Loopy Belief Propagation

- does not require \( q(x) \) to be valid distribution, only "locally consistent" (adjacent nodes agree on marginals)

- relation to "turbo codes" in information theory

- algorithm: apply sum-product updates repeatedly (exact for trees)

- convergence issues, but now exist convex/convergent vectors

- LBP is only for Gaussian/Discrete, generalization is "expectation propagation"

\[ \begin{align*}
\text{Pseudo-likelihood} & \\
\text{For learning parameters of UGM, common and consistent convex approximation is "pseudo-likelihood"}
\end{align*} \]

\[ p(x) \propto \prod_{c} y_{c}(x) \propto \prod_{d} p(x_{d} | x_{\bar{d}}) \]

- training is now easy as in DAGs

Convex Relaxation to Decoding

\[
\begin{align*}
\max_{x \in \{0,1\}} & \sum_{i=1}^{n} \phi_{i}(x_{i}) + \sum_{i=1}^{n} \theta_{i}(x_{i} \rightarrow x_{\bar{i}}) \\
\text{subject to} & \sum_{i=1}^{n} z_{i}^{j} = 1 & (\text{can only pick one state}) \\
& z_{i}^{j} \in \{0,1\} & \text{(edge labelings agree with node labelings)} \\
& z_{i}^{j} \in \{0,1\} & \text{(integer linear program)} \\
& z_{i}^{j} = z_{i}^{j} & \text{(edge labelings agree with node labelings)}
\end{align*}
\]

"Linear Programming Relaxation"
Monte Carlo Methods

Methods for intractable inference problems

Variational: approximate $p(x)$ with simple "simple" $q(x)$

Markov: approximate $p(x)$ with $S$ samples $x^1, x^2, \ldots, x^S$

- use $x^1, x^2, \ldots, x^S$ to approximate
  - $p(x)$
  - $p(B|D)$
  - $\int p(x)$

Ex. p-relevant

\[
\begin{matrix}
\theta_1 & \theta_2 & \theta_3 \\
0.1 & 0.1 & 0 \\
0.3 & 0.8 & 0 \\
0.2 & 0.7 & 0.1 \\
0.1 & 0.7 & 0.2 \\
\end{matrix}
\]

$E[\theta] = \frac{(0.1 + 0.3 + 0.1 + 0.1)}{4} = 0.15$

But the hard part is selecting good samples.

Variational: fast but biased
Markov: slow but consistent (all more complex - slower but better)

Sampling from Standard Distribution

Assume you can generate $u \sim U(0,1)$

If $x \sim \text{Multihit}(0.7, 0.1, 0.2)$

\[
\begin{matrix}
0.7 & 0.1 & 0.2 \\
\end{matrix}
\]

Generate $u \sim U(0,1)$, if it falls in "A" then $x^5 = 2$

For univariate continuous, use inverse CDF

For R.V. $X$, CDF is $F(x) = P(X \leq x)$

Inverse CDF: $F^{-1}(p) = \{ x : F(x) = p \}$

"quantile function"

ex: $F(0.5) = \text{median}$

Sampling from DAG

- Sample last
  - Sample conditioned on parents
  - Sample first

"Ancstral Sampling"

Special case:

$x$ is a multivariate normal $x \sim N(\mu, \Sigma)$

- Generate $x \sim N(0, I)$
- Set: $y = \mu + Lx$, where $LL^T = \Sigma$
Rejection Sampling

- Sample from proposal \( q(x) \), where \( \frac{Mq(x)}{p(x)} \), \( M \in \mathbb{R}^+ \)
  
Ex: Sample from posterior: \( p(d|\theta) = p(d|\theta)p(\theta)/p(d) \)
Target: \( p(d|\theta) \)
Proposal: \( q(d|\theta) = p(d|\theta) \)

We have: \( M = \frac{p(d|\theta)p(\theta)}{p(d|\theta)p(\theta)} \) (use: satisfies inequality)

“Adaptive Rejection Sampling”

Importance Sampling

Want to compute \( E[h(x)] = \int h(x)p(x)dx \)

- Represent as: \( \sum_{i=1}^{N} h(x_i)p(x_i)dx_i \)
- Sample from \( q(x) \), weight each sample by \( \frac{p(x_i)}{q(x_i)} \)

\( E[h(x)] \approx \sum_{i=1}^{N} w_i h(x_i) \)

Importance Sampling (SIR, iterative filter)

- Generate from sequence of proposal distributions \( q_k(x) \)

MCML (Markov Chain Monte Carlo)

High-dimensional integration

Key idea: Construct Markov chain with \( p(x) \) as stationary distribution
- We will do a random walk through \( X \), where asymptotically we spend \( p(x) \) time at \( x \).
- Use these samples for Monte Carlo integration
Gibbs Sampling

(sampling variant of coordinate descent)

- sample $x_1 \sim p(x_1 | x_2, x_3)$
- sample $x_2 \sim p(x_2 | x_1, x_3)$
- sample $x_3 \sim p(x_3 | x_1, x_2)$
- repeat

- ELABORATE ON HIGH-DIMENSIONAL SAMPLING TO LO SAMPLING

- may require "burn-in" (if you start in areas of low probability)
- (don't want to be biased by start point)

- for graphical models, only depend on neighbors (not co-parents for DAGs)

- "block Gibbs sampling"

- "Rao-Blackwellized"

Metropolis-Hastings Algorithm

- generalization that allows arbitrary "proposal"

Metropolis algorithm for symmetric proposal:
- at some $x$, propose $x'$
- accept with probability min $(1, \frac{p(x')}{p(x)})$
- sample from arbitrary density $p(x)$
- adapted to asymmetric

Simulated Annealing

\[
\min \left( 1, \exp \left( \frac{E(x) - E(x')}{T} \right) \right)
\]

- $T$ large, can go anywhere
- $T$ small, "only uphill"