Learning the Structure of Probabilistic SDDs

Jessa Bekker Arthur Choi Guy Van den Broeck
jessa.bekker@cs.kuleuven.be

UCLA — KATHOLIEKE UNIVERSITEIT LEUVEN

Motivation and Contribution

Inference in probabilistic graphical models (PGMs) is NP-hard
⇒ No guarantees about sampled results
⇒ Learning is also hard

Tractable learning only learns model that allow efficient inference by using a tractable representation.

We propose the first tractable learning algorithm that uses PSDDs as its representation. It learns maximally tractable models that are interpretable and allow incorporation of domain knowledge/constraints.

Tractable learning

A **tractable representation** represents the inference calculation and provides therefore a measure for the inference complexity (its size).

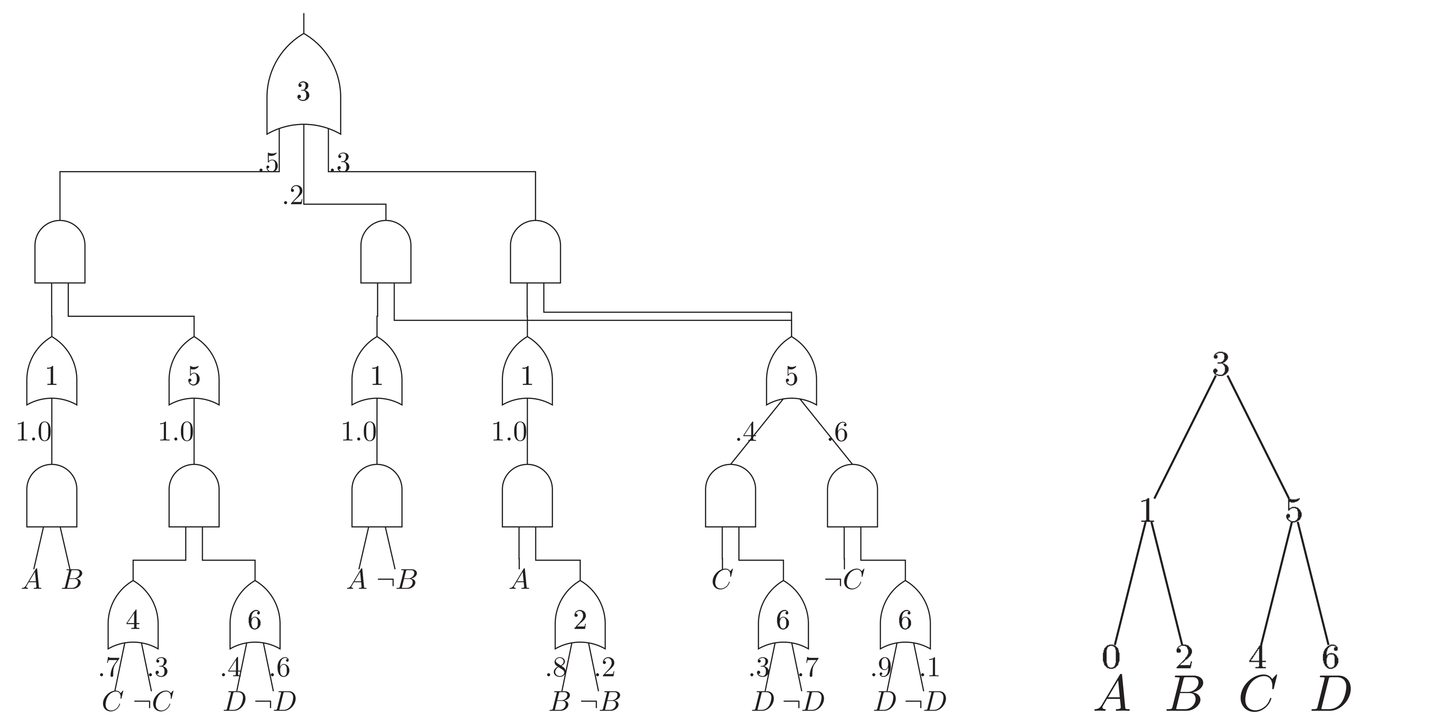
Tractable learning incrementally updates a model while keeping the tractable representation small.

The choice of tractable representation is critical:

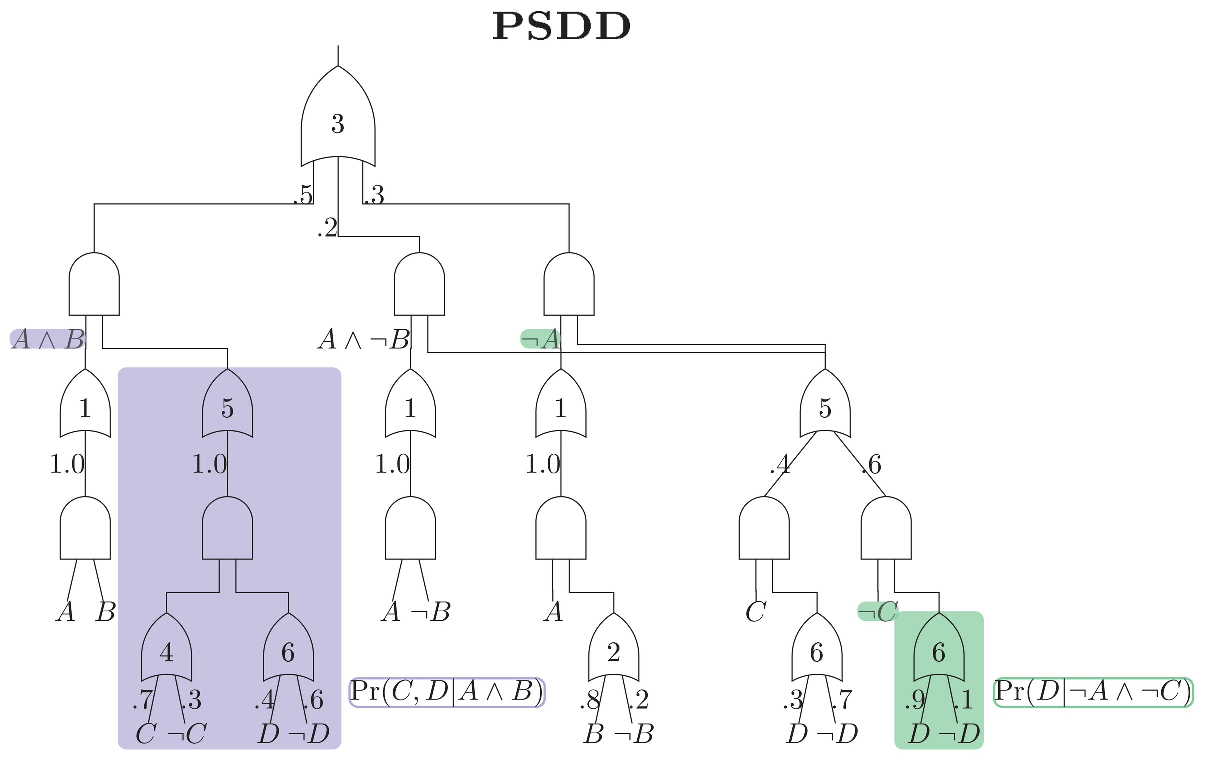
- SDDs most tractable: classic and complex symmetric queries.
- ACs and SPNs: less restricted and more stable learning.
- PSDDs combine all qualities.

Probabilistic Sentential Decision Diagrams (PSDDs)

PSDDs represent probability distributions and allow linear-time inference



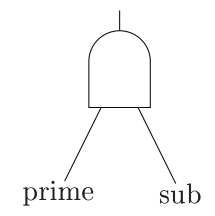
PSDD **Vtree**



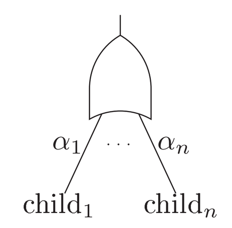
C	D	$\Pr(C,D \mid A\wedge B)$
1	1	$0.7\cdot 0.4 = 0.28$
1	0	$0.7\cdot 0.6 = 0.42$
0	1	$0.3\cdot 0.4 = 0.12$
0	0	$0.3\cdot 0.6 = 0.18$

D	$\Pr(D \mid \neg A\wedge\neg C)$
1	0.9
0	0.1

Each node $\mathcal{N}$ = probability distribution
$f(\mathcal{N})$: possible worlds (non-zero probability) as logical formula

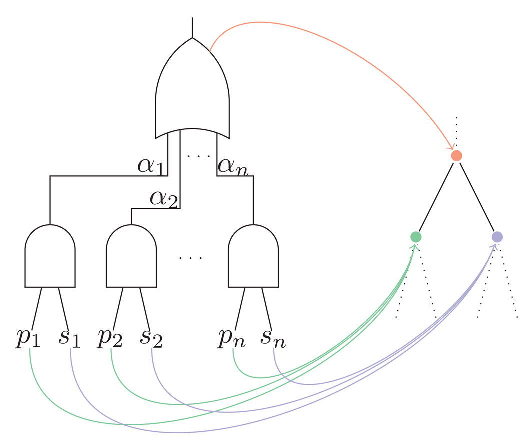


AND/product node
$\Pr(x|\mathcal{N}) = \Pr(x|\text{prime})\cdot\Pr(x|\text{sub})$
$f(\mathcal{N}) = (x|\text{prime}) \wedge f(\text{sub})$
Prime and sub are OR nodes or variables



OR/sum node
$\Pr(x|\mathcal{N}) = \sum_i \alpha_i \Pr(x|\text{child}_i)$
$f(\mathcal{N}) = \bigvee_i \Pr(x|\text{child}_i)$
Children are AND nodes or variables

Rules for a valid PSDD



Every PSDD is normalized for a **vtree**
⇒ **Decomposability**: prime and sub have no variables in common

Determinism: $f(p_i)\wedge f(p_j) = \perp$ if $i \neq j$
⇒ OR node decomposes on prime formulas $f(p_i)$

PSDDs are ...

- ✔ Efficient (marginal and complex queries)
- ✔ Interpretable
- ✔ Able to integrate expert knowledge and domain constraints
- ✔ Compact

PSDD Structure Learning

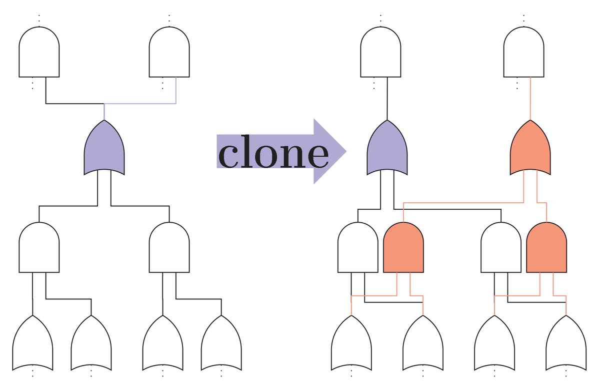
Greedy search algorithm:

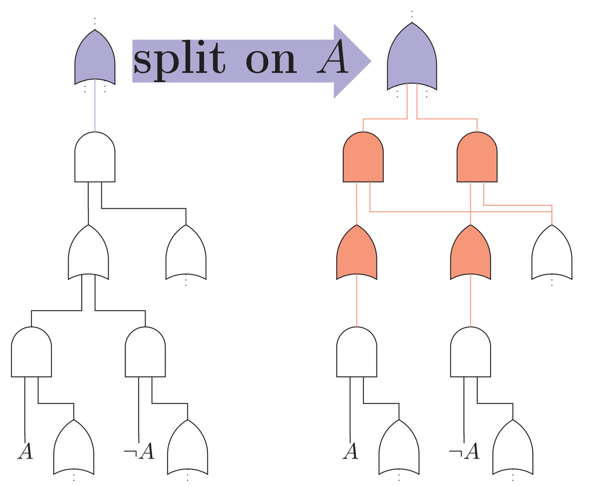
```
N = initial model
while |N| < maxSize and not converged :
  ops = candidateOperations(N)
  op = argmax_{op ∈ ops} score(op)
  N = N.apply(op)
return N
```

Small local operations: very stable

Optimization: nodes that will not change in the future are cached and reused where possible

Operations:





$$\text{Score} = \frac{\Delta\text{likelihood}}{\Delta\text{size}}$$

Parameter learning:
Count in data

Possible initial models:

- Domain constraints
- Expert knowledge
- Independent variables (Bayesian net without edges)

Related work

PSDDs
Y. Shen, A. Choi and A. Darwiche. Tractable Operations for Arithmetic Circuits of Probabilistic Models. NIPS16
A. Choi, G. Van den Broeck and A. Darwiche. Tractable Learning for Structured Probability Spaces: A Case Study in Learning Preference Distributions. IJCAI15

Other Tractable Learners
J. Bekker and J. Davis and A. Choi and A. Darwiche and G. Van den Broeck. Tractable Learning for Complex Probability Queries. NIPS15
D. Lowd and A.Rooshenas. Learning Markov Networks with Arithmetic Circuits. AISTATS13
R. Gens and P. Domingos. Learning the Structure of Sum-Product Networks. ICML13
A. Dennis and V. Ventura. Greedy Structure Search for Sum-Product Networks. IJCAI15