Lecture 19:

Monte Carlo Methods (Koller & Friedman ch 9)

Kevin Murphy
23 November 2004

Monte Carlo sampling

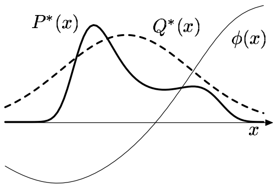
- Goal: estimate $Ef(X)$ where $X \sim P(\cdot)$.
- If $f(X) = \delta(X_i = x_i)$, then $Ef(X) = P(X_i = x_i)$.
- Draw M samples $x^m \sim P$, then compute

$$Ef(X) \approx \frac{1}{M} \sum_{m=1}^{M} f(x^m)$$

- Key problem: drawing samples from $P()$.
- For a Bayes net, we can easily sample from the prior $P(X)$ following topological order.
- To sample from posterior, $P(X|e)$, we can sample from $P(X)$ and reject samples inconsistent with the evidence, but this is inefficient.

Unnormalized importance sampling

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



$$\begin{aligned} E_P f(X) &= \sum_x P(x) f(x) \\ &= \sum_x Q(x) f(x) \frac{P(x)}{Q(x)} \\ &\approx \sum_{m=1}^{M} f(x^m) \frac{P(x^m)}{Q(x^m)} \\ &= \sum_{m=1}^{M} f(x^m) w^m \end{aligned}$$

Normalized importance sampling

- Suppose we can only evaluate $P'(x) = \alpha P(x)$ (eg for an MRF).
- $w(x) = \frac{P'(x)}{Q(x)}$, so $E_Q w(X) = \sum_x Q(x) \frac{P'(x)}{Q(x)} = \sum_x P'(x) = \alpha$.
- We have to slightly modify the estimator:

$$\begin{aligned} E_P f(X) &= \sum_x P(x) f(x) = \sum_x Q(x) f(x) \frac{P(x)}{Q(x)} \\ &= \frac{1}{\alpha} \sum_x Q(x) f(x) \frac{P'(x)}{Q(x)} \\ &= \frac{1}{\alpha} E_Q f(X) w(X) \\ &= \frac{E_Q f(X) w(X)}{E_Q w(X)} \\ &= \frac{\sum_m w_m f(x^m)}{\sum_m w_m} \end{aligned}$$

Normalized vs unnormalized importance sampling

- Unormalized importance sampling is unbiased:

$$E_Q f(X) w(X) = E_Q f(X) P(X)/Q(X) = E_P f(X)$$

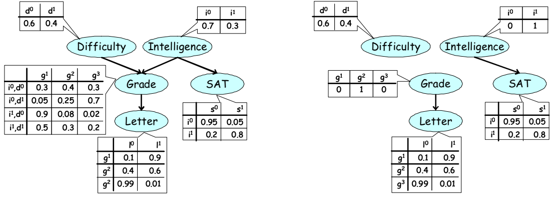
- Normalized importance sampling is biased, eg for $M = 1$:

$$E_Q \frac{f(x^1) w(x^1)}{w(x^1)} = E_Q f(x^1)$$

- 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.

Likelhood 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.



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^1,d^0	0.9	0.08	0.02
i^1,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

d^0	d^1
0.6	0.4

i^0	i^1
0	1

g^1	g^2	g^3
0	1	0

	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^3	0.99	0.01

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

Likelhood 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.

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:

$$\begin{aligned} E_{P(X|e)} f(X) &= \sum_{x_p, x_d} P(x_p, x_d|e) f(x_p, x_d) \\ &= \sum_{x_p} P(x_p|e) \sum_{x_d} P(x_d|x_p, e) f(x_p, x_d) \\ &= \sum_{x_p} P(x_p|e) E_{P(X_d|x_p,e)} f(x_p, X_d) \end{aligned}$$

- This has lower variance, because of the identity: $\text{Var}[\tau(X_p, X_d)] = \text{Var}[E(\tau(X_d, X_p)|X_p)] + E[\text{Var}(\tau(X_d, X_p)|X_p)]$
- Hence $\text{Var}[E(\tau(X_d, X_p)|X_p)] \le \text{Var}[\tau(X_d, X_p)]$, so $\tau'(X_d, X_p) = E(\tau(X_d, X_p)|X_p)$ is a lower variance estimator.

Rao-Blackwellised importance sampling

- Each sample is a setting x_p^m and a distribution over X_d conditioned on x_p^m and the evidence e.
- The simplest case is to sample from an upwardly closed subset of nodes in the BN (roots and some of their children).
- The estimate is

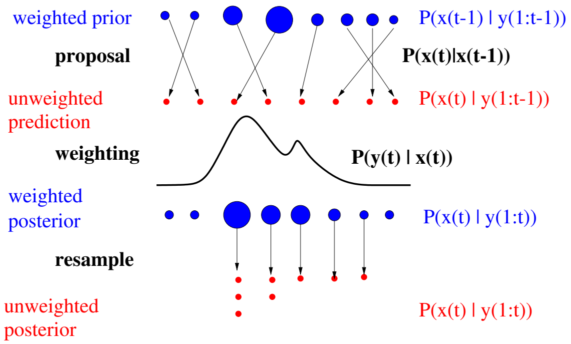
$$E_{P(X|e)} f(X) \approx \frac{\sum_m w^m E_{P(X_d|x_p^m,e)} f(x_p^m, X_d)}{\sum_m w^m}$$

where $w(x_p) = \frac{P(x_p,e_p)}{Q(x_p)} P(e_d|x_p, e_p)$.

- The term $\frac{P(x_p,e_p)}{Q(x_p)}$ is computed using likelihood weighting on X_p.
- The second term $P(e_d|x_p, e_p)$ is computed using exact inference.

Particle filtering (sequential Monte Carlo)

- PF is sequential importance sampling with resampling (SISR).
- Goal is to estimate $P(x_{1:t}|y_{1:t})$ recursively (online) for a state-space model for which Kalman filter/ HMM filter is inapplicable.



Sequential importance sampling

- Suppose the target density is $P(x_{1:t}|y_{1:t})$ and the proposal is $q(x_{1:t}|y_{1:t})$, so $w_t^i \propto P(x_{1:t}^i|y_{1:t})/Q(x_{1:t}^i|y_{1:t})$.
- The probability of a sample path can be computed recursively using Bayes' rule:

$$\begin{aligned} w_t^i &\propto \frac{P(y_t|x_t^i)P(x_t^i|x_{t-1}^i)P(x_{1:t-1}^i|y_{1:t-1})}{Q(x_t^i|x_{1:t-1}^i, y_{1:t})Q(x_{1:t-1}^i|y_{1:t-1})} \\ &= \frac{P(y_t|x_t^i)P(x_t^i|x_{t-1}^i)}{Q(x_t^i|x_{1:t-1}^i, y_{1:t})} w_{t-1}^i \\ &= \hat{w}_t^i w_{t-1}^i \end{aligned}$$

- For online problems, we typically use $Q(x_t|x_{1:t-1}^i, y_{1:t}) = Q(x_t|x_{t-1}^i, y_{1:t})$ so we don't have to store the entire history. Hence

$$\hat{w}_t^i = \frac{P(y_t|x_t^i)P(x_t^i|x_{t-1}^i)}{Q(x_t^i|x_{t-1}^i, y_{1:t})}$$

Sequential Importance sampling with resampling

- As time increases, one sample path will turn out to be exponentially more likely than any other, so all the weights except one go to 0.
- This is called sample impoverishment.
- Whenever the effective number of samples $N_{eff} = 1/\sum_i (w_t^i)^2$ drops below a threshold, we resample with replacement.
- The resampled weights are set to $1/N$, since the past weights are reflected in the empirical frequency.
- There are various ways to do the resampling in $O(N)$ time.

Pseudo code for particle filter

```
function $[\{x_t^i, w_t^i\}_{i=1}^N] = \mathrm{PF}(\{x_{t-1}^i, w_{t-1}^i\}_{i=1}^N, y_t)$
for $i = 1 : N$
    Sample $x_t^i \sim Q(\cdot | x_{t-1}^i, y_{1:t})$
    Compute $\hat{w}_t^i = \frac{P(y_t|x_t^i)P(x_t^i|x_{t-1}^i)}{Q(x_t^i|x_{t-1}^i, y_{1:t})}$
    $w_t^i = \hat{w}_t^i \times w_{t-1}^i$
Compute $w_t = \sum_{i=1}^N w_t^i$
Normalize $w_t^i := w_t^i / w_t$
Compute $N_{eff} = 1/\sum_i (w_t^i)^2$.
if $N_{eff} <$ threshold
  $\pi =$ resample$(\{w_t^i\}_{i=1}^N)$
  $x_t^\cdot = x_t^\pi$
  $w_t^i = 1/N$
```

Simplest proposal distribution for PF

- The simplest proposal is to sample from the prior $Q(x_t|x_{t-1}^i, y_{1:t}) = P(X_t|x_{t-1}^i)$.
- This is like likelihood weighting, where the evidence is at the leaves.
- In vision, this is called the condensation algorithm.
- Recall that the incremental weight is

$$\hat{w}_t^i = \frac{P(y_t|x_t^i)P(x_t^i|x_{t-1}^i)}{Q(x_t^i|x_{t-1}^i, y_{1:t})}$$

- So for condensation, $\hat{w}_t^i = P(y_t|x_t^i)$.

Optimal proposal distribution for PF

- It is better to look at the evidence before proposing:

$$q(x_t|x_{t-1}^i, y_t) = P(x_t|x_{t-1}^i, y_t) = \frac{P(y_t|x_t)P(x_t|x_{t-1}^i)}{\int dx_t P(y_t|x_t)P(x_t|x_{t-1}^i)}$$

- This is optimal in the sense that in minimizes the variance of the weights.
- In this case, the incremental weight is the denominator $\hat{w}_t^i = P(y_t|x_{t-1}^i)$.
- This requires integrating out x_t, which may be hard.

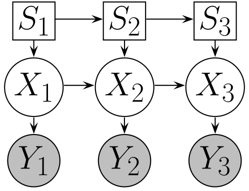
Unscented particle filtering

- Often it is too hard to compute the optimal proposal $P(X_t|x^i_{t-1}, y_{1:t})$ exactly.
- But sometimes we can approximate this.
- Consider a nonlinear system with Gaussian process noise and linear-Gaussian observations:

$$\begin{aligned} P(X_t|x^i_{t-1}) &= \mathcal{N}(X_t; f_t(x^i_{t-1}), Q_t) \\ P(Y_t|X_t) &= \mathcal{N}(y_t; C_t X_t, R_t) \end{aligned}$$

- Then we can compute $Q(X_t|x^i_{t-1}, y_{1:t})$ using an EKF/UKF (with a delta function prior on x^i_{t-1}), and sample from this.

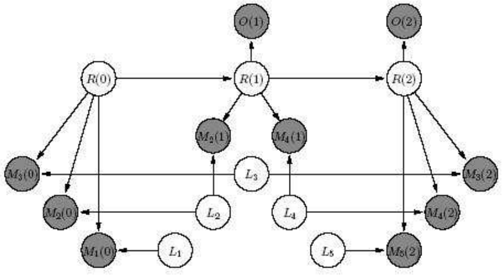
RBPF for switching LDS



- Recall that the belief state has $O(2^t)$ Gaussian modes:
- 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 $S^m_t \sim P(S_t|s^m_{t-1})$ from the prior, apply the KF (using parameters for S^m_t) to the old belief state $(\hat{x}^m_{t-1|t-1}, P^m_{t-1|t-1})$ to get an approximation to $P(X_t|y_{1:t}, s^m_{1:t})$.
- Useful for fault diagnosis.

RBPF for SLAM ("FastSLAM")

- Key idea: if you always know the robot's location, the posterior over landmarks factorizes, so KF takes $O(N_L)$ time instead of $O(N_L^2)$.
- So sample $R_{1:t}$, and for each particle/ trajectory, run a Kalman filter.



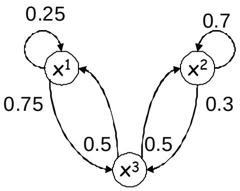
Markov chain Monte Carlo (MCMC)

- Importance sampling does not scale well to high dimensions.
- Rao-Blackwellisation not always possible.
- MCMC is an alternative.
- Construct a Markov chain whose stationary distribution is the target density $\pi = P(X|e)$.
- Run for T samples (burn-in time) until the chain converges/ mixes/ reaches stationary distribution.
- Then collect M (correlated) samples $x^m \sim \pi$.
- Key issues:
  - Designing proposals so that the chain mixes rapidly.
  - Diagnosing convergence.

Markov chains: definitions

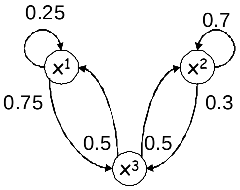
- $\pi(x)$ is a stationary distribution if $\pi(x') = \sum_x \pi(x) T(x \to x')$, i.e., π is a left eigenvector of the transition matrix $\pi^T = \pi^T A$.

$$\begin{pmatrix}0.2 & 0.5 & 0.3\end{pmatrix} = \begin{pmatrix}0.2 & 0.5 & 0.3\end{pmatrix} \begin{pmatrix} 0.25 & 0 & 0.75 \\ 0 & 0.7 & 0.3 \\ 0.5 & 0.5 & 0 \end{pmatrix}$$



Markov chains: definitions

- An MC is *periodic* if it cycles through the state space without converging.
- An MC is *reducible* if the stationary distribution reached depends on the starting state (different one-way traps).
- An MC is *ergodic (regular)* if you can get from state x to x' in a finite number of steps.
- Thm: a finite state MC has a unique stationary distribution iff it is regular.



Gibbs sampling

- Gibbs sampling is an MCMC algorithm that is especially appropriate for inference in graphical models.
- The transition matrix updates each node one at a time: $T((u_i, x_i) \to (u_i, x_i')) = P(x_i'|u_i)$.
- This is efficient since $P(x_i|u_i) = P'(x_i, u_i)/P'(u_i)$ only depends on the values in X_i's Markov blanket

```
function $[\{x_{1:n}^m\}_{m=1}^M]$ = Gibbs(Potentials, $T$)
sample $x^0$ from $P(X|e)$
for $t = 1 : T$
    $x^t = x^{t-1}$
    for each $X_i$
        $u_i$ = values of $MB(X_i)$ in $x^t$
        Sample $x_i^t \sim P(\cdot|u_i)$
```

Gibbs sampling

- Gibbs sampling can fail if there are deterministic constraints, eg $X \to Z \leftarrow Y$ where Z is xor. Suppose we observe $Z = 1$. The posterior has 2 modes: $P(X = 1, Y = 0|Z = 1)$ and $P(X = 0, Y = 1|Z = 1)$. However, if we start in mode 1, $P(X|y = 0, z = 1)$ leaves $X = 1$, so we can't move (Reducible Markov chain).
- If all states have non-zero probability, the MC is guaranteed to be regular.
- Sampling blocks of variables at a time can help improve mixing.

Metropolis Hastings

- Gibbs sampling is only applicable when we can sample one variable given all the others.
- MH is more general.
- It constructs a reversible MC.
- Defn: An MC is *reversible* if $\exists! \pi$ st. $\pi(x)T(x \to x') = \pi(x')T(x' \to x)$ (detailed balance).
- Thm: if the MC is regular and satisfies detailed balance, then π is the unique stationary distribution.
- MH will construct T.

Metropolis Hastings

- MH proposes moves according to $Q(x \to x')$ and accepts them with probability $A(x \to x')$.
- The induced transition matirx is

$$\begin{aligned} T(x \to x') &= Q(x \to x')A(x \to x') \text{if } x \neq x' \\ T(x \to x) &= Q(x \to x) \sum_{x' \neq x} Q(x \to x')(1 - A(x \to x')) \end{aligned}$$

- Detailed balance means

$$\pi(x)Q(x \to x')A(x \to x') = \pi(x')Q(x' \to x)A(x' \to x)$$

- Hence the acceptance ratio is

$$A(x \to x') = \min\left(1, \frac{\pi(x')Q(x' \to x)}{\pi(x)Q(x \to x')}\right)$$

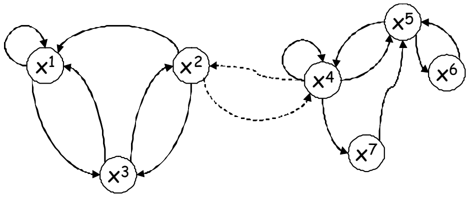
Gibbs sampling is a special case of Metropolis Hastings

- Suppose we use the proposal $Q((u_i, x_i) \to (u_i, x_i')) = P(x_i'|u_i)$
- Then the acceptance ratio is

$$\begin{aligned} A((u_i, x_i) \to (u_i, x_i')) &= \min(1, \frac{P(x_i'|u_i)Q((u_i, x_i') \to (u_i, x_i))}{P(x_i|u_i)Q((u_i, x_i) \to (u_i, x_i'))}) \\ &= \min(1, \frac{P(x_i'|u_i)P(x_i|u_i)}{P(x_i|u_i)P(x_i'|u_i)}) \\ &= \min(1, 1) \end{aligned}$$

Mixing time

- The ϵ mixing time T_ϵ is the minimal number of steps (from any starting distribution) until $D_{var}(P^{(T)}, \pi) \leq \epsilon$, where D_{var} is variational distance.
- Chains with low bandwidth (conductance) regions of space take a long time to mix.
- This arises for GMs with deterministic or highly skewed potentials.



Convergence diagnosis (CODA)

- How can we tell when burn-in is over?
- Run multiple chains from different starting conditions, wait until they start "behaving similarly".
- Various heuristics have been proposed.
- See the CODA package in R.