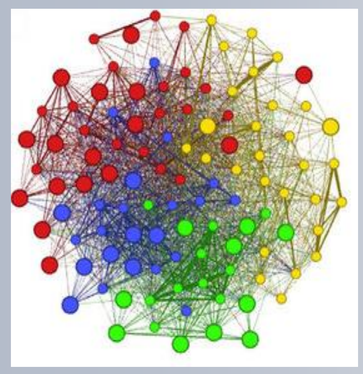
Lazy Arithmetic Circuits

Seyed Mehran Kazemi & David Poole

http://cs.ubc.ca/~{smkazemi, poole}

Problem



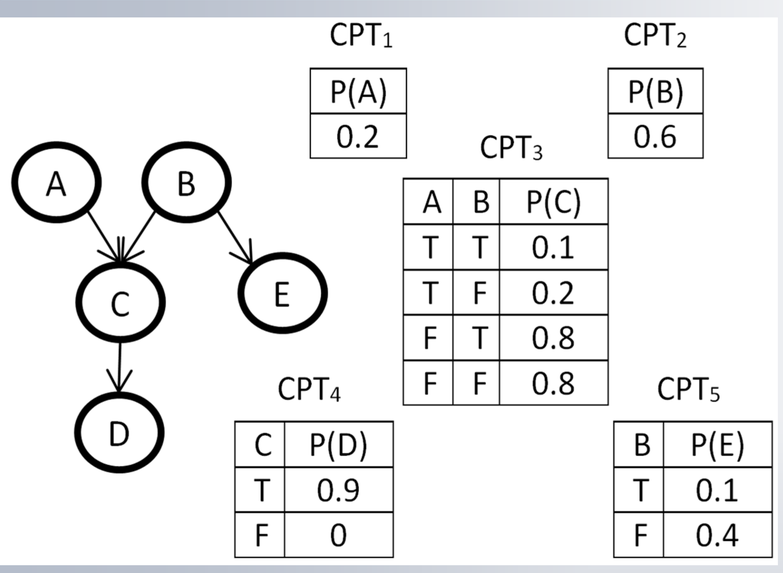
Huge Bayesian network with local structure

Offline time with high computing power

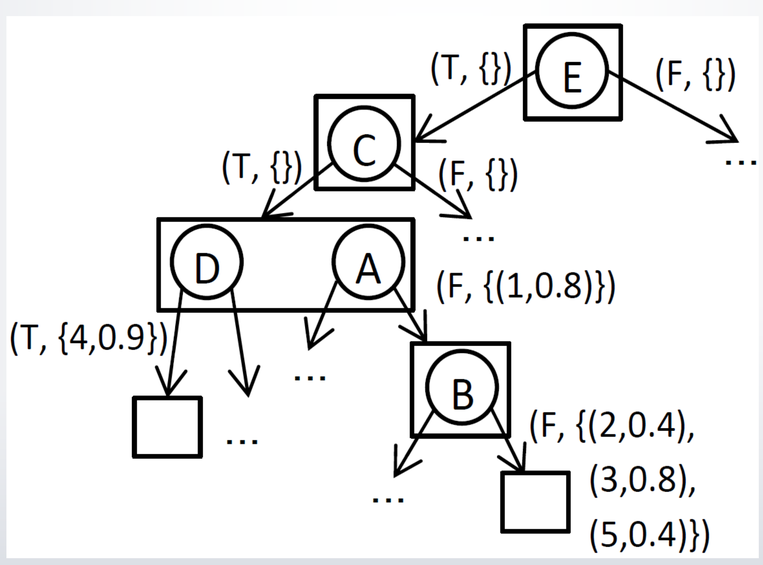
Given observation, query multiple nodes

	Local Structure	Offline Computations	Multiple Queries One Run	Pruning
Contextual Variable Elimination	✓	✗	✗	✓
Recursive Conditioning	✓	✗	✗	✓
Junction Tree	✗	✓	✓	✗
Lazy Junction Tree	✗	✓	✓	✓
Arithmetic Circuits	✓	✓	✓	✗
Lazy Arithmetic Circuits	✓	✓	✓	✓

An Example Bayesian Network



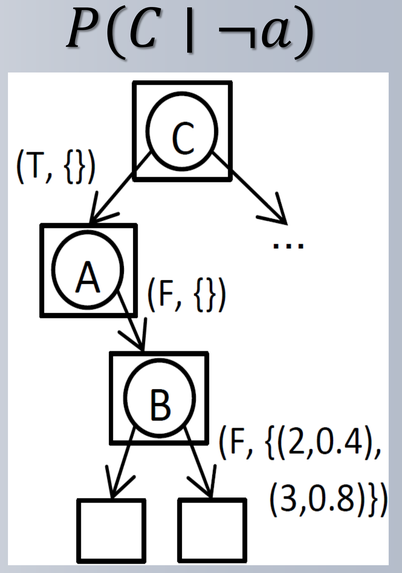
Lazy RCGraph (LRCGraph)



Pruning LRCGraphs

1. Run d-separation and find irrelevant variables (IVs).
2. Replace nodes having IVs with either one of their children.
3. Remove all <CPTIndex,Value> pairs where CPTIndex belongs to an IV.
4. For the evidence variables, only keep the child whose value is consistent with the observation.
5. During pruning, if #(root nodes) > 1, remove the nodes (and their subgraphs) from the root having only non-query variables in their subgraph.

$P(C \mid \neg a)$



From LRCGraphs to Lazy Arithmetic Circuits

- An LRCGraph can be converted to an AC by replacing each circle-node by a decision structure, each rectangle by a product over its circle-nodes, and each <CPTIndex, Value> pair on the edges by a node under appropriate branch of decision structure.
- We call these ACs **"lazy ACs (LACs)"** because they postpone multiplying evaluations at one branch to the query time.
- Pruning LACs is linear in the size of the *pruned LAC* and the number of irrelevant random variables
  - ➔ **Sub-linear** in the size of the original LAC.
- The pruned LAC can be exponentially smaller than the original LAC
  - ➔ Inference is **sub-linear** in the size of the original LAC.
- LACs ≡ decision-DNNF ⊂ d-DNNF ≡ ACs
  - ➔ Open problem: Can we generalize the pruning to more general ACs than decision-DNNFs?

REFERENCES

1. Darwiche, A. 2003. A differential approach to inference in Bayesian networks. *JACM* 50(3).
   Arithmetic circuits.
2. Poole, D., & Zhang, N. L. 2003. Exploiting contextual independence in probabilistic inference. *JAIR* 18.
   Variable elimination exploiting local structure.
3. Darwiche, A. 2001. Recursive Conditioning. *Artificial Intelligence* 126(1-2).
   Recursive conditioning
4. Madsen, A. L., and Jensen, F. V. 1999. Lazy propagation: a junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence.* 113(1).
   Junction tree algorithm with lazy evaluation/propagation.