Rao-Blackwellized Estimators & Collapsed Samplers
Probabilistic Mixture Models

- Cluster weights: \( \pi \sim \text{Dir}(\alpha) \)
- Cluster params: \( \theta_k \sim H(\lambda) \) \( \theta_k = \{\mu_k, \Sigma_k\} \)
- Cluster assign: \( p(z_i \mid \pi) = \text{Cat}(z_i \mid \pi) \) \( z_i \in \{1, \ldots, K\} \)
- Observations: \( p(x_i \mid z_i, \mu, \Sigma) = \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i}) \)
A Standard Gibbs Sampler

- Given fixed mixture weights and mixture parameters, the cluster assignments are conditionally independent:

\[
p(z \mid x, \pi, \theta) = \prod_{i=1}^{N} p(z_i \mid x_i, \pi, \theta)
\]

\[
p(z_i = k \mid x_i, \pi, \theta) \propto \pi_k f(x_i \mid \theta_k)
\]

Sample from these categorical distributions, once per variable, in arbitrary order.

- Given fixed cluster assignments \(z\), all parameters are conditionally independent:

\[
p(\pi \mid x, z, \theta) = p(\pi \mid z) = \text{Dir}(\pi \mid N_1 + \alpha, \ldots, N_K + \alpha)
\]

\[
N_k = \sum_{i=1}^{N} \delta(z_i, k)
\]

\[
p(\theta \mid x, z, \pi) = \prod_{k=1}^{K} p(\theta_k \mid x, z) = \prod_{k=1}^{K} p(\theta_k \mid X_k)
\]

\[
X_k = \{x_i \mid z_i = k\}
\]

- Compared to the EM algorithm for finite mixture models:
  - Form same assignment distributions as in E-step, but then draw a single sample from each
  - Sample, rather than taking mode, of parameter distributions from M-step
Snapshots of Mixture Gibbs Sampler

Initialization A

Initialization B

2 Iterations 10 Iterations 50 Iterations
Rao-Blackwellized Estimation

- Basic Monte Carlo estimation for joint distribution of \( x, z \):
  \[
  (x^{(\ell)}, z^{(\ell)}) \sim p(x, z) \quad \ell = 1, 2, \ldots, L
  \]
  \[
  \mathbb{E}_p[f(x, z)] = \int_Z \int_{\mathcal{X}} f(x, z)p(x, z) \, dx \, dz \approx \frac{1}{L} \sum_{\ell=1}^{L} f(x^{(\ell)}, z^{(\ell)}) = \mathbb{E}_{\tilde{p}}[f(x, z)]
  \]

- But suppose that the conditional distribution \( p(x \mid z) \) is tractable:
  \[
  \mathbb{E}_p[f(x, z)] = \int_Z \int_{\mathcal{X}} f(x, z)p(x \mid z) \, p(z) \, dx \, dz
  \]
  \[
  = \int_Z \left[ \int_{\mathcal{X}} f(x, z)p(x \mid z) \, dx \right] p(z) \, dz
  \]
  \[
  \approx \frac{1}{L} \sum_{\ell=1}^{L} \int_{\mathcal{X}} f(x, z^{(\ell)})p(x \mid z^{(\ell)}) \, dx = \mathbb{E}_{\tilde{p}}[\mathbb{E}_p[f(x, z) \mid z]]
  \]

- This estimator is guaranteed to have lower variance!
A Collapsed Sampling Algorithm

\[ z_i \sim \text{Cat}(\pi) \]
\[ x_i \sim F(\theta_{z_i}) \]
\[ \theta_k \sim G(\beta) \]

Conjugate priors allow exact marginalization of parameters, to make an equivalent model with fewer variables.
Bayesian Learning of Probabilities

**Posterior Predictive Distribution:** For the next observation,

\[
p(z_{N+1} = k \mid z_1, \ldots, z_N) = \int \mu_k p(\mu \mid z_1, \ldots, z_N) \, d\mu
\]

\[
= \frac{N_k + \alpha_k}{N + \alpha_0} = \mathbb{E}[\mu_k \mid z_1, \ldots, z_N]
\]

**Dirichlet Prior Distribution:**

\[
p(\mu) = \text{Dir}(\mu \mid \alpha) \propto \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}
\]

**Dirichlet Posterior Distribution (Conjugate):**

\[
p(\mu \mid x) \propto \prod_{k=1}^{K} \mu_k^{N_k + \alpha_k - 1} \propto \text{Dir}(\mu \mid N_1 + \alpha_1, \ldots, N_K + \alpha_K)
\]
A Collapsed Gibbs Sampler

- **Collapsed mixture model representation:**
  
  \[
  p(z \mid x) \propto p(z)p(x \mid z)
  \]
  
  \[
  \propto \int p(z \mid \pi)p(\pi \mid \alpha) \, d\pi \int p(x \mid z, \theta)p(\theta \mid \lambda) \, d\theta
  \]

- **Apply standard Gibbs sampling updates:**

  \[
  p(z_i \mid z_{\backslash i}, x) \propto p(z_i \mid z_{\backslash i})p(x \mid z_i, z_{\backslash i})
  \]

  - **Conditional prior:**

    \[
    N_{k}^{\backslash i} = \sum_{j=1, j \neq i}^{N} \delta(z_j, k)
    \]

    \[
    p(z_i = k \mid z_{\backslash i}) = \frac{N_{k}^{\backslash i} + \alpha/K}{N - 1 + \alpha}
    \]

  - **Conditional likelihood:**

    \[
    X_{k}^{\backslash i} \triangleq \{ x_j \mid z_j = k, j \neq i \}\]

    \[
    p(x_i \mid z_i = k, z_{\backslash i}, x_{\backslash i}) = \int_{\Theta_k} p(x_i \mid \theta_k)p(\theta_k \mid X_{k}^{\backslash i}) \, d\theta_k
    \]

Conjugate analysis given “other” data assigned to this cluster
Gibbs: Representation & Mixing

Multiple Trials on 2D Mixture Data from Earlier Slide

Standard Gibbs: Alternatively sample assignments, parameters
Collapsed Gibbs: Marginalize parameters, sample assignments
Blocked Gibbs Samplers
Sum-Product for Blocked Tree Sampling

Global Directed Factorization:
- Choose some node as the root of the tree, order other nodes by depth
- Directed factorization from root to leaves:
  \[ p(x) = p(x_{\text{Root}}) \prod_s p(x_s \mid x_{\text{Pa}(s)}) \]

Bottom-Up Message Passing:
- Pass messages from leaves to root
- Compute marginal of root node:
  \[ m_{ts}(x_s) \propto \sum_{x_t} \psi_{st}(x_s, x_t) \psi_t(x_t) \prod_{u \in \Gamma(t) \setminus s} m_{ut}(x_t) \]
  \[ q_t(x_t) \propto \psi_t(x_t) \prod_{u \in \Gamma(t)} m_{ut}(x_t) \]

Top-Down Recursive Sampling:
- Sample root from marginal, then sample by depth given parent:
  \[ p(x_s \mid X_t = \hat{x}_t, t = \text{Pa}(s)) \propto \psi_{ts}(\hat{x}_t, x_s) \psi_s(x_s) \prod_{u \in \Gamma(s) \setminus t} m_{us}(x_s) \]
Example: Hidden Markov Model

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

    \[
    (x_1^{(\ell)}, x_2^{(\ell)}, \ldots, x_T^{(\ell)}) \sim p(x \mid y)
    \]

- Justification from Markov properties of HMM:

\[
p(x_t \mid x_{t+1}, y) = p(x_t \mid x_{t+1}, y_1, \ldots, y_t) \propto p(x_t \mid y_1, \ldots, y_t)p(x_{t+1} \mid x_t)
\]
Gibbs Sampling for Learning HMMs

- Given a fixed state sequence \( z \), the state transition and state emission parameters are conditionally independent
  - Compute posterior given states, as in the M-step of the EM algorithm for HMMs
  - Sample from posterior (often easy)

- **Standard Gibbs:** Sample state variables \( z_t \) one at a time, given parameters and other states:
  \[
  p(z_t \mid z_{\backslash t}, \theta, \pi, y) \propto p(z_t \mid z_{t-1}, \pi)p(z_{t+1} \mid z_t, \pi)p(y_t \mid z_t, \theta)
  \]

- **Blocked Gibbs:** Sample entire state sequence, given parameters and entire observation sequence, use sum-product dynamic programming
  \[
  p(z \mid \theta, \pi, y) \propto p(z \mid \pi)p(y \mid z, \theta)
  \]
  - Similar to use of sum-product to compute marginals in E-step of EM algorithm
Gibbs Sampling for Learning HMMs

- Given a fixed state sequence $z$, the state transition and state emission parameters are conditionally independent
  - Compute posterior given states, as in the M-step of the EM algorithm for HMMs
  - Sample from posterior (often easy)

**Experiment with Synthetic Data**

![Graphs showing observations, estimated mode sequence, and normalized Hamming distance](image)

**Standard Gibbs**

- Normalized Hamming Distance vs. Iteration
- Parameters

**Blocked Gibbs**

- Normalized Hamming Distance vs. Iteration
- Parameters
Blocked Gibbs for Markov Random Fields

- For general graphs with cycles, cannot use sum-product BP to draw exact samples.
- A standard Gibbs sampler iteratively resamples the values of single variables, in some order:

- A blocked Gibbs sampler iteratively resamples subsets of the original variables that are tractable (form a sub-graph without cycles):

- Draw exact samples by applying blocked sampler to junction tree. Intractable?