Latent Tree Models and
Approximate Inference in Bayesian Networks

Yi Wang    Nevin L. Zhang    Tao Chen

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology

July 15, 2008
1. The Problem
2. Our Solution
3. Experimental Results
4. Summary
The Problem

- Probabilistic inference in Bayesian networks (BNs)
  - Given a BN $\mathcal{N}$ over $\mathbf{X}$ with joint distribution $P_\mathcal{N}(\mathbf{X})$
  - Observing evidence $\mathbf{E} = \mathbf{e}$
  - What is $P_\mathcal{N}(Q|\mathbf{E} = \mathbf{e})$?

- Useful for prediction and diagnosis
- Intractable for general BNs
- Our contribution: A new approximate method
Latent Tree Models: Definition

- Tree-structured Bayesian networks with discrete variables
- Variables at leaf nodes are observed
  - Manifest variables
- Variables at internal nodes are hidden
  - Latent variables
Latent Tree Models: Properties

- Computationally simple
  - Probabilistic inference takes linear time
- Represent complex relationships among manifest variables
  - Eliminating all latent variables \( \Rightarrow \text{complete graph} \)
  - Need fully-connected BN without latent variables to represent all relationships
Our Solution

- Two-phase approach
  1. Offline: Approximate BN $\mathcal{N}$ using a latent tree model $\mathcal{M}$
     - $\mathcal{M}$ uses $\mathbf{X}$ as manifest variables
     - $P_{\mathcal{M}}(\mathbf{X})$ should be close to $P_{\mathcal{N}}(\mathbf{X})$
  2. Online: Make inference with $\mathcal{M}$
     - Return $P_{\mathcal{M}}(Q|E = e)$ instead of $P_{\mathcal{N}}(Q|E = e)$
     - Use any exact inference methods

- Low online cost and good approximation accuracy
Offline Phase

- Learn an LTM to approximate given BN
- Input: (1) BN $\mathcal{N}$ over random variables $\mathbf{X}$; (2) Parameter $C$
- Output: An LTM $\mathcal{M}$
  - Using $\mathbf{X}$ as manifest variables
  - Cardinalities of latent variables upper bounded by $C$
  - Small KL divergence

$$D[\mathcal{P}_\mathcal{N}(\mathbf{X})\|\mathcal{P}_\mathcal{M}(\mathbf{X})] = \sum_{\mathbf{X}} \mathcal{P}_\mathcal{N}(\mathbf{X}) \log \frac{\mathcal{P}_\mathcal{N}(\mathbf{X})}{\mathcal{P}_\mathcal{M}(\mathbf{X})}$$

Three steps
1. Determine tree structure using hierarchical clustering
2. Set cardinalities of latent variables at $C$ and simplify models
3. Optimize parameters using EM algorithm
Step 1: Structure Learning

- **Basic ideas**
  1. Siblings are more closely correlated than nodes located far apart
  2. If $\mathcal{M}$ approximates $\mathcal{N}$ well, then
     Nodes closely correlated in $\mathcal{M} \iff$ they closely correlated in $\mathcal{N}$

- **The procedure**
  1. Compute pairwise mutual information based on $P_{\mathcal{N}}(X)$
  2. Hierarchically cluster manifest variables
  3. Introduce a latent variable for each cluster
Step 2: Cardinalities of Latent Variables

- Choice of $C$ influences performance
- Consider two extreme cases
  1. $C$ is very large ($\geq \prod_{X \in \mathbf{X}} |X|$)
     - Each latent variable represents a joint variable of $\mathbf{X}$
     - Represent BN $\mathcal{N}$ exactly
     - High inferential complexity
  2. $C = 1$
     - Independent model
     - All interactions among $\mathbf{X}$ lost
     - Poorest approximation to $\mathcal{N}$
     - Lowest inferential complexity
Step 2: Cardinalities of Latent Variables

- Starting from large $C$ and gradually decreasing it
  - Less interactions among $X$ captured
  - Poorer approximation quality
  - More efficient online inference

- **Tradeoff between efficiency and accuracy** by changing $C$
Step 3: Parameter Learning

- Given model structure, find optimal parameter

$$\theta^* = \arg\min_{\theta} D[P_N(X) \| P_M(X|\theta)]$$

- No closed-form solution due to latent variables in $\mathcal{M}$
Step 3: Parameter Learning

- Solve equivalent maximum likelihood estimate problem
  1. Generate a data set $\mathcal{D}$ with $N$ samples from BN $\mathcal{N}$
  2. Find the MLE

$$\hat{\theta} = \arg \max_{\theta} \log P_M(\mathcal{D}|\theta)$$

- As $N \to \infty$, $\hat{\theta}$ almost surely converges to $\theta^*$
  - Use large $N$ for better approximation
- Values of latent variables are missing from $\mathcal{D}$
- Use EM algorithm for parameter learning
  - Converges slowly, especially for large sample sizes
  - Long offline phase
Experiments

- Evaluated on 8 networks
- Examined impact of $C$
- Compared with
  - Clique tree propagation
  - Loopy belief propagation
  - Method based on Chow-Liu tree
  - Method based on latent class model
Impact of $C$

- Increase $C$
  - Higher approximation accuracy
  - Longer running time

- Exchange efficiency for accuracy by increasing $C$
Versus Clique Tree Propagation

- Good approximation accuracy
  - Average KL less than $10^{-2}$
- Low inferential cost
  - Faster by one to two orders of magnitude
Pearl’s tree propagation on general BNs
Succeeds in many real-world applications
Our method v.s. LBP
  Comparable or better accuracy
  Much more efficient
Summary

A novel approximate method for probabilistic inference

- Exploits merits of LTM
- Can tradeoff between online efficiency and accuracy
- Achieves good accuracy at low online cost

Offline phase takes long time

- EM is time consuming

Suitable for applications

- Allow long offline phase
- Demand good online performance
Thank You

Welcome to our poster presentation!