The need for multimodal belief states in dynamic models

- An LDS defines only unimodal belief states
  \[ \hat{x}_{t+1|t} = \hat{x}_{t+1} + K_{t+1}(y_{t+1} - C\hat{x}_{t+1}) \]
  \[ P_{t+1|t} = P_{t+1} - KCP_{t+1} \]

- (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

- Discrete variables:
  - Mixture model (e.g., mixture of multinomials)
  - HMM (for discrete sequential data, e.g., text)
  - Factorial HMM

- Continuous variables:
  - Mixture model (e.g., mixture of Gaussians)
  - HMM (for continuous sequential data, e.g., speech signal)
  - State space model
  - Factor analysis
  - Switching SSM

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 $k$-nary chains.

- The common observed child $Y_t$ couples all the parents (explaining away).

- But the parameterization cost for fHMM is $O(kd^k)$ for $k$ chain-specific transition models $p(Q_t^{(i)} | Q_{t-1}^{(i-1)})$ rather than $O(d^k)$ for $p(X_t | X_{t-1})$.
Special case: switching HMM

- 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

- 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
    \[ p(X_t = x_t \mid X_{t-1} = x_{t-1}, S_{t-1} = i) = N(x_t; A x_{t-1}, Q) \]
    \[ p(Y_t = y_t \mid X_t = x_t) = N(x_t; C x_t, R) \]
    \[ p(S_t = j \mid S_{t-1} = i) = M(i, j) \]
  - 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 genetic 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

\[ p(D) = \sum_{\{z_{n,m}\}} \prod_{n} p(x_{n,m} | \phi_{z_{n,m}}) p(z_{n,m} | \pi_{z_{n,m}}) p(\pi_{z_{n,m}} | \alpha) \]
\[ p(\pi_z | D) = \sum_{\{z_{n,m}\}} \prod_{n} p(x_{n,m} | \phi_{z_{n,m}}) p(z_{n,m} | \pi_{z_{n,m}}) p(\pi_{z_{n,m}} | \alpha) \]

Inference is very hard in M3, all hidden variables are coupled and not factorizable!

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 $q^*(\cdot)$ in a tractable family to approximate the original joint $p()$

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

- There can be many different choices of $\mathcal{F}$ and $F()$.
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 $\tilde{\alpha}_t \in \mathcal{F}$. Then do a projection step to find the closest approximation in the family:
  $$\tilde{\alpha}_t \in \arg \min_{q \in \mathcal{F}} \text{KL}(\tilde{\alpha}_t \parallel q)$$
- 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 expectations can be approximated using sample-based averages
    $$E[f(x)] = \frac{1}{N} \sum_{i=1}^{N} f(x^{(i)})$$
- 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 equally 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, \( B_0 \). Same for \( E_0 \). \( P(A|B_0, E_0) = \{0.001, 0.999\} \) suppose it is false...
2) Frequency counting: In the samples right, \( P(J|A_0) = P(J,A_0)/P(A_0) = \{1/9, 8/9\} \).

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 \( P(J|A_1) \)? we have only one sample ... \( P(J|A_1) = P(J,A_1)/P(A_1) = \{0, 1\} \).
4) what if we want to compute \( P(J|B_1) \)? No such sample available! \( P(J|A_1) = P(J,B_1)/P(B_1) \) can not be defined.

For a model with hundreds or more variables, rare events will be very hard to garner enough samples even after a long time or sampling ...
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
    \[ x^* \sim Q(X), \quad \text{accept } x^* \text{ w.p. } \frac{\Pi'(x^*)}{kQ(x^*)} \]

- Correctness:
  \[
p(x) = \frac{[\Pi'(x)/kQ(x)]Q(x)}{\int [\Pi'(x)/kQ(x)]Q(x)dx} \quad = \quad \frac{\Pi'(x)}{\Pi'(x)dx} = \Pi(x)
\]

- Pitfall …
Rejection sampling

- Pitfall:
  - Using $Q = \mathcal{N}(\mu, \sigma_q)$ to sample $P = \mathcal{N}(\mu, \sigma_p)$
  - If $\sigma_q$ exceeds $\sigma_p$ 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 $R(\cdot)$ is hard.
- Suppose we can sample from a "simpler" proposal distribution $Q(\cdot)$ instead.
- If $Q$ dominates $P$ (i.e., $Q(x) > 0$ whenever $R(x) > 0$), we can sample from $Q$ and reweight:

\[
\langle f(X) \rangle = \int f(x)P(x)dx = \int f(x) \frac{P(x)}{Q(x)}Q(x)dx = \frac{1}{M} \sum_{n=1}^{M} f(x^n) \frac{P(x^n)}{Q(x^n)} \text{ where } x^n \sim Q(X) = \frac{1}{M} \sum_{n=1}^{M} f(x^n)w^n
\]
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 $\alpha$ as follows:
  - Let $r(X) = \frac{P'(x)}{Q(x)} \Rightarrow \langle r(X) \rangle_Q = \int \frac{P(x)}{Q(x)} Q(x) dx = \int P(x) dx = \alpha$
  - Now

$$\langle f(X) \rangle_p = \left[ f(x) \frac{P'(x)}{Q(x)} dx = \frac{1}{\alpha} \int f(x) \frac{P'(x)}{Q(x)} Q(x) dx \right]$$

$$= \left[ \frac{f(x) r(x) Q(x)}{\alpha} dx \right]$$

$$= \sum_m f(x_m) w_m$$

where $x_m \sim Q(x)$

- 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 $r_m = \frac{P(x_m)}{Q(x_m)}$ can be small even if the samples come from low-prob region of $P$ and potentially erroneous.

**Solution**

- Use heavy tail $Q$
- Weighted resampling $w_m = \frac{P(x_m)}{Q(x_m)/\sum r_m}$
Weighted resampling

- Sampling importance resampling (SIR):
  1. Draw $N$ samples from $Q$: $X_1^* \ldots X_N^*$
  2. Constructing weights: $w_1 \ldots w_N$, 
  $$w^n = \frac{p(x^n | y_t)}{\sum_{k=1}^{N} p(x^k | y_t)}$$
  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
  $$p(X_t | Y_{1:t}) = \frac{p(X_t | Y_{1:t-1})p(Y_t | X_t)}{\int p(X_t | Y_{1:t-1})p(Y_t | X_t)dX_t}$$
  - Thus $p(X_t | Y_{1:t})$ is represented by
    $$X_t^* \sim p(X_t | Y_{1:t}), \quad w_t^* = \frac{p(Y_t | X_t^*)}{\sum_{k=1}^{N} p(Y_t | X_k^*)}$$

- A sequential weighted resampler
  - Time update
    $$p(X_{t+1} | Y_{1:t+1}) = p(X_{t+1} | Y_t)\frac{p(Y_t | X_{t+1})}{p(Y_t | X_t)}$$
  - Measurement update
    $$p(X_{t+1} | Y_{1:t+1}) = \frac{p(X_{t+1} | Y_t)\frac{p(Y_t | X_{t+1})}{p(Y_t | X_t)}dX_{t+1}}{\sum_{k=1}^{N} w_k^n}$$
    $$(sample \ from \ a \ mixture \ model)$$
    $$(reweight)$$
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:

$$E_{\pi \mid \theta}(f(X)) = \int p(x_p \mid \theta) \left( \int p(x_d \mid x_p) f(x_p, x_d) \, dx_d \right) \, dx_p$$

$$= \int p(x_p \mid \theta) E_{\pi \mid \theta}(f(x_p, X_d) \mid x_p) \, dx_p$$

$$= \frac{1}{M} \sum_{m=1}^{M} E_{\pi \mid \theta}(f(x_p^m, X_d)) \quad x_p^m = p(x_p \mid \theta)$$

- This has lower variance, because of the identity:

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

- Hence $E[\tau(X_p, X_d) \mid X_p]$ 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 $P(X \mid \theta)$.
- 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.