CSE 592
Applications of Artificial Intelligence

Winter 2003

Probabilistic Reasoning

Basics

- **Random variable**
  Cavity: yes or no
  P(Cavity) = 0.1
- **Conditional Probability**
  P(A|B)
  P(Cavity | Toothache) = 0.8
- **Joint Probability Distribution**
  (# variables)$^{(\text{\# values})}$ numbers
- **Bayes Rule**
  P(B|A) = P(A|B)P(B) / P(A)
- **(Conditional) Independence**
  P(A|C) = P(A)
  P(A | P,C) = P(A | C)

	Ache	No Ache
Cavity	0.04	0.06
No Cavity	0.01	0.89

Bayesian networks

A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

Syntax:
a set of nodes, one per variable
a directed, acyclic graph (link ≈ "directly influences")
a conditional distribution for each node given its parents:
$\mathbf{P}(X_i|Parents(X_i))$

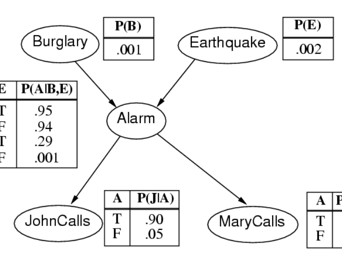
In the simplest case, conditional distribution represented as a conditional probability table (CPT) giving the distribution over X_i for each combination of parent values

Example

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

Variables: $Burglar$, $Earthquake$, $Alarm$, $JohnCalls$, $MaryCalls$
Network topology reflects "causal" knowledge:
- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call

Example contd.



P(B)
.001

P(E)
.002

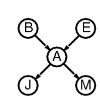
B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Compactness

A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values



Each row requires one number p for $X_i = true$ (the number for $X_i = false$ is just $1 - p$)

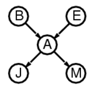
If each variable has no more than k parents, the complete network requires $O(n \cdot 2^k)$ numbers

I.e., grows linearly with n, vs. $O(2^n)$ for the full joint distribution

For burglary net, $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. $2^5 - 1 = 31$)

Global semantics

Global semantics defines the full joint distribution as the product of the local conditional distributions:



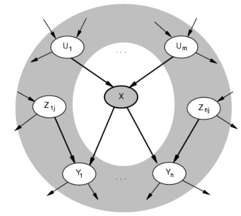
$$\mathbf{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i | Parents(X_i))$$

e.g., $P(j \land m \land a \land \neg b \land \neg e)$

$=$

Markov blanket

Each node is conditionally independent of all others given its Markov blanket: parents + children + children's parents



Constructing Bayesian networks

Need a method such that a series of locally testable assertions of conditional independence guarantees the required global semantics

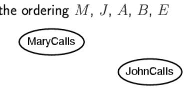
1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For $i = 1$ to n
   add X_i to the network
   select parents from $X_1, \ldots, X_{i-1}$ such that
   $\mathbf{P}(X_i | Parents(X_i)) = \mathbf{P}(X_i | X_1, \ldots, X_{i-1})$

This choice of parents guarantees the global semantics:

$$\begin{aligned} \mathbf{P}(X_1, \ldots, X_n) &= \prod_{i=1}^{n} \mathbf{P}(X_i | X_1, \ldots, X_{i-1}) \quad \text{(chain rule)} \\ &= \prod_{i=1}^{n} \mathbf{P}(X_i | Parents(X_i)) \quad \text{(by construction)} \end{aligned}$$

Example

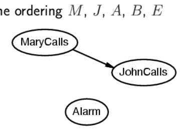
Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$?

Example

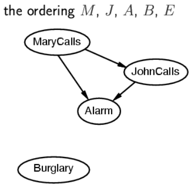
Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$? No
$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$?

Example

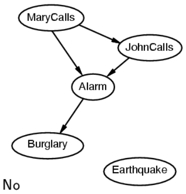
Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$? No
$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$? No
$P(B|A,J,M) = P(B|A)$?
$P(B|A,J,M) = P(B)$?

Example

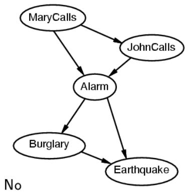
Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$? No
$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$? No
$P(B|A,J,M) = P(B|A)$? Yes
$P(B|A,J,M) = P(B)$? No
$P(E|B,A,J,M) = P(E|A)$?
$P(E|B,A,J,M) = P(E|A,B)$?

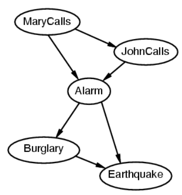
Example

Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$? No
$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$? No
$P(B|A,J,M) = P(B|A)$? Yes
$P(B|A,J,M) = P(B)$? No
$P(E|B,A,J,M) = P(E|A)$? No
$P(E|B,A,J,M) = P(E|A,B)$? Yes

Example contd.



Deciding conditional independence is hard in noncausal directions

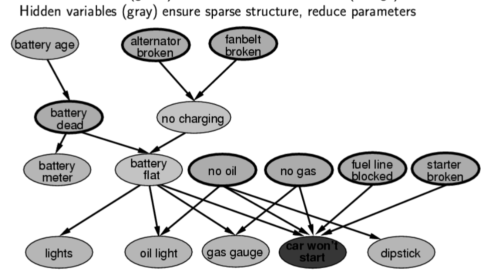
(Causal models and conditional independence seem hardwired for humans!)

Assessing conditional probabilities is hard in noncausal directions

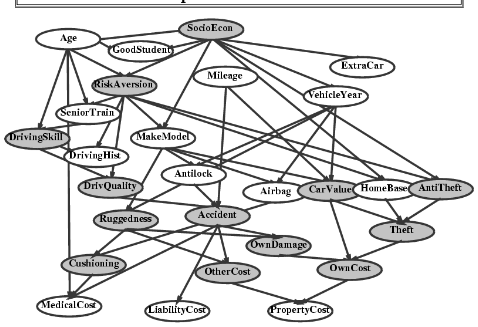
Network is less compact: $1+2+4+2+4=13$ numbers needed

Example: Car diagnosis

Initial evidence: car won't start
Testable variables (green), "broken, so fix it" variables (orange)
Hidden variables (gray) ensure sparse structure, reduce parameters



Example: Car insurance



Compact conditional distributions

CPT grows exponentially with no. of parents
CPT becomes infinite with continuous-valued parent or child

Solution: canonical distributions that are defined compactly

Deterministic nodes are the simplest case:
$X = f(Parents(X))$ for some function f

E.g., Boolean functions
$NorthAmerican \Leftrightarrow Canadian \vee US \vee Mexican$

E.g., numerical relationships among continuous variables
$\frac{\partial Level}{\partial t} =$ inflow + precipitation - outflow - evaporation

Compact conditional distributions contd.

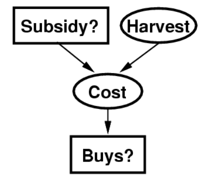
Noisy-OR distributions model multiple noninteracting causes
1) Parents $U_1 \ldots U_k$ include all causes (can add leak node)
2) Independent failure probability q_i for each cause alone
$\Rightarrow P(X|U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = 1 - \prod_{i=1}^{j} q_i$

$Cold$	Flu	$Malaria$	$P(Fever)$	$P(\neg Fever)$
F	F	F	**0.0**	1.0
F	F	T	0.9	**0.1**
F	T	F	0.8	**0.2**
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	**0.6**
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Number of parameters **linear** in number of parents

Hybrid (discrete+continuous) networks

Discrete ($Subsidy?$ and $Buys?$); continuous ($Harvest$ and $Cost$)



Option 1: discretization—possibly large errors, large CPTs
Option 2: finitely parameterized canonical families

1) Continuous variable, discrete+continuous parents (e.g., $Cost$)
2) Discrete variable, continuous parents (e.g., $Buys?$)

Continuous child variables

Need one conditional density function for child variable given continuous parents, for each possible assignment to discrete parents

Most common is the linear Gaussian model, e.g.,:

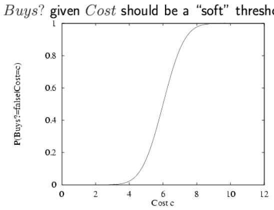
$$\begin{aligned} & P(Cost = c | Harvest = h, Subsidy? = true) \\ = \; & N(a_t h + b_t, \sigma_t)(c) \\ = \; & \frac{1}{\sigma_t \sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{c - (a_t h + b_t)}{\sigma_t}\right)^2\right) \end{aligned}$$

Mean $Cost$ varies linearly with $Harvest$, variance is fixed

Linear variation is unreasonable over the full range
but works OK if the **likely** range of $Harvest$ is narrow

Discrete variable w/ continuous parents

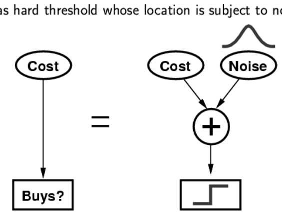
Probability of $Buys?$ given $Cost$ should be a "soft" threshold:



Probit distribution uses integral of Gaussian:
$\Phi(x) = \int_{-\infty}^{x} N(0,1)(x)dx$
$P(Buys? = true \mid Cost = c) = \Phi((-c + \mu)/\sigma)$

Why the probit?

1. It's sort of the right shape
2. Can view as hard threshold whose location is subject to noise



Summary

Bayes nets provide a natural representation for (causally induced) conditional independence

Topology + CPTs = compact representation of joint distribution

Generally easy for (non)experts to construct

Canonical distributions (e.g., noisy-OR) = compact representation of CPTs

Continuous variables $\Rightarrow$ parameterized distributions (e.g., linear Gaussian)

In-Class Exercise

- In groups of 2 or 3, sketch the structure of Bayes net that would be useful for diagnosing printing problems with Powerpoint
- How could the network be used by a Help wizard?
- 15 minutes

Outline

◇ Exact inference by enumeration

◇ Exact inference by variable elimination

◇ Approximate inference by stochastic simulation

◇ Approximate inference by Markov chain Monte Carlo

Inference tasks

Simple queries: compute posterior marginal $\mathbf{P}(X_i|\mathbf{E}=\mathbf{e})$
e.g., $P(NoGas|Gauge=empty, Lights=on, Starts=false)$

Conjunctive queries: $\mathbf{P}(X_i, X_j|\mathbf{E}=\mathbf{e}) = \mathbf{P}(X_i|\mathbf{E}=\mathbf{e})\mathbf{P}(X_j|X_i, \mathbf{E}=\mathbf{e})$

Optimal decisions: decision networks include utility information;
probabilistic inference required for $P(outcome|action, evidence)$

Value of information: which evidence to seek next?

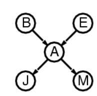
Sensitivity analysis: which probability values are most critical?

Explanation: why do I need a new starter motor?

Inference by enumeration

Slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation

Simple query on the burglary network:



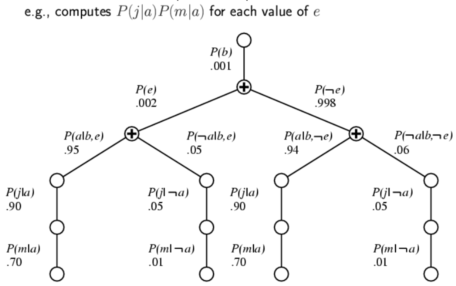
$\mathbf{P}(B|j, m)$
$= \mathbf{P}(B, j, m)/P(j, m)$
$= \alpha\mathbf{P}(B, j, m)$
$= \alpha\Sigma_e\Sigma_a\mathbf{P}(B, e, a, j, m)$

Rewrite full joint entries using product of CPT entries:
$\mathbf{P}(B|j, m)$
$= \alpha\Sigma_e\Sigma_a\mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a)$
$= \alpha\mathbf{P}(B)\Sigma_e P(e)\Sigma_a\mathbf{P}(a|B, e)P(j|a)P(m|a)$

Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time

Evaluation tree

Enumeration is inefficient: repeated computation
e.g., computes $P(j|a)P(m|a)$ for each value of e



Inference by variable elimination

Variable elimination: carry out summations right-to-left,
storing intermediate results (factors) to avoid recomputation

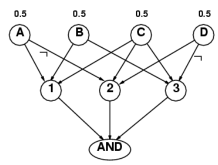
$$\begin{aligned}
&\mathbf{P}(B|j, m) \\
&\quad = \alpha\, \underbrace{\mathbf{P}(B)}_{B}\, \Sigma_e\, \underbrace{P(e)}_{E}\, \Sigma_a\, \underbrace{\mathbf{P}(a|B, e)}_{A}\, \underbrace{P(j|a)}_{J}\, \underbrace{P(m|a)}_{M} \\
&\quad = \alpha\mathbf{P}(B)\Sigma_e P(e)\Sigma_a\mathbf{P}(a|B, e)P(j|a)f_M(a) \\
&\quad = \alpha\mathbf{P}(B)\Sigma_e P(e)\Sigma_a\mathbf{P}(a|B, e)f_J(a)f_M(a) \\
&\quad = \alpha\mathbf{P}(B)\Sigma_e P(e)\Sigma_a f_A(a, b, e)f_J(a)f_M(a) \\
&\quad = \alpha\mathbf{P}(B)\Sigma_e P(e)f_{\bar{A}JM}(b, e) \text{ (sum out } A\text{)} \\
&\quad = \alpha\mathbf{P}(B)f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E\text{)} \\
&\quad = \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
\end{aligned}$$

Complexity of exact inference

Singly connected networks (or polytrees):
- any two nodes are connected by at most one (undirected) path
- time and space cost of variable elimination are $O(d^k n)$

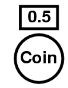
Multiply connected networks:
- can reduce 3SAT to exact inference $\Rightarrow$ NP-hard
- equivalent to **counting** 3SAT models $\Rightarrow$ #P-complete

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



Inference by stochastic simulation

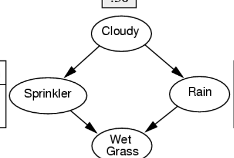
Basic idea:
1) Draw N samples from a sampling distribution S
2) Compute an approximate posterior probability $\hat{P}$
3) Show this converges to the true probability P



Outline:
- Sampling from an empty network
- Rejection sampling: reject samples disagreeing with evidence
- Likelihood weighting: use evidence to weight samples
- Markov chain Monte Carlo (MCMC): sample from a stochastic process whose stationary distribution is the true posterior

Example

P(C)
.50



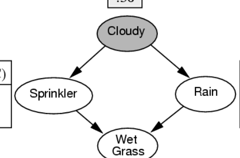
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example

P(C)
.50



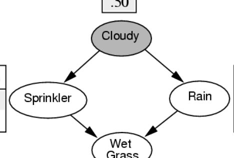
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example

P(C)
.50



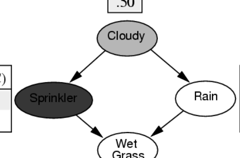
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example

P(C)
.50

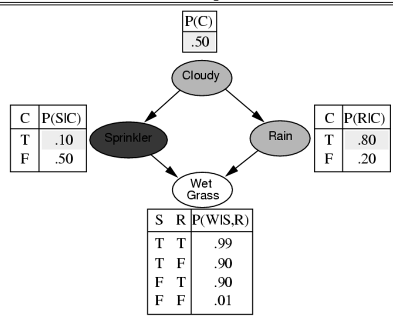


C	P(S|C)
T	.10
F	.50

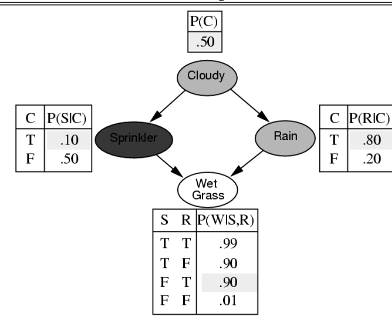
C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

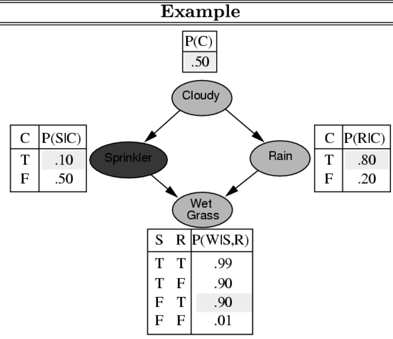
Example



Example



Example



Sampling from an empty network contd.

Probability that PRIORSAMPLE generates a particular event
$S_{PS}(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i | Parents(X_i)) = P(x_1 \ldots x_n)$
i.e., the true prior probability

E.g., $S_{PS}(t, f, t, t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 = P(t, f, t, t)$

Let $N_{PS}(x_1 \ldots x_n)$ be the number of samples generated for event $x_1, \ldots, x_n$

Then we have

$$\begin{aligned}\lim_{N\to\infty} \hat{P}(x_1, \ldots, x_n) &= \lim_{N\to\infty} N_{PS}(x_1, \ldots, x_n)/N \\ &= S_{PS}(x_1, \ldots, x_n) \\ &= P(x_1 \ldots x_n)\end{aligned}$$

That is, estimates derived from PRIORSAMPLE are consistent

Shorthand: $\hat{P}(x_1, \ldots, x_n) \approx P(x_1 \ldots x_n)$

Rejection sampling

$\hat{\mathbf{P}}(X|\mathbf{e})$ estimated from samples agreeing with $\mathbf{e}$

```
function REJECTION-SAMPLING(X, e, bn, N) returns an estimate of P(X|e)
   local variables: N, a vector of counts over X, initially zero
   for j = 1 to N do
       x ← PRIOR-SAMPLE(bn)
       if x is consistent with e then
           N[x] ← N[x]+1 where x is the value of X in x
   return NORMALIZE(N[X])
```

E.g., estimate $\mathbf{P}(Rain|Sprinkler = true)$ using 100 samples
27 samples have $Sprinkler = true$
Of these, 8 have $Rain = true$ and 19 have $Rain = false$.

$\hat{\mathbf{P}}(Rain|Sprinkler = true) = \text{NORMALIZE}(\langle 8, 19\rangle) = \langle 0.296, 0.704\rangle$

Similar to a basic real-world empirical estimation procedure

Analysis of rejection sampling

$\hat{\mathbf{P}}(X|\mathbf{e}) = \alpha \mathbf{N}_{PS}(X, \mathbf{e})$ (algorithm defn.)
$= \mathbf{N}_{PS}(X, \mathbf{e})/N_{PS}(\mathbf{e})$ (normalized by $N_{PS}(\mathbf{e})$)
$\approx \mathbf{P}(X, \mathbf{e})/P(\mathbf{e})$ (property of PRIORSAMPLE)
$= \mathbf{P}(X|\mathbf{e})$ (defn. of conditional probability)

Hence rejection sampling returns consistent posterior estimates

Problem: hopelessly expensive if $P(\mathbf{e})$ is small

$P(\mathbf{e})$ drops off exponentially with number of evidence variables!

Markov Chain Monte Carlo

MCMC with Gibbs Sampling

Fix the values of observed variables

Set the values of all non-observed variables randomly

Perform a random walk through the space of complete variable assignments. On each move:

1. **Pick a variable X**
2. **Calculate Pr(X=true | all other variables)**
3. **Set X to true with that probability**

Repeat many times. Frequency with which any variable X is true is it's posterior probability.

Converges to true posterior when frequencies stop changing significantly

- **stable distribution, mixing**

Markov Blanket Sampling

How to calculate Pr(X=true | all other variables) ?

Recall: a variable is independent of all others given it's Markov Blanket

- **parents**
- **children**
- **other parents of children**

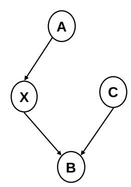
So problem becomes calculating Pr(X=true | MB(X))

- **We solve this sub-problem exactly**
- **Fortunately, it is easy to solve**

$$P(X) = \alpha P(X \mid Parents(X)) \prod_{Y \in Children(X)} P(Y \mid Parents(Y))$$

Example

$$P(X) = \alpha P(X \mid Parents(X)) \prod_{Y \in Children(X)} P(Y \mid Parents(Y))$$



$$P(X \mid A,B,C) = \frac{P(X,A,B,C)}{P(A,B,C)}$$

$$= \frac{P(A)P(X \mid A)P(C)P(B \mid X,C)}{P(A,B,C)}$$

$$= \left[\frac{P(A)P(C)}{P(A,B,C)}\right] P(X \mid A)P(B \mid X,C)$$

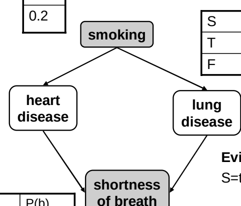
$$= \alpha P(X \mid A)P(B \mid X,C)$$

Example

P(s)
0.2

S	P(l)
T	0.8
F	0.1

S	P(s)
T	0.6
F	0.1



Evidence:
S=true, B=true

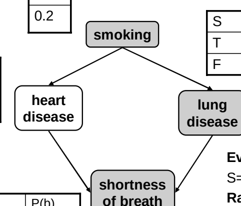
H	L	P(b)
T	T	0.9
T	F	0.8
F	T	0.7
F	F	0.1

Example 2

P(s)
0.2

S	P(l)
T	0.8
F	0.1

S	P(h)
T	0.6
F	0.1



Evidence:
S=true, B=true
Randomly set H=false, L=true

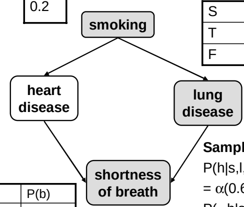
H	L	P(b)
T	T	0.9
T	F	0.8
F	T	0.7
F	F	0.1

Example 3

P(s)
0.2

S	P(l)
T	0.8
F	0.1

S	P(h)
T	0.6
F	0.1



H	L	P(b)
T	T	0.9
T	F	0.8
F	T	0.7
F	F	0.1

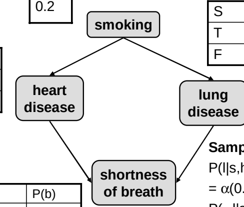
Sample H:
$P(h|s,l,b)=\alpha P(h|s)P(b|h,l)$
$= \alpha(0.6)(0.9)= \alpha\ 0.54$
$P(\neg h|s,l,b)=\alpha P(\neg h|s)P(b|\neg h,l)$
$= \alpha(0.4)(0.7)= \alpha\ 0.28$
Normalize: 0.54/(0.54+0.28)=0.66
Flip coin: H becomes true (maybe)

Example 4

P(s)
0.2

S	P(l)
T	0.8
F	0.1

S	P(h)
T	0.6
F	0.1



H	L	P(b)
T	T	0.9
T	F	0.8
F	T	0.7
F	F	0.1

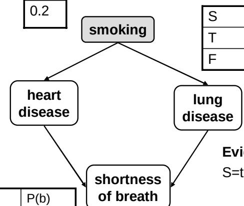
Sample L:
$P(l|s,h,b)=\alpha P(l|s)P(b|h,l)$
$= \alpha(0.8)(0.9)= \alpha\ 0.72$
$P(\neg l|s,h,b)=\alpha P(\neg l|s)P(b|h, \neg l)$
$= \alpha(0.2)(0.8)= \alpha\ 0.16$
Normalize: 0.72/(0.72+0.16)=0.82
Flip coin: …

Example 5: Different Evidence

P(s)
0.2

S	P(l)
T	0.8
F	0.1

S	P(s)
T	0.6
F	0.1



H	L	P(b)
T	T	0.9
T	F	0.8
F	T	0.7
F	F	0.1

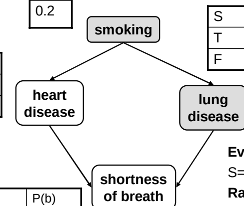
Evidence:
S=true, B=false

Example 6

P(s)
0.2

S	P(l)
T	0.8
F	0.1

S	P(h)
T	0.6
F	0.1



H	L	P(b)
T	T	0.9
T	F	0.8
F	T	0.7
F	F	0.1

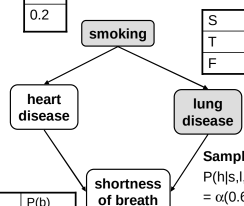
Evidence:
S=true, B=false
Randomly set H=false, L=true

Example 7

P(s)
0.2

S	P(l)
T	0.8
F	0.1

S	P(h)
T	0.6
F	0.1



H	L	P(b)
T	T	0.9
T	F	0.8
F	T	0.7
F	F	0.1

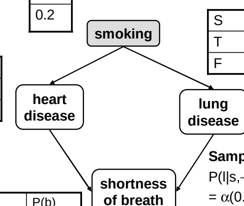
Sample H:
$P(h|s,l,\neg b)=\alpha P(h|s)P(\neg b|h,l)$
$= \alpha(0.6)(0.1)= \alpha\ 0.06$
$P(\neg h|s,l,\neg b)=\alpha P(\neg h|s)P(\neg b|\neg h,l)$
$= \alpha(0.4)(0.3)= \alpha\ 0.12$
Normalize: 0.06/(0.06+0.12)=0.33
Flip coin: H stays false (maybe)

Example 8

P(s)
0.2

S	P(l)
T	0.8
F	0.1

S	P(h)
T	0.6
F	0.1



H	L	P(b)
T	T	0.9
T	F	0.8
F	T	0.7
F	F	0.1

Sample L:
$P(l|s,\neg h,\neg b)=\alpha P(l|s)P(\neg b|\neg h,l)$
$= \alpha(0.8)(0.3)= \alpha\ 0.24$
$P(\neg l|s,\neg h,\neg b)=\alpha P(\neg l|s)P(\neg b|\neg h,\neg l)$
$= \alpha(0.2)(0.9)= \alpha\ 0.18$
Normalize: 0.24/(0.24+0.18)=0.75
Flip coin: …

Summary

Exact inference by variable elimination:
– polytime on polytrees, NP-hard on general graphs
– space = time, very sensitive to topology

Approximate inference by LW, MCMC: (and rejection sampling)
– LW does poorly when there is lots of (downstream) evidence
– LW, MCMC generally insensitive to topology
– Convergence can be very slow with probabilities close to 1 or 0
– Can handle arbitrary combinations of discrete and continuous variables

Outline

◇ Time and uncertainty

◇ Inference: filtering, prediction, smoothing

◇ Hidden Markov models (later on – next lecture)

◇ Kalman filters (a brief mention)

◇ Dynamic Bayesian networks

◇ Particle filtering

Time and uncertainty

The world changes; we need to track and predict it

Diabetes management vs vehicle diagnosis

Basic idea: copy state and evidence variables for each time step

$\mathbf{X}_t$ = set of unobservable state variables at time t
e.g., $BloodSugar_t$, $StomachContents_t$, etc.

$\mathbf{E}_t$ = set of observable evidence variables at time t
e.g., $MeasuredBloodSugar_t$, $PulseRate_t$, $FoodEaten_t$

This assumes **discrete time**; step size depends on problem

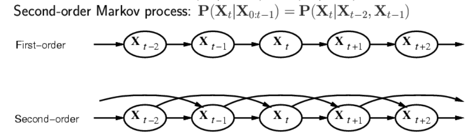
Notation: $\mathbf{X}_{a:b} = \mathbf{X}_a, \mathbf{X}_{a+1}, \ldots, \mathbf{X}_{b-1}, \mathbf{X}_b$

Markov processes (Markov chains)

Construct a Bayes net from these variables: parents?

Markov assumption: $\mathbf{X}_t$ depends on **bounded** subset of $\mathbf{X}_{0:t-1}$

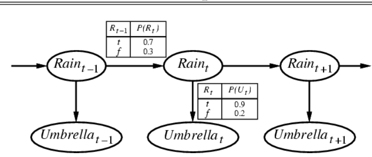
First-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
Second-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2}, \mathbf{X}_{t-1})$



Sensor Markov assumption: $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t}, \mathbf{E}_{0:t-1}) = \mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$

Stationary process: transition model $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$ and
sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$ fixed for all t

Example



First-order Markov assumption not exactly true in real world!

Possible fixes:
1. **Increase order** of Markov process
2. **Augment state**, e.g., add $Temp_t$, $Pressure_t$

Example: robot motion.
Augment position and velocity with $Battery_t$

Inference tasks

Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
belief state—input to the decision process of a rational agent

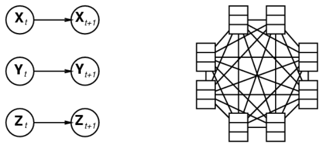
Prediction: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$ for $k > 0$
evaluation of possible action sequences;
like filtering without the evidence

Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \leq k < t$
better estimate of past states, essential for learning

Most likely explanation: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$
speech recognition, decoding with a noisy channel

DBNs vs. HMMs

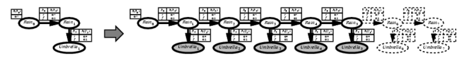
Every HMM is a single-variable DBN; every discrete DBN is an HMM



Sparse dependencies $\Rightarrow$ exponentially fewer parameters;
e.g., 20 state variables, three parents each
DBN has $20 \times 2^3 = 160$ parameters, HMM has $2^{20} \times 2^{20} \approx 10^{12}$

Exact inference in DBNs

Naive method: unroll the network and run any exact algorithm



Problem: inference cost for each update grows with t

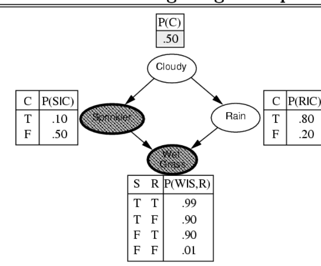
Rollup filtering: add slice $t+1$, "sum out" slice t using variable elimination

Largest factor is $O(d^{n+1})$, update cost $O(d^{n+2})$
(cf. HMM update cost $O(d^{2n})$)

Approximate Inference in DBN's

- Most popular technique today is particle filtering
- Modification of a sampling technique called Likelihood Weighting
- Idea:
  - Fix evidence variables
  - Sample non-evidence variables
  - Weight each sample by the likelihood it accords the evidence

Likelihood weighting example



P(C)
.50

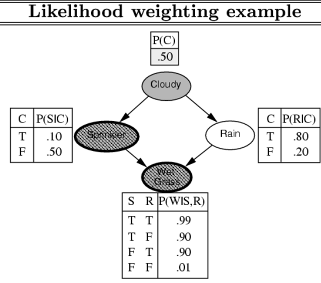
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

$w = 1.0$

Likelihood weighting example



P(C)
.50

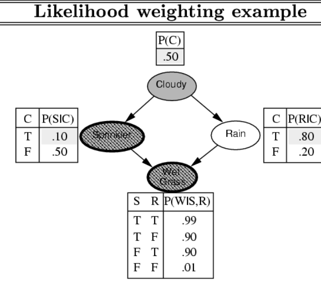
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

$w = 1.0$

Likelihood weighting example



P(C)
.50

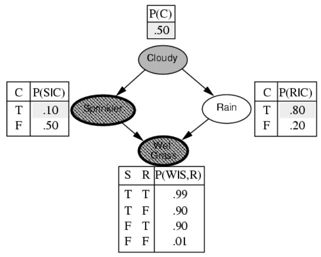
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

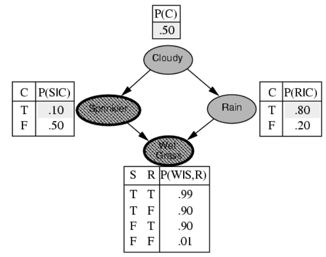
$w = 1.0$

Likelihood weighting example



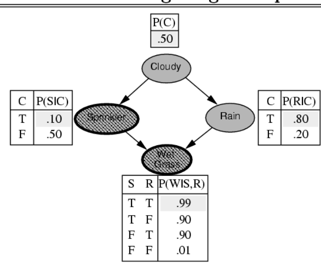
$w = 1.0 \times 0.1$

Likelihood weighting example



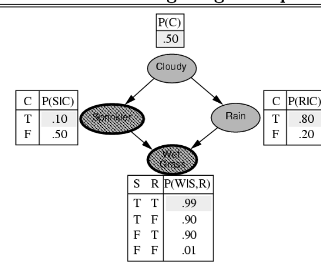
$w = 1.0 \times 0.1$

Likelihood weighting example



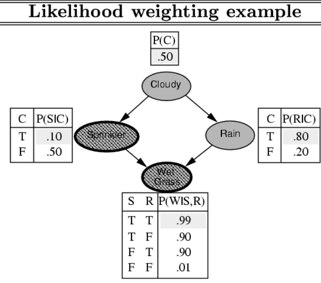
$w = 1.0 \times 0.1$

Likelihood weighting example



$w = 1.0 \times 0.1$

Likelihood weighting example

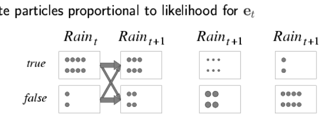


$w = 1.0 \times 0.1 \times 0.99 = 0.099$

Particle filtering

Basic idea: ensure that the population of samples ("particles") tracks the high-likelihood regions of the state-space

Replicate particles proportional to likelihood for $\mathbf{e}_t$



(a) Propagate (b) Weight (c) Resample

Widely used for tracking nonlinear systems, esp. in vision

Also used for simultaneous localization and mapping in mobile robots
10^5-dimensional state space

Particle filtering contd.

Assume consistent at time t: $N(\mathbf{x}_t|\mathbf{e}_{1:t})/N = P(\mathbf{x}_t|\mathbf{e}_{1:t})$

Propagate forward: populations of $\mathbf{x}_{t+1}$ are

$N(\mathbf{x}_{t+1}|\mathbf{e}_{1:t}) = \Sigma_{\mathbf{x}_t} P(\mathbf{x}_{t+1}|\mathbf{x}_t) N(\mathbf{x}_t|\mathbf{e}_{1:t})$

Weight samples by their likelihood for $\mathbf{e}_{t+1}$:

$W(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1}) = P(\mathbf{e}_{t+1}|\mathbf{x}_{t+1}) N(\mathbf{x}_{t+1}|\mathbf{e}_{1:t})$

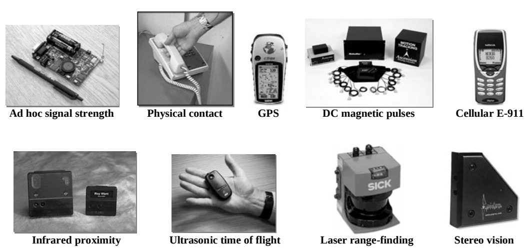
Resample to obtain populations proportional to W:

$$\begin{aligned} N(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1})/N &= \alpha W(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1}|\mathbf{x}_{t+1}) N(\mathbf{x}_{t+1}|\mathbf{e}_{1:t}) \\ &= \alpha P(\mathbf{e}_{t+1}|\mathbf{x}_{t+1}) \Sigma_{\mathbf{x}_t} P(\mathbf{x}_{t+1}|\mathbf{x}_t) N(\mathbf{x}_t|\mathbf{e}_{1:t}) \\ &= \alpha' P(\mathbf{e}_{t+1}|\mathbf{x}_{t+1}) \Sigma_{\mathbf{x}_t} P(\mathbf{x}_{t+1}|\mathbf{x}_t) P(\mathbf{x}_t|\mathbf{e}_{1:t}) \\ &= P(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1}) \end{aligned}$$

The Location Stack: Design and Sensor-Fusion for Location-Aware Ubicomp

Jeffrey Hightower

A survey & taxonomy of location technologies

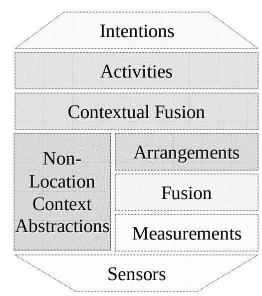


[Hightower and Borriello, *IEEE Computer*, Aug 2001]

The Location Stack

5 Principles

1. *There are fundamental measurement techniques.*
2. *There are standard ways to combine measurements.*
3. *There are standard object relationship queries.*
4. *Applications are concerned with activities.*
5. *Uncertainty is important.*



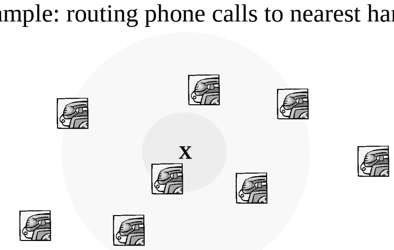
[Hightower, Brumitt, and Borriello, *WMCSA*, Jan 2002]

Principle 4: *Applications are concerned with activities.*

- Dinner is in progress.
- A presentation is going on in Mueller 153.
- Jeff is walking through his house listening to The Beatles.
- Jane is dispensing ethylene-glycol into beaker #45039.
- Elvis has left the building.

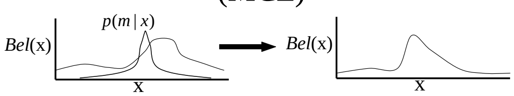
Principle 5: *Uncertainty is important.*

Example: routing phone calls to nearest handset

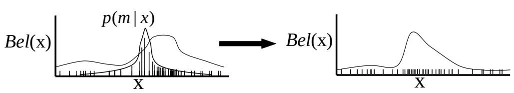


[Hightower and Borriello, *Ubicomp LMUC Workshop*, Sep 2001]

Fusion using Monte Carlo localization (MCL)



$$Bel(x_t) = p(x_t \mid m_t ... m_0)$$
$$Bel(x_t) = \eta p(m_t \mid x_t) \int p(x_t \mid x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

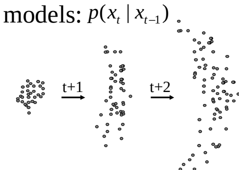


79

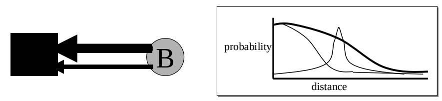
MCL details

Motion models: $p(x_t \mid x_{t-1})$

Stochastically shift all particles



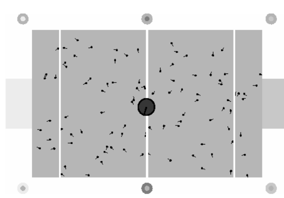
Sensor likelihood models: $p(m_t \mid x_t)$



80

2D MCL Example: Robocup

- 1 Object
- 2 types of Measurements
  - Vision marker distance
  - Odometry
- Red dot is most likely state.
  - (x,y,orientation)



[Fox et al., *Sequential Monte Carlo Methods in Practice*, 2000]

81

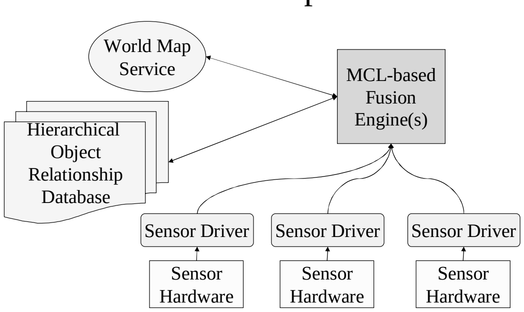
Adaptive MCL

- Performance improvement: adjust sample count to best represent the posterior.
  1. Assume we know the true *Bel(x)* represented as a multinomial distribution.
  2. Determine number of samples such that with probability (*1-p)*, the Kullback-Leibler distance between the true posterior and the particle filter representation is less than ε

[Fox, *NIPS*, 2002]

82

Location Stack Implementation



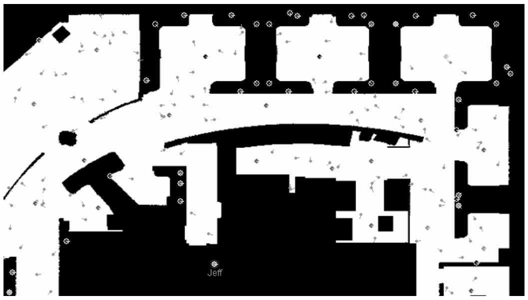
83

Location Stack Supported Technologies

1. VersusTech commercial infrared badge proximity system
2. RF Proximity using the Berkeley motes
3. SICK LMS-200 180º infrared laser range finders
4. MIT Cricket ultrasound range beacons
5. Indoor harmonic radar, *in progress*
6. 802.11b WiFi triangulation system, *in progress*
7. Cellular telephone E-OTD, *planned*

84

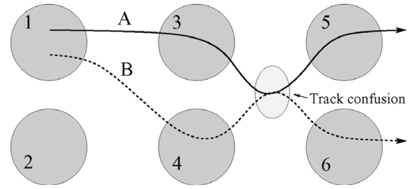
The Location Stack in action



85

Person Tracking with Anonymous and Id-Sensors: Motivation

- Accurate anonymous sensors exist
- Id-sensors are less accurate but provide explicit object identity information.



86

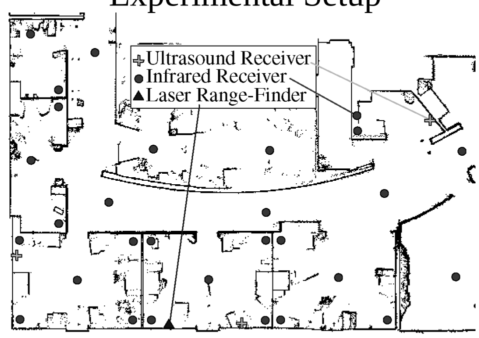
Person Tracking with Anonymous and Id-Sensors: Concept

- Use Rao-Blackwellised particle filters to efficiently estimate locations
  1. Each particle is an association history between Kalman filter object tracks and observations.
  2. Due to initial id uncertainty, starts by tracking using only anonymous sensors and estimating object id's with sufficient statistics.
  3. Once id estimates are certain enough, sample id them using a fully Rao-Blackwellised particle filter over both object tracks and id assignments.

[Fox, Hightower, and Schulz., *Submitted to IJCAI*, 2003]

87

Experimental Setup



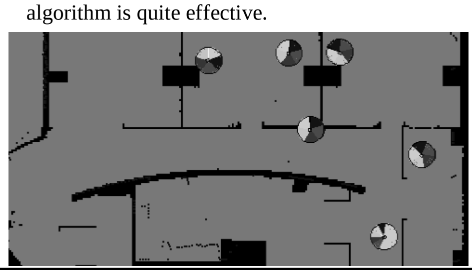
88

Experimental Setup

89

Person Tracking with Anonymous and Id-Sensors: Result

- Our 2 phase Rao-Blackwellised particle filter algorithm is quite effective.



90

Conclusion

*Relying on a single location technology to support all UbiComp applications is inappropriate. Instead, the **Location Stack** provides:*

1. The ability to fuse measurements from many technologies including both anonymous and id-sensors while preserving sensor uncertainty models.
2. Design abstractions enabling system evolution as new sensor technologies are created.
3. A common vocabulary to partition the work and research problems appropriately.

91

Future Work

- Further evaluate the Location Stack through use in real research and commercial applications.
- Collaboration with machine learning community to work on contextual fusion and activity inference.

92