Probabilistic Graphical Models

10-708

Towards Complex Graphical Models and Approximate Inference

Eric Xing

Lecture 16, Nov 7, 2005

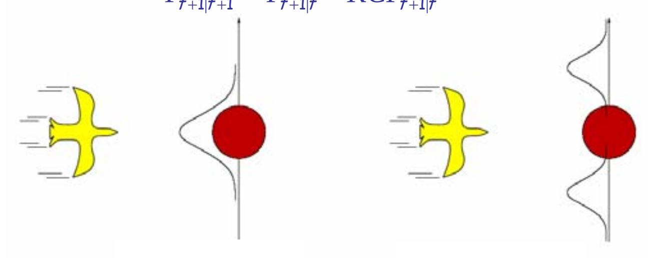
Reading: MJ-Chap. 21

The need for multimodal belief states in dynamic models

- An LDS defines only unimodal belief states

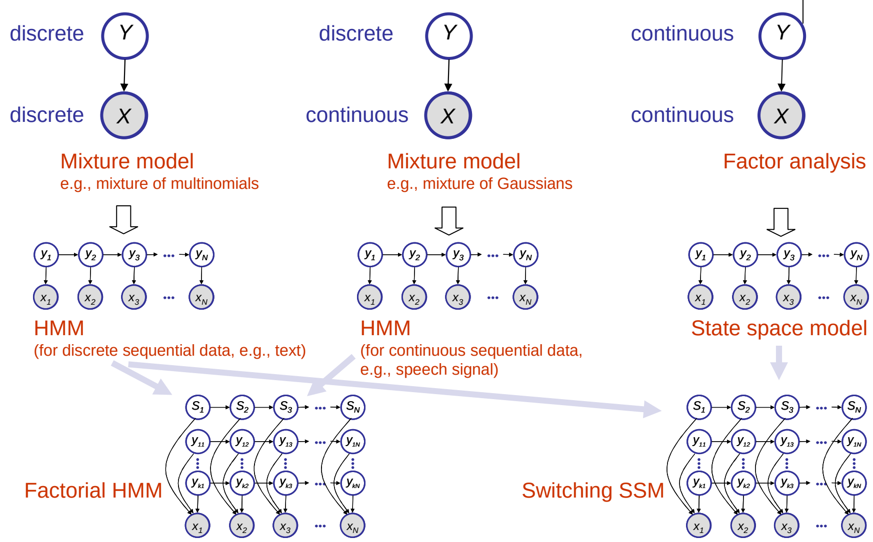
$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + K_{t+1}(\mathrm{y}_{t+1} - \mathrm{C}\hat{\mathrm{x}}_{t+1|t})$$

$$P_{t+1|t+1} = P_{t+1|t} - KCP_{t+1|t}$$



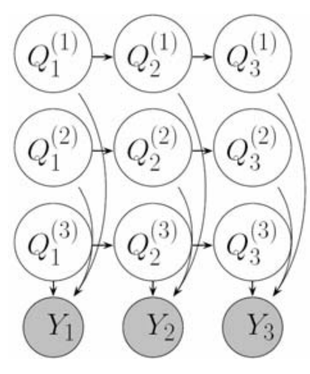
- (a) A Kalman filter will predict the location of the bird using a single Gaussian centered on the obstacle.
- (b) A more realistic model allows for the bird's evasive action, predicting that it will fly to one side or the other.

A road map to more complex dynamic models

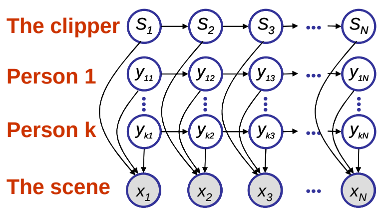


Factorial HMM

- The belief state at each time is $X_t = \{Q_t^{(1)}, \ldots, Q_t^{(k)}\}$ and in the most general case has a state space $O(d^k)$ for k d-nary chains
- The common observed child Y_t couples all the parents (explaining away).
- But the parameterization cost for fHMM is $O(kd^2)$ for k chain-specific transition models $p(Q_t^{(i)} \mid Q_{t-1}^{(i)})$ rather than $O(d^{2k})$ for $p(X_t \mid X_{t-1})$



Special case: switching HMM

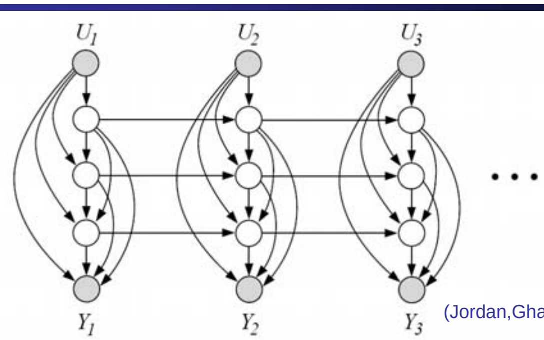


Multi-View Face Tracking with Factorial and Switching HMM

Peng Wang , Qiang Ji
Department of Electrical, Computer and System Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180

- Different chains have different state space and different semantics
- The exact calculation is intractable and we must use approximate inference methods

Hidden Markov decision trees

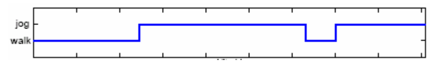


(Jordan,Ghahramani,&Saul,197)

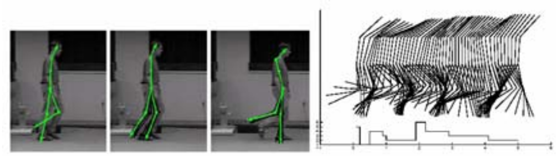
- A combination of decision trees with factorial HMMs
- This gives a "command structure" to the factorial representation
- Appropriate for multi-resolution time series
- Again, the exact calculation is intractable and we must use approximate inference methods

Switching LDS

- Possible world:
  - multiple motion state:



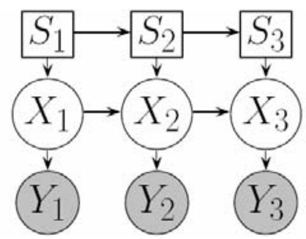
- Task:
  - Trajectory prediction



- Model:
  - Combination of HMM and LDS

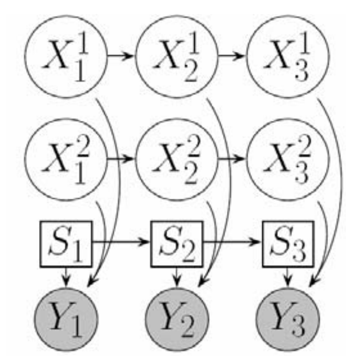
$$p(X_t = x_t \mid X_{t-1} = x_{t-1}, S_t = i) = \mathcal{N}(x_t; A_i x_{t-1}, Q_i)$$
$$p(Y_t = y_t \mid X_t = x_t) = \mathcal{N}(t_t; C x_t, R)$$
$$p(S_t = j \mid S_{t-1} = i) = M(i, j)$$

  - Belief state has O(k^t) Gaussian modes:



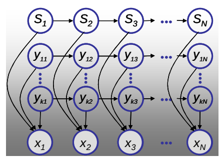
Data association (correspondence problem)

- Optimal belief state has O(k^t) modes.
- Common to use nearest neighbor approximation.
- For each time slice, can enforce that at most one source causes each observation
- Correspondence problem also arises in shape matching and stereo vision.

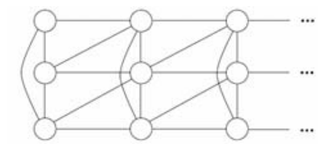


Triangulating fHMM

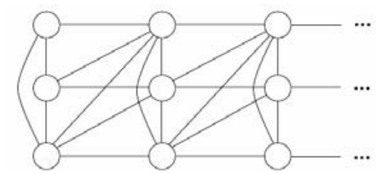
- Is the following triangulation correct?



?⇒



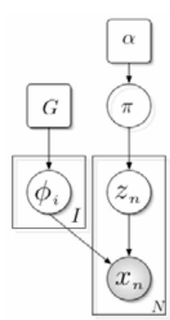
- Here is a triangulation



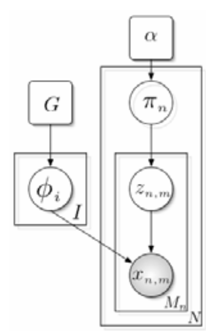
- We have created cliques of size *k*+1, and there are O(*kT*) of them. The junction tree algorithm is not efficient for factorial HMMs.

Mixed Membership Model (M³)

- Mixture versus admixture



⇒

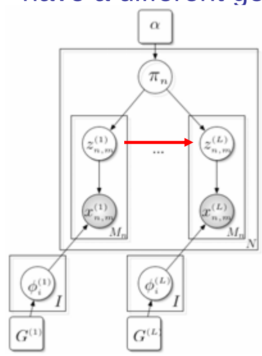


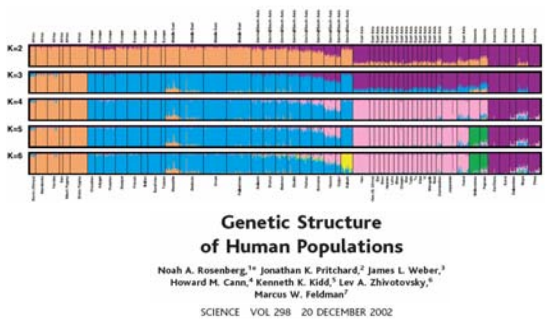
A Bayesian mixture model

A Bayesian admixture model:
Mixed membership model

Population admixture: M³ in genetics

- The genetic materials of each modern individual are inherited from multiple ancestral populations, each DNA locus may have a different generic origin …

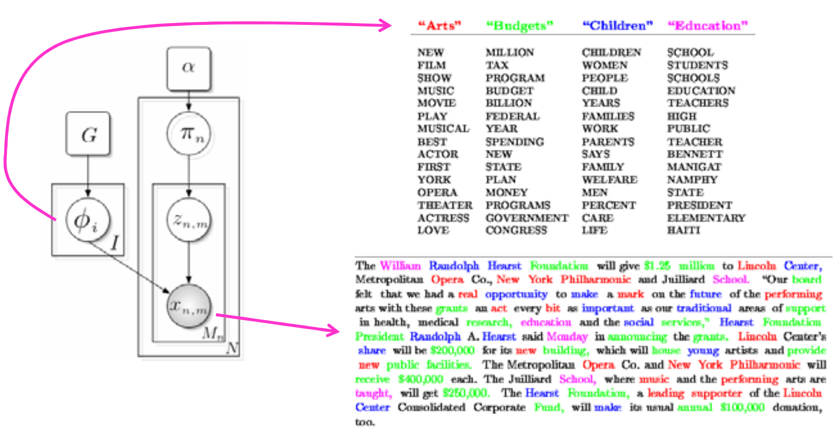




- Ancestral labels may have (e.g., Markovian) dependencies

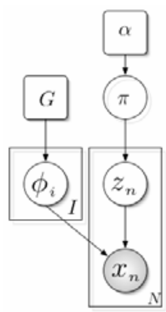
Latent Dirichlet Allocation: M³ in text mining

- A document is a bag of words each generated from a randomly selected topic

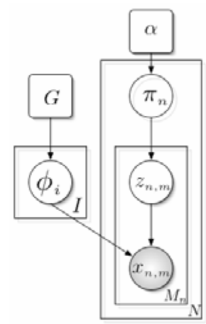


Inference in Mixed Membership Models

- Mixture versus admixture



$\Rightarrow$



$$p(D) = \sum_{\{z_{n,m}\}} \int \cdots \int \left(\prod_n \left(\prod_m p(x_{n,m} \mid \phi_{z_n}) p(z_{n,m} \mid \pi_n) \right) p(\pi_n \mid \alpha) \right) p(\phi \mid G) d\pi_1 \cdots d\pi_N d\phi$$

  - Inference is very hard in M^3, all hidden variables are coupled and not factorizable!

$$p(\pi_n \mid D) \sim \sum_{\{z_{n,m}\}} \int \left(\prod_n \left(\prod_m p(x_{n,m} \mid \phi_{z_n}) p(z_{n,m} \mid \pi_n) \right) p(\pi_n \mid \alpha) \right) p(\phi \mid G) d\pi_{-i} d\phi$$

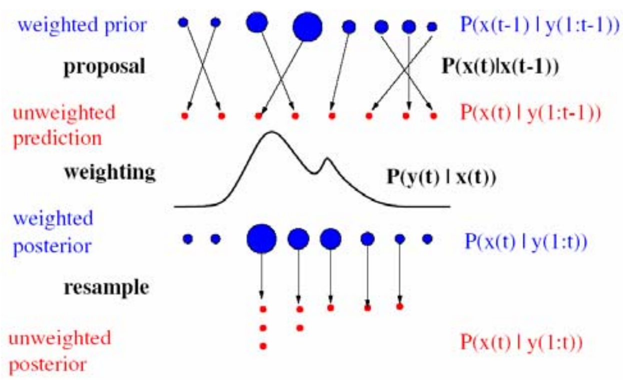
Approaches to inference

- Exact inference algorithms
  - The elimination algorithm
  - The junction tree algorithms

- Approximate inference techniques
  - Monte Carlo algorithms:
    - Stochastic simulation / sampling methods
    - Markov chain Monte Carlo methods
  - Variational algorithms:
    - Belief propagation
    - Assumed density filtering
    - Variational inference

Example: Particle filtering (sequential Monte Carlo)

- Represent belief state as weighted set of samples (non-parametric).
- Can handle nonlinear transition/emission and multi-modality.
- Easy to implement.
- Only works well in small dimensions.



Example: Structured Variational approximation

- Finds an optimal $q^*()$ in a tractable family to approximate the original joint $p()$

$$q^*() \in \arg\min_{q \in \mathcal{F}} \mathrm{F}(q \,\|\, p)$$

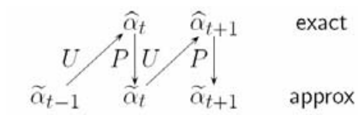
- There can be many different choices of $\mathcal{F}$ and F().

fHMM　　Mean field approx.　　Structured variational approx.

Example: Assumed density filtering (ADF)

- ADF forces the belief state to live in some restricted family $\mathcal{F}$, e.g., product of histograms, Gaussian.
- Given a prior $\tilde{\alpha}_{t-1} \in \mathcal{F}$, do one step of exact Bayesian updating to get $\hat{\alpha}_t \notin \mathcal{F}$. Then do a projection step to find the closest approximation in the family:

$$\tilde{\alpha}_t \in \arg\min_{q \in \mathcal{F}} \mathrm{KL}(\hat{\alpha}_t \,\|\, q)$$



- The Boyen-Koller (BK) algorithm is ADF applied to a DBN
  - e.g., let $\mathcal{F}$ be a product of (singleton) marginals:
- This is also a variational method, and the updating step can still be intractable

Monte Carlo methods

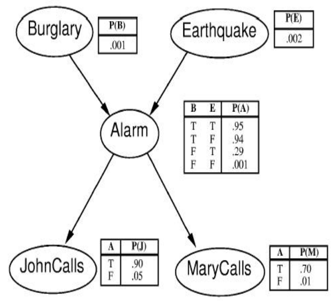
- Draw random samples from the desired distribution
- Yield a stochastic representation of a complex distribution
  - marginals and other expections can be approximated using sample-based averages

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

- **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

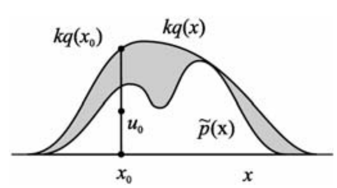
- Suppose we wish to sample from dist. $\Pi(X)=\Pi'(X)/Z$.
  - $\Pi(X)$ is difficult to sample, but $\Pi'(X)$ is easy to evaluate
  - Sample from a simpler dist $Q(X)$
  - Rejection sampling

$$x^* \sim Q(X), \qquad \text{accept } x^* \text{ w.p. } \Pi'(x^*)/kQ(x^*)$$

  - Correctness:

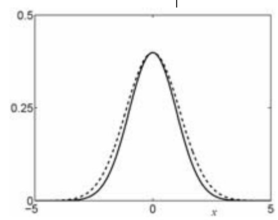
$$p(x) = \frac{[\Pi'(x)/kQ(x)]Q(x)}{\int [\Pi'(x)/kQ(x)]Q(x)dx} = \frac{\Pi'(x)}{\int \Pi'(x)dx} = \Pi(x)$$

  - Pitfall …

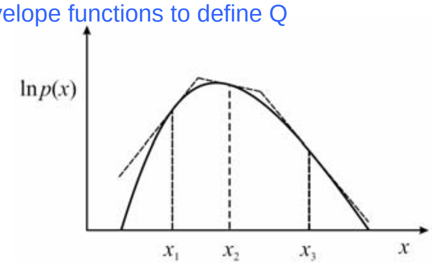


Rejection sampling

- Pitfall:
  - Using $Q=\mathcal{N}(\mu,\sigma_q I)$ to sample $P=\mathcal{N}(\mu,\sigma_p I)$
  - 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!

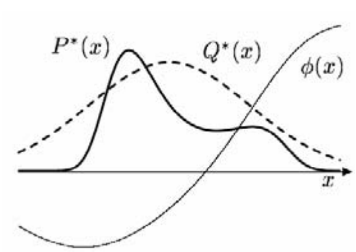


- Adaptive rejection sampling
  - Using envelope functions to define Q



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:



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

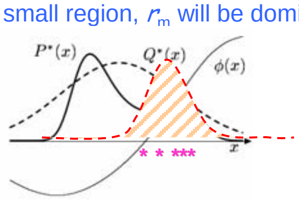
Normalized importance sampling

- 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

$$
\begin{aligned}
\langle f(X) \rangle_P = \int f(x) P(x) dx &= \frac{1}{\alpha} \int f(x) \frac{P'(x)}{Q(x)} Q(x) dx \\
&= \frac{\int f(x) r(x) Q(x) dx}{\int r(x) Q(x) dx} \\
&\approx \frac{\sum_m f(x^m) r^m}{\sum_m r^m} \quad \text{where } x^m \sim Q(X) \\
&= \sum_m f(x^m) w^m \quad \text{where } w^m = \frac{r^m}{\sum_m r^m}
\end{aligned}
$$

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

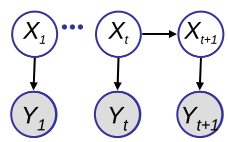


  - Note that if the high-prob mass region of Q falls into the low-prob mass region of P, the variance of $r^m = P(x^m)/Q(x^m)$ can be small even if the samples come from low-prob region of P and potentially erroneous .
- Solution
  - Use heavy tail Q.
  - Weighted resampling $\quad w^m = \frac{P(x^m)/Q(x^m)}{\sum_l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

Weighted resampling

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

- 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

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

- A sequential weighted resampler
  - Time update

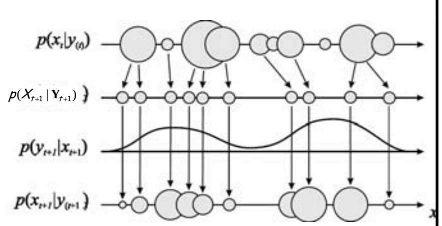
$$p(X_{t+1} | \mathbf{Y}_{1:t}) = \int p(X_{t+1} | X_t) p(X_t | \mathbf{Y}_{1:t}) dX_t$$

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

  - Measurement update

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

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



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:

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

- This has lower variance, because of the identity:

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

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

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 $= P(X|e)$.
- Run for T samples (burn-in time) until the chain converges/mixes/reaches stationary distribution.
- Then collect M (correlated) samples x_m .
- Key issues:
  - Designing proposals so that the chain mixes rapidly.
  - Diagnosing convergence.