CS598JHM: **Advanced NLP** (Spring '10)

Sampling (Koller/Friedman '09, Ch.12)

Julia Hockenmaier
juliahmr@illinois.edu
3324 Siebel Center

http://www.cs.uiuc.edu/class/sp10/cs598jhm

Sampling methods

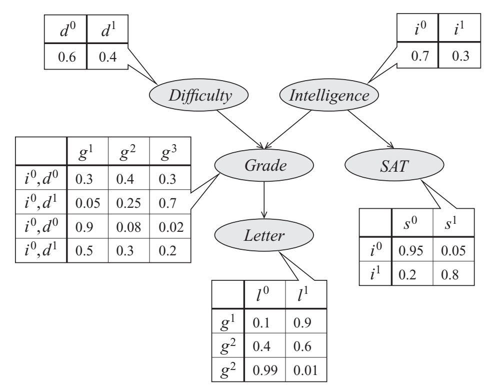
Task: Compute the expectation *f(x)* relative to *P(x)*

Approximate this through sampling:
Draw a finite number of samples from *P(x)*

Also known as particle-based approximate inference

Forward sampling

The *Student* network: writing letters of recommendation



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^0, d^0	0.9	0.08	0.02
i^0, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^2	0.99	0.01

Forward sampling in a Bayesian Netowrk

1. Sort the nodes $X_1 \dots X_n$ topologically
   (i.e. such that parents precede their descendants)

2. For $i=1 \dots n$:
   2.1. Let $\boldsymbol{u}_i$ be the current assignment to the parents of X_i
   2.2. Sample x_i from $P(x_i \mid \boldsymbol{u}_i)$

3. Return $(x_1,\dots,x_n)$

Sampling from a conditional distribution $P(\,y \mid e)$

Rejection Sampling
Sample from $P(x)$ and reject when $E \mathrel{!=} e$

Problem: $P(e)$ may be very low.
Now we require $P(e)^{-1}$ more samples

Likelihood weighting

Task: sample $P(\boldsymbol{y}|\boldsymbol{e})$ given multiple observations $e_1 \dots e_n$

We can use forward sampling, but need to take the probability $\mathrm{P}(e_i|\dots)$ into account.

Likelihood weighting:
- Weight each sample by $w = \prod_i \mathrm{P}(e_i|\dots)$
- Estimate conditional probability $P(\boldsymbol{y}|\boldsymbol{e})$
  as a weighted average of samples

Likelihood-weighted sampling in a Bayesian Network

1. Sort the nodes $X_1 \dots X_n$ topologically
   (i.e. such that parents precede their descendants)
2. Initialize $w = 1$
3. For $i=1 \dots n$:
   3.1. Let $\boldsymbol{u}_i$ be the current assignment to the parents of X_i
   3.2. If $x_i \notin \boldsymbol{e}$: sample x_i from $P(x_i \mid \boldsymbol{u}_i)$
   3.3. If $x_i \in \boldsymbol{e}$: 1) set x_i to e_i. 2) multiply w by $P(e_i \mid \boldsymbol{u}_i)$

4. Return $(x_1,\dots,x_n)$, w

Importance sampling

Likelihood weighted sampling is a special case of **importance sampling**

- We cannot always sample efficiently from $P(\mathrm{x})$
- But we may be able to **evaluate** $P(\mathrm{x})$ efficiently
- And we may be able to sample efficiently from some **proposal distribution** $Q(\mathrm{x})$
- If $Q(\mathrm{x}) \neq 0$ whenever $P(\mathrm{x}) \neq 0$, we can compute $E_{P(x)}[f(x)]$

$$\begin{aligned} E_{P(x)}[f(x)] &= E_{Q(x)}\left[f(x)\frac{P(x)}{Q(x)}\right] \\ &= \sum_x Q(x) f(x) \frac{P(x)}{Q(x)} \\ &= \sum_x f(x)P(x) \end{aligned}$$

Normalized importance sampling

Instead of $P(x)$, we often know only some **unnormalized** probability $P'(x)$ with $P(x) = P'(x)/Z$

We may want to sample from $P(x \mid e)$, but only have $P(x,e)$

Define a weight $w(x) = P'(x)/Q(x)$

Now we can compute $\mathrm{E}_{\mathrm{Q}}[w(x)]$…..

$$\begin{aligned} E_{Q(x)}[w(x)] &= \sum_x Q(x)\frac{P'(x)}{Q(x)} \\ &= \sum_x P'(x) = Z \end{aligned}$$

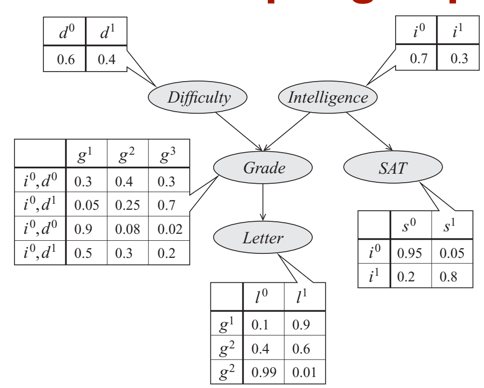
…. and hence estimate Z

Putting things together…:

Normalized/Weighted importance sampling

$$\begin{aligned} E_{P(x)}[f(x)] &= \sum_x f(x)P(x) \\ &= \sum_x Q(x) f(x) \frac{P(x)}{Q(x)} \\ &= \frac{1}{Z}\sum_x Q(x) f(x) \frac{P'(x)}{Q(x)} \\ &= \frac{1}{Z} E_{Q(x)}[f(x)w(x)] \\ &= \frac{E_{Q(x)}[f(x)w(x)]}{E_{Q(x)}[w(x)]} \end{aligned}$$

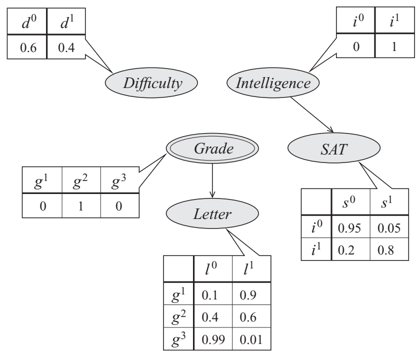
Importance sampling in practice



Sample from $P(D,I,S,L \mid G=g2)$
What is a good proposal distribution Q?

Constructing Q

- The proposal distribution sets all (conditioning) variables in Z to their known value.
- It also decouples all variables in Z from their parents



Limitations of likelihood weighting

- Evidence nodes affect sampling only for their descendants
- The effect of the evidence on other nodes is only captured by the weight
- When evidence is mostly at the leaf nodes, we effectively sample from the prior distribution (which can be very different from the posterior)
- Markov Chain Monte Carlo sampling methods generate a sequence of samples which may start out as the prior, but will approximate the posterior