Pearl's algorithm

Message passing algorithm for exact inference in polytree BBNs

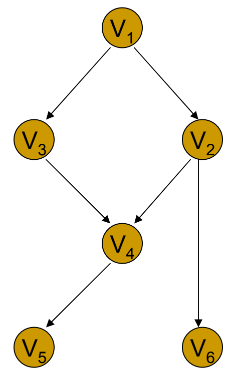
Tomas Singliar, tomas@cs.pitt.edu

Outline

- Bayesian belief networks
- The inference task
- Idea of belief propagation
- Messages and incorporation of evidence
- Combining messages into belief
- Computing messages
- Algorithm outlined
- What if the BBN is not a polytree?

Bayesian belief networks

- *(G, P)* directed acyclic graph with the joint p.d. *P*
- each node is a variable of a multivariate distribution
- links represent causal dependencies
  - CPT in each node
  - d-separation corresponds to independence
- polytree
  - at most one path between V_i and V_k
  - implies each node separates the graph into two disjoint components
  - (this graph is not a polytree)



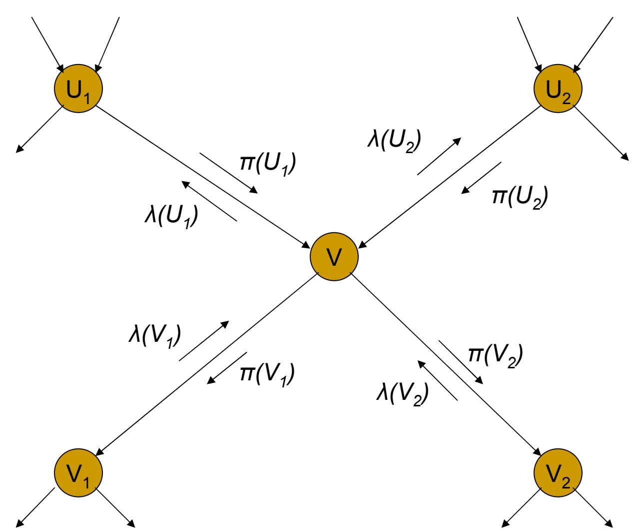
The inference task

- we observe the values of some variables $V_{e_1},\dots,V_{e_{|E|}}$
- they are the *evidence variables* E
- Inference - to compute the conditional probability $P(X_i\,|E)$ for all *non-evidential* nodes X_i
- Exact inference algorithm
  - Variable elimination
  - Join tree
  - **Belief propagation**
- Computationally intensive – (NP-hard)
- Approximate algorithms

Pearl's belief propagation

- We have the evidence *E*
- Local computation for one node *V* desired
- Information flows through the links of *G*
  - *flows as messages of two types – λ and π*
- *V* splits network into two *disjoint* parts
  - Strong independence assumptions induced – crucial!
- Denote E_V^+ the part of evidence accessible through the parents of *V* (*causal*)
  - passed downward in *π messages*
- Analogously, let E_V^- be the *diagnostic* evidence
  - passed upwards in *λ messages*

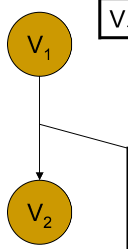
Pearl's Belief Propagation



The π Messages

- What are the messages?
- For simplicity, let the nodes be binary

V_1=T	0.8
V_1=F	0.2



P	V_1=T	V_1=F
V_2=T	0.4	0.9
V_2=F	0.6	0.1

The message passes on information.

What information? Observe:

$$P(V_2 | V_1) = P(V_2 | V_1=T)P(V_1=T) + P(V_2 | V_1=F)P(V_1=F)$$

The information needed is the CPT of $V_1 = \pi_V(V_1)$

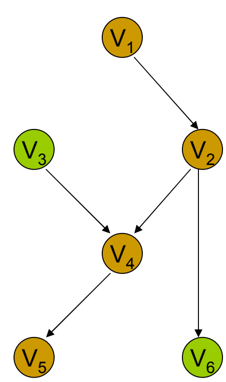
π Messages capture information passed from parent to child

The Evidence

- Evidence – values of observed nodes
  - V_3 = T, V_6 = 3
- Our belief in what the value of V_i 'should' be changes.
- This belief is propagated
- As if the CPTs became

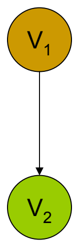
V_3=T	1.0
V_3=F	0.0

P	V_2=T	V_2=F
V_6=1	0.0	0.0
V_6=2	0.0	0.0
V_6=3	1.0	1.0



The Messages

- We know what the π messages are
- What about λ?



Assume E = { V_2 } and compute by Bayes rule:

$$P(V_1 \mid V_2) = \frac{P(V_1)P(V_2 \mid V_1)}{P(V_2)} = \alpha P(V_1)P(V_2 \mid V_1)$$

The information not available at V_1 is the $P(V_2|V_1)$. To be passed upwards by a λ-message. Again, this is not in general exactly the CPT, but the *belief* based on evidence down the tree.

- The messages are $\pi(V)=P(V|E^+)$ and $\lambda(V)=P(E^-|V)$

Combination of evidence

- Recall that $E_V = E_V^+ \cup E_V^-$ and let us compute

$$P(V \mid E) = P(V \mid E_V^+, E_V^-) = \alpha' P(E_V^+, E_V^- \mid V)P(V) =$$
$$\alpha' P(E_V^- \mid V)P(E_V^+ \mid V)P(V) = \alpha P(E_V^- \mid V)P(V \mid E_V^+) =$$
$$\alpha\lambda(V)\pi(V) = BEL(V)$$

- α is the normalization constant
- normalization is not necessary (can do it at the end)
- but may prevent numerical underflow problems

Messages

- Assume *X* received λ-messages from neighbors
- How to compute λ(x) = p(e^-|x)?
- Let $Y_1, \dots, Y_c$ be the children of *X*
- $\lambda_{XY}(x)$ denotes the λ-message sent between X and Y

$$\lambda(x) = \prod_{j=1}^{c} \lambda_{Y_j X_i}(x)$$

Messages

- Assume *X* received *π* -messages from neighbors
- How to compute $\pi(x) = p(x|e^+)$?
- Let $U_1, \dots, U_p$ be the parents of *X*
- $\pi_{XY}(x)$ denotes the π-message sent between X and Y
- summation over the CPT

$$\pi(x) = \sum_{u_1,\dots,u_p} P(x \mid u_1,\dots,u_p) \prod_{j=1}^{p} \pi_{U_j X_i}(u_j)$$

Messages to pass

- We need to compute $\pi_{XY}(x)$

$$\pi_{XY_J}(x) = \alpha \pi_X(x) \prod_{k \neq j} \lambda_{Y_k X}(x)$$

- Similarly, $\lambda_{XY}(x)$, X is parent, Y child
- Symbolically, group other parents of Y into V = $V_1, \ldots, V_q$

$$\lambda_{Y_j X}(x) = \sum_{y_j} \lambda_{Y_j}(y_j) \sum_{v_1,\ldots,y_q} p(y \mid v_1,\ldots,v_q) \prod_{k=1}^{q} \pi_{V_k Y_j}(v_k)$$

The Pearl Belief Propagation Algorithm

- We can summarize the algorithm now:
  - Initialization step
    - For all $V_i=e_i$ in E:
      - $\lambda(x_i) = 1$ wherever $x_i = e_i$; 0 otherwise
      - $\pi(x_i) = 1$ wherever $x_i = e_i$; 0 otherwise
    - For nodes without parents
      - $\pi(x_i) = p(x_i)$ - prior probabilities
    - For nodes without children
      - $\lambda(x_i) = 1$ uniformly (normalize at end)

The Pearl Belief Propagation Algorithm

- Iterate until no change occurs
  - (For each node X) if X has received all the π messages from its parents, calculate $\pi(x)$
  - (For each node X) if X has received all the λ messages from its children, calculate $\lambda(x)$
  - (For each node X) if $\pi(x)$ has been calculated and X received all the λ-messages from all its children (except Y), calculate $\pi_{XY}(x)$ and send it to Y.
  - (For each node X) if $\lambda(x)$ has been calculated and X received all the π-messages from all parents (except U), calculate $\lambda_{XU}(x)$ and send it to U.
- Compute BEL(X) = $\lambda(x)\pi(x)$ and normalize

Complexity

- On a polytree, the BP algorithm converges in time proportional to diameter of network – at most linear
- Work done in a node is proportional to the size of CPT
- Hence BP is linear in number of network parameters
- For general BBNs
  - Exact inference is NP-hard
  - Approximate inference is NP-hard

Most Graphs are not Polytrees

- Cutset conditioning
  - Instantiate a node in cycle, absorb the value in child's CPT.
  - Do it with all possible values and run belief propagation.
  - Sum over obtained conditionals
  - Hard to do
    - Need to compute P(c)
    - Exponential explosion - minimal cutset desirable (also NP-complete)
- Clustering algorithm
- Approximate inference
  - MCMC methods
  - Loopy BP

Thank you

- Questions welcome
- References

  - Pearl, J. : Probabilistic reasoning in intelligent systems – Networks of plausible inference, Morgan – Kaufmann 1988
  - Castillo, E., Gutierrez, J. M., Hadi, A. S. : Expert Systems and Probabilistic Network Models, Springer 1997
    - Derivations shown in class are from this book, except that we worked with π instead of ρ messages. They are related by factor of $p(e^+)$.
  - www.cs.kun.nl/~peterl/teaching/CS45CI/bbn3-4.ps.gz
  - Murphy, K.P., Weiss, Y., Jordan, M. : Loopy belief propagation for approximate inference – an empirical study, UAI 99