Inference Part II

This unit complements [RN] inference slides, adding details for poly-trees and MCMC inference. Slides use material from [RN95, RN02] text and [M99] paper.

MCMC Inference

• Markov Chain whose states are samples from network.
  \[ \pi_{t+1}(x') = \sum_x \pi_t(x) Q\{x' | x\} \]
  (Under some conditions) has unique stationary distribution, i.e., \[ \pi_{t+1}(x) = \pi_t(x) \].
  Main idea: build MC so that stationary distribution is the one we want to sample from.

Gibbs Sampling

• Fix all evidence nodes to their values.
• Notation: \( \hat{X} \) includes all variables but \( X_i \) (this includes evidence nodes).
• Define \( Q\{\} \) as follows: Pick \( i \) at random and pick value for \( X_i \) based on \( \Pr\{X_i | \hat{X}\} = \Pr\{X_i | \text{Markov Blanket} (X_i)\} \alpha \Pr\{X_i | \text{Pa} (X_i)\} \prod \Pr\{Y_{i,j} | \text{Pa} (Y_{i,j})\} \).
• Fact: \( \Pr\{X | E\} \) is in detailed balance with \( Q\{\} \).

Metropolis Hastings Sampling

• Assumes: desired \( p() \) can be evaluated (but not easily sampled).
  Or we can at least calculate \( p(x')/p(x) \) for fixed neighbors \( x, x' \).
  \( Q\{x'|x\} \) is a next state distribution which is not necessarily in detailed balance with \( p() \). Define a new \( Q\{\} \) as follows:
  • Given \( x \) draw \( x' \) using \( Q\{x'|x\} \).
  • With probability \( \min(1,\frac{p(x')Q\{x|x'\}}{p(x)Q\{x'|x\}}) \) return \( x' \) and otherwise return \( x \).
• Fact: \( p() \) is in detailed balance with \( Q\{\} \).
• Fact: Gibbs sampling (detailed balance) a special case.

MCMC Inference

• Detailed Balance between \( p() \) and \( Q\{\} \) provides an easy way to guarantee that \( p() \) is the stationary distribution.
  • DB: \( \forall x, x': p(x)Q\{x'|x\} = p(x')Q\{x|x'\} \)
  • \( \pi_{t+1}(x') = \sum_x \pi_t(x) Q\{x' | x\} = \sum_x \pi_t(x') Q\{x' | x\} = \pi_t(x') \sum_x Q\{x'|x\} = \pi_t(x') \).

Need "burn in" time to get to stationary distribution.

Subsequent samples are correlated. Using just one sample per burn in is expensive.

Various compromises e.g., one burn in followed by sampling every \( k \) steps (to reduce correlation).
Poly-Tree Networks

- Can order variables so that Variable elimination runs in linear time (in network representation size)
- [P88] gives message passing algorithm (see [M99])
- [RN95] gives a variant as a recursive algorithm

Inference in Poly-Tree Networks [RN95]

A singly connected network. There are no cycles even ignoring the direction of edges.

- **Pr** \(X \mid E\)
- **Can order variables so that Variable elimination runs in linear time**
- **E** + \(X\) is the evidence reachable through parents of \(X\)
- **E** − \(X\) is the evidence reachable through children of \(X\)
- If \(E\) is empty then \(Pr(X \mid E) = Pr(X)\)
- \(Pr(X \mid E) = Pr(X \mid E^{-} X^{+}) = \frac{Pr(E^{-} X^{+} \mid X) Pr(X)}{Pr(E^{-} X^{+})}\)
- But \(X\) d-separates \(E^{-} X^{+}\).
- \(Pr(X \mid E) = Normalize(Pr(E^{-} X^{+}) Pr(X \mid E^{-} X^{+}))\)

Inference in Poly-Tree Networks [RN95]

- Computing \(Pr(X \mid E^{+})\)
- If \(parents(X)\) is empty then so is \(E^{+}\) so \(Pr(X \mid E^{+}) = Pr(X)\)
- If \(U = \{U_{1}, \ldots\} = parents(X)\) and \(V = \{V_{1}, \ldots\} = children(X)\) and \(E_{U} \setminus X\) is evidence reachable from \(U_{1}\) but not through \(X\) then:
  \[Pr(X \mid E^{-} X^{+}) = \sum_{u} Pr(X \mid U = u) \prod_{i} Pr(U_{i} = u_{i} \mid E_{U_{i} \setminus X})\]
- \(Pr(X \mid U = u)\) given in CPT
- If \(U_{i}\) is an evidence node then \(Pr(U_{i} = u_{i} \mid E_{U_{i} \setminus X}) = 1\) for the correct value and 0 otherwise.
- Otherwise, recursive call (with less evidence).

Inference in Poly-Tree Networks

- Computing \(Pr\{E^{-} X^{+} \mid X\}\):
- If \(E^{-} X^{+}\) is empty then return \((1, \ldots, 1)\)
- \(Pr\{\}\) = 1 for each value of \(X\)
- Let \(Y = \{Y_{1}, \ldots\} = children(X)\)
  and \(Z_{i} = \{Z_{i1}, \ldots\} = parents(Y_{i}) \setminus X \setminus E\)
  and \(Z_{i}' = parents(Y_{i}) \setminus X \cap E\)
- For each \(Y_{i}\) compile its contribution \(contrib(Y_{i})\)
- Finally return \(Pr\{E^{-} X^{+} \mid X\} = \prod_{i} contrib(Y_{i})\)
Message Passing Algorithm [M99]

- \( \Pr(X|E) = \frac{\Pr(E_X|X)\Pr(X|E_X)}{\Pr(E_X|X)\Pr(X|E_X) + \Pr(E_{\neg X}|X)\Pr(X|E_{\neg X})} \)
- \( \Pr(x|E) = \alpha \Pr(E_X|X = x) \Pr(X = x|E_X) \)
- Define \( \lambda_X(x) = \Pr(E_X|X = x) \)
- Define \( \pi_X(x) = \Pr(X = x|E_X) \)
- \( \Pr(x|E) = \alpha \lambda_X(x) \pi_X(x) \)

Example

- Compute \( \Pr(Cold | Sneezing = 1) \)
- \( E_X = \phi \) and \( E_{\neg X} = (Sneezing = 1) \)
- Want to compute: Normalize(\( \Pr(E_X|X)\Pr(X|E_X) \))
- \( \Pr(Cold|\phi) = \Pr(Cold) \) and available in CPT:
  - \( \Pr(Cold = 0) = 0.90 \) and \( \Pr(Cold = 1) = 0.10 \)
- Next we need \( \Pr(Sneezing = 1|Cold) \)
- \( Cold \) has only one child \( Y_1 \) = \( Sneezing \) and it is an evidence node.
- \( Z_1 = \{Z_1, \} = \{\text{Allergy}\} \) and \( Z_1 = \phi \)
- We need to recursively compute \( \Pr(\text{Allergy}|\phi) = \Pr(\text{Allergy}) \)
- Imagine we have done that to result in:

Message Passing Algorithm

- Calculating \( \lambda_X(x) = \Pr(E_X|X = x) \):
- \( \lambda_X(x) = \Pr(E_X|X = x) \prod \Pr(E_{\neg X}|X) = \Pr(X|E_{\neg X}) \)
  - \( E_X \) is the evidence at node \( X \) (if any)
  - \( E_{\neg X} \) is evidence reachable via \( Y_j \) excluding paths via \( X \)

- Define message \( \lambda_{C->P}(p) = \Pr(E_{P->C}|P = p) \)

- \( \lambda_X(x) = \lambda_{X->X}(x) \prod \lambda_{Y_j->X}(x) \)
  - \( \lambda_{X->X}(x) \) is unit vector if evidence and all 1’s vector otherwise.
- Need \( \lambda \) messages from all children.
Message Passing Algorithm - the messages

• Calculating $\lambda_{C \rightarrow P}(p_i)$: $P$ is all parents; $P_i$ single parent

$\lambda_{C \rightarrow P}(p_i) = \Pr\{E_{P \rightarrow C} \mid P_i = p_i\} =
\alpha \sum_c \lambda_C(c) \left( \sum_{\hat{P}_i=\hat{p}} \Pr\{C = \hat{c} \mid P = \hat{p}\} \prod_{k \neq i} \pi_{C_k \rightarrow P}(p_k) \right)$

• Need $\lambda$ messages from all children and $\pi$ messages from all parents but $i$.

• Calculating $\pi_{P \rightarrow C_j}(p)$: $C$ are all children; $C_j$ single child

$\pi_{P \rightarrow C_j}(p) = \Pr\{P = p \mid E_{P \rightarrow C_j}\} =
\alpha \pi_P(p) \lambda_{P \rightarrow C_j}(p) \prod_{k \neq j} \lambda_{C_k \rightarrow P}(p)$

• Need $\pi$ messages from all parents and $\lambda$ messages from all children but $j$.

Message Passing Algorithm

• Poly-tree network: messages can be ordered and algorithm converges.

• Loopy belief propagation for general DAGs: repeat sending messages until (hopefully) converge.

• Example: . . .