLE}(bn)$

if \mathbf{x} is consistent with \mathbf{e} **then**

$\mathbf{C}[j] \leftarrow \mathbf{C}[j] + 1$ where x_j is the value of X in \mathbf{x}

return NORMALIZE(\mathbf{C})

Importance Sampling

function LIKELIHOOD-WEIGHTING(X, \mathbf{e}, bn, N) **returns** an estimate of $\mathbf{P}(X | \mathbf{e})$

inputs: X , the query variable

\mathbf{e} , observed values for variables \mathbf{E}

bn , a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \dots, X_n)$

N , the total number of samples to be generated

local variables: \mathbf{W} , a vector of weighted counts for each value of X , initially zero

for $j = 1$ **to** N **do**

$\mathbf{x}, w \leftarrow \text{WEIGHTED-SAMPLE}(bn, \mathbf{e})$

$\mathbf{W}[j] \leftarrow \mathbf{W}[j] + w$ where x_j is the value of X in \mathbf{x}

return $\text{NORMALIZE}(\mathbf{W})$

function WEIGHTED-SAMPLE(bn, \mathbf{e}) **returns** an event and a weight

$w \leftarrow 1$; $\mathbf{x} \leftarrow$ an event with n elements, with values fixed from \mathbf{e}

for $i = 1$ **to** n **do**

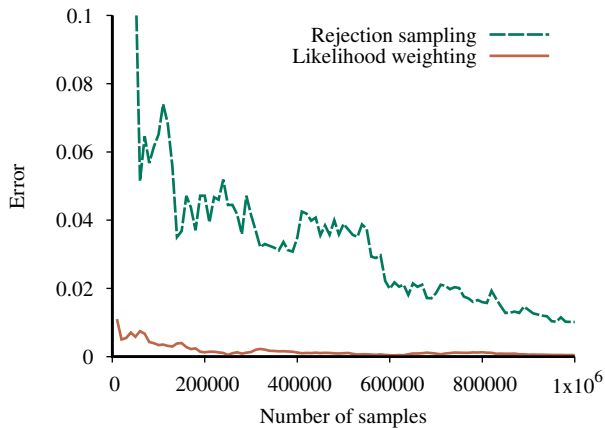
if X_i is an evidence variable with value x_{ij} in \mathbf{e}

then $w \leftarrow w \times P(X_i = x_{ij} | \text{parents}(X_i))$

else $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i | \text{parents}(X_i))$

return \mathbf{x}, w

Rejection vs. Importance Sampling



Markov Chain Monte Carlo (MCMC) Algorithms

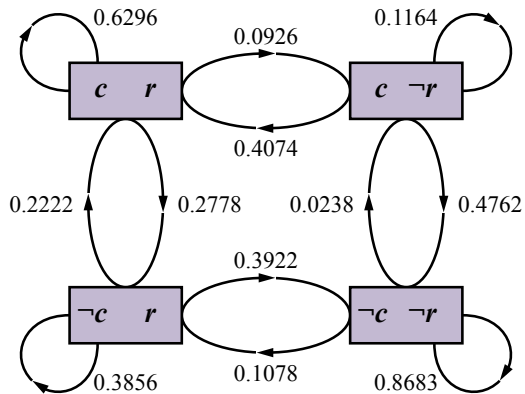
Instead of generating each sample from scratch, MCMC algorithms generate a sample by making a random change to the preceding sample. Think of an MCMC algorithm as being in a particular current state that specifies a value for every variable and generating a next state by making random changes to the current state.

Gibbs Sampling

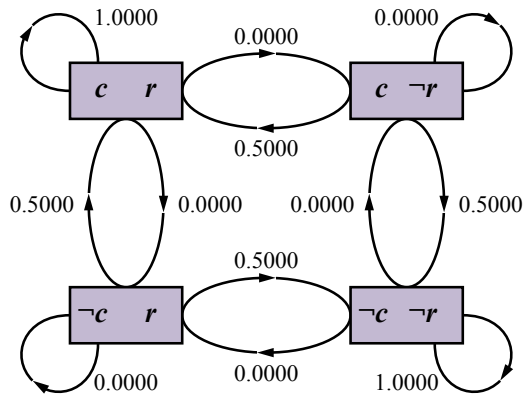
function GIBBS-ASK(X, \mathbf{e}, bn, N) **returns** an estimate of $\mathbf{P}(X | \mathbf{e})$
 local variables: \mathbf{C} , a vector of counts for each value of X , initially zero
 \mathbf{Z} , the nonevidence variables in bn
 \mathbf{x} , the current state of the network, initialized from \mathbf{e}

 initialize \mathbf{x} with random values for the variables in \mathbf{Z}
 for $k = 1$ **to** N **do**
 choose any variable Z_i from \mathbf{Z} according to any distribution $\rho(i)$
 set the value of Z_i in \mathbf{x} by sampling from $\mathbf{P}(Z_i | mb(Z_i))$
 $\mathbf{C}[j] \leftarrow \mathbf{C}[j] + 1$ where x_j is the value of X in \mathbf{x}
 return NORMALIZE(\mathbf{C})

Markov Chains

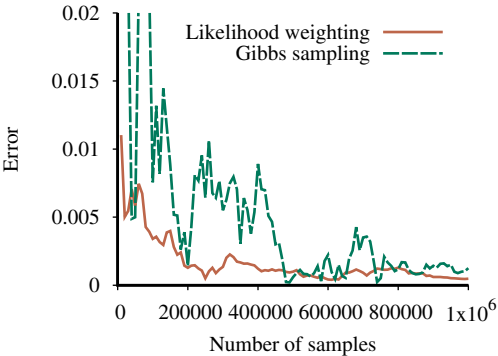


(a)

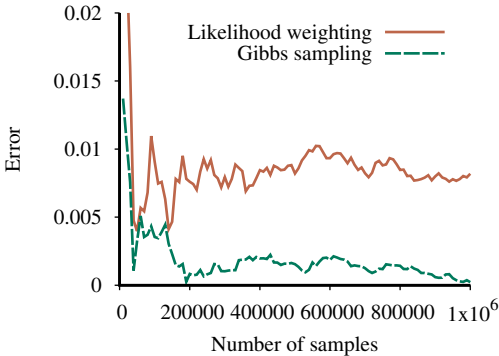


(b)

Gibbs Sampling vs. Importance Sampling



(a)



(b)