Probabilistic Circuits

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)

\[ X = \{ \text{Day, Time, } \text{Jam}_\text{Alma}, \text{Jam}_\text{Str2}, \ldots, \text{Jam}_\text{StrN} \} \]
Complete evidence (EVI)

\[ X = \{ \text{Day, Time, Jam}_{\text{Alma}}, \text{Jam}_{\text{Str2}}, \ldots, \text{Jam}_{\text{StrN}} \} \]

\( q_1: \) What is the probability that today is a Monday at 12.00 and there is a traffic jam only on Alma Str.?
Complete evidence (EVI)

\[ X = \{ \text{Day, Time, Jam}_{\text{Alma}}, \text{Jam}_{\text{Str2}}, \ldots, \text{Jam}_{\text{StrN}} \} \]

\( q_1: \) What is the probability that today is a Monday at 12.00 and there is a traffic jam only on Alma Str.?

\[ q_1(m) = p_m(X = \{ \text{Mon, 12.00, 1, 0, \ldots, 0} \}) \]
Complete evidence (EVI)

\[ X = \{ \text{Day, Time, } \text{Jam}_{\text{Alma}}, \text{Jam}_{\text{Str2}}, \ldots, \text{Jam}_{\text{StrN}} \} \]

\( q_1 \): What is the probability that today is a Monday at 12.00 and there is a traffic jam only on Alma Str.?

\( q_1(m) = p_m(X = \{ \text{Mon}, 12.00, 1, 0, \ldots, 0 \}) \)

...fundamental in maximum likelihood learning

\[ \theta_{m}^{\text{MLE}} = \arg\max_{\theta} \prod_{x \in D} p_m(x; \theta) \]
**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.?
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.?

$q_2(m) = p_m(\text{Day} = \text{Mon}, \text{Jam}_{\text{Alma}} = 1)$
**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.?

$q_2(m) = p_m(\text{Day} = \text{Mon}, \text{Jam}_{\text{Alma}} = 1)$

General: $p_m(e) = \int p_m(e, H) \, dH$

where $E \subset X$, $H = X \setminus E$
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.?

$q_2(m) = p_m(\text{Day} = \text{Mon}, \text{Jam}_{\text{Alma}} = 1)$

General: $p_m(e) = \int p_m(e, H) \, dH$

and if you can answer MAR queries, then you can also do **conditional queries** (CON):

$$p_m(q \mid e) = \frac{p_m(q, e)}{p_m(e)}$$
Maximum A Posteriori (MAP)
aka Most Probable Explanation (MPE)

q₃: Which combination of roads is most likely to be jammed on Monday at 9am?
**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?

$q_3(m) = \arg\max_j p_m(j_1, j_2, \ldots | \text{Day} = M, \text{Time} = 9)$
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?

$q_3(m) = \arg\max_j p_m(j_1, j_2, \ldots | \text{Day} = M, \text{Time} = 9)$

General: $\arg\max_q p_m(q | e)$

where $Q \cup E = X$
Tractable Probabilistic Inference

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

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

\[ p(x) = \prod_{i=1}^{n} p(x_i) \]

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 $O(|X| \cdot 2^w)$, 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 \( O(|X| \cdot 2^w) \), which is linear for fixed width \( w \) [Dechter 1998; Koller et al. 2009].

\[\Rightarrow \text{what about bounding the treewidth by design?}\]
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 \( O(|X| \cdot 2^w) \), which is linear for fixed width \( w \) [Dechter 1998; Koller et al. 2009].

⇒ what about bounding the treewidth by design?

⇒ Bounded-treewidth PGMs cannot represent all possible distributions.
### Summary so far...

<table>
<thead>
<tr>
<th></th>
<th>EVI</th>
<th>MAR</th>
<th>MAP</th>
<th>expressive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully-factorized</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>Bounded-treewidth PGMs</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>PGMs</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
</tr>
</tbody>
</table>
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 $k$
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!

$$\max_q p_m(q \mid e) = \max_q \sum_i w_i p_i(q \mid e)$$

$$\neq \sum_i w_i \max_q p_i(q \mid e)$$
Expressiveness and efficiency

Expressiveness: Ability to represent rich and effective classes of functions

⇒ mixture of Gaussians can approximate any distribution!

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

⇒ mixture of Gaussians can approximate any distribution!

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

⇒ but how many components does a Gaussian mixture need?

---

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...

<table>
<thead>
<tr>
<th></th>
<th>EVI</th>
<th>MAR</th>
<th>MAP</th>
<th>expressive</th>
<th>expr. efficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully-factorized</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Bounded-treewidth PGMs</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>PGMs</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Mixtures</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>
Expressive models are not very tractable...
and tractable ones are not very expressive...
probabilistic circuits are at the "sweet spot"
Probabilistic Circuits
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

How do you build a probabilistic circuit?
Base case: Univariate distributions

A single node encoding a distribution

\[ \neg X \]

\( \Rightarrow \)  e.g., indicators for \( X \) or \( \neg X \) for Boolean random variable
Base case: Univariate distributions

\[ p_X(x) \]

A single node encoding a distribution

\[ x \xrightarrow{\bigwedge} p_X(x) \]

\( X \)

\( X \)

- e.g., indicators for \( X \) or \( \neg X \) for Boolean random variable
- More generally, PDFs for continuous random variable
Base case: Univariate distributions

A single node encoding a distribution

⇒ e.g., indicators for \(X\) or \(\neg X\) for Boolean random variable
⇒ More generally, PDFs for continuous random variable
**Base case: Univariate distributions**

1.3 $\overline{\cup}$ .33

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*

\[ p(x_1, x_2, x_3) = p(x_1) \cdot p(x_2) \cdot p(x_3) \]

\[
\begin{array}{c}
0.8 \\
X_1
\end{array}
\quad \begin{array}{c}
0.5 \\
X_2
\end{array}
\quad \begin{array}{c}
0.9 \\
X_3
\end{array}
\]

⇒ *feedforward evaluation*
**Factorizations as product nodes**

**Divide and conquer complexity**

\[ p(x_1, x_2, x_3) = p(x_1) \cdot p(x_2) \cdot p(x_3) \]
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

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

⇒ by stacking them we increase expressive efficiency
A grammar for tractable models

Recursive semantics of probabilistic circuits

$X_1$
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

\[ X_1 \times X_2 \times X_3 \]

\[ X_1 \times X_1 \times X_3 \]

\( \Rightarrow \) just like in factorization!

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

⇒ otherwise not accounting for some variables

smooth circuit

non-smooth circuit

Darwiche et al., “A knowledge compilation map”, 2002
Smoothness + decomposability = tractable MAR
\[ \text{Smoothness} + \text{decomposability} = \text{tractable MAR} \]

If \( p(x) = \sum_i w_i p_i(x) \), (smoothness):

\[
\int p(x) \, dx = \int \sum_i w_i p_i(x) \, dx = \sum_i w_i \int p_i(x) \, dx
\]

\[ \Rightarrow \text{integrals are “pushed down” to children} \]
Smoothness + decomposability = tractable MAR

If $p(x, y, z) = p(x)p(y)p(z)$, (decomposability):

$$\int \int \int p(x, y, z) dx dy dz =$$

$$= \int \int \int p(x)p(y)p(z) dx dy dz =$$

$$= \int p(x)dx \int p(y)dy \int p(z)dz$$

$\implies$ integrals decompose into easier ones
Smoothness + decomposability = tractable MAR

To compute \( p(x_2, x_4) \):

- leaves over \( X_1 \) and \( X_3 \) output
  \[ Z_i = \int p(x_i) \, dx_i \]
  \[ \implies \text{for normalized leaf distributions: } 1.0 \]
- leaves over \( X_2 \) and \( X_4 \) output \text{EVI}
- feedforward evaluation (bottom-up)
Smoothness + decomposability = tractable MAR

To compute \( p(x_2, x_4) \):

- leaves over \( X_1 \) and \( X_3 \) output
  \[ Z_i = \int p(x_i) \, dx_i \]
  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
  \[
  Z_i = \int p(x_i) \, dx_i
  \]
  - for normalized leaf distributions: 1.0

- leaves over $X_2$ and $X_4$ output **EVI**

- feedforward evaluation (bottom-up)
Smoothness and decomposability are **sufficient** conditions for a circuit to compute marginals.
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 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 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

\[ w_1 \times X_1 \leq \theta \times X_2 \]  
\[ w_2 \times X_1 > \theta \times X_2 \]

deterministic circuit

\[ x_1 \leq \theta \]  
\[ x_2 \]  
\[ x_1 > \theta \]  
\[ x_2 \]

non-deterministic circuit

\[ x_1 \]  
\[ x_2 \]  
\[ x_1 \]  
\[ x_2 \]

e.g. *if their distributions have disjoint support*
A product node is consistent if any variable shared between its children appears in a single leaf node.

\[ X_1 \land X_2 \land X_3 \]

\[ w_1 \quad w_2 \quad w_3 \quad w_4 \]

consistent circuit

\[ X_1 \land X_2 \land X_3 \]

\[ w_1 \quad w_2 \quad w_3 \quad w_4 \]

inconsistent circuit

\[ X_1 \land X_2 \leq \theta \land X_2 > \theta \land X_3 \]
$Determinism + consistency = tractable \text{MAP}$
If \( p(q, e) = \sum_i w_i p_i(q, e) = \max_i w_i p_i(q, e) \),
(determinism):

\[
\max_q p(q, e) = \max_q \sum_i w_i p_i(q, e) = \max_q \max_i w_i p_i(q, e)
\]

\[
= \max_i \max_q w_i p_i(q, e)
\]

\[
\Rightarrow \text{ one non-zero child term, thus sum is max}
\]
**Determinism** + **consistency** = **tractable MAP**

If \( \max_{q_{\text{shared}}} p(q, e) = \max_{q_{\text{shared}}} p(q_x, e_x) \cdot \max_{q_{\text{shared}}} p(q_y, e_y) \) (consistent):

\[
\max_q p(q, e) = \max_{q_x, q_y} p(q_x, e_x, q_y, e_y) = \max_{q_x} p(q_x, e_x) \cdot \max_{q_y} p(q_y, e_y)
\]

\[ \implies \text{solving optimization independently} \]
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 $M_i = \max p(x_i)$
- leaves over $X_2$ and $X_4$ output $\text{EVI}$
- 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 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 ⇒ cannot maximize correctly without summations.
- Inconsistent node ⇒ MAP assignments of children conflict with each other.
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 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

smooth & Decomp.

[Darwiche et al. 2002b]

? det. & Decomp.

det. & cons.

: strictly more succinct
Succinctness of circuits

- **smooth & Decomp.**
  - [Peharz et al. 2015] (strictly more succinct)
  - [Darwiche et al. 2002b] (equally succinct)
- **det. & Decomp.**
- **det. & cons.**

Directed edges:
- : strictly more succinct
- : equally succinct
Succinctness of circuits

- smooth & Decomp.
- smooth & cons.
- det. & Decomp.
- det. & cons.

[Darwiche et al. 2002b]: strictly more succinct

[Peherz et al. 2015]: equally succinct

: strictly more succinct
: equally succinct
Succinctness of circuits

Consider following circuit over Boolean variables:

\[ \prod_i (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in X \]

- Size linear in the number of variables
- Deterministic and consistent
- Marginal (with no evidence) is the solution to #P-hard SAT problem [Valiant 1979] ⇒ no tractable circuit for marginals!
Consider the following circuit over Boolean variables:

\[ \prod_i (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in X \]

- Size linear in the number of variables
- Deterministic and consistent
- Marginal (with no evidence) is the solution to \#P-hard SAT' problem [Valiant 1979] \implies no tractable circuit for marginals!
Consider following circuit over Boolean variables:
\[ \prod_i^r (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in X \]

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

- Strictly more succinct
- Equally succinct
Consider following circuit over Boolean variables:
\[
\prod_i^r (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in X
\]
- Size linear in the number of variables
- Deterministic and consistent
- Marginal (with no evidence) is the solution to \#P-hard SAT\' problem [Valiant 1979] ⇒ no tractable circuit for marginals!
Consider the following circuit over Boolean variables:

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

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

Consider following circuit over Boolean variables:
\[ \prod_i (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in X \]

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

: strictly more succinct
--- : equally succinct
### Succinctness of circuits

Consider the following circuit over Boolean variables:

\[ \prod_i (Y_i \cdot Z_{i1} + (\neg Y_i) \cdot Z_{i2}), \quad Z_{ij} \in X \]

- 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!**

\[ \Rightarrow \text{ implies hardness of smoothing consistent circuits!} \]
Succinctness of circuits

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

- Linear-size smooth and decomposable circuit
- MAP of $p(X)$ solves marginal MAP of $p(X, C)$ which is NP-hard [Campos 2011] ⇒ no tractable circuit for MAP!
Consider the marginal distribution $p(X)$ from a naive Bayes distribution $p(X, C)$:

- Linear-size smooth and decomposable circuit
- MAP of $p(X)$ solves marginal MAP of $p(X, C)$ which is NP-hard [Campos 2011] $\Rightarrow$ no tractable circuit for MAP!
Consider the marginal distribution $p(X)$ from a naive Bayes distribution $p(X, C)$:

- Linear-size smooth and decomposable circuit
- MAP of $p(X)$ solves marginal MAP of $p(X, C)$ which is NP-hard \cite{Campos2011} \implies no tractable circuit for MAP!
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 \[\text{[Campos 2011]} \Rightarrow \text{no tractable circuit for MAP!}\]
Structured decomposability

A product node is structured decomposable if decomposes according to a node in a \textit{vtree}.

\[ X_1 \times X_2 \times X_3 \]

\[ \text{vtree} \]

\[ \text{structured decomposable circuit} \]
Structured decomposability

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

\[ X_3 \]
\[ X_1 \]
\[ X_2 \]

\[ \text{vtree} \]

\[ \implies \text{stronger requirement than decomposability} \]

\[ \text{non structured decomposable circuit} \]
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
<table>
<thead>
<tr>
<th>Representation</th>
<th>Smooth</th>
<th>Decomposable</th>
<th>Deterministic</th>
<th>Structured-Decomposable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic Circuits (ACs) [Darwiche 2003]</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>Sum-Product Networks (SPNs) [Poon et al. 2011]</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Cutset Networks (C Nets) [Rahman et al. 2014]</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>PSDDs [Kisa et al. 2014a]</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>AndOrGraphs [Dechter et al. 2007]</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Building circuits
Compiling PGMs to probabilistic circuits

Example: from BN trees to circuits
## Learning probabilistic circuits

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generative</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Deterministic</strong></td>
<td><strong>Greedy</strong></td>
</tr>
<tr>
<td>closed-form MLE</td>
<td>top-down</td>
</tr>
<tr>
<td>[Kisa et al. 2014b; Peharz et al. 2014]</td>
<td>[Gens et al. 2013; Rooshenas et al. 2014]</td>
</tr>
<tr>
<td><strong>Non-deterministic</strong></td>
<td></td>
</tr>
<tr>
<td>EM</td>
<td>bottom-up</td>
</tr>
<tr>
<td>[Poon et al. 2011; Peharz 2015; Zhao et al. 2016a]</td>
<td>[Rahman et al. 2014; Vergari et al. 2015]</td>
</tr>
<tr>
<td>SGD</td>
<td><strong>Hill climbing</strong></td>
</tr>
<tr>
<td>Bayesian</td>
<td></td>
</tr>
<tr>
<td>[Jaini et al. 2016; Rashwan et al. 2016]</td>
<td>[Dennis et al. 2015; Liang et al. 2017a]</td>
</tr>
<tr>
<td><strong>Discriminative</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Deterministic</strong></td>
<td><strong>Greedy</strong></td>
</tr>
<tr>
<td>convex-opt MLE</td>
<td>top-down</td>
</tr>
<tr>
<td>[Liang et al. 2019]</td>
<td>[Shao et al. 2019]</td>
</tr>
<tr>
<td><strong>Non-deterministic</strong></td>
<td><strong>Hill climbing</strong></td>
</tr>
<tr>
<td>EM</td>
<td>[Rooshenas et al. 2016]</td>
</tr>
<tr>
<td>[Rashwan et al. 2018]</td>
<td></td>
</tr>
<tr>
<td>SGD</td>
<td></td>
</tr>
<tr>
<td>[Gens et al. 2012; Sharir et al. 2016]</td>
<td></td>
</tr>
<tr>
<td>[Peharz et al. 2019]</td>
<td></td>
</tr>
</tbody>
</table>
How expressive are probabilistic circuits?

density estimation benchmarks

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


Darwiche, Adnan (2003). “A Differential Approach to Inference in Bayesian Networks”. In: JACM.
References II


References IV


References V


