Sequential Monte Carlo: Importance Sampling & Particles

\[ m_{12}(x_2) \quad m_{23}(x_3) \]
Standard Monte Carlo

\[
E[f] = \int f(x)p(x \mid y) \, dx \approx \frac{1}{L} \sum_{\ell=1}^{L} f(x^{(\ell)}) \quad x^{(\ell)} \sim p(x \mid y)
\]

- Suppose interested in some complex, global function of state:

- Can efficiently draw joint samples from posterior marginals:
  
  - Forward Message Passing: \( p(x_t \mid y), p(x_t, x_{t+1} \mid y) \)
  
  - Backwards Sampling:
    
    \[
    \begin{align*}
    x_T^{(\ell)} & \sim p(x_T \mid y) \\
    x_{T-1}^{(\ell)} & \sim p(x_{T-1} \mid x_T^{(\ell)}, y) \\
    x_{T-2}^{(\ell)} & \sim p(x_{T-2} \mid x_{T-1}^{(\ell)}, y)
    \end{align*}
    \]
Standard Monte Carlo

- Procedure only tractable for a limited class of models:
  - Discrete states: Sum-product belief propagation
  - Gaussian continuous states: Kalman filter

- Can efficiently draw joint samples from posterior marginals:
  - Forward Message Passing: \( p(x_t \mid y), p(x_t, x_{t+1} \mid y) \)
  - Backwards Sampling:
    \[
    x_T^{(\ell)} \sim p(x_T \mid y) \\
    x_{T-1}^{(\ell)} \sim p(x_{T-1} \mid x_T^{(\ell)}, y) \\
    x_{T-2}^{(\ell)} \sim p(x_{T-2} \mid x_{T-1}^{(\ell)}, y) \\
    (x_1^{(\ell)}, x_2^{(\ell)}, \ldots, x_T^{(\ell)}) \sim p(x \mid y)
    \]
Standard Importance Sampling

- Suppose interested in some complex, global function of state:

\[ \mathbb{E}[f] = \int f(x)p(x \mid y) \, dx \approx \sum_{\ell=1}^{L} w_{\ell} f(x^{(\ell)}) \quad w_{\ell} \propto \frac{p(x^{(\ell)} \mid y)}{q(x^{(\ell)} \mid y)} \quad x^{(\ell)} \sim q(x \mid y) \]

- Could use Markov structure to construct efficient proposal:

\[ q(x \mid y) = q(x_0) \prod_{t=1}^{T} q(x_t \mid x_{t-1}, y_t) \]

\[ q(x_t \mid x_{t-1}, y_t) \approx p(x_t \mid x_{t-1}, y) \]

Valid in principle, but small local errors give large global estimator variance.
Letting $p_{t−1,t}^r(x_t) \propto p(x_t \mid y_{t−1}^t)$

$m_{t−1,t}(x_t)p(y_t \mid x_t) \propto p(x_t \mid y_t) = q_t(x_t)$

**Inference (Product step of BP):**

$q_t(x_t) = \frac{1}{Z_t} m_{t−1,t}(x_t)p(y_t \mid x_t)$

**Prediction (Integral/Sum step of BP):**

$m_{t,t+1}(x_{t+1}) \propto \int_{x_t} p(x_{t+1} \mid x_t)q_t(x_t) \, dx_t$

$y_{t} = \{y_1, \ldots, y_t\}$

As discussed in Sec. 2.3.2, analytic evaluation of BP's message update integral is typically intractable for non-linear or non-Gaussian dynamic systems. For high-dimensional state spaces, like those arising in visual tracking and other tree-structured graphs, the belief factor propagation (BP) algorithm is based on a series of steps. In Markov chains and other tree-structured graphs, the belief factor factorizes as in eq. (3.1). As described in Sec. 2.3.2, particle filters approximate the forward BP messages passed between neighboring nodes. In such HMMs, these BP messages have an interesting probabilistic interpretation.

In this section, we suppose that the hidden states take value in a finite alphabet. For such HMMs, these BP messages have an interesting probabilistic interpretation. In the current observation's likelihood of an observation sequence is required. This approach is widely used in online tracking applications, where causal processing estimates of the state variables. For high-dimensional state spaces, like those arising in visual tracking and other tree-structured graphs, the belief factor propagation (BP) algorithm is based on a series of steps.
Particle-Based Density Estimates

Particle-Based Posterior State Estimates:
- Approximate density by set of (possibly weighted) samples
- Dynamically move samples to the most probable parts of space
- Do this in a way which minimizes bias

\[
m_{t-1,t}(x_t) \approx \sum_{\ell=1}^{L} w_{t-1,t}^{(\ell)} \delta(x_t, x_t^{(\ell)})
\]

\[
\sum_{\ell=1}^{L} w_{t-1,t}^{(\ell)} = 1
\]
Particle Filtering Algorithms

- Represent state estimates using a set of samples
- Propagate over time using sequential importance sampling with resampling

Sample-based density estimate

Weight by observation likelihood

Resample & propagate by dynamics
The Bootstrap Particle Filter

- Represent state estimates using a set of samples
- Propagate over time using sequential importance sampling with resampling
- Assume sample-based approximation of incoming message:
  \[ m_{t-1,t}(x_t) = p(x_t | y_{t-1}, \ldots, y_1) \approx \sum_{\ell=1}^{L} \frac{1}{L} \delta_{x_t^{(\ell)}}(x_t) \]
- Account for observation via importance weights:
  \[ p(x_t | y_t, y_{t-1}, \ldots, y_1) \approx \sum_{\ell=1}^{L} w_t^{(\ell)} \delta_{x_t^{(\ell)}}(x_t) \quad w_t^{(\ell)} \propto p(y_t | x_t^{(\ell)}) \]
- Sample from corresponding conditional distribution of next state:
  \[ m_{t,t+1}(x_{t+1}) \approx \sum_{m=1}^{L} \frac{1}{L} \delta_{x_{t+1}^{(m)}}(x_{t+1}) \quad x_{t+1}^{(m)} \sim \sum_{\ell=1}^{L} w_t^{(\ell)} p(x_{t+1} | x_t^{(\ell)}) \]
Markov Chain Monte Carlo Methods (MCMC)

Construct a biased random walk that explores the target distribution $P^\star(x)$. Markov steps, $x_t \sim T(x_t \leftarrow x_{t-1})$.

MCMC gives approximate, correlated samples from $P^\star(x)$. 
Markov Chain Monte Carlo (MCMC)

- At each time point, state $z^{(t)}$ is a configuration of all the variables in the model: parameters, hidden variables, etc.
- We design the transition distribution $q(z | z^{(t)})$ so that the chain is irreducible and ergodic, with a unique stationary distribution $p^*(z)$

$$p^*(z) = \int_{\mathcal{Z}} q(z | z') p^*(z') \, dz'$$

- For learning, the target equilibrium distribution is usually the posterior distribution given data $x$: $p^*(z) = p(z | x)$
- Popular recipes: Metropolis-Hastings and Gibbs samplers
Importance Sampling for Regression

- Model: Gaussian noise around some unknown straight line
- Propose from prior on lines, weight by data likelihood
Metropolis Algorithm for Regression

- Perturb parameters: $Q(\theta' ; \theta)$, e.g. $\mathcal{N}(\theta, \sigma^2)$
- Accept with probability $\min\left(1, \frac{\tilde{P}(\theta'|D)}{\tilde{P}(\theta|D)}\right)$
- Otherwise keep old parameters

Detail: Metropolis, as stated, requires $Q(\theta'; \theta) = Q(\theta; \theta')$

This subfigure from PRML, Bishop (2006)
Markov Chain Monte Carlo (MCMC)

Construct a biased random walk that explores target dist $P^*(x)$

Markov steps, $x_t \sim T(x_t \leftarrow x_{t-1})$

MCMC gives approximate, correlated samples from $P^*(x)$
Transition (Proposal) Distributions

Discrete example

\[ P^* = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} \quad T = \begin{pmatrix} 2/3 & 1/2 & 1/2 \\ 1/6 & 0 & 1/2 \\ 1/6 & 1/2 & 0 \end{pmatrix} \quad T_{ij} = T(x_i \leftarrow x_j) \]

\[ P^* \] is an invariant distribution of \( T \) because \( TP^* = P^* \), i.e.

\[
\sum_x T(x' \leftarrow x) P^*(x) = P^*(x')
\]

Also \( P^* \) is the equilibrium distribution of \( T \):

To machine precision: \( T^{100} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} = P^* \)

Ergodicity requires: \( T^K(x' \leftarrow x) > 0 \) for all \( x' : P^*(x') > 0 \), for some \( K \)
Detailed balance means $\rightarrow x \rightarrow x'$ and $\rightarrow x' \rightarrow x$ are equally probable:

$$T(x' \leftarrow x)P^*(x) = T(x \leftarrow x')P^*(x')$$

**Detailed balance implies the invariant condition:**

$$\sum_x T(x' \leftarrow x)P^*(x) = P^*(x')\sum_x T(x \leftarrow x')$$

Enforcing detailed balance is easy: it only involves isolated pairs.
If $T$ satisfies stationarity, we can define a reverse operator

\[
\tilde{T}(x \leftarrow x') \propto T(x' \leftarrow x) P^*(x) = \frac{T(x' \leftarrow x) P^*(x)}{\sum_x T(x' \leftarrow x) P^*(x)} = \frac{T(x' \leftarrow x) P^*(x)}{P^*(x')}
\]

**Generalized balance condition:**

\[
T(x' \leftarrow x) P^*(x) = \tilde{T}(x \leftarrow x') P^*(x')
\]

also implies the invariant condition and is necessary.

Operators satisfying detailed balance are their own reverse operator.