

Approximate inference in Bayes nets. Temporal reasoning.

Ch. 14.5 and 15.1- 3

Wed, Oct 22, 2008

COMP 131, Lecture 12

1

Announcements

- A5 due today 11:59pm
- A6 out
- A4 solutions available next Monday in class

Wed, Oct 22, 2008

COMP 131, Lecture 12

2

Last time

- Bayes nets
- Exact inference

Wed, Oct 22, 2008

COMP 131, Lecture 12

3

Today

- Approximate inference in Bayes nets
- Temporal information and reasoning

Wed, Oct 22, 2008

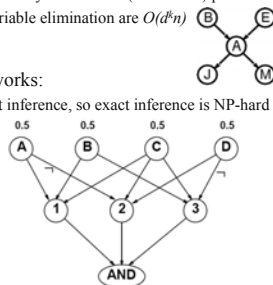
COMP 131, Lecture 12

4

Complexity of exact inference

- Singly connected networks (**polytrees**):
 - any two nodes are connected by at most one (undirected) path
 - time and space cost of variable elimination are $O(d^n)$
- Multiply connected networks:
 - can reduce 3SAT to exact inference, so exact inference is NP-hard

1. $A \vee B \vee C$
2. $C \vee D \vee \neg A$
3. $B \vee C \vee \neg D$



Wed, Oct 22, 2008

COMP 131, Lecture 12

5

Approximate inference

- Inference by stochastic simulation (Monte Carlo methods)
 - generate samples from a distribution
 - approximate probabilities with sample frequencies
 - answer accuracy depends on the number of samples
- Direct sampling
- Rejection sampling
- Likelihood weighting
- Markov Chain Monte Carlo (MCMC)

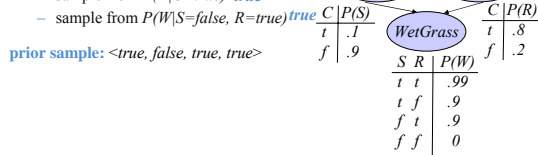
Wed, Oct 22, 2008

COMP 131, Lecture 12

6

Direct sampling

- Sample a distribution = generate an event from it
 - flip a coin
 - observe an event like a phone call by Mary or an alarm ringing
- In a Bayes net
 - sample from $P(C)$ *true*
 - sample from $P(S|C=true)$ *false*
 - sample from $P(R|C=true)$ *true*
 - sample from $P(W|S=false, R=true)$ *true*



Wed, Oct 22, 2008

COMP 131, Lecture 12

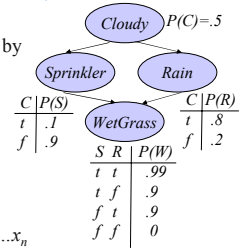
7

Direct sampling (cont.)

- Probability of an event generated by sampling

$$S_{ps}(x_1 \dots x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i)) = P(x_1 \dots x_n)$$

joint prior distribution



- If there are N samples total, and sample frequency for an event $x_1 \dots x_n$ is $N_{ps}(x_1 \dots x_n)$:

$$\lim_{N \rightarrow \infty} N_{ps}(x_1 \dots x_n) / N = S_{ps}(x_1 \dots x_n) = P(x_1 \dots x_n)$$

more samples = better estimate

if out of 1000 samples, 511 have $\text{Rain}=\text{true}$, we can estimate $P(\text{Rain}=\text{true})=0.511$

Wed, Oct 22, 2008

COMP 131, Lecture 12

8

Rejection sampling

- First, generate prior samples directly
- Then, reject all those that do not match the evidence
- Estimate $\bar{P}(X=x|e)$ by counting how often $X=x$ occurs in remaining samples
- Consistent estimate of true probability:
 - $\bar{P}(X|e) = N_{ps}(X, e) / N_{ps}(e)$ by algorithm definition
 - has $P(X|e)$ as its limit as $N_{ps}(X, e)$ increases
- Similar to a basic empirical estimation procedure

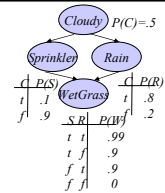
Wed, Oct 22, 2008

COMP 131, Lecture 12

9

Likelihood weighting

- But we're rejecting so many samples!
- **Likelihood weighting algorithm:**
 - fix evidence variables E first
 - sample only from remaining variables X, Y
 - before tallying event counts, weigh each event by the **likelihood** it accords the evidence (product of conditional probabilities for each E given its parents)
 - events in which evidence appears unlikely will be given less weight



query: $P(\text{Rain} | S=\text{true}, WG=\text{true})$? initially, $w=1$

1. sample from $P(C)$, suppose *true*
 2. $S = \text{true}$
 3. sample from $P(R|C=\text{true})$, *true*
 4. $WG = \text{true}$
- so, set $w := w P(S=\text{true}|C=\text{true}) = 0.1$ $w := w P(WG | s, c) = 0.099$

Wed, Oct 22, 2008

COMP 131, Lecture 12

10

Markov Chain Monte Carlo (MCMC)

- Remember “full state description” search (e.g., simulated annealing)?
- MCMC generates the next sample by modifying one variable value (for non-evidence vars) in the current sample
 - “state” = atomic event, sample (all variables assigned)
 - process is a **random walk** in the space of possible complete assignments

query: $P(\text{Rain} | S=t, WG=t)$?

1. Fix evidence vars $S=t, WG=t$

2. Assign hidden vars randomly, say $R=f, C=t$

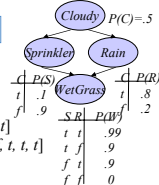
initial state: $[t, t, f, t]$

loop: 1. Sample C from $P(C | S=t, WG=t)$, say $[t, f, f, t]$

2. Sample R from $P(R | C=f, S=t, WG=t)$, say $[f, t, t, t]$

normalize count of $R=t$ and $R=f$ in all samples

Markov blanket



Wed, Oct 22, 2008

COMP 131, Lecture 12

11

Temporal reasoning

- So far, we've assumed *static* worlds
 - car diagnosis: whatever was broken stays broken while the agent is inferring what broke...
- Patient diagnosis and treatment selection
- Playing a video game
- Driving a car
- Navigating unknown terrain

all require reasoning about changing states and predicting the future

Wed, Oct 22, 2008

COMP 131, Lecture 12

12

States and observations

- Similar to situation calculus ideas: **time slices** or snapshots of a process
- X_t – unobserved (hidden) variables at time t
- E_t – observed (evidence) variables at time t : $E_t = e_t$
- Order variables chronologically to specify dependencies
- Problems:
 - unbounded number of conditional probabilities
 - unbounded number of parents

Wed, Oct 22, 2008

COMP 131, Lecture 12

13

Stationary Markov processes

- Process is governed by laws that don't themselves change with time (not to be confused with *static*)
 - $P(X_t | \text{Parents}(X_t))$ for all t

- Markov assumption: current state depends on *finite history* only

First-order Markov process:

$$P(X_t | X_{0:t-1}) = P(X_t | X_{t-1})$$

transition model

$$P(E_t | X_{0:t}, E_{0:t-1}) = P(E_t | X_t)$$

sensor model

$$P(X_0)$$



Andrei Markov
1856 – 1922

Wed, Oct 22, 2008

COMP 131, Lecture 12

14

1st order MP joint distribution

Complete joint distribution over all variables:

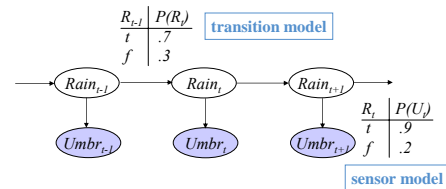
$$P(X_0, X_1, \dots, X_t, E_1, \dots, E_t) = P(X_0) \prod_{i=1}^t P(X_i | X_{i-1}) P(E_i | X_i)$$

Wed, Oct 22, 2008

COMP 131, Lecture 12

15

Rain/Umbrella world



a dynamic Bayes net (DBN)

Wed, Oct 22, 2008

COMP 131, Lecture 12

16

Inference in temporal models

- Filtering: $P(X_t | e_{1:t})$
 - computing the **belief state** (input into the decision process of a rational agent)
- Prediction: $P(X_{t+k} | e_{1:t})$
 - computing distribution over future states, perhaps for evaluating action sequences
- Smoothing (hindsight): $P(X_{t+k} | e_{1:t})$
 - better estimates of past states, perhaps for learning
- Most likely explanation: $\arg \max_{x_{1:t}} P(x_{1:t} | e_{1:t})$
 - given a sequence of observations, compute the most likely sequence of states that could have generated them

Wed, Oct 22, 2008

COMP 131, Lecture 12

17

Filtering

- A **recursive** state estimation algorithm
 - given our estimate at time t , and some evidence e_{t+1} , compute estimate for time $t+1$

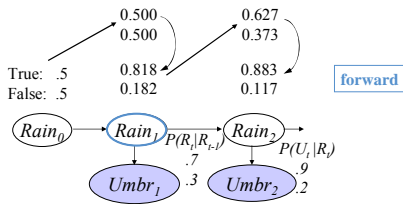
$$\begin{aligned}
 P(X_{t+1} | e_{1:t+1}) &= P(X_{t+1} | e_{1:t}, e_{t+1}) && \text{divide up evidence} \\
 &= \alpha P(e_{t+1} | X_{t+1}, e_{1:t}) P(X_{t+1} | e_{1:t}) && \text{Bayes rule} \\
 &= \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | e_{1:t}) && \text{Markov property} \\
 &= \alpha P(e_{t+1} | X_{t+1}) \sum_{x_t} P(X_{t+1} | x_t) P(x_t | e_{1:t}) && \text{sensor model} \quad \text{one-step prediction} \\
 &= \alpha P(e_{t+1} | X_{t+1}) \sum_{x_t} P(X_{t+1} | x_t) P(x_t | e_{1:t}) && \text{Markov property} \\
 f_{1:t+1} &= \alpha \text{FORWARD}(f_{1:t}, e_{t+1})
 \end{aligned}$$

Wed, Oct 22, 2008

COMP 131, Lecture 12

18

Rain/umbrella example



step $t=1$: $P(R_1) = \sum_0 P(R_1|r_0)P(r_0) = [0.7 \cdot 0.5 + [0.3 \cdot 0.5] = [0.5 \cdot 0.5]$
 $P(R_1|u_1) = \alpha P(u_1|R_1)P(R_1) = \alpha [0.2][0.5] = \alpha [0.1]$

step $t=2$: $P(R_2|u_1) = \sum_{r_1} P(R_2|r_1)P(r_1|u_1) = [0.7 \cdot 0.3] + [0.3 \cdot 0.7] = [0.627 \cdot 0.373]$
 $P(R_2|u_1, u_2) = \alpha P(u_2|R_2)P(R_2|u_1) = \alpha [0.2][0.627 \cdot 0.373]$
 $= \alpha [0.565 \cdot 0.075] \approx [0.883 \cdot 0.117]$

Wed, Oct 22, 2008 COMP 131, Lecture 12 19

Prediction

- Like filtering but without any new evidence
- We already know how to predict one step ahead:

$$P(X_{t+1}, e_{1:t}) = \sum_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t}) \quad \text{no new evidence } e_{t+1}$$

transition model

- If we know the state distribution for k steps ahead, we can predict the distribution for $k+1$ steps ahead recursively:

$$P(X_{t+k+1}, e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1}|x_{t+k})P(x_{t+k}|e_{1:t})$$

Wed, Oct 22, 2008 COMP 131, Lecture 12 20

As we predict more and more...

$$T = \begin{bmatrix} P(x_1|x_1) & \dots & P(x_1|x_n) \\ \vdots & \ddots & \vdots \\ P(x_n|x_1) & \dots & P(x_n|x_n) \end{bmatrix} \quad P = \begin{bmatrix} P(x_1) \\ \vdots \\ P(x_n) \end{bmatrix}$$

transition model belief state (state distribution)

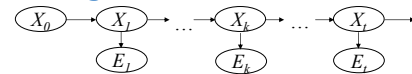
$$P(X_{t+1}) = \sum_{x_t} P(X_{t+1}|x_t)P(x_t) = \sum_{x_t} P(X_{t+1}|x_t) \sum_{x_{t-1}} P(X_t|x_{t-1})P(x_{t-1}) = \dots$$

$$P_{t+1} = TP_t = T(TP_{t-1}) = \dots = T^t P_1$$

- Stationary distribution P^* after a mixing time t^* so long as matrix elements are probabilities (columns sum up to 1)

Wed, Oct 22, 2008 COMP 131, Lecture 12 21

Smoothing



- Divide evidence between $e_{1:k}$ and $e_{k+1:t}$
 $P(X_k, e_{1:t}) = P(X_k|e_{1:k}, e_{k+1:t})$
 $= \alpha P(X_k|e_{1:k})P(e_{k+1:t}|X_k, e_{1:k})$ Bayes rule
 $= \alpha P(X_k|e_{1:k})P(e_{k+1:t}|X_k)$ conditional independence
 $= \alpha f_{1:k} b_{k+1:t}$

- BACKWARD message $b_{k+1:t}$ computed by backward recursion

$$P(e_{k+1:t}|X_k) = \sum_{x_{k+1}} P(e_{k+1:t}|X_k, x_{k+1})P(x_{k+1}|X_k) \quad \text{condition on } X_{k+1}$$

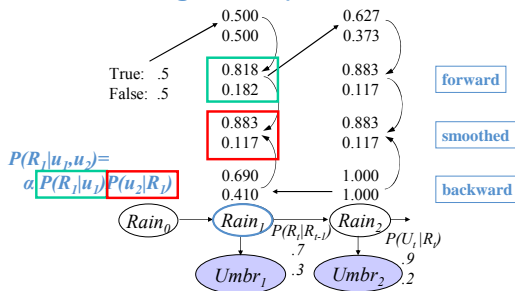
$$= \sum_{x_{k+1}} P(e_{k+1:t}|x_{k+1})P(x_{k+1}|X_k) \quad \text{cond. independence}$$

$$= \sum_{x_{k+1}} P(e_{k+1}, e_{k+2:t}|x_{k+1})P(x_{k+1}|X_k)$$

$$= \sum_{x_{k+1}} P(e_{k+1}|x_{k+1})P(e_{k+2:t}|x_{k+1})P(x_{k+1}|X_k)$$

Wed, Oct 22, 2008 COMP 131, Lecture 12 22

Smoothing example



Forward-backward algorithm: cache forward filtering results $O(t)$ time; $O(t|f)$ space

Wed, Oct 22, 2008 COMP 131, Lecture 12 23

Most likely explanation

- Most likely sequence \neq sequence of most likely states!
- Most likely path to each x_{t+1}
 $=$ most likely path to some x_t plus one more step
 $\max_{x_1 \dots x_t} P(x_1 \dots x_t, x_{t+1}|e_{1:t+1}) =$
 $= P(e_{1:t+1}|x_{t+1}) \max_{x_t} (P(x_{t+1}|x_t) \max_{x_1 \dots x_{t-1}} P(x_1 \dots x_{t-1}, x_t|e_{1:t}))$

- Identical to filtering, except $f_{1:t}$ replaced by

$$m_{1:t} = \max_{x_1 \dots x_{t-1}} P(x_1, \dots, x_{t-1}, X_t|e_{1:t})$$

$m_{1:t}(i)$ gives the probability of the most likely path to state i .

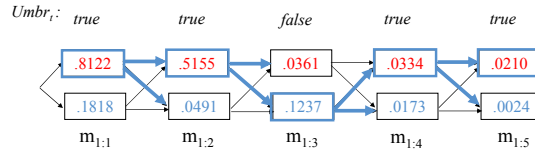
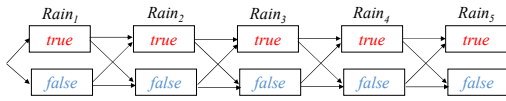
Update has sum replaced by max, giving the **Viterbi algorithm**:

$$m_{1:t+1} = P(e_{1:t+1}|X_{t+1}) \max_{x_t} (P(X_{t+1}|x_t) m_{1:t})$$

Wed, Oct 22, 2008 COMP 131, Lecture 12 24

Viterbi example

$$m_{1:t+1} = P(e_{t+1}|X_{1:t}) \max_{x_t} (P(X_{t+1}|x_t)m_{1:t})$$



Wed, Oct 22, 2008

COMP 131, Lecture 12

25

Summary

- Approximate inference less complex but not exact
- Sampling methods: stochastic simulation of the process
- Temporal info can be encoded as a time-slice process (variables at a point in time), which should be **stationary** and have the **Markov property**
- Inference in temporal BNs includes **filtering, prediction, smoothing, and most likely sequence**

Wed, Oct 22, 2008

COMP 131, Lecture 12

26