Probabilistic Circuits

Representations
Inference
Learning

YooJung Choi
University of California, Los Angeles

based on joint AAAI-2020 and UAI-2019 tutorials with

Guy Van den Broeck
University of California, Los Angeles

Antonio Vergari
University of California, Los Angeles

Robert Peharz
TU Eindhoven

Nicola Di Mauro
University of Bari

December 19th, 2019 - ***KOCOON workshop*** **- Arras, France**

Why tractable inference?

or expressiveness vs tractability

Probabilistic circuits

a unified framework for tractable models

Building circuits

learning them from data and compiling other models

Why tractable inference?

or the inherent trade-off of tractability vs. expressiveness

Complete evidence (EVI)

$\mathbf{X} = \{\text{Day}, \text{Time}, \text{Jam}_{\text{Alma}}, \text{Jam}_{\text{Str2}}, \ldots, \text{Jam}_{\text{StrN}}\}$

© fineartamerica.com

Complete evidence (EVI)

$\mathbf{X} = \{\text{Day}, \text{Time}, \text{Jam}_{\text{Alma}}, \text{Jam}_{\text{Str2}}, \ldots, \text{Jam}_{\text{StrN}}\}$

$\mathbf{q}_1$: *What is the probability that today is a Monday at 12.00 and there is a traffic jam only on Alma Str.?*

© fineartamerica.com

Complete evidence (EVI)

$\mathbf{X} = \{\text{Day}, \text{Time}, \text{Jam}_{\text{Alma}}, \text{Jam}_{\text{Str2}}, \ldots, \text{Jam}_{\text{StrN}}\}$

$\mathbf{q}_1$: *What is the probability that today is a Monday at 12.00 and there is a traffic jam only on Alma Str.?*

$\mathbf{q}_1(\mathbf{m}) = p_{\mathbf{m}}(\mathbf{X} = \{\text{Mon}, 12.00, 1, 0, \ldots, 0\})$

© fineartamerica.com

Complete evidence (EVI)

$\mathbf{X} = \{\text{Day}, \text{Time}, \text{Jam}_{\text{Alma}}, \text{Jam}_{\text{Str2}}, \ldots, \text{Jam}_{\text{StrN}}\}$

$\mathbf{q}_1$: *What is the probability that today is a Monday at 12.00 and there is a traffic jam only on Alma Str.?*

$$\mathbf{q}_1(\mathbf{m}) = p_{\mathbf{m}}(\mathbf{X} = \{\text{Mon}, 12.00, 1, 0, \ldots, 0\})$$

...fundamental in ***maximum likelihood learning***

$$\theta_{\mathbf{m}}^{\text{MLE}} = \text{argmax}_\theta \prod_{\mathbf{x} \in \mathcal{D}} p_{\mathbf{m}}(\mathbf{x}; \theta)$$

© fineartamerica.com

Marginal queries (MAR)

q_2: *What is the probability that today is a Monday ~~at 12.00~~ and there is a traffic jam ~~only~~ on Alma Str.?*

© fineartamerica.com

Marginal queries (MAR)

$\mathbf{q}_2$: *What is the probability that today is a Monday ~~at 12.00~~ and there is a traffic jam ~~only~~ on Alma Str.?*

$$\mathbf{q}_2(\mathbf{m}) = p_{\mathbf{m}}(\mathsf{Day} = \mathsf{Mon}, \mathsf{Jam}_{\mathsf{Alma}} = 1)$$

© fineartamerica.com

Marginal queries (MAR)

$\mathbf{q}_2$: *What is the probability that today is a Monday ~~at 12.00~~ and there is a traffic jam ~~only~~ on Alma Str.?*

$$\mathbf{q}_2(\mathbf{m}) = p_{\mathbf{m}}(\mathsf{Day} = \mathsf{Mon}, \mathsf{Jam}_{\mathsf{Alma}} = 1)$$

General: $p_{\mathbf{m}}(\mathbf{e}) = \int p_{\mathbf{m}}(\mathbf{e}, \mathbf{H})\, d\mathbf{H}$

where $\mathbf{E} \subset \mathbf{X}, \quad \mathbf{H} = \mathbf{X} \setminus \mathbf{E}$

© fineartamerica.com

Marginal queries (MAR)

$\mathbf{q}_2$: *What is the probability that today is a Monday ~~at 12.00~~ and there is a traffic jam ~~only~~ on Alma Str.?*

$$\mathbf{q}_2(\mathbf{m}) = p_{\mathbf{m}}(\mathsf{Day} = \mathsf{Mon}, \mathsf{Jam}_{\mathsf{Alma}} = 1)$$

General: $p_{\mathbf{m}}(\mathbf{e}) = \int p_{\mathbf{m}}(\mathbf{e}, \mathbf{H})\, d\mathbf{H}$
and if you can answer MAR queries,
then you can also do ***conditional queries*** (CON):

$$p_{\mathbf{m}}(\mathbf{q} \mid \mathbf{e}) = \frac{p_{\mathbf{m}}(\mathbf{q}, \mathbf{e})}{p_{\mathbf{m}}(\mathbf{e})}$$

© fineartamerica.com

Maximum A Posteriori (MAP)

aka Most Probable Explanation (MPE)

q_3: *Which combination of roads is most likely to be jammed on Monday at 9am?*

© fineartamerica.com

Maximum A Posteriori (MAP)

aka Most Probable Explanation (MPE)

$\mathbf{q}_3$: *Which combination of roads is most likely to be jammed on Monday at 9am?*

$$\mathbf{q}_3(\mathbf{m}) = \operatorname{argmax}_{\mathbf{j}} p_{\mathbf{m}}(\mathbf{j}_1, \mathbf{j}_2, \ldots \mid \mathsf{Day}=\mathsf{M}, \mathsf{Time}=9)$$

Maximum A Posteriori (MAP)

aka Most Probable Explanation (MPE)

$\mathbf{q}_3$: *Which combination of roads is most likely to be jammed on Monday at 9am?*

$$\mathbf{q}_3(\mathbf{m}) = \operatorname{argmax}_{\mathbf{j}} p_{\mathbf{m}}(\mathbf{j}_1, \mathbf{j}_2, \ldots \mid \text{Day}=\text{M}, \text{Time}=9)$$

General: $\operatorname{argmax}_{\mathbf{q}} p_{\mathbf{m}}(\mathbf{q} \mid \mathbf{e})$

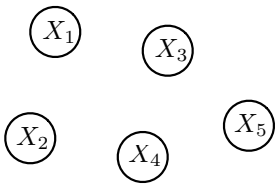
where $\mathbf{Q} \cup \mathbf{E} = \mathbf{X}$

Tractable Probabilistic Inference

A class of queries $\mathcal{Q}$ *is* tractable *on a family of probabilistic models* $\mathcal{M}$
iff for any query $\mathbf{q} \in \mathcal{Q}$ *and model* $\mathbf{m} \in \mathcal{M}$
exactly *computing* $\mathbf{q}(\mathbf{m})$ *runs in time* $O(\text{poly}(|\mathbf{m}|))$.

Fully factorized models

A completely disconnected graph. Example: Product of Bernoullis (PoBs)

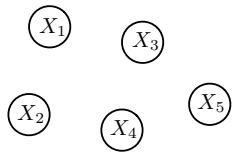


$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i)$$

Complete evidence, marginals and MAP, MMAP inference is ***linear***!

What do we lose?

Expressiveness: Ability to represent rich and complex classes of distributions

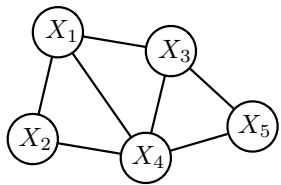


Fully factorized models cannot represent *all possible distributions*.

Probabilistic Graphical Models (PGMs)

Declarative semantics: a clean separation of modeling assumptions from inference

Nodes: random variables
Edges: dependencies



+

Inference:

- conditioning *[Darwiche 2001; Sang et al. 2005]*
- elimination *[Zhang et al. 1994; Dechter 1998]*
- message passing *[Yedidia et al. 2001; Dechter et al. 2002; Choi et al. 2010; Sontag et al. 2011]*

Complexity of MAR on PGMs

Exact complexity: Computing MAR is *#P-complete* *[Cooper 1990; Roth 1996]*

Complexity of MAR on PGMs

Exact complexity: Computing MAR is *#P-complete* *[Cooper 1990; Roth 1996]*

Fixed-parameter tractable: MAR on a graphical model with treewidth w takes time $O(|\mathbf{X}| \cdot 2^w)$, which is linear for fixed width w *[Dechter 1998; Koller et al. 2009].*

Complexity of MAR on PGMs

Exact complexity: Computing MAR is *#P-complete* *[Cooper 1990; Roth 1996]*

Fixed-parameter tractable: MAR on a graphical model with treewidth w takes time $O(|\mathbf{X}| \cdot 2^w)$, which is linear for fixed width w *[Dechter 1998; Koller et al. 2009].*

$\Rightarrow$ *what about bounding the treewidth by design?*

Complexity of MAR on PGMs

Exact complexity: Computing MAR is *#P-complete* *[Cooper 1990; Roth 1996]*

Fixed-parameter tractable: MAR on a graphical model with treewidth w takes time $O(|\mathbf{X}| \cdot 2^w)$, which is linear for fixed width w *[Dechter 1998; Koller et al. 2009].*

$\Rightarrow$ *what about bounding the treewidth by design?*

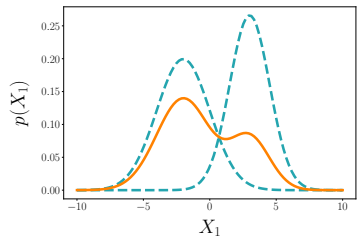
$\Rightarrow$ *Bounded-treewidth PGMs cannot represent all possible distributions.*

Summary so far...

	EVI	*MAR*	*MAP*	**expressive**
Fully-factorized	✔	✔	✔	✘
Bounded-treewidth PGMs	✔	✔	✔	✘
PGMs	✔*	✘	✘	✔

Mixtures

Mixtures as a convex combination of k (simpler) probabilistic models

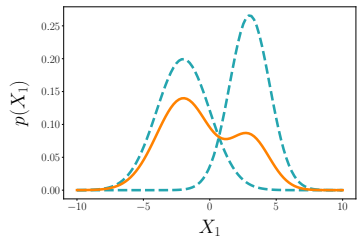


EVI, MAR, CON queries scale linearly in k

$$p(X) = w_1 \cdot p_1(X) + w_2 \cdot p_2(X)$$

Mixtures

Mixtures as a convex combination of k (simpler) probabilistic models



$$p(X) = w_1 \cdot p_1(X) + w_2 \cdot p_2(X)$$

EVI, MAR, CON queries scale linearly in k

...MAP is intractable!

$$\max_{\mathbf{q}} p_{\mathbf{m}}(\mathbf{q} \mid \mathbf{e}) = \max_{\mathbf{q}} \sum_i w_i p_i(\mathbf{q} \mid \mathbf{e})$$

$$\neq \sum_i w_i \max_{\mathbf{q}} p_i(\mathbf{q} \mid \mathbf{e})$$

Expressiveness and efficiency

Expressiveness: Ability to represent rich and effective classes of functions

$\Rightarrow$ *mixture of Gaussians can approximate any distribution!*

Cohen et al., "On the expressive power of deep learning: A tensor analysis", 2016
Martens et al., "On the Expressive Efficiency of Sum Product Networks", 2014

Expressiveness and efficiency

Expressiveness: Ability to represent rich and effective classes of functions

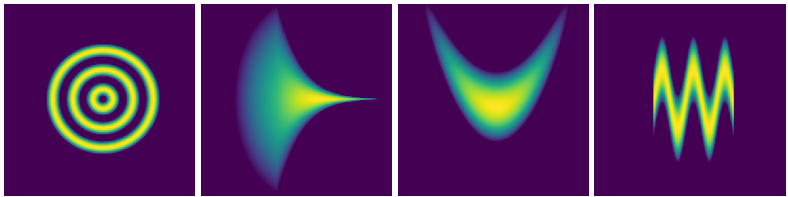
$\Rightarrow$ *mixture of Gaussians can approximate any distribution!*

Expressive efficiency (succinctness): Ability to represent rich and effective classes of functions **compactly**

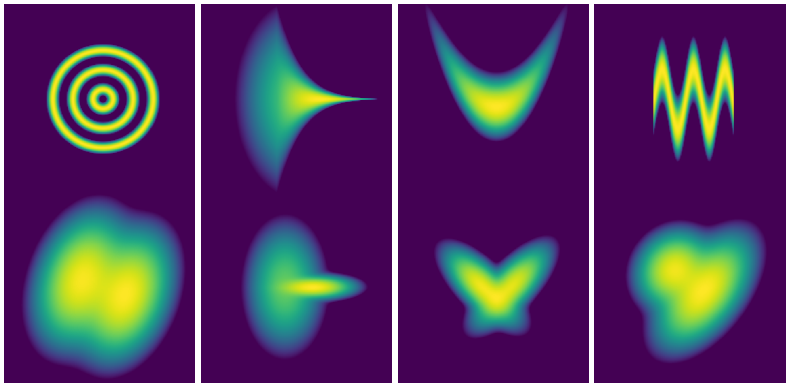
$\Rightarrow$ *but how many components does a Gaussian mixture need?*

Cohen et al., "On the expressive power of deep learning: A tensor analysis", 2016
Martens et al., "On the Expressive Efficiency of Sum Product Networks", 2014

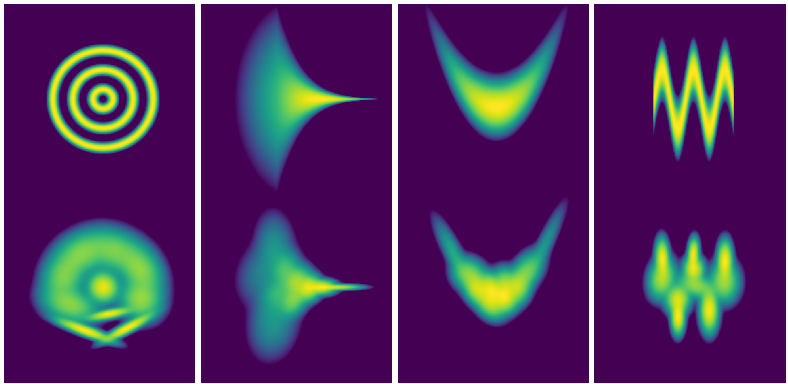
How expressive efficient are mixture?



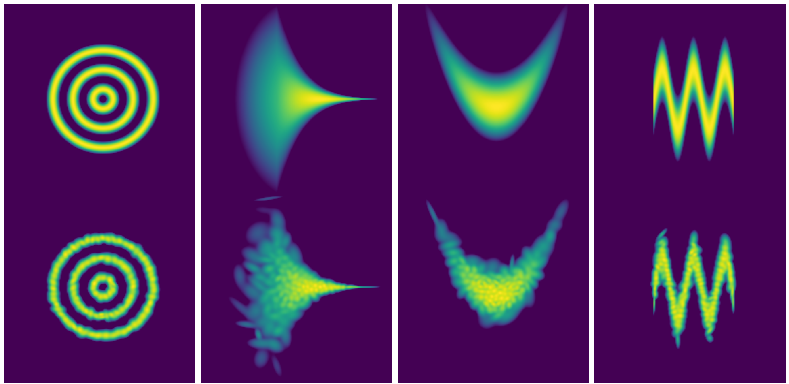
How expressive efficient are mixture?



How expressive efficient are mixture?

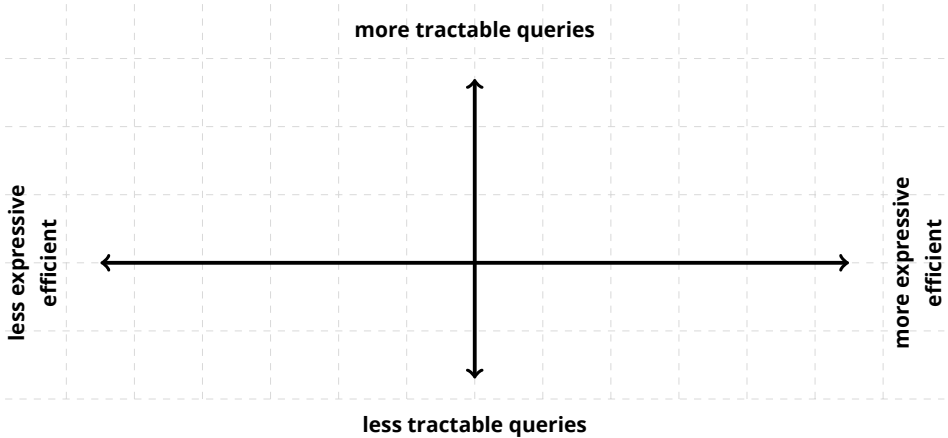


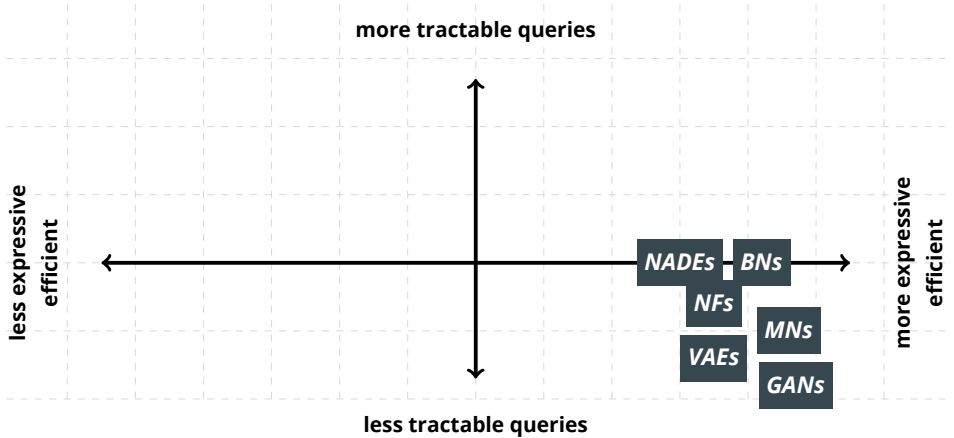
How expressive efficient are mixture?



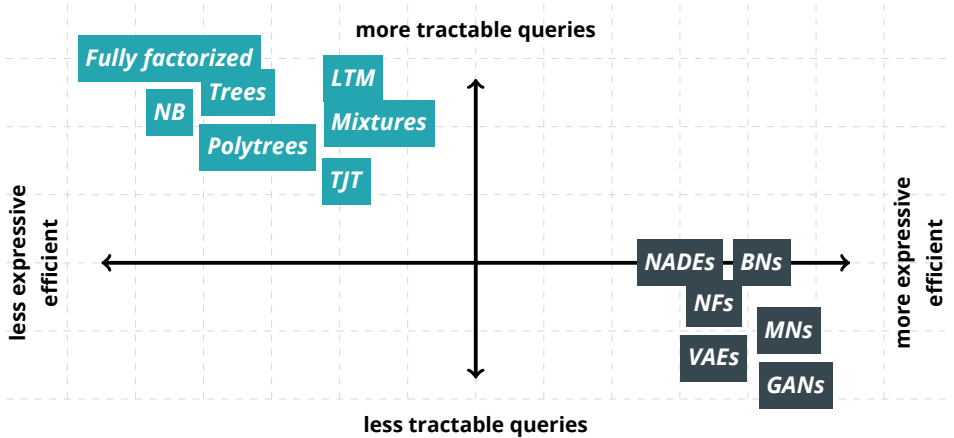
Summary so far...

	EVI	MAR	MAP	expressive	expr.efficient
Fully-factorized	✔	✔	✔	✘	✘
Bounded-treewidth PGMs	✔	✔	✔	✘	✘
PGMs	✔*	✘	✘	✔	✔
Mixtures	✔	✔	✘	✔	✘

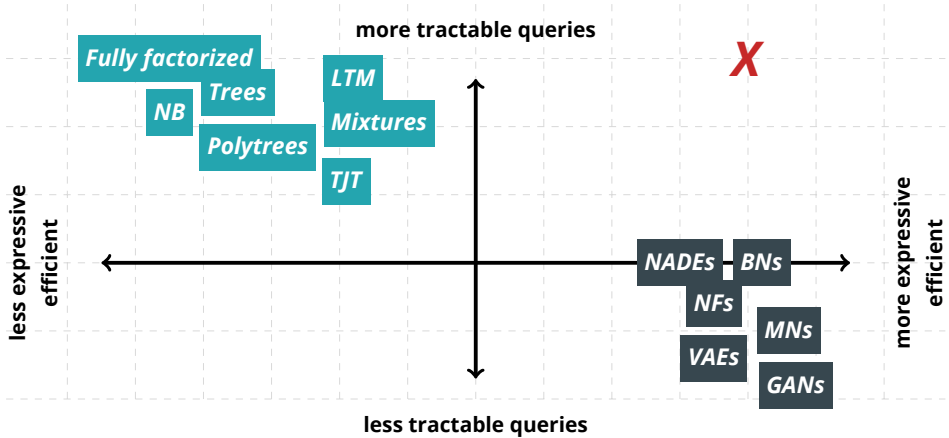

more tractable queries
less expressive
efficient
more expressive
efficient
less tractable queries



Expressive models are not very tractable...



and tractable ones are not very expressive...



probabilistic circuits are at the "sweet spot"

Probabilistic Circuits

Stay tuned for...

Next:

1. *What are the building blocks of probabilistic circuits?*
   $\Rightarrow$ *How to build a tractable computational graph?*
2. For which queries are probabilistic circuits tractable?
   $\Rightarrow$ *tractable classes induced by structural properties*

After: *How do you build a probabilistic circuit?*

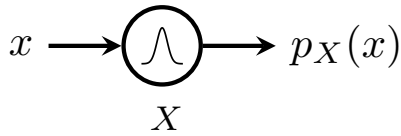
Base case: Univariate distributions

$\neg X$

A single node encoding a distribution

$\Rightarrow$ *e.g., indicators for* X *or* $\neg X$ *for Boolean random variable*

Base case: Univariate distributions

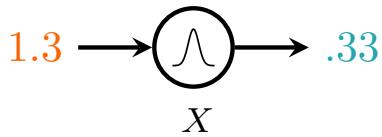


A single node encoding a distribution

$\Rightarrow$ *e.g., indicators for X or $\neg X$ for Boolean random variable*

$\Rightarrow$ *More generally, PDFs for continuous random variable*

Base case: Univariate distributions

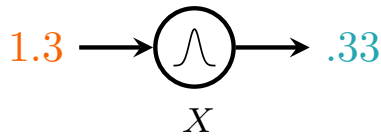


A single node encoding a distribution

$\Rightarrow$ *e.g., indicators for* X *or* $\neg X$ *for Boolean random variable*

$\Rightarrow$ *More generally, PDFs for continuous random variable*

Base case: Univariate distributions



A single node encoding a distribution

$\Rightarrow$ *e.g., indicators for* X *or* $\neg X$ *for Boolean random variable*

$\Rightarrow$ *More generally, PDFs for continuous random variable*

Assumption: tractable for EVI, MAR, MAP

Factorizations as product nodes

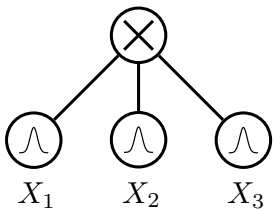
Divide and conquer complexity

$$p(X_1, X_2, X_3) = p(X_1) \cdot p(X_2) \cdot p(X_3)$$

Factorizations as product nodes

Divide and conquer complexity

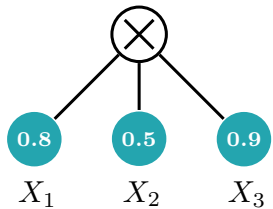
$$p(X_1, X_2, X_3) = p(X_1) \cdot p(X_2) \cdot p(X_3)$$



Factorizations as product nodes

Divide and conquer complexity

$$p(x_1, x_2, x_3) = p(x_1) \cdot p(x_2) \cdot p(x_3)$$

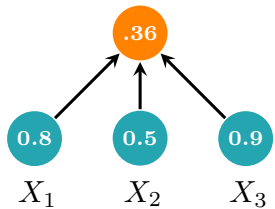


$\Rightarrow$ *feedforward evaluation*

Factorizations as product nodes

Divide and conquer complexity

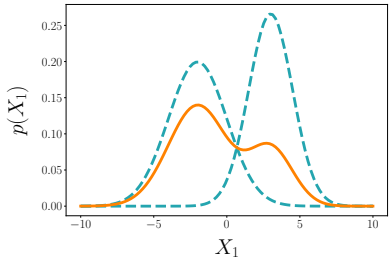
$$p(x_1, x_2, x_3) = p(x_1) \cdot p(x_2) \cdot p(x_3)$$



$\Rightarrow$ *feedforward evaluation*

Mixtures as sum nodes

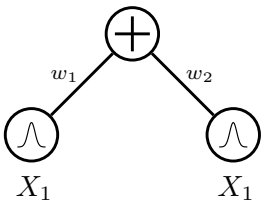
Enhance expressiveness



$$p(X) = w_1 \cdot p_1(X) + w_2 \cdot p_2(X)$$

Mixtures as sum nodes

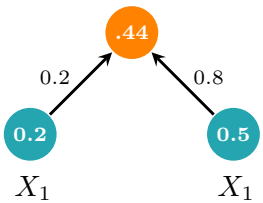
Enhance expressiveness



$$p(x) = 0.2 \cdot p_1(x) + 0.8 \cdot p_2(x)$$

Mixtures as sum nodes

Enhance expressiveness



$$p(x) = 0.2 \cdot p_1(x) + 0.8 \cdot p_2(x)$$

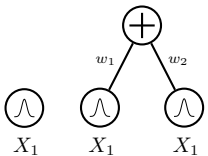
$\Rightarrow$ *by **stacking** them we increase expressive efficiency*

A grammar for tractable models

Recursive semantics of probabilistic circuits

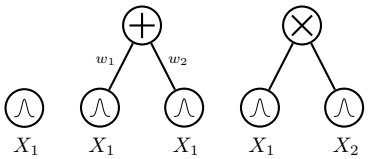
A grammar for tractable models

Recursive semantics of probabilistic circuits



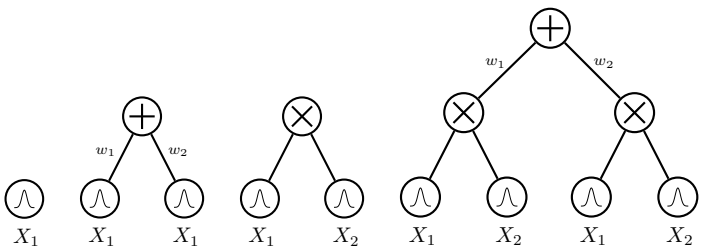
A grammar for tractable models

Recursive semantics of probabilistic circuits



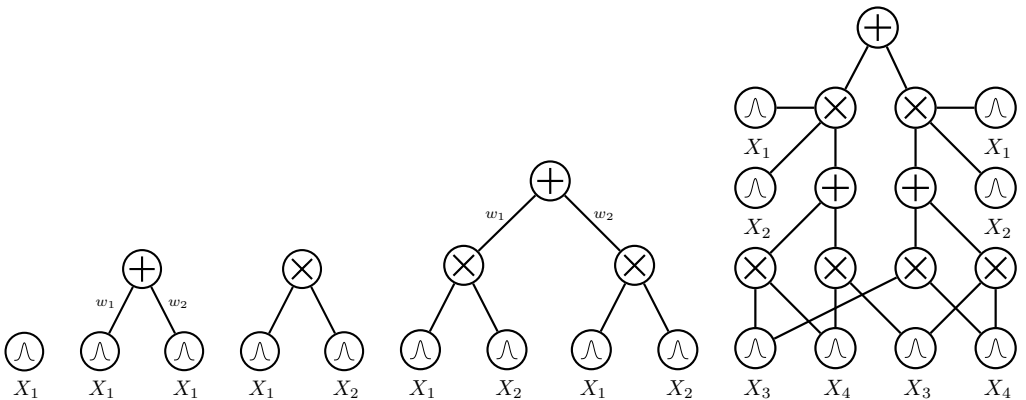
A grammar for tractable models

Recursive semantics of probabilistic circuits



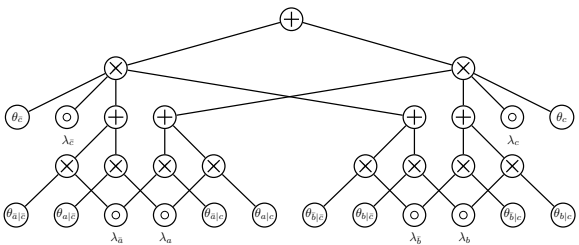
A grammar for tractable models

Recursive semantics of probabilistic circuits

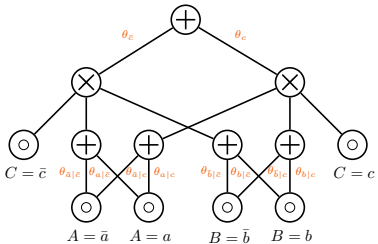


Logical circuits

connection to probabilistic circuits through WMC



Compiled circuit of WMC encoding



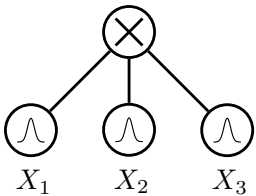
Equivalent probabilistic circuit

Which structural constraints to ensure tractability?

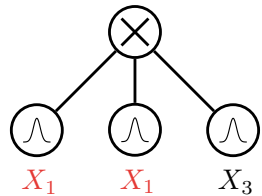
Decomposability

A product node is decomposable if its children depend on disjoint sets of variables

⇒ *just like in factorization!*



decomposable circuit



***non*-decomposable** circuit

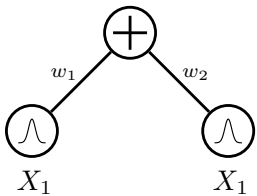
Darwiche et al., "A knowledge compilation map", 2002

Smoothness

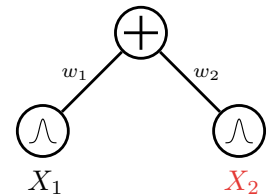
aka completeness

A sum node is smooth if its children depend of the same variable sets

$\Rightarrow$ *otherwise not accounting for some variables*



smooth circuit



***non*-smooth** circuit

Darwiche et al., "A knowledge compilation map", 2002

Smoothness + ***decomposability*** = ***tractable MAR***

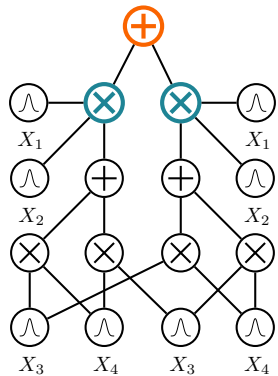
Smoothness + **decomposability** = **tractable MAR**

If $\boldsymbol{p}(\mathbf{x}) = \sum_i w_i \boldsymbol{p}_i(\mathbf{x})$, (***smoothness***):

$$\int \boldsymbol{p}(\mathbf{x}) d\mathbf{x} = \int \sum_i w_i \boldsymbol{p}_i(\mathbf{x}) d\mathbf{x} =$$

$$= \sum_i w_i \int \boldsymbol{p}_i(\mathbf{x}) d\mathbf{x}$$

$\Rightarrow$ *integrals are "pushed down" to children*

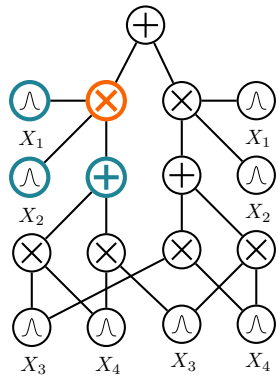


Smoothness + **decomposability** = **tractable MAR**

If $p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{x})p(\mathbf{y})p(\mathbf{z})$, (***decomposability***):

$$\int\int\int p(\mathbf{x}, \mathbf{y}, \mathbf{z}) d\mathbf{x} d\mathbf{y} d\mathbf{z} =$$
$$= \int\int\int p(\mathbf{x})p(\mathbf{y})p(\mathbf{z}) d\mathbf{x} d\mathbf{y} d\mathbf{z} =$$
$$= \int p(\mathbf{x}) d\mathbf{x} \int p(\mathbf{y}) d\mathbf{y} \int p(\mathbf{z}) d\mathbf{z}$$

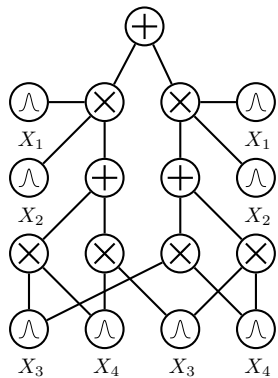
$\Longrightarrow$ *integrals decompose into easier ones*



Smoothness + **decomposability** = **tractable MAR**

To compute $p(x_2, x_4)$:

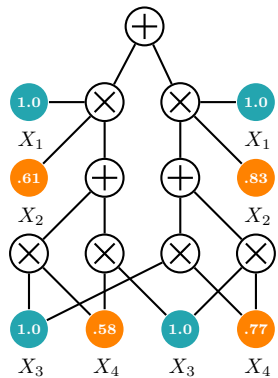
- leaves over X_1 and X_3 output $Z_i = \int p(x_i) dx_i$
  $\Longrightarrow$ *for normalized leaf distributions:* 1.0
- leaves over X_2 and X_4 output *EVI*
- feedforward evaluation (bottom-up)



Smoothness + **decomposability** = **tractable MAR**

To compute $p(x_2, x_4)$:

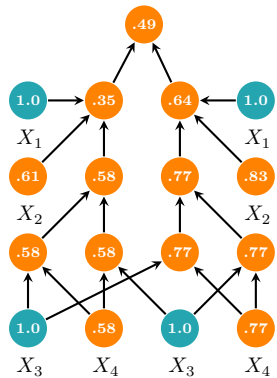
- leaves over X_1 and X_3 output $Z_i = \int p(x_i) dx_i$
  $\Longrightarrow$ *for normalized leaf distributions:* 1.0
- leaves over X_2 and X_4 output **EVI**
- feedforward evaluation (bottom-up)



Smoothness + **decomposability** = **tractable MAR**

To compute $p(x_2, x_4)$:

- leaves over X_1 and X_3 output
  $Z_i = \int p(x_i) dx_i$
  $\Rightarrow$ *for normalized leaf distributions:* 1.0
- leaves over X_2 and X_4 output **EVI**
- feedforward evaluation (bottom-up)



Smoothness **+** ***decomposability*** **=** ***tractable MAR***

*Smoothness and decomposability are **sufficient** conditions for a circuit to compute marginals.*

Smoothness **+** ***decomposability*** **=** ***tractable MAR***

Smoothness and decomposability are ***necessary*** *and* ***sufficient*** *conditions for a circuit to compute marginals.*

- Non-smooth node $\Rightarrow$ a variable is unaccounted for $\Rightarrow$ lower-bounds the marginal
- Non-decomposable node $\Rightarrow$ integral does not decompose.

Smoothness **+** ***decomposability*** **=** ***tractable MAR***

Smoothness and decomposability are ***necessary*** *and* ***sufficient*** *conditions for a circuit to compute marginals.*

- Non-smooth node $\Rightarrow$ a variable is unaccounted for $\Rightarrow$ lower-bounds the marginal
- Non-decomposable node $\Rightarrow$ integral does not decompose.

Smoothness + **decomposability** = **tractable MAR**

*Smoothness and decomposability are **necessary** and **sufficient** conditions for a circuit to compute marginals.*

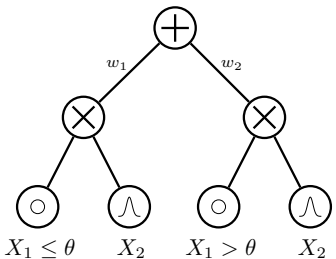
- Non-smooth node $\Rightarrow$ a variable is unaccounted for $\Rightarrow$ lower-bounds the marginal
- Non-decomposable node $\Rightarrow$ integral does not decompose.

Determinism

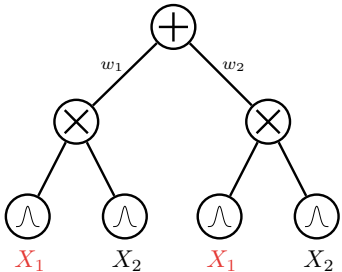
aka selectivity

A sum node is deterministic if the output of only one children is non zero for any input

$\Rightarrow$ *e.g. if their distributions have disjoint support*



deterministic circuit

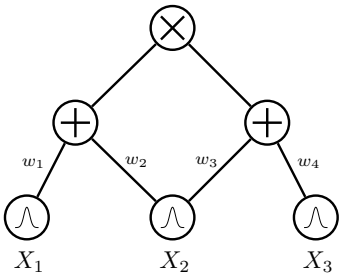


***non*-deterministic circuit**

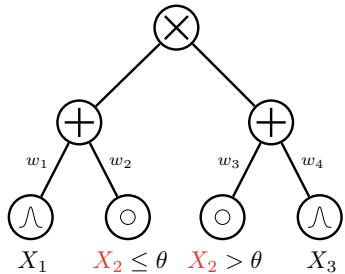
Consistency

A product node is consistent if any variable shared between its children appears in a single leaf node

$\Rightarrow$ *decomposability implies consistency*



consistent circuit



inconsistent circuit

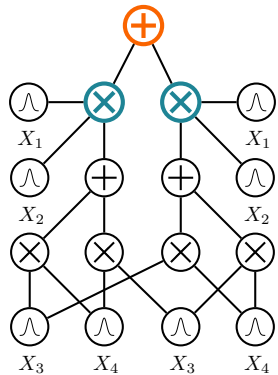
Determinism **+** ***consistency*** **=** ***tractable MAP***

Determinism + **consistency** = **tractable MAP**

If $\boldsymbol{p}(\mathbf{q}, \mathbf{e}) = \sum_i w_i \boldsymbol{p}_i(\mathbf{q}, \mathbf{e}) = \max_i w_i \boldsymbol{p}_i(\mathbf{q}, \mathbf{e})$, (***determinism***):

$$\begin{aligned}\max_{\mathbf{q}} \boldsymbol{p}(\mathbf{q}, \mathbf{e}) &= \max_{\mathbf{q}} \sum_i w_i \boldsymbol{p}_i(\mathbf{q}, \mathbf{e}) \\ &= \max_{\mathbf{q}} \max_i w_i \boldsymbol{p}_i(\mathbf{q}, \mathbf{e}) \\ &= \max_i \max_{\mathbf{q}} w_i \boldsymbol{p}_i(\mathbf{q}, \mathbf{e})\end{aligned}$$

$\Rightarrow$ *one non-zero child term, thus sum is max*

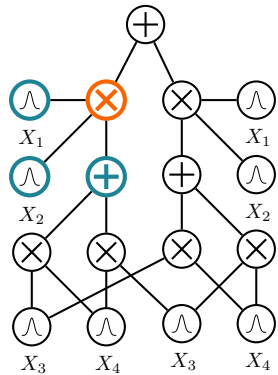


Determinism + **consistency** = **tractable MAP**

If $\max_{\mathbf{q}_{\text{shared}}} p(\mathbf{q}, \mathbf{e}) = \max_{\mathbf{q}_{\text{shared}}} p(\mathbf{q}_{\mathbf{x}}, \mathbf{e}_{\mathbf{x}}) \cdot \max_{\mathbf{q}_{\text{shared}}} p(\mathbf{q}_{\mathbf{y}}, \mathbf{e}_{\mathbf{y}})$ (***consistent***):

$$\max_{\mathbf{q}} p(\mathbf{q}, \mathbf{e}) = \max_{\mathbf{q}_{\mathbf{x}}, \mathbf{q}_{\mathbf{y}}} p(\mathbf{q}_{\mathbf{x}}, \mathbf{e}_{\mathbf{x}}, \mathbf{q}_{\mathbf{y}}, \mathbf{e}_{\mathbf{y}})$$
$$= \max_{\mathbf{q}_{\mathbf{x}}} p(\mathbf{q}_{\mathbf{x}}, \mathbf{e}_{\mathbf{x}}) \cdot \max_{\mathbf{q}_{\mathbf{y}}} p(\mathbf{q}_{\mathbf{y}}, \mathbf{e}_{\mathbf{y}})$$

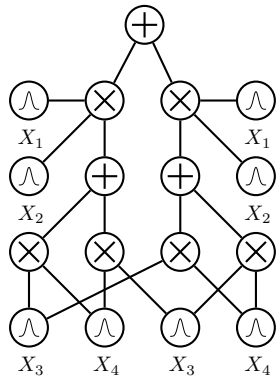
$\Rightarrow$ *solving optimization independently*



Determinism + **consistency** = **tractable MAP**

To compute $\max_{x_1,x_3} p(x_1, x_2, x_3, x_4)$:

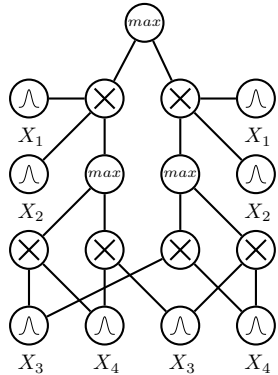
- turn sum into max nodes
- leaves over X_1 and X_3 output $M_i = \max p(x_i)$
- leaves over X_2 and X_4 output EVI
- feedforward evaluation (bottom-up)



Determinism + **consistency** = **tractable MAP**

To compute $\max_{x_1,x_3} p(x_1, x_2, x_3, x_4)$:

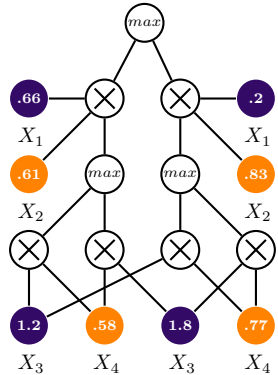
- turn sum into max nodes
- leaves over X_1 and X_3 output $M_i = \max p(x_i)$
- leaves over X_2 and X_4 output EVI
- feedforward evaluation (bottom-up)



Determinism + **consistency** = **tractable MAP**

To compute $\max_{x_1,x_3} p(x_1, x_2, x_3, x_4)$:

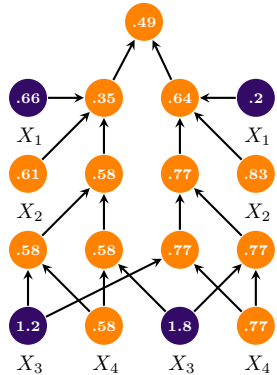
- turn sum into max nodes
- leaves over X_1 and X_3 output $\boldsymbol{M}_i = \max p(x_i)$
- leaves over X_2 and X_4 output **EVI**
- feedforward evaluation (bottom-up)



Determinism + **consistency** = **tractable MAP**

To compute $\max_{x_1,x_3} p(x_1, x_2, x_3, x_4)$:

- turn sum into max nodes
- leaves over X_1 and X_3 output $\boldsymbol{M}_i = \max p(x_i)$
- leaves over X_2 and X_4 output **EVI**
- feedforward evaluation (bottom-up)



Determinism **+** ***consistency*** **=** ***tractable MAP***

*Determinism and consistency are **sufficient** conditions for a circuit to compute MAP.*

Determinism + **consistency** = **tractable MAP**

*Determinism and consistency are **necessary** and **sufficient** conditions for a circuit to compute MAP.*

- Non-deterministic node $\Rightarrow$ cannot maximize correctly without summations.
- Inconsistent node $\Rightarrow$ MAP assignments of children conflict with each other.

Determinism **+** ***consistency*** **=** ***tractable MAP***

*Determinism and consistency are **necessary** and **sufficient** conditions for a circuit to compute MAP.*

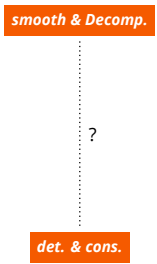
- Non-deterministic node $\Rightarrow$ cannot maximize correctly without summations.
- Inconsistent node $\Rightarrow$ MAP assignments of children conflict with each other.

Determinism **+** ***consistency*** **=** ***tractable MAP***

*Determinism and consistency are **necessary** and **sufficient** conditions for a circuit to compute MAP.*

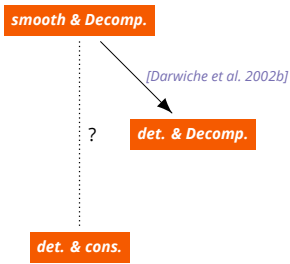
- Non-deterministic node $\Rightarrow$ cannot maximize correctly without summations.
- Inconsistent node $\Rightarrow$ MAP assignments of children conflict with each other.

Succinctness of circuits



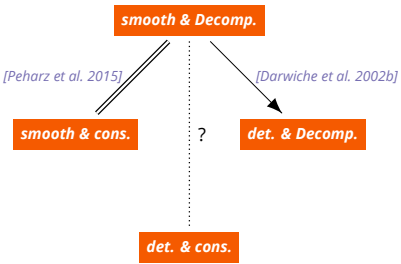
Are smooth & decomposable circuits as succinct as deterministic & consistent ones, or vice versa?

Succinctness of circuits

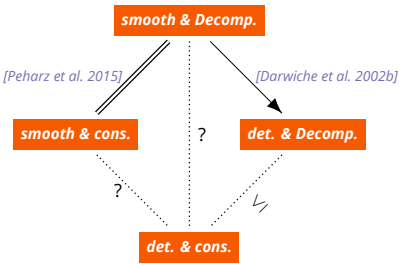


: strictly more succinct

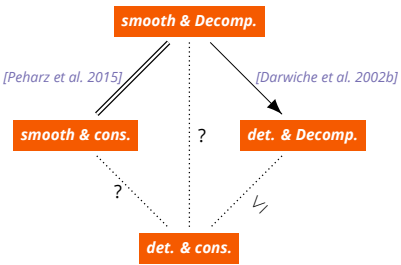
Succinctness of circuits



Succinctness of circuits



Succinctness of circuits

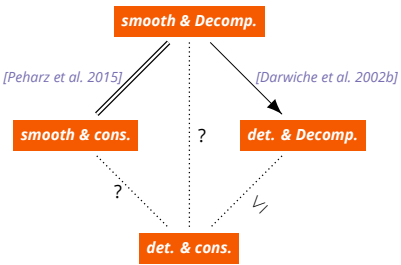


Consider following circuit over Boolean variables:

$$\prod_i^r (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in \mathbf{X}$$

- Size linear in the number of variables
- Deterministic and consistent
- Marginal (with no evidence) is the solution to #P-hard SAT′ problem *[Valiant 1979]* $\Rightarrow$ **no tractable circuit for marginals!**

Succinctness of circuits

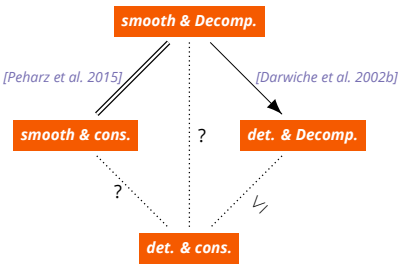


Consider following circuit over Boolean variables:

$$\prod_i^r (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in \mathbf{X}$$

- Size linear in the number of variables
- Deterministic and consistent
- Marginal (with no evidence) is the solution to #P-hard SAT′ problem *[Valiant 1979]* $\Rightarrow$ **no tractable circuit for marginals!**

Succinctness of circuits

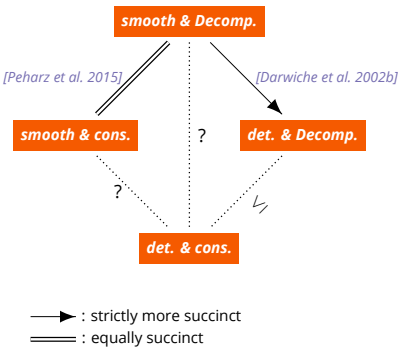


Consider following circuit over Boolean variables:

$$\prod_i^r (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in \mathbf{X}$$

- Size linear in the number of variables
- Deterministic and consistent
- Marginal (with no evidence) is the solution to #P-hard SAT′ problem *[Valiant 1979]* $\Rightarrow$ **no tractable circuit for marginals!**

Succinctness of circuits

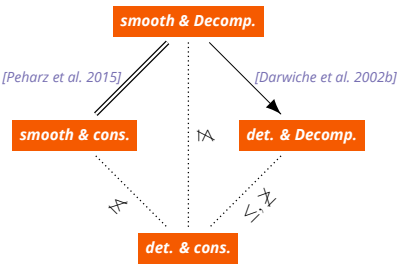


Consider following circuit over Boolean variables:

$$\prod_i^r (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in \mathbf{X}$$

- Size linear in the number of variables
- Deterministic and consistent
- Marginal (with no evidence) is the solution to #P-hard SAT′ problem *[Valiant 1979]* $\Rightarrow$ **no tractable circuit for marginals!**

Succinctness of circuits

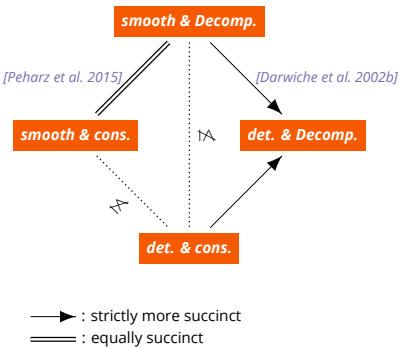


Consider following circuit over Boolean variables:

$$\prod_i^r (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in \mathbf{X}$$

- Size linear in the number of variables
- Deterministic and consistent
- Marginal (with no evidence) is the solution to #P-hard SAT′ problem *[Valiant 1979]* $\Rightarrow$ **no tractable circuit for marginals!**

Succinctness of circuits

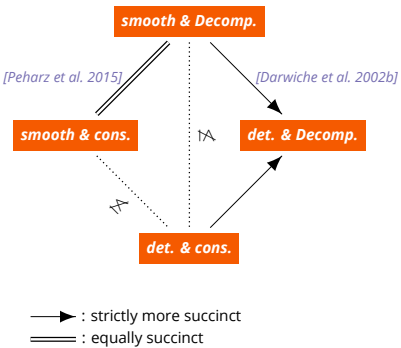


Consider following circuit over Boolean variables:

$$\prod_i^r (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in \mathbf{X}$$

- Size linear in the number of variables
- Deterministic and consistent
- Marginal (with no evidence) is the solution to #P-hard SAT′ problem *[Valiant 1979]* $\Rightarrow$ **no tractable circuit for marginals!**

Succinctness of circuits



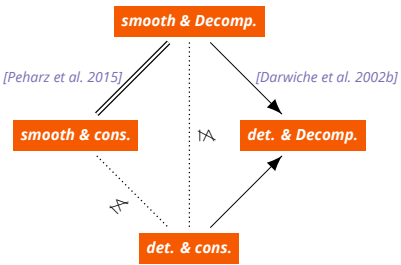
Consider following circuit over Boolean variables:

$$\prod_i^r (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in \mathbf{X}$$

- Size linear in the number of variables
- Deterministic and consistent
- Marginal (with no evidence) is the solution to #P-hard SAT′ problem *[Valiant 1979]* $\Rightarrow$ **no tractable circuit for marginals!**

$\Rightarrow$ *implies hardness of smoothing consistent circuits!*

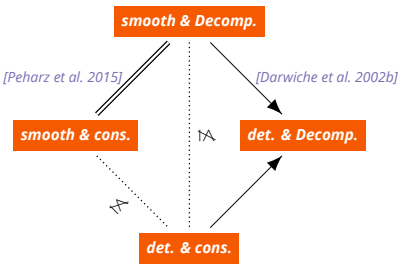
Succinctness of circuits



Consider the marginal distribution $p(\mathbf{X})$ from a naive Bayes distribution $p(\mathbf{X}, C)$:

- Linear-size smooth and decomposable circuit
- MAP of $p(\mathbf{X})$ solves marginal MAP of $p(\mathbf{X}, C)$ which is NP-hard *[Campos 2011]* $\Rightarrow$ **no tractable circuit for MAP!**

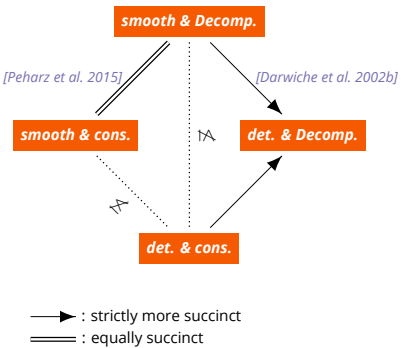
Succinctness of circuits



Consider the marginal distribution $p(\mathbf{X})$ from a naive Bayes distribution $p(\mathbf{X}, C)$:

- Linear-size smooth and decomposable circuit
- MAP of $p(\mathbf{X})$ solves marginal MAP of $p(\mathbf{X}, C)$ which is NP-hard *[Campos 2011]* $\Rightarrow$ **no tractable circuit for MAP!**

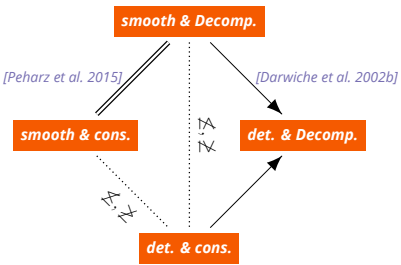
Succinctness of circuits



Consider the marginal distribution $p(\mathbf{X})$ from a naive Bayes distribution $p(\mathbf{X}, C)$:

- Linear-size smooth and decomposable circuit
- MAP of $p(\mathbf{X})$ solves marginal MAP of $p(\mathbf{X}, C)$ which is NP-hard *[Campos 2011]* $\Rightarrow$ **no tractable circuit for MAP!**

Succinctness of circuits



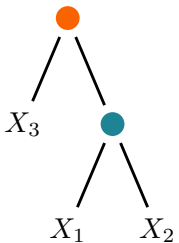
Consider the marginal distribution $p(\mathbf{X})$ from a naive Bayes distribution $p(\mathbf{X}, C)$:

- Linear-size smooth and decomposable circuit
- MAP of $p(\mathbf{X})$ solves marginal MAP of $p(\mathbf{X}, C)$ which is NP-hard *[Campos 2011]* $\Rightarrow$ **no tractable circuit for MAP!**

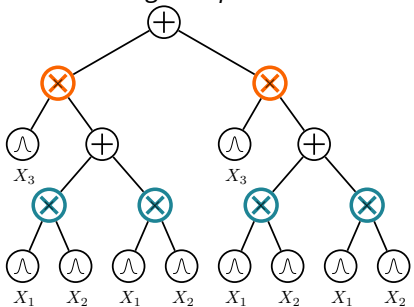
Structured decomposability

A product node is structured decomposable if decomposes according to a node in a ***vtree***

$\Rightarrow$ *stronger requirement than decomposability*



vtree

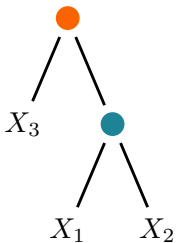


structured decomposable circuit

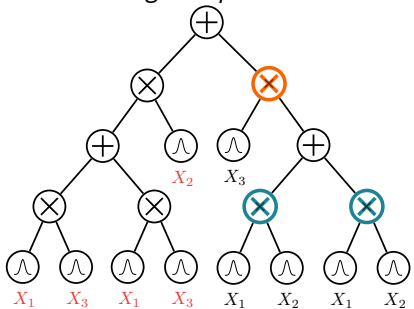
Structured decomposability

A product node is structured decomposable if decomposes according to a node in a ***vtree***

$\Rightarrow$ *stronger requirement than decomposability*



vtree



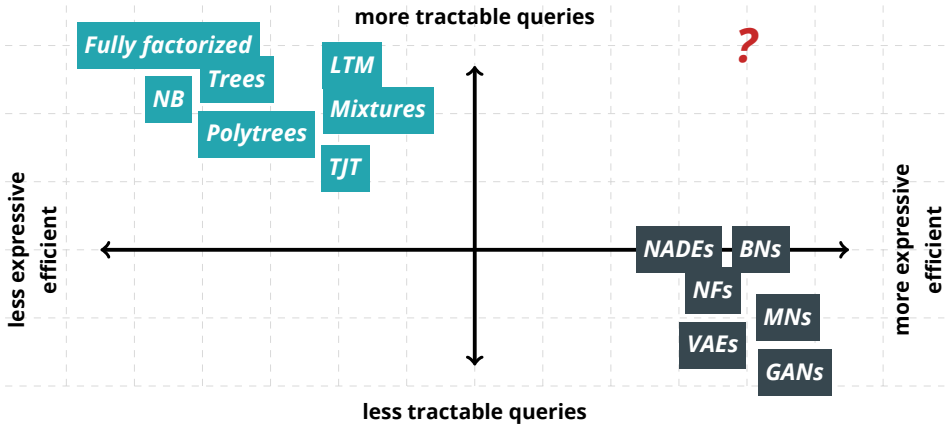
non structured decomposable circuit

structured decomposability = ***tractable…***

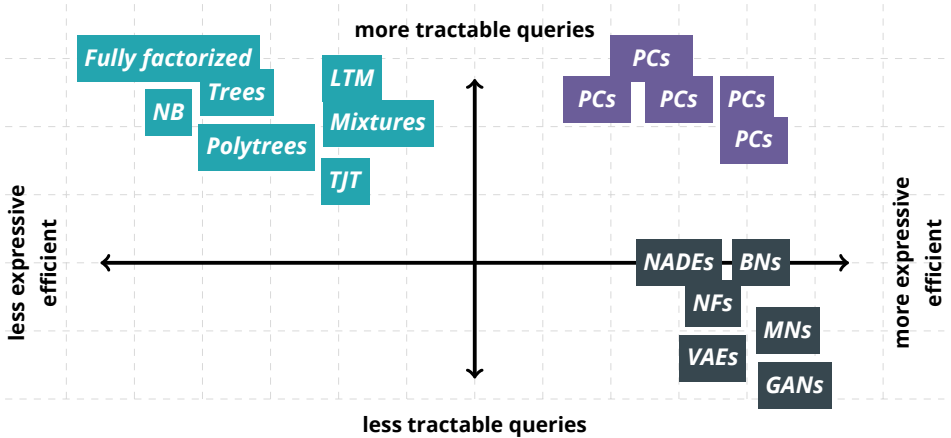
- ***Symmetric*** and ***group queries*** (exactly-k, odd-number, etc.) *[Bekker et al. 2015]*

For the "right" vtree

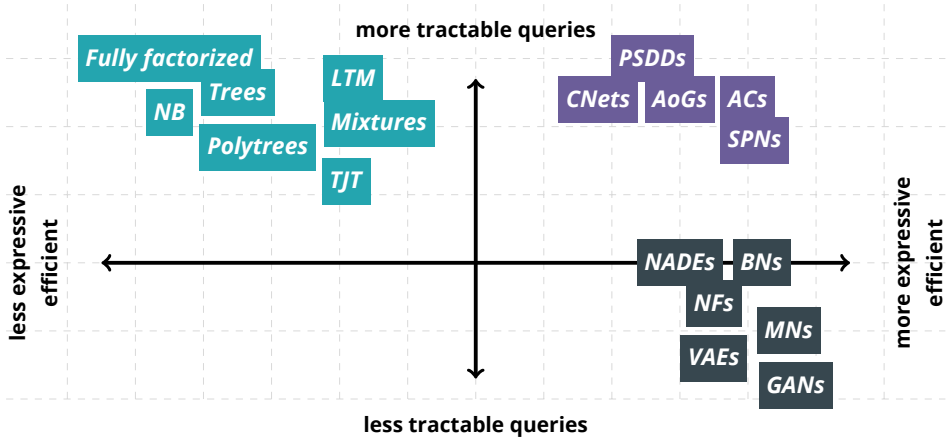
- ***Marginal MAP*** queries
- Probability of logical circuit event in probabilistic circuit *[Choi et al. 2015]*
- ***Multiply*** two probabilistic circuits *[Shen et al. 2016]*
- ***KL Divergence*** between probabilistic circuits *[Liang et al. 2017b]*
- ***Same-decision probability*** *[Oztok et al. 2016]*
- ***Expected same-decision probability*** *[Choi et al. 2017]*
- ***Expected classifier agreement*** *[Choi et al. 2018]*
- ***Expected predictions*** *[Khosravi et al. 2019]*



where are probabilistic circuits?



tractability vs expressive efficiency



tractability vs expressive efficiency

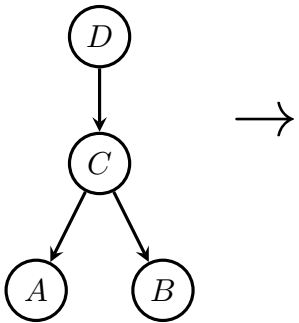
Smooth $\vee$ ***decomposable*** $\vee$ ***deterministic***
$\vee$ ***structured decomposable*** ***PCs*?**

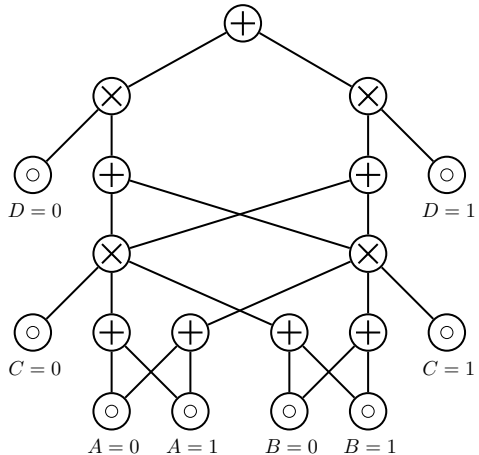
	smooth	***dec.***	**det.**	**str.dec.**
Arithmetic Circuits (ACs) *[Darwiche 2003]*	✔	✔	✔	✘
Sum-Product Networks (SPNs) *[Poon et al. 2011]*	✔	✔	✘	✘
Cutset Networks (CNets) *[Rahman et al. 2014]*	✔	✔	✔	✘
PSDDs *[Kisa et al. 2014a]*	✔	✔	✔	✔
AndOrGraphs *[Dechter et al. 2007]*	✔	✔	✔	✔

Building circuits

Compiling PGMs to probabilistic circuits

Example: from BN trees to circuits





Learning probabilistic circuits

	Parameters	Structure
Generative	***deterministic*** closed-form MLE *[Kisa et al. 2014b; Peharz et al. 2014]* ***non-deterministic*** EM *[Poon et al. 2011; Peharz 2015; Zhao et al. 2016a]* SGD *[Sharir et al. 2016; Peharz et al. 2019]* Bayesian *[Jaini et al. 2016; Rashwan et al. 2016]* *[Zhao et al. 2016b; Trapp et al. 2019; Vergari et al. 2019]*	***greedy*** top-down *[Gens et al. 2013; Rooshenas et al. 2014]* *[Rahman et al. 2014; Vergari et al. 2015]* bottom-up *[Peharz et al. 2013]* ***hill climbing*** *[Lowd et al. 2008, 2013; Peharz et al. 2014]* *[Dennis et al. 2015; Liang et al. 2017a]* ***random*** RAT-SPNs *[Peharz et al. 2019]* XCNet *[Di Mauro et al. 2017]*
Discriminative	***deterministic*** convex-opt MLE *[Liang et al. 2019]* ***non-deterministic*** EM *[Rashwan et al. 2018]* SGD *[Gens et al. 2012; Sharir et al. 2016]* *[Peharz et al. 2019]*	***greedy*** top-down *[Shao et al. 2019]* ***hill climbing*** *[Rooshenas et al. 2016]*

How expressive are probabilistic circuits?

density estimation benchmarks

dataset	best circuit	BN	MADE	VAE	dataset	best circuit	BN	MADE	VAE
nltcs	**-5.99**	-6.02	-6.04	**-5.99**	***dna***	**-79.88**	-80.65	-82.77	-94.56
msnbc	**-6.04**	**-6.04**	-6.06	-6.09	***kosarek***	**-10.52**	-10.83	-	-10.64
kdd	-2.12	-2.19	**-2.07**	-2.12	***msweb***	-9.62	-9.70	**-9.59**	-9.73
plants	**-11.84**	-12.65	-12.32	-12.34	***book***	-33.82	-36.41	-33.95	**-33.19**
audio	-39.39	-40.50	-38.95	**-38.67**	***movie***	-50.34	-54.37	-48.7	**-47.43**
jester	-51.29	**-51.07**	-52.23	-51.54	***webkb***	-149.20	-157.43	-149.59	**-146.9**
netflix	-55.71	-57.02	-55.16	**-54.73**	***cr52***	-81.87	-87.56	-82.80	**-81.33**
accidents	-26.89	**-26.32**	-26.42	-29.11	***c20ng***	-151.02	-158.95	-153.18	**-146.9**
retail	**-10.72**	-10.87	-10.81	-10.83	***bbc***	**-229.21**	-257.86	-242.40	-240.94
pumbs*	-22.15	**-21.72**	-22.3	-25.16	***ad***	-14.00	-18.35	**-13.65**	-18.81

Conclusions

- You can be both tractable and expressive – ***probabilistic circuits!***
- Exploit connections to logical circuits.
- Many interesting probabilistic queries – necessary and sufficient conditions?

Juice.jl a library for advanced logical and probabilistic inference with circuits in Julia ***SOON!***

References I

- Valiant, Leslie G (1979). "The complexity of enumeration and reliability problems". In: *SIAM Journal on Computing* 8.3, pp. 410–421.
- Cooper, Gregory F (1990). "The computational complexity of probabilistic inference using Bayesian belief networks". In: *Artificial intelligence* 42.2-3, pp. 393–405.
- Zhang, Nevin Lianwen and David Poole (1994). "A simple approach to Bayesian network computations". In: *Proceedings of the Biennial Conference-Canadian Society for Computational Studies of Intelligence*, pp. 171–178.
- Roth, Dan (1996). "On the hardness of approximate reasoning". In: *Artificial Intelligence* 82.1–2, pp. 273–302.
- Dechter, Rina (1998). "Bucket elimination: A unifying framework for probabilistic inference". In: *Learning in graphical models*. Springer, pp. 75–104.
- Darwiche, Adnan (2001). "Recursive conditioning". In: *Artificial Intelligence* 126.1-2, pp. 5–41.
- Yedidia, Jonathan S, William T Freeman, and Yair Weiss (2001). "Generalized belief propagation". In: *Advances in neural information processing systems*, pp. 689–695.
- Darwiche, Adnan and Pierre Marquis (2002a). "A knowledge compilation map". In: *Journal of Artificial Intelligence Research* 17, pp. 229–264.
- — (2002b). "A knowledge compilation map". In: *Journal of Artificial Intelligence Research* 17.1, pp. 229–264.
- Dechter, Rina, Kalev Kask, and Robert Mateescu (2002). "Iterative join-graph propagation". In: *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 128–136.
- Darwiche, Adnan (2003). "A Differential Approach to Inference in Bayesian Networks". In: *J.ACM*.

References II

- Sang, Tian, Paul Beame, and Henry A Kautz (2005). "Performing Bayesian inference by weighted model counting". In: *AAAI*. Vol. 5, pp. 475–481.
- Dechter, Rina and Robert Mateescu (2007). "AND/OR search spaces for graphical models". In: *Artificial intelligence* 171.2-3, pp. 73–106.
- Lowd, Daniel and Pedro Domingos (2008). "Learning Arithmetic Circuits". In: *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*. UAI'08. Helsinki, Finland: AUAI Press, pp. 383–392. ISBN: 0-9749039-4-9. URL: `http://dl.acm.org/citation.cfm?id=3023476.3023522`.
- Koller, Daphne and Nir Friedman (2009). ***Probabilistic Graphical Models: Principles and Techniques***. MIT Press.
- Choi, Arthur and Adnan Darwiche (2010). "Relax, compensate and then recover". In: *JSAI International Symposium on Artificial Intelligence*. Springer, pp. 167–180.
- Campos, Cassio P de (2011). "New complexity results for MAP in Bayesian networks". In: *IJCAI*. Vol. 11, pp. 2100–2106.
- Poon, Hoifung and Pedro Domingos (2011). "Sum-Product Networks: a New Deep Architecture". In: *UAI 2011*.
- Sontag, David, Amir Globerson, and Tommi Jaakkola (2011). "Introduction to dual decomposition for inference". In: *Optimization for Machine Learning* 1, pp. 219–254.
- Gens, Robert and Pedro Domingos (2012). "Discriminative Learning of Sum-Product Networks". In: *Advances in Neural Information Processing Systems 25*, pp. 3239–3247.
- — (2013). "Learning the Structure of Sum-Product Networks". In: *Proceedings of the ICML 2013*, pp. 873–880.
- Lowd, Daniel and Amirmohammad Rooshenas (2013). "Learning Markov Networks With Arithmetic Circuits". In: *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*. Vol. 31. JMLR Workshop Proceedings, pp. 406–414.
- Peharz, Robert, Bernhard Geiger, and Franz Pernkopf (2013). "Greedy Part-Wise Learning of Sum-Product Networks". In: *ECML-PKDD 2013*.

References III

- Kisa, Doga et al. (July 2014a). "Probabilistic sentential decision diagrams". In: *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Vienna, Austria.
- — (July 2014b). "Probabilistic sentential decision diagrams". In: *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Vienna, Austria. URL: `http://starai.cs.ucla.edu/papers/KisaKR14.pdf`.
- Martens, James and Venkatesh Medabalimi (2014). "On the Expressive Efficiency of Sum Product Networks". In: *CoRR* abs/1411.7717.
- Peharz, Robert, Robert Gens, and Pedro Domingos (2014). "Learning Selective Sum-Product Networks". In: *Workshop on Learning Tractable Probabilistic Models*. LTPM.
- Rahman, Tahrima, Prasanna Kothalkar, and Vibhav Gogate (2014). "Cutset Networks: A Simple, Tractable, and Scalable Approach for Improving the Accuracy of Chow-Liu Trees". In: *Machine Learning and Knowledge Discovery in Databases*. Vol. 8725. LNCS. Springer, pp. 630–645.
- Rooshenas, Amirmohammad and Daniel Lowd (2014). "Learning Sum-Product Networks with Direct and Indirect Variable Interactions". In: *Proceedings of ICML 2014*.
- Bekker, Jessa et al. (2015). "Tractable Learning for Complex Probability Queries". In: *Advances in Neural Information Processing Systems 28 (NIPS)*.
- Choi, Arthur, Guy Van den Broeck, and Adnan Darwiche (2015). "Tractable learning for structured probability spaces: A case study in learning preference distributions". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*.
- Dennis, Aaron and Dan Ventura (2015). "Greedy Structure Search for Sum-product Networks". In: *IJCAI'15*. Buenos Aires, Argentina: AAAI Press, pp. 932–938. ISBN: 978-1-57735-738-4.
- Peharz, Robert (2015). "Foundations of Sum-Product Networks for Probabilistic Modeling". PhD thesis. Graz University of Technology, SPSC.

References IV

- Peharz, Robert et al. (2015). "On Theoretical Properties of Sum-Product Networks". In: *The Journal of Machine Learning Research*.
- Vergari, Antonio, Nicola Di Mauro, and Floriana Esposito (2015). "Simplifying, Regularizing and Strengthening Sum-Product Network Structure Learning". In: *ECML-PKDD 2015*.
- Cohen, Nadav, Or Sharir, and Amnon Shashua (2016). "On the expressive power of deep learning: A tensor analysis". In: *Conference on Learning Theory*, pp. 698–728.
- Jaini, Priyank et al. (2016). "Online Algorithms for Sum-Product Networks with Continuous Variables". In: *Probabilistic Graphical Models - Eighth International Conference, PGM 2016, Lugano, Switzerland, September 6-9, 2016. Proceedings*, pp. 228–239. URL: `http://jmlr.org/proceedings/papers/v52/jaini16.html`.
- Oztok, Umut, Arthur Choi, and Adnan Darwiche (2016). "Solving PP-PP-complete problems using knowledge compilation". In: *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*.
- Rashwan, Abdullah, Han Zhao, and Pascal Poupart (2016). "Online and Distributed Bayesian Moment Matching for Parameter Learning in Sum-Product Networks". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 1469–1477.
- Rooshenas, Amirmohammad and Daniel Lowd (2016). "Discriminative Structure Learning of Arithmetic Circuits". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 1506–1514.
- Sharir, Or et al. (2016). "Tractable generative convolutional arithmetic circuits". In: *arXiv preprint arXiv:1610.04167*.
- Shen, Yujia, Arthur Choi, and Adnan Darwiche (2016). "Tractable Operations for Arithmetic Circuits of Probabilistic Models". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3936–3944.
- Zhao, Han, Pascal Poupart, and Geoffrey J Gordon (2016a). "A Unified Approach for Learning the Parameters of Sum-Product Networks". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., pp. 433–441.

References V

- Zhao, Han et al. (2016b). "Collapsed Variational Inference for Sum-Product Networks". In: *In Proceedings of the 33rd International Conference on Machine Learning*. Vol. 48.
- Choi, YooJung, Adnan Darwiche, and Guy Van den Broeck (2017). "Optimal feature selection for decision robustness in Bayesian networks". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.
- Di Mauro, Nicola et al. (2017). "Fast and Accurate Density Estimation with Extremely Randomized Cutset Networks". In: *ECML-PKDD 2017*.
- Liang, Yitao, Jessa Bekker, and Guy Van den Broeck (2017a). "Learning the structure of probabilistic sentential decision diagrams". In: *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*.
- Liang, Yitao and Guy Van den Broeck (Aug. 2017b). "Towards Compact Interpretable Models: Shrinking of Learned Probabilistic Sentential Decision Diagrams". In: *IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI)*. URL: `http://starai.cs.ucla.edu/papers/LiangXAI17.pdf`.
- Choi, YooJung and Guy Van den Broeck (2018). "On robust trimming of Bayesian network classifiers". In: *arXiv preprint arXiv:1805.11243*.
- Rashwan, Abdullah, Pascal Poupart, and Chen Zhitang (2018). "Discriminative Training of Sum-Product Networks by Extended Baum-Welch". In: *International Conference on Probabilistic Graphical Models*, pp. 356–367.
- Khosravi, Pasha et al. (2019). "What to Expect of Classifiers? Reasoning about Logistic Regression with Missing Features". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*.
- Liang, Yitao and Guy Van den Broeck (2019). "Learning Logistic Circuits". In: *Proceedings of the 33rd Conference on Artificial Intelligence (AAAI)*.
- Peharz, Robert et al. (2019). "Random Sum-Product Networks: A Simple and Effective Approach to Probabilistic Deep Learning". In: *Uncertainty in Artificial Intelligence*.

References VI

- Shao, Xiaoting et al. (2019). “Conditional Sum-Product Networks: Imposing Structure on Deep Probabilistic Architectures”. In: *arXiv preprint arXiv:1905.08550*.
- Trapp, Martin et al. (2019). “Bayesian Learning of Sum-Product Networks”. In: *Advances in neural information processing systems (NeurIPS)*.
- Vergari, Antonio et al. (2019). “Automatic Bayesian density analysis”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 5207–5215.