Informatics 2D: Reasoning and Agents

Alex Lascarides

School of informatics

The University of Edinburgh

Lecture 25b: Approximate inference in BNs:
Monte Carlo sampling methods

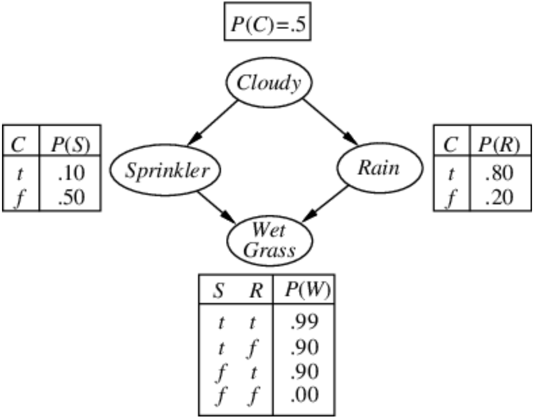
Where are we?

Last time...

- Approximate inference in BNs: Direct sampling
- Rejection sampling for queries $\mathbf{P}(X|e)$
- Very wasteful!
- Today: **Likelihood weighting and MCMC**

Reminder

- $[Cloudy, Sprinkler, Rain, WetGrass]$:



C	P(S)
t	.10
f	.50

C	P(R)
t	.80
f	.20

S	R	P(W)
t	t	.99
t	f	.90
f	t	.90
f	f	.00

- Query $\mathbf{P}(X|\mathbf{e})$;
  where $\mathbf{Y}$ are the non-query and non-evidence variables.

Likelihood weighting

- A **direct sampling** method that avoids inefficiency of rejection sampling,
  by *generating only samples consistent with evidence*
- Fixes the values for evidence variables $\mathbf{E}$ and samples only the remaining variables X and $\mathbf{Y}$
- Since not all events are equally probable, each event has to be weighted by its **likelihood** that it accords to the evidence
- Likelihood is measured by product of conditional probabilities for each evidence variable, given its parents

Likelihood weighting

- Consider query $\mathbf{P}(Rain|Sprinkler = true, WetGrass = true)$ in our example; initially set weight $w = 1$, then event is generated:
  - Sample from $\mathbf{P}(Cloudy) = \langle 0.5, 0.5 \rangle$, suppose this returns *true*
  - *Sprinkler* is evidence variable with value *true*, we set

    $$w \leftarrow w \times P(Sprinkler = true|Cloudy = true) = 0.1$$
  - Sample from $\mathbf{P}(Rain|Cloudy = true) = \langle 0.8, 0.2 \rangle$, suppose this returns *true*
  - *WetGrass* is evidence variable with value *true*, we set

    $$w \leftarrow w \times P(WetGrass = true|Sprinkler = true, Rain = true) = 0.099$$
- Sample returned=$[true, true, true, true]$ with weight 0.099 tallied under $Rain = true$

Likelihood weighting – why it works

- $S(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{l} P(z_i | parents(Z_i))$
- S's sample values for each Z_i is influenced by the evidence among Z_i's ancestors
- But S pays no attention when sampling Z_i's value to evidence from Z_i's non-ancestors; so it's not sampling from the true posterior probability distribution!
- But the likelihood weight w makes up for the difference between the actual and desired sampling distributions:

$$w(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{m} P(e_i | parents(E_i))$$

Likelihood weighting – why it works

- Since two products cover all the variables in the network, we can write

$$P(\mathbf{z},\mathbf{e}) = \underbrace{\prod_{i=1}^{l} P(z_i|parents(Z_i))}_{S(\mathbf{z},\mathbf{e})} \underbrace{\prod_{i=1}^{m} P(e_i|parents(E_i))}_{w(\mathbf{z},\mathbf{e})}$$

- With this, it is easy to derive that likelihood weighting is consistent (tutorial exercise)
- Problem: most samples will have very small weights as the number of evidence variables increases

The Markov chain Monte Carlo (MCMC) algorithm

- MCMC algorithm: create an event from a previous event, rather than generate all events from scratch
- Helpful to think of the BN as having a **current state** specifying a value for each variable
- Consecutive state is generated by sampling a value for one of the non-evidence variables X_i conditioned on the current values of variables in the Markov blanket of X_i
- Recall that Markov blanket consists of parents, children, and children's parents
- Algorithm randomly wanders around state space flipping one variable at a time and keeping evidence variables fixed

The MCMC algorithm

- Consider query $\mathbf{P}(Rain|Sprinkler = true, WetGrass = true)$ once more
- *Sprinkler* and *WetGrass* (evidence variables) are fixed to their observed values, hidden variables *Cloudy* and *Rain* are initialised randomly (e.g. *true* and *false*)
- Initial state is $[true, true, false, true]$
- Execute repeatedly:
    - Sample *Cloudy* given values of Markov blanket, i.e. sample from $\mathbf{P}(Cloudy|Sprinkler = true, Rain = false)$
    - Suppose result is *false*, new state is $[false, true, false, true]$
    - Sample *Rain* given values of Markov blanket, i.e. sample from $\mathbf{P}(Rain|Sprinkler = true, Cloudy = false, WetGrass = true)$
    - Suppose we obtain $Rain = true$, new state $[false, true, true, true]$

The MCMC algorithm – why it works

- Each state is a sample, contributes to estimate of query variable *Rain* (count samples to compute estimate as before)
- Basic idea of proof that MCMC is consistent:

  The sampling process settles into a "dynamic equilibrium" in which the long-term fraction of time spent in each state is exactly proportional to its posterior probability
- MCMC is a very powerful method used for all kinds of things involving probabilities

Summary

- Approximate inference in BNs
- rejection sampling (last time)
- Likelihood working and why it works
- MCMC algorithm and why it works
- Next time: **Time and Uncertainty I**