A Differential Approach to Inference in Bayesian Networks

Presented by Yanna Shen
shenyn@cbmi.pitt.edu

Outline

- Introduction
  - Overview of algorithms for inference in Bayesian networks (BN)
  - Proposed new approach
- How to represent BN as multi-variate polynomial?
- How to answer queries?
- How to represent polynomial using arithmetic circuits?
- How to generate arithmetic circuits?
- Conclusions

Background

- Bayesian network
  - Directed acyclic graph (DAG)
  - Conditional probability tables (CPT)
- Review of three classes of inference algorithms
  - Conditioning
  - Variable elimination
  - Tree clustering

A BN with n nodes and tree width w

$O(n\ exp(w))$ in time and space

Introduction

- A new approach to inference in BN
  - The probability distribution of a BN is represented as a polynomial
  - Probabilistic queries are answered by evaluating and differentiating the polynomial
  - Polynomial is represented as an arithmetic circuit, which can be evaluated and differentiated in time and space linear in its size.

A BN with n nodes and tree width w, a circuit can be built in

$O(n\ exp(w))$ in time and space

Network Polynomial

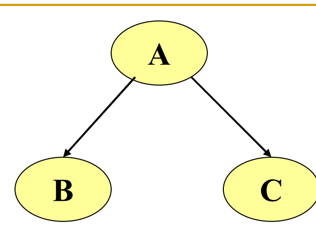
Let X be a variable; $\mathbf{U}$ be its parents in a BN

- *Evidence indicators* λ_x $\qquad \lambda_x = \begin{cases} 1 & \text{if } x \sim \mathbf{e} \text{ (evidence)} \\ 0 & \text{otherwise} \end{cases}$
- *Network parameters* $\theta_{x|\mathbf{u}}$
  - Represent the conditional probability $P_r(x \mid \mathbf{u})$

By the *Chain Rule*,

$$P_r(\mathbf{x}) = \prod_{x\mathbf{u} \sim \mathbf{x}} \theta_{x|\mathbf{u}}$$

Network Polynomial



A	θ_A
true	**0.5**
false	0.5

A	B	$\theta_{B\|A}$
true	true	1
true	false	**0**
false	true	0
false	false	1

A	C	$\theta_{C\|A}$
true	true	0.8
true	false	**0.2**
false	true	0.2
false	false	0.8

$$P_r(\mathbf{x}) = \prod_{x\mathbf{u} \sim \mathbf{x}} \theta_{x|\mathbf{u}}$$

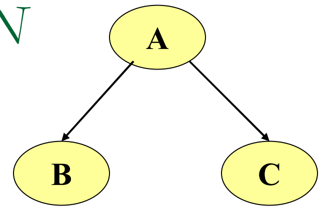
$$P_r(a\bar{b}\bar{c}) = \theta_a \theta_{\bar{b}|a} \theta_{\bar{c}|a} = 0.5 \times 0 \times 0.2 = 0$$

Polynomial of Network N

$$f = \sum_{\mathbf{x}} \prod_{x\mathbf{u} \sim \mathbf{x}} \lambda_x \theta_{x|\mathbf{u}}$$

$$f = \lambda_a \lambda_b \lambda_c \theta_a \theta_{b|a} \theta_{c|a} +$$
$$\lambda_a \lambda_b \lambda_{\bar{c}} \theta_a \theta_{b|a} \theta_{\bar{c}|a} +$$
$$\lambda_a \lambda_{\bar{b}} \lambda_c \theta_a \theta_{\bar{b}|a} \theta_{c|a} +$$
$$\lambda_a \lambda_{\bar{b}} \lambda_{\bar{c}} \theta_a \theta_{\bar{b}|a} \theta_{\bar{c}|a} +$$
$$\ldots$$
$$\lambda_{\bar{a}} \lambda_{\bar{b}} \lambda_{\bar{c}} \theta_{\bar{a}} \theta_{\bar{b}|\bar{a}} \theta_{\bar{c}|\bar{a}}$$

$$f(\mathbf{e}) = \Pr(\mathbf{e})$$



A	θ_A
true	0.5
false	0.5

A	B	$\theta_{B\|A}$
		1
		0
false	true	0
false	false	1

A	C	$\theta_{C\|A}$
true	true	0.8
true	false	0.2
false	true	0.2
false	false	0.8

A BN with n binary nodes

2^n terms (instantiations)

Evidence $\mathbf{e} = a\bar{c}$

Replace $\lambda_a = 1, \lambda_{\bar{a}} = 0, \lambda_b = 1, \lambda_{\bar{b}} = 1, \lambda_c = 0, \lambda_{\bar{c}} = 1$

$$\Pr(a\bar{c}) = f(a\bar{c}) = \theta_a \theta_{b|a} \theta_{\bar{c}|a} + \theta_a \theta_{\bar{b}|a} \theta_{\bar{c}|a} = 0.1$$

Derivatives wrt. Evidence Indicators

$$f = \lambda_a \lambda_b \lambda_c \theta_a \theta_{b|a} \theta_{c|a} +$$
$$\lambda_a \lambda_b \lambda_{\bar{c}} \theta_a \theta_{b|a} \theta_{\bar{c}|a} +$$
$$\lambda_a \lambda_{\bar{b}} \lambda_c \theta_a \theta_{\bar{b}|a} \theta_{c|a} +$$
$$\lambda_a \lambda_{\bar{b}} \lambda_{\bar{c}} \theta_a \theta_{\bar{b}|a} \theta_{\bar{c}|a} +$$
$$\ldots$$
$$\lambda_{\bar{a}} \lambda_{\bar{b}} \lambda_{\bar{c}} \theta_{\bar{a}} \theta_{\bar{b}|\bar{a}} \theta_{\bar{c}|\bar{a}}$$

How to compute $\dfrac{\partial f}{\partial \lambda_a}$?

Conditioning f on event a

Set indicator $\lambda_a = 1, \lambda_{\bar{a}} = 0$

$$\frac{\partial f}{\partial \lambda_a} = \lambda_b \lambda_c \theta_a \theta_{b|a} \theta_{c|a} + \lambda_b \lambda_{\bar{c}} \theta_a \theta_{b|a} \theta_{\bar{c}|a} +$$
$$\lambda_{\bar{b}} \lambda_c \theta_a \theta_{\bar{b}|a} \theta_{c|a} + \lambda_{\bar{b}} \lambda_{\bar{c}} \theta_a \theta_{\bar{b}|a} \theta_{\bar{c}|a}$$

$$\frac{\partial f}{\partial \lambda_a}(a\bar{c}) = \theta_a \theta_{b|a} \theta_{\bar{c}|a} + \theta_a \theta_{\bar{b}|a} \theta_{\bar{c}|a} = 0.5 \times 1 \times 0.2 + 0.5 \times 0 \times 0.2 = 0.1$$

v	λ_a	$\lambda_{\bar{a}}$	λ_b	$\lambda_{\bar{b}}$	λ_c	$\lambda_{\bar{c}}$	θ_a	$\theta_{\bar{a}}$	$\theta_{b\|a}$	$\theta_{b\|\bar{a}}$	$\theta_{\bar{b}\|a}$	$\theta_{\bar{b}\|\bar{a}}$	$\theta_{c\|a}$	$\theta_{c\|\bar{a}}$	$\theta_{\bar{c}\|a}$	$\theta_{\bar{c}\|\bar{a}}$
$\partial f / \partial v$	**0.1**	0.4	0.1	0	0.4	0.1	0.2	0	0.1	0	0.1	0	0	0	0.5	0

Partial derivatives of the network polynomial f at evidence $a\bar{c}$

Derivatives wrt. Evidence Indicators

- For every variable X and evidence $\mathbf{e}$ in a Bayesian network, $$\frac{\partial f}{\partial \lambda_x}(\mathbf{e}) = \Pr(x, \mathbf{e} - X)$$

Where, $\mathbf{e} - X$ denotes the subset of instantiation $\mathbf{e}$ pertaining to variables not appearing in X.

Evidence $\mathbf{e} = a\bar{c}$

$$\frac{\partial f}{\partial \lambda_b}(a\bar{c}) = \Pr(b, a\bar{c} - B) = \Pr(ab\bar{c})$$

Derivatives wrt. Evidence Indicators (posterior marginals)

- For every variable X and evidence $\mathbf{e}$, $X \notin \mathbf{E}$:

$$\Pr(x \mid \mathbf{e}) = \frac{1}{f(\mathbf{e})} \frac{\partial f}{\partial \lambda_x}(\mathbf{e})$$

Evidence $\mathbf{e} = a\bar{c}$

$$\Pr(b \mid \mathbf{e}) = \frac{1}{f(\mathbf{e})} \frac{\partial f}{\partial \lambda_b}(\mathbf{e}) = \frac{1}{0.1} \times 0.1 = 1$$

$$\Pr(\bar{b} \mid \mathbf{e}) = \frac{1}{f(\mathbf{e})} \frac{\partial f}{\partial \lambda_{\bar{b}}}(\mathbf{e}) = \frac{1}{0.1} \times 0 = 0$$

v	λ_a	$\lambda_{\bar{a}}$	λ_b	$\lambda_{\bar{b}}$	λ_c	$\lambda_{\bar{c}}$	θ_a	$\theta_{\bar{a}}$	$\theta_{b\mid a}$	$\theta_{b\mid \bar{a}}$	$\theta_{\bar{b}\mid a}$	$\theta_{\bar{b}\mid \bar{a}}$	$\theta_{c\mid a}$	$\theta_{c\mid \bar{a}}$	$\theta_{\bar{c}\mid a}$	$\theta_{\bar{c}\mid \bar{a}}$
$\partial f / \partial v$	**0.1**	0.4	**0.1**	**0**	0.4	0.1	0.2	0	0.1	0	0.1	0	0	0	0.5	0

Partial derivatives of the network polynomial f at evidence $a\bar{c}$, $f(a\bar{c}) = 0.1$

Derivatives wrt. Evidence Indicators (posterior marginals)

- For every variable X and evidence $\mathbf{e}$:

$$\Pr(\mathbf{e} - X) = \sum_x \frac{\partial f}{\partial \lambda_x}(\mathbf{e})$$

$$\Pr(x' \mid \mathbf{e} - X) = \frac{\frac{\partial f}{\partial \lambda_{x'}}(\mathbf{e})}{\sum_x \frac{\partial f}{\partial \lambda_x}(\mathbf{e})}$$

Evidence $\mathbf{e} = a\bar{c}$

$$\Pr(\mathbf{e} - A) = \Pr(\bar{c}) = \frac{\partial f}{\partial \lambda_a}(\mathbf{e}) + \frac{\partial f}{\partial \lambda_{\bar{a}}}(\mathbf{e}) = 0.1 + 0.4 = 0.5$$

Derivatives wrt. Network Parameters and Second Partial Derivatives

For every family $X\mathbf{U}$, and evidence $\mathbf{e}$,

$$\theta_{x|\mathbf{u}} \frac{\partial f}{\partial \theta_{x|\mathbf{u}}}(\mathbf{e}) = \Pr(x, \mathbf{u}, \mathbf{e}) \qquad (1)$$

For every pair of variables X, Y, and evidence $\mathbf{e}$, when $X \neq Y$,

$$\frac{\partial^2 f}{\partial \lambda_x \partial \lambda_y}(\mathbf{e}) = \Pr(x, y, \mathbf{e} - XY) \qquad (2)$$

For every family $X\mathbf{U}$, variable Y, and evidence $\mathbf{e}$,

$$\theta_{x|\mathbf{u}} \frac{\partial^2 f}{\partial \theta_{x|\mathbf{u}} \partial \lambda_y}(\mathbf{e}) = \Pr(x, \mathbf{u}, y, \mathbf{e} - Y) \qquad (3)$$

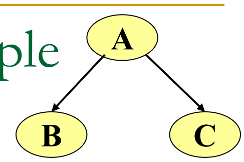
For every pair of families $X\mathbf{U}$, $Y\mathbf{V}$, and evidence $\mathbf{e}$, when $x\mathbf{u} \neq y\mathbf{v}$,

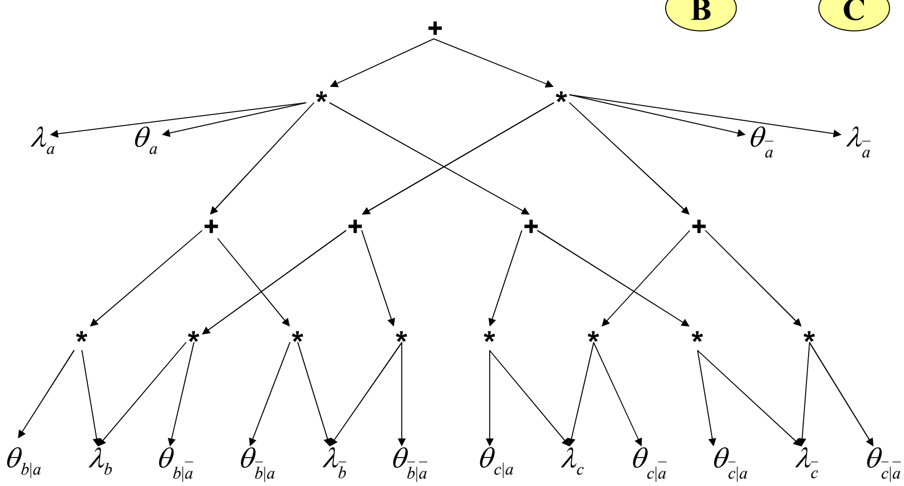
$$\theta_{x|\mathbf{u}} \theta_{y|\mathbf{v}} \frac{\partial^2 f}{\partial \theta_{x|\mathbf{u}} \partial \theta_{y|\mathbf{v}}}(\mathbf{e}) = \Pr(x, \mathbf{u}, y, \mathbf{v}, \mathbf{e}) \qquad (4)$$

How to Represent Polynomial Using an Arithmetic Circuit?

- An arithmetic circuit over variables $\sum$ is a rooted, directed acyclic graph.
  - Leaf nodes: numeric constants or variables in $\sum$
  - Other nodes: multiplication and addition operations
- Size: # of edges

An Arithmetic Circuit Example





How to Differentiate the Circuit?

- Two registers $vr(v$
  - Initialization: $dr(v)$ is
    where $dr(v) = 1$
  - Upward-pass: At no
    store it in $vr(v)$
  - Downward-pass: At node v and for each parent p, increment $dr(v)$ by
    - $dr(p)$ if p is an addition node;
    - $dr(p)\prod_{v'} vr(v')$ if p is a multiplication node, where v' are the other children of p.

If v is not the root node, and has parent p, by chain rule,

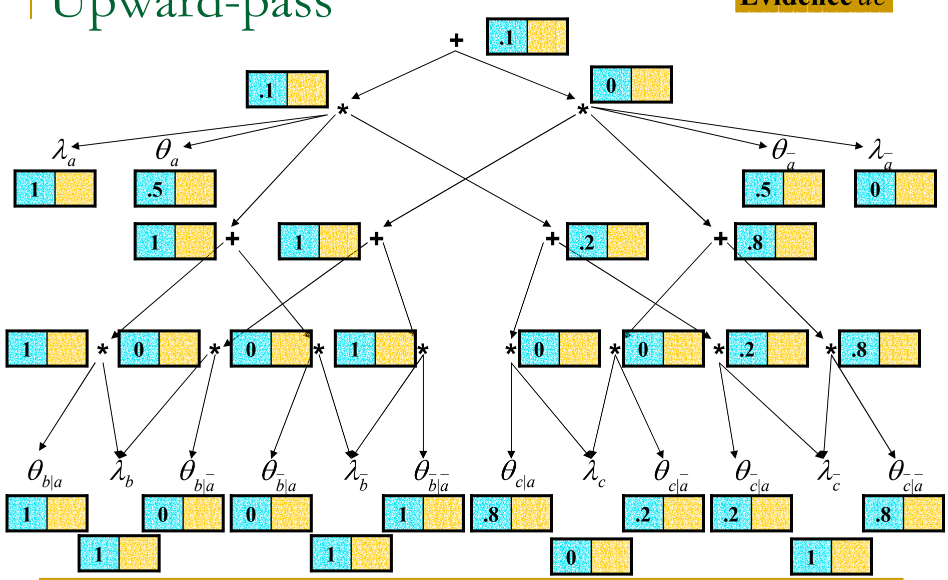
$$\frac{\partial f}{\partial v} = \sum_p \frac{\partial f}{\partial p}\frac{\partial p}{\partial v}$$

If v' are other children of parent p.

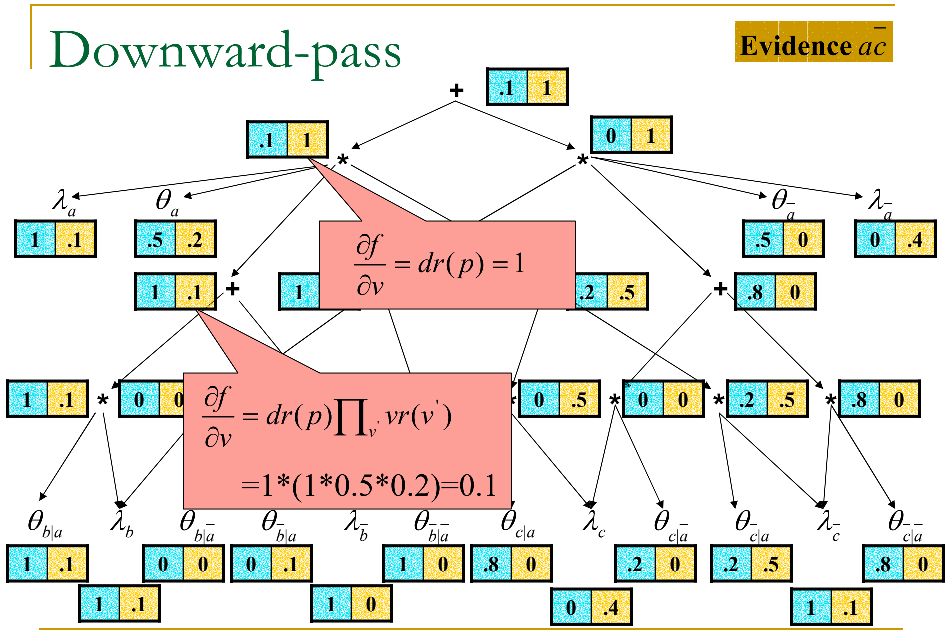
- p is a multiplication node, then $\frac{\partial p}{\partial v} = \frac{\partial\left(v\prod_{v'} v'\right)}{\partial v} = \prod_{v'} v'$
- p is an addition node, then $\frac{\partial p}{\partial v} = \frac{\partial\left(v+\sum_{v'} v'\right)}{\partial v} = 1$

Upward-pass

Evidence $a\bar{c}$



Downward-pass



The Complexity of Differentiating Circuits

- Upward-pass:
  - Time: linear in the circuit size
- Downward-pass
  - Time is linear only when each multiplication node has a bounded number of children

$$\prod_{v'} vr(v') = \frac{vr(p)}{vr(v)} \text{ when } vr(v) \neq 0$$

Time:
of v' – 1 ⟶ 1

If $vr(v) = 0$,

need two additional bits per multipication node to guarantee the method takes time which is linear in the circuit size

How to Generate Arithmetic Circuit?

- Goal: generate the smallest circuit possible; Offer guarantees on the complexity of circuits
- Two classes of methods:
  - Exploit the global structure of a BN
  - Exploit the local structure (the specific values of conditional probabilities)

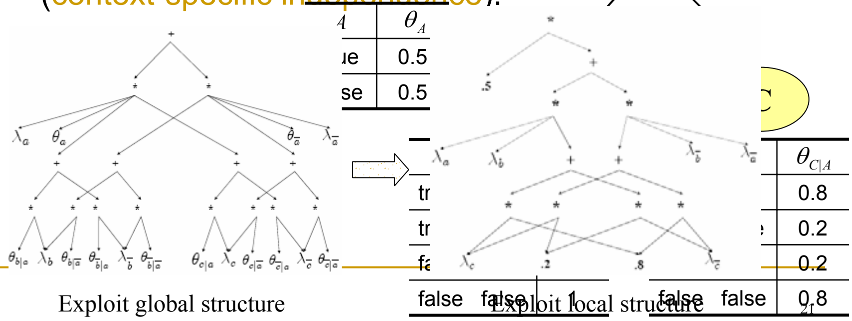
Circuits that Exploit Global Structure

- Each jointree embeds an arithmetic circuit that computes the network polynomial.
- Assuming we have a jointree for the given network, refer to *Definition 5* for generating circuits based on jointrees.
- If a network has n nodes and treewidth w, then the circuit complexity is $O(n\ exp(w))$

If the jointree has a cluster of large size, say 40, then the embedded arithmetic circuit will be intractable.

Circuits that Exploit Local Structure

- If the conditional probabilities of the BN exhibit some local structure:
  - whether some probabilities = 0 or 1 (logical constraint)
  - whether some probabilities in the same [CPT a]re equal (context-specific independence).



Circuits that Exploit Local Structure (reducing the problem to logical reasoning)

- Three conceptual steps:
  - Encoding a multi-linear function using a propositional theory
  - Factoring the propositional encoding (logical form d-DNNF, refer to [Darwiche 2002b])
  - Extracting an arithmetic circuit

Net	Vars#	d-DNNF–based ⋆/+nodes	Jointree–based ⋆/+nodes
Poker	7	302	685
Golf	8	143	676
Boblo	22	393	494
6nt	58	1377	12378
6hj	58	1814	29176
3nt	58	6328	35902

Logical constraints lead to a significant reductions in the size of circuits

Conclusions

- A new approach for inference in Bayesian networks which is based on evaluating and differentiating arithmetic circuits
- Subsumes the jointree approach
- The complexity of inference is sensitive to both the global and local structure of Bayesian networks