PROBABILISTIC AND DIFFERENTIABLE PROGRAMMING

V10: Probabilistic Circuits I

Özgür L. Özçep
Universität zu Lübeck
Institut für Informationssysteme
Today’s Agenda

Probabilistic Circuits

1. Motivation
2. Building Blocks
3. Structural Properties for Tractability
MOTIVATION
Have the cake and eat it too

- Be as expressive as possible
  - Express arbitrary distributions

- Be as efficient as possible in inferencing/answering queries

- For exact (not just approximative) answering
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 $O(poly(|m|))$.

- Often poly will in fact be linear
- Note: if $M$ and $Q$ are compact in the number of random variables $X$, i.e., $|m|, |q| \in O(poly(|X|))$, then query time is $O(poly(|X|))$. 
Why exact inference?

1. No need for approximations when we can be exact
2. We can do exact inference in approximate models (e.g., Dechter et al. 2002)
3. Approximations shall come with guarantees
4. Approximate inference (even with guarantees) can mislead learners (Kulesza/Pereira 2007)
5. Approximations can be intractable as well (Dagum/Luby1993)
Complete Evidence (EVI)

• \( q_3: \) What is the probability that today is a Monday at 12.00 and there is a traffic jam only on 5th Avenue?

• \( X = \{\text{Day, Time, Jam}_{5th}, Jam_{Str2}, \ldots, Jam_{StrN}\} \)

• \( q_3(m) = p_m(X = \{\text{Mon, 12.00,1,0, \ldots, 0}\}) \)

• Fundamental in maximum likelihood learning
  \[ \theta^M_{MLE} = \arg\max_\theta (\prod_{x \in D} p_m(x; \theta)) \]
Marginal (MAR) and Conditional (CON) queries

- $q_1$: What is the probability that today is a Monday at 12:00 and there is a traffic jam only on 5th Avenue?

- $q_1(m) = p_m(\text{Day} = \text{Mon}, \text{Jam}_{5th} = 1)$

- General:

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

- With this can answer conditional queries too

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

- \( q_5 \): Which combination of roads is most likely to be jammed on Monday at 9am?

\[
q_5(m) = \arg\max_j p_m(j_1, j_2, ... | \text{Day} = \text{Mon}, \text{Time} = 9)
\]

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

where \( Q \cup E = X \)
Maximum A Posteriori (MAP) (aka Most Probable Explanation (MPE))

- $q_5$: Which combination of roads is most likely to be jammed on Monday at 9am?

- ... Intractable for latent variable models

\[
\max_{q} p_m(q | e) = \max_{q} \sum_{z} p_m(q, z | e) \\
\neq \sum_{z} \max_{q} p_m(q, z | e)
\]
Marginal MAP (MMAP) (aka Bayesian network MAP)

- $q_6$: Which combination of roads is most likely to be jammed on Monday at 9am?
- $q_6(m) = \arg\max_j p_m(j_1, j_2, \ldots \mid Time = 9)$

- General: $\arg\max_q p_m(q \mid e) = \arg\max_q \Sigma_h p_m(q, h \mid e)$
  
  where $Q \cup H \cup E = X$

- $\text{NP}^{\text{PP}}$-complete (Park/Darwiche)
- $\text{NP}$-hard for trees (Campos 2011)
Advanced Queries (ADV)

- $q_2$: Which day is most likely to have a traffic jam on my route to work?
  
  $q_2(m) = \arg\max_d p_m(Day = d, \wedge \bigvee_{i \in route} Jam_{Stri})$
  
  - $\implies$ Marginals + MAP + logical events

- $q_7$: What is the probability of seeing more traffic jams in Uptown than Midtown?
  
  - $\implies$ counts + group comparison

- And more
  
  - expected classification agreement (Oztok et al. 2016)
  
  - Expected predictions (Khosravi et al. 2019b)
tractable bands

1) (Kobyzev et al. 19)
OK, fully factorized models have broadest tractability spectrum, but ...

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**!

⇒ but definitely not expressive…

(no dependencies representable)
Expressiveness and efficiency

- **Expressiveness**: Ability to represent rich and effective classes of functions

- **Mixture of Gaussians can approximate any distribution!**
  - See (Cohen et al. 15)

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

- ⇒ but how many components does a Gaussian mixture need?
Use these as building blocks

„Eat the cake and have it“ *tractable bands*
probabilistic circuits are at the “sweet spot”
BUILDING BLOCKS
A probabilistic circuit $C$ over variables $X$ is a computational graph encoding a (possibly unnormalized) probability distribution $p(X)$.

- Note that we have an operational semantics here
- By constraining the graph one can make inference tractable
Distributions as computational graphs

Base case: a single node encoding a distribution

- A Gaussian PDF continuous variable
- Indicators for $X$ or $\neg X$ for Boolean RVs
Simple distributions are tractable “black boxes” for:

- **EVI**: output $p(x)$ (density or mass)
- **MAR**: output 1 (normalized) or $Z$ (unnormalized)
- **MAP**: output the mode
Simple distributions are tractable „black boxes“ for:
- **EVI**: output $p(x)$ (density or mass)
- **MAR**: output 1 (normalized) or $Z$ (unnormalized)
- **MAP**: output the mode
Reminder: Partition function $Z$

\[ P(X_1, \ldots, X_n) = \frac{1}{Z} \prod_j \phi_j(X_1, \ldots, X_n) \]

- Bottleneck: Summing out variables
- E.g.: Partition function

Sum of exponentially many products

\[ Z = \sum_x \prod_j \phi_j \]
Factorizations as product nodes
( Divide and Conquer complexity)

\[ p(X_1, X_2, X_3) = p(X_1) \cdot p(X_2) \cdot p(X_3) \]

e.g. modeling a multivariate Gaussian with diagonal covariance matrix by a product node of univariate Gaussians
Factorizations as product nodes
(Divide and Conquer complexity)

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

Feed forward evaluation
Mixtures as sum nodes
(enhance expressiveness)

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

\( \Rightarrow \) e.g. modeling a mixture of Gaussians...
Mixtures as sum nodes
(enhance expressiveness)

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

⇒ \text{...as weighted sum node over Gaussian input distributions}
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 for probabilistic circuits)
Probabilistic Circuits are not PGMs

They are *probabilistic* and *graphical*, however ...

<table>
<thead>
<tr>
<th></th>
<th><strong>PGMs</strong></th>
<th><strong>Circuits</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nodes:</strong></td>
<td>random variables</td>
<td>unit of computations</td>
</tr>
<tr>
<td><strong>Edges:</strong></td>
<td>dependencies</td>
<td>order of execution</td>
</tr>
<tr>
<td><strong>Inference:</strong></td>
<td>conditioning</td>
<td>feedforward pass</td>
</tr>
<tr>
<td></td>
<td>elimination</td>
<td>backward pass</td>
</tr>
<tr>
<td></td>
<td>message passing</td>
<td></td>
</tr>
</tbody>
</table>

⇒ they are *computational graphs*, more like neural networks
Control on the graph

- We do not arbitrarily compose the building blocks as in neural networks
- But define *structural constraints for tractability*
Side note: Compare this with descriptive complexity

Correspondence of logics and complexity classes

- Arithmetic Hierarchy
  - FO(N)
  - FO\exists(N)
  - Recursive

- Primitive Recursive
  - SO[2^{O(1)}]
  - EXPTIME
  - SO(LFP)

- Polynomial-Time Hierarchy
  - FO[n^{O(1)}]
  - SO[n^{O(1)}]
  - PSPACE
  - SO(FP)
  - SO(TC)

- NP and co-NP
  - FO[n^{O(1)}]
  - "truly feasible"
  - P
  - FO(LFP)
  - SO–Horn

- Logarithmic-Time Hierarchy
  - FO[(\log n)^{O(1)}]
  - NC
  - \text{NC}^2
  - log(CFL)
  - \text{sAC}^1
  - FO(TC)
  - \text{NSPACE}[\log n]
  - \text{SO–Krom}
  - FO(DTC)
  - \text{DSPACE}[\log n]

- Regular Languages
  - FO(M)
  - \text{Regular}
  - \text{NC}^1
  - \text{ThC}^0

- Logarithmic–Time Hierarchy
  - FO
  - \text{AC}^0

The Descriptive World
STRUCTURAL PROPERTIES FOR TRACTABILITY
Decomposability

A product node is decomposable if its children depend on disjoint sets of variables \(\textit{just like in factorization}\).

\[
\begin{align*}
\text{decomposable circuit} & \\
\text{non-decomposable circuit}
\end{align*}
\]

(Darwiche/Marquis 01)
Smoothness (aka as completeness)

A sum node is smooth iff its children depend on the same variable sets (otherwise not accounting for some variables)

\[
\begin{align*}
\text{smooth circuit} & \quad \begin{array}{c}
\bigwedge \quad w_1 \\
X_1 \\
\bigwedge \quad w_2 \\
X_1
\end{array} \\
\text{non-smooth circuit} & \quad \begin{array}{c}
\bigwedge \quad w_1 \\
X_1 \\
\bigwedge \quad w_2 \\
X_2
\end{array}
\end{align*}
\]

(Darwiche/Marquis 01)
Smoothness + decomposability = tractable MAR

Computing arbitrary integrations (or summations)
\[ \Rightarrow \text{linear in circuit size!} \]

E.g., suppose we want to compute Z:
\[ \int p(x)dx \]
Smoothness + decomposability = tractable MAR

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

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

$\Rightarrow$ integrals are “pushed down” to children
Smoothness + decomposability = tractable MAR

If \( p(x, y, z) = p(x)p(y)p(z) \), *(decomposability)*:

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

\( \Rightarrow \) integrals decompose into easier ones
Smoothness + decomposability = tractable MAR

Forward pass evaluation for MAR
⇒ linear in circuit size!

E.g. 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)

Analogously one can show:
Smoothness + decomposability = tractable CON

Note: Nodes with the same RV-label may have different probabilities associated with them. Hence, e.g., the left bottom $X_4$ may get a different value than the right bottom $X_4$. 
Smoothness + decomposability $\neq$ tractable MAP

We cannot decompose bottom-up a MAP query:

$$\arg\max_q p(q \mid e)$$

since for a sum node we are marginalizing out a latent variable

$$\arg\max_q \sum_i w_i p_i(q, e) = \arg\max_q \sum p(q, z, e) \neq \sum \arg\max_q p(q, z, e)$$

$\implies$ MAP for latent variable models is intractable [Conaty et al. 2017]
Determinism (aka selectivity)

A sum node is **deterministic** if the output of only one of its children is non zero for any input *(e.g. if their distributions have disjoint support)*
Determinism + decomposability = tractable MAP

Computing maximization with arbitrary evidence $e$  
\[ \max_{\mathbf{q}} p(\mathbf{q} \mid \mathbf{e}) \]

\[ \Rightarrow \text{linear in circuit size!} \]

E.g., suppose we want to compute:

\[ \max_{\mathbf{q}} p(\mathbf{q} \mid \mathbf{e}) \]
Determinism + decomposability = tractable MAP

If \( p(q, e) = \sum_i w_i p_i(q, e) = \max_i w_i p_i(q, e) \),
(deterministic sum node):

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

\( \Rightarrow \) one non-zero child term, thus sum is max
Determinism + decomposability = tractable MAP

If \( p(q, e) = p(q_x, e_x, q_y, e_y) = p(q_x, e_x)p(q_y, e_y) \) (decomposable product node):

\[
\max_q p(q | e) = \max_q p(q, e) \cdot 1/p(e) \\
= \max_{q_x, q_y} p(q_x, e_x, q_y, e_y) \cdot 1/p(e) \\
= \max_{q_x} p(q_x, e_x) \cdot \max_{q_y} p(q_y, e_y) \cdot 1/p(e)
\]

\[\Rightarrow \quad \text{solving optimization independently}\]
Determinism + decomposability = tractable MAP

Evaluating the circuit twice: **bottom-up** and **top-down** $\implies$ still linear in circuit size!

E.g., for $\text{argmax}_{x_1, x_3} p(x_1, x_3 \mid x_2, x_4)$:

1. turn sum into max nodes and distributions into max distributions
2. evaluate $p(x_2, x_4)$ bottom-up
3. retrieve max activations top-down
4. compute **MAP states** for $X_1$ and $X_3$ at leaves
Determinism + decomposability = tractable MAP

Evaluating the circuit twice:

**bottom-up** and **top-down**  \[\Rightarrow\]  **still linear in circuit size**!

E.g., for \(\text{argmax}_{x_1,x_3} p(x_1,x_3 \mid x_2,x_4)\):

1. turn sum into max nodes and distributions into max distributions
2. evaluate \(p(x_2,x_4)\) bottom-up
3. retrieve max activations top-down
4. compute \(\text{MAP states}\) for \(X_1\) and \(X_3\) at leaves
Determinism + decomposability = tractable MAP

Evaluating the circuit twice: **bottom-up** and **top-down** still linear in circuit size!

E.g., for $\text{argmax}_{x_1, x_3} p(x_1, x_3 \mid x_2, x_4)$:

1. turn sum into max nodes and distributions into max distributions
2. evaluate $p(x_2, x_4)$ bottom-up
3. retrieve max activations top-down
4. compute MAP states for $X_1$ and $X_3$ at leaves
Determinism + decomposability = tractable MAP

Evaluating the circuit twice:
*bottom-up* and *top-down* \(\Rightarrow\) *still linear in circuit size!*

E.g., for \(\text{argmax}_{x_1, x_3} p(x_1, x_3 | x_2, x_4)\):
1. turn sum into max nodes and distributions into max distributions
2. evaluate \(p(x_2, x_4)\) bottom-up
3. retrieve max activations top-down
4. compute **MAP states** for \(X_1\) and \(X_3\) at leaves
Semantic segmentation is MAP over joint pixel and label space.

Even approximate MAP for non-deterministic circuits (SPNs) delivers good performances.

Rathke et al., “Locally adaptive probabilistic models for global segmentation of pathological oct scans”, 2017


Friesen et al., “Submodular Sum-product Networks for Scene Understanding”, 2016
Determinism + decomposability ≠ tractable MMAP

We **cannot** decompose a MMAP query!

\[
\text{argmax}_q \sum_z p(q, z \mid e)
\]

we still have latent variables to marginalize...

This will be discussed in lecture V12 (when considering advanced queries)
tractability vs expressive efficiency
How expressive are probabilistic circuits?

Measuring average test set log-likelihood on 20 density estimation benchmarks

Comparing against intractable models:

- Bayesian networks (BN) [Chickering 2002] with sophisticated context-specific CPDs
- MADEs [Germain et al. 2015]
- VAEs [Kingma et al. 2014] (IWAE ELBO [Burda et al. 2015])

Gens et al., “Learning the Structure of Sum-Product Networks”, 2013
### 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><strong>-5.99</strong></td>
<td>-6.02</td>
<td>-6.04</td>
<td><strong>-5.99</strong></td>
<td>dna</td>
<td><strong>-79.88</strong></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>-</td>
<td>-10.64</td>
</tr>
<tr>
<td>plants</td>
<td><strong>-11.84</strong></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><strong>-38.67</strong></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><strong>-146.9</strong></td>
</tr>
<tr>
<td>netflix</td>
<td>-55.71</td>
<td>-57.02</td>
<td>-55.16</td>
<td><strong>-54.73</strong></td>
<td>cr52</td>
<td>-81.87</td>
<td>-87.56</td>
<td>-82.80</td>
<td><strong>-81.33</strong></td>
</tr>
<tr>
<td>accidents</td>
<td>-26.89</td>
<td><strong>-26.32</strong></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><strong>-146.9</strong></td>
</tr>
<tr>
<td>retail</td>
<td><strong>-10.72</strong></td>
<td>-10.87</td>
<td>-10.81</td>
<td>-10.83</td>
<td>bbc</td>
<td><strong>-229.21</strong></td>
<td>-257.86</td>
<td>-242.40</td>
<td>-240.94</td>
</tr>
<tr>
<td>pumbs*</td>
<td>-22.15</td>
<td><strong>-21.72</strong></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>

(Best negative log-likelihoods in bold)
Uhhh, a lecture with a hopefully useful

APPENDIX
Probability theory basics reminder

Random variable (RV)

- possible worlds defined by assignment of values to random variables.
- **Boolean** random variables
e.g., Cavity (do I have a cavity?). Domain is < true, false >
- **Discrete** random variables
e.g., possible value of Weather is one of < sunny, rainy, cloudy, snow >
- Domain values must be exhaustive and mutually exclusive
- **Elementary propositions** are constructed by assignment of a value to a random variable: e.g.,
  - Cavity = false (abbreviated as \( \neg \text{cavity} \))
  - Cavity = true (abbreviated as \( \text{cavity} \))
- **(Complex) propositions** formed from elementary propositions and standard logical connectives, e.g., Weather = sunny \( \lor \) Cavity = false

Probabilities

- Axioms (for propositions \( a, b, T = (a \lor \neg a) \), and \( \bot = \neg T \)):
  - \( 0 \leq P(a) \leq 1; P(T) = 1; P(\bot) = 0 \)
  - \( P(a \lor b) = P(a) + P(b) - P(a \land b) \)
- **Joint probability distribution of** \( X = \{X_1, \ldots, X_n\} \)
  - \( P(x_1,\ldots,x_n) \)
  - gives the probability of every atomic event on \( X \)
- **Conditional probability** \( P(a \mid b) = P(a \land b) / P(b) \text{ if } P(b) > 0 \)
- Chain rule
  \[
P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, \ldots, X_{i-1})
\]
- **Marginalization:** \( P(Y) = \sum_{z \in Z} P(Y,z) \)
- **Conditioning on** \( Z \):
  - \( P(Y) = \sum_{z \in Z} P(Y \mid z) P(z) \) (discrete)
  - \( P(Y) = \int P(Y \mid z) P(z) dz \) (continuous)
    \[
    = \mathbb{E}_{z \sim P(z)} P(Y \mid z)
    \]
    (expected value notation)
- **Bayes’ Rule**
  \[
P(H \mid D) = \frac{P(D \mid H) \cdot P(H)}{P(D)} = \frac{P(D \mid H) \cdot P(H)}{\sum_h P(D \mid h) P(h)}
  \]
Color Convention in this Course

- Formulae, when occurring inline
- Newly introduced terminology and definitions
- Important results (observations, theorems) as well as emphasizing some aspects
- Examples are given with standard orange with possibly light orange frame
- Comments and notes in nearly opaque post-it
- Algorithms and program code
- Reminders (in the grey fog of your memory)
Today’s lecture is based on the following

- A. Vergari, Y. Choi, R. Peharz, G. van den Broeck: Probabilistic Circuits, Tutorial at AAAI 2020, pp.1 – 80,
References