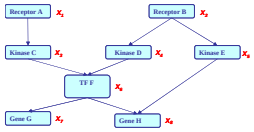
School of Computer Science
Carnegie Mellon

Approximate Inference: Monte Carlo Inference

Probabilistic Graphical Models (10-708)

Lecture 18, Nov 19, 2007



Eric Xing

Reading: J-Chap. 1, KF-Chap. 11

Monte Carlo methods

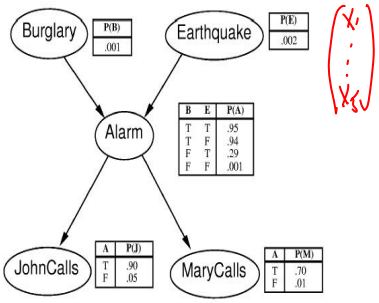
- Draw random samples from the desired distribution
- Yield a stochastic representation of a complex distribution
  - marginals and other expections can be approximated using sample-based averages

$$E[f(x)] = \frac{1}{N}\sum_{t=1}^{N} f(x^{(t)})$$

- **Asymptotically** exact and easy to apply to arbitrary models
- Challenges:
  - how to draw samples from a given dist. (not all distributions can be trivially sampled)?
  - how to make better use of the samples (not all sample are useful, or eqally useful, see an example later)?
  - how to know we've sampled enough?

Example: naive sampling

- Construct samples according to probabilities given in a BN.



Alarm example: (Choose the right sampling sequence)
1) Sampling:P(B)=<0.001, 0.999> suppose it is false, B0. Same for E0. P(A|B0, E0)=<0.001, 0.999> suppose it is false...
2) Frequency counting: In the samples right,
P(J|A0)=P(J,A0)/P(A0)=<1/9, 8/9>.

E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E1	B0	A1	M1	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0

Example: naive sampling

- Construct samples according to probabilities given in a BN.

Alarm example: (Choose the right sampling sequence)

3) what if we want to compute P(J|A1) ?
we have only one sample ...
P(J|A1)=P(J,A1)/P(A1)=<0, 1>.

4) what if we want to compute P(J|B1) ?
No such sample available!
P(J|A1)=P(J,B1)/P(B1) can not be defined.

For a model with hundreds or more variables, rare events will be very hard to garner evough samples even after a long time or sampling ...

E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E1	B0	A1	M1	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0

Monte Carlo methods (cond.)

- Direct Sampling
  - We have seen it.
  - Very difficult to populate a high-dimensional state space
- Rejection Sampling
  - Create samples like direct sampling, only count samples which is consistent with given evidences.
- Likelihood weighting, ...
  - Sample variables and calculate evidence weight. Only create the samples which support the evidences.
- Markov chain Monte Carlo (MCMC)
  - Metropolis-Hasting
  - Gibbs

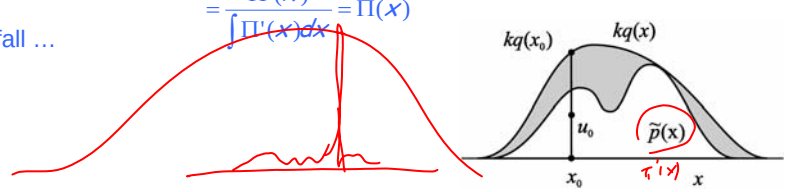
Rejection sampling

- Suppose we wish to sample from dist. $\Pi(X)=\Pi'(X)/Z$.
  - $\Pi(X)$ is difficult to sample, but $\Pi'(X)$ is easy to evaluate
  - Sample from a simpler dist $Q(X)$
  - Rejection sampling

    (1) $x^* \sim Q(X)$, (2) accept x^* w.p. $\Pi'(x^*)/kQ(x^*)$
  - Correctness:

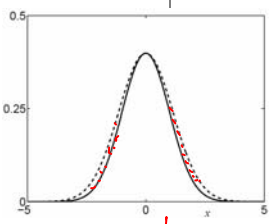
$$p(x)=\frac{[\Pi'(x)/kQ(x)]Q(x)}{\int[\Pi'(x)/kQ(x)]Q(x)dx}=\frac{\Pi'(x)}{\int\Pi'(x)dx}=\Pi(x)$$

  - Pitfall …

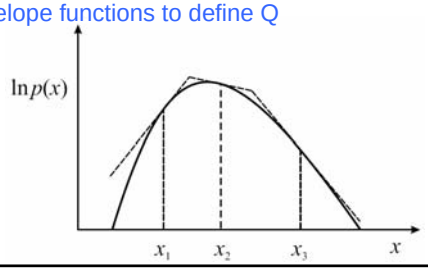


Rejection sampling

- Pitfall:
  - Using $Q=\mathcal{N}(\mu,\sigma_q^2 I)$ to sample $P=\mathcal{N}(\mu,\sigma_p^2 I)$
  - If σ_q exceeds σ_p by 1%, and dimensional=1000,
  - The optimal acceptance rate $k=(\sigma_q/\sigma_p)^d \approx 1/20{,}000$
  - Big waste of samples!

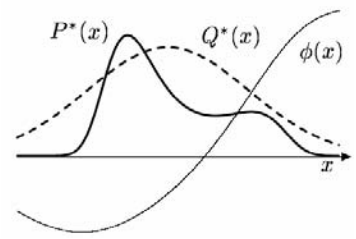


- Adaptive rejection sampling
  - Using envelope functions to define Q



Unnormalized importance sampling

- Suppose sampling from $P(\cdot)$ is hard.
- Suppose we can sample from a "simpler" proposal distribution $Q(\cdot)$ instead.
- If Q dominates P (i.e., $Q(x) > 0$ whenever $P(x) > 0$), we can sample from Q and reweight:



$$\langle f(X)\rangle = \int f(x)P(x)dx$$
$$= \int f(x)\frac{P(x)}{Q(x)}Q(x)dx$$
$$\approx \frac{1}{M}\sum_m f(x^m)\frac{P(x^m)}{Q(x^m)} \quad \text{where } x^m \sim Q(X)$$
$$= \frac{1}{M}\sum_m f(x^m)w^m$$

Normalized importance sampling

- Suppose we can only evaluate $P'(x) = \alpha P(x)$ (e.g. for an MRF).
- We can get around the nasty normalization constant α as follows:
  - Let $r(X) = \frac{P'(x)}{Q(x)} \quad \Rightarrow \quad \langle r(X) \rangle_Q = \int \frac{P'(x)}{Q(x)} Q(x) dx = \int P'(x) dx = \alpha$
- Now

$$\langle f(X) \rangle_P = \int f(x) P(x) dx = \frac{1}{\alpha} \int f(x) \frac{P'(x)}{Q(x)} Q(x) dx$$

$$= \frac{\int f(x) r(x) Q(x) dx}{\int r(x) Q(x) dx}$$

$$\approx \frac{\sum_m f(x^m) r^m}{\sum_m r^m} \quad \text{where } x^m \sim Q(X)$$

$$= \sum_m f(x^m) w^m \quad \text{where } w^m = \frac{r^m}{\sum_m r^m}$$

Normalized vs unnormalized importance sampling

- Unormalized importance sampling is unbiased:

$$E_Q[f(X)w(X)] = \int f(x) w(x) Q(x) dx = \int f(x) \frac{P(x)}{Q(x)} Q(x) dx = \int f(x) P(x) dx = E_P(f(x))$$

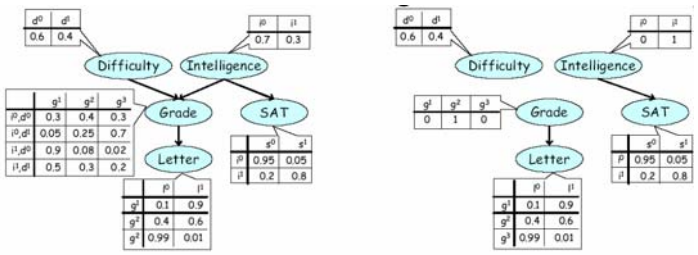
- Normalized importance sampling is biased, eg for M = 1:

$$E_Q\left[\frac{f(x^1)w(x^1)}{w(x^1)}\right] = E_Q(f(x^1)) \neq E_P[f(x)]$$

- However, the variance of the normalized importance sampler is usually lower in practice.
- Also, it is common that we can evaluate P'(x) but not P(x), e.g. P(x|e) = P'(x, e)/P(e) for Bayes net, or P(x) = P'(x)/Z for MRF.

Likelihood weighting

- We now apply normalized importance sampling to a Bayes net.
- The proposal Q is gotten from the mutilated BN where we clamp evidence nodes, and cut their incoming arcs. Call this P_M.



- The unnormalized posterior is P'(x) = P(x, e).
- So for $f(X_i) = \delta(X_i = x_i)$, we get $\hat{P}(X_i = x_i \mid e) = \frac{\sum_m w_m \delta(x_i^m = x_i)}{\sum_m w_m}$ where $w_m = P'(x^m, e) / P_M(x^m)$.

Likelihood weighting algorithm

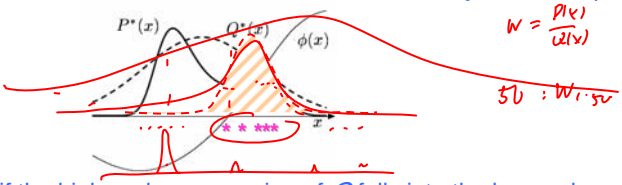
$[x_{1:n}, w]$ = function LW(CPDs, G, E)
let $X_1, \ldots, X_n$ be a topological ordering of G
$w = 1$
$x = (0, \ldots, 0)$
for $i = 1 : n$
  let $u_i = x(Pa_i)$
  if $X_i \notin E$
  then sample x_i from $P(X_i|u_i)$
  else
    $x_i = e(X_i)$
    $w = w * P(x_i|u_i)$

Efficiency of likelihood weighting

- The efficiency of importance sampling depends on how close the proposal Q is to the target P.
- Suppose all the evidence is at the roots. Then Q = P(X|e), and all samples have weight 1.
- Suppose all the evidence is at the leaves. Then Q is the prior, so many samples might get small weight if the evidence is unlikely.
- We can use arc reversal to make some of the evidence nodes be roots instead of leaves, but the resulting network can be much more densely connected.

Weighted resampling

- Problem of importance sampling: depends on how well Q matches P
  - If P(x)f(x) is strongly varying and has a significant proportion of its mass concentrated in a small region, r_m will be dominated by a few samples



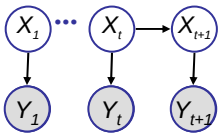
  - Note that if the high-prob mass region of Q falls into the low-prob mass region of P, the variance of $r^m = P(x^m)/Q(x^m)$ can be small even if the samples come from low-prob region of P and potentially erroneous .
- Solution
  - Use heavy tail Q.
  - Weighted resampling

$$w^m = \frac{P(x^m)/Q(x^m)}{\sum_l P(x^l)/Q(x^l)} = \frac{r^m}{\sum_m r^m}$$

Weighted resampling

- Sampling importance resampling (SIR):
  1. Draw N samples from Q: $X_1 \dots X_N$
  2. Constructing weights: $w_1 \dots w_N$, $w^m = \frac{P(x^m)/Q(x^m)}{\sum_l P(x^l)/Q(x^l)} = \frac{r^m}{\sum_m r^m}$
  3. Sub-sample x from $\{X_1 \dots X_N\}$ w.p. $(w_1 \dots w_N)$

- Particular Filtering
  - A special weighted resampler
  - Yield samples from posterior $p(X_t | Y_{1:t})$



Sketch of Particle Filters

- The starting point

$$p(X_t | \mathbf{Y}_{1:t}) = p(X_t | Y_t, \mathbf{Y}_{1:t-1}) = \frac{p(X_t | \mathbf{Y}_{1:t-1}) p(Y_t | X_t)}{\int p(X_t | \mathbf{Y}_{1:t-1}) p(Y_t | X_t) dX_t}$$

  - Thus $p(X_t | Y_{1:t})$ is represented by

$$\left\{ X_t^m \sim p(X_t | \mathbf{Y}_{1:t-1}),\ w_t^m = \frac{p(Y_t | X_t^m)}{\sum_{m=1}^{M} p(Y_t | X_t^m)} \right\}$$

- A sequential weighted resampler
  - Time update

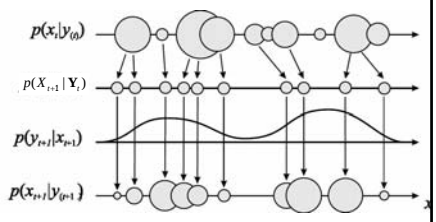
$$p(X_{t+1} | \mathbf{Y}_{1:t}) = \int p(X_{t+1} | X_t) p(X_t | \mathbf{Y}_{1:t}) dX_t$$

$$= \sum_m w_t^m p(X_{t+1} | X_t^m) \quad \text{(sample from a mixture model)}$$

  - Measurement update

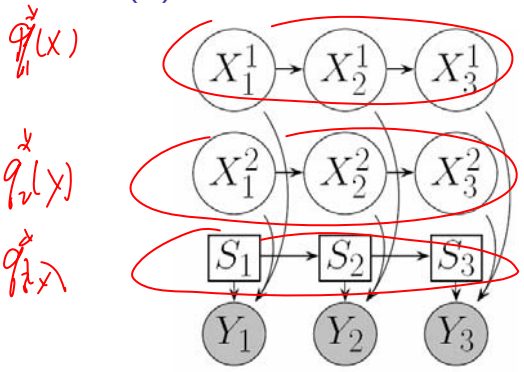
$$p(X_{t+1} | \mathbf{Y}_{1:t+1}) = \frac{p(X_{t+1} | \mathbf{Y}_{1:t}) p(Y_{t+1} | X_{t+1})}{\int p(X_{t+1} | \mathbf{Y}_{1:t}) p(Y_{t+1} | X_{t+1}) dX_{t+1}}$$

$$\Rightarrow \left\{ X_{t+1}^m \sim p(X_{t+1} | \mathbf{Y}_{1:t}),\ w_{t+1}^m = \frac{p(Y_{t+1} | X_{t+1}^m)}{\sum_{m=1}^{M} p(Y_{t+1} | X_{t+1}^m)} \right\} \quad \text{(reweight)}$$



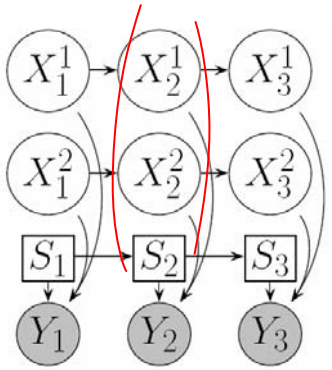
PF for switching SSM

- Recall that the belief state has O(2t) Gaussian modes



PF for switching SSM

- Key idea: if you knew the discrete states, you can apply the right Kalman filter at each time step.
- So for each old particle *m*, sample $S_t^m \sim P(S_t \mid S_{t-1}^m)$ from the prior, apply the KF (usomg parameters for S_t^m) to the old belief state $(\hat{x}_{t-1|t-1}^m, P_{t-1|t-1}^m)$ to get an approximation to $P(X_t \mid y_{1:t}, s_{1:t}^m)$
- Useful for online tracking, fault diagnosis, etc.



Rao-Blackwellised sampling

- Sampling in high dimensional spaces causes high variance in the estimate.
- RB idea: sample some variables X_p, and conditional on that, compute expected value of rest X_d analytically:

$$
\begin{aligned}
E_{p(X|e)}\left[f(X)\right] &= \int p(x_p, x_d \mid e) f(x_p, x_d) dx_p dx_d \\
&= \int_{x_p} p(x_p \mid e) \left(\int_{x_d} p(x_d \mid x_p, e) f(x_p, x_d) dx_d \right) dx_p \\
&= \int_{x_p} p(x_p \mid e) E_{p(X_d|x_p,e)}\left[f(x_p, X_d)\right] dx_p \\
&= \frac{1}{M} \sum_m E_{p(X_d|x_p^m,e)}\left[f(x_p^m, X_d)\right], \qquad x_p^m \sim p(x_p \mid e)
\end{aligned}
$$

- This has lower variance, because of the identity:

$$
\mathrm{var}\left[\tau(X_p, X_d)\right] = \mathrm{var}\left[E\left[\tau(X_p, X_d) \mid X_p\right]\right] + E\left[\mathrm{var}\left[\tau(X_p, X_d) \mid X_p\right]\right]
$$

Rao-Blackwellised sampling

- Sampling in high dimensional spaces causes high variance in the estimate.
- RB idea: sample some variables X_p, and conditional on that, compute expected value of rest X_d analytically:

$$
\begin{aligned}
E_{p(X|e)}[f(X)] &= \int p(x_p, x_d \mid e) f(x_p, x_d) dx_p dx_d \\
&= \int_{x_p} p(x_p \mid e) \left(\int_{x_d} p(x_d \mid x_p, e) f(x_p, x_d) dx_d \right) dx_p \\
&= \int_{x_p} p(x_p \mid e) E_{p(X_d|x_p,e)}[f(x_p, X_d)] dx_p \\
&= \frac{1}{M} \sum_m E_{p(X_d|x_p^m,e)}[f(x_p^m, X_d)], \qquad x_p^m \sim p(x_p \mid e)
\end{aligned}
$$

- This has lower variance, because of the identity:

$$\mathrm{var}[\tau(X_p, X_d)] = \mathrm{var}[E[\tau(X_p, X_d) \mid X_p]] + E[\mathrm{var}[\tau(X_p, X_d) \mid X_p]]$$

- Hence $\mathrm{var}[E[\tau(X_p, X_d) \mid X_p]] \le \mathrm{var}[\tau(X_p, X_d)]$, so $\tau(X_p, X_d) = E[f(X_p, X_d) \mid X_p]$ is a lower variance estimator.

Summary: Monte Carlo Methods

- Direct Sampling
  - Very difficult to populate a high-dimensional state space
- Rejection Sampling
  - Create samples like direct sampling, only count samples which is consistent with given evidences.
- Likelihood weighting, ...
  - Sample variables and calculate evidence weight. Only create the samples which support the evidences.
- Markov chain Monte Carlo (MCMC)
  - Metropolis-Hasting
  - Gibbs