Einsum Networks

Fast and Scalable Learning of Tractable Probabilistic Circuits

Robert Peharz
Eindhoven University of Technology

Karl Stelzner
Technical University of Darmstadt

Guy Van den Broeck
University of California, Los Angeles

Steven Lang
Technical University of Darmstadt

Alejandro Molina
Technical University of Darmstadt

Kristian Kersting
Technical University of Darmstadt

Antonio Vergari
University of California, Los Angeles

Martin Trapp
Graz University of Technology

Zoubin Ghahramani
University of Cambridge; Uber AI Labs

International Conference on Machine Learning *(ICML), July 2020*

In This Paper

Probabilistic Circuits (PCs) **— Just a special type of neural network**

Yet, they are slow

- Computational graphs highly sparse and cluttered
- Operations implemented in the log-domain
- ~ 50 times slower than neural net of comparable size

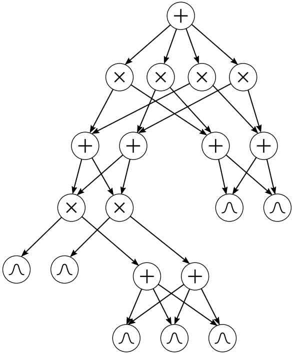
We propose ***Einsum Networks (EiNets)***

- PC architecture using a few monolithic `einsum` operations
- Run and train PCs up to **two orders of magnitude faster**
- Scale PCs to datasets previously out of reach (CelebA, SVHN)

Probabilistic Circuits

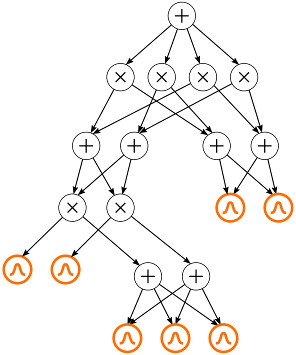
Probabilistic Circuits

Computational graph containing 3 types of operations:
Distributions (leaves), products, and weighted sums.



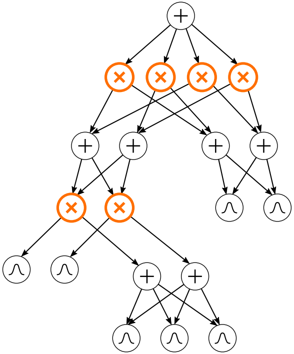
Probabilistic Circuits

Computational graph containing 3 types of operations:
Distributions (leaves), products, and weighted sums.



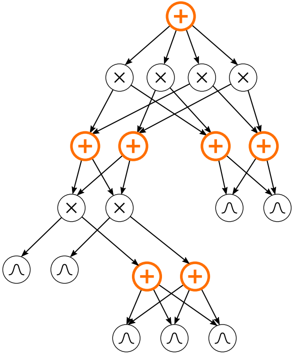
Probabilistic Circuits

Computational graph containing 3 types of operations:
Distributions (leaves), **products**, and weighted sums.

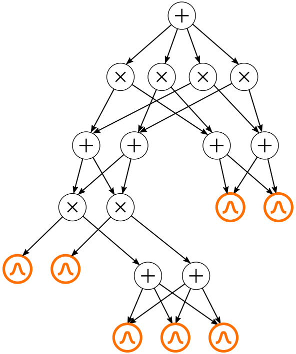


Probabilistic Circuits

Computational graph containing 3 types of operations:
Distributions (leaves), products, and **weighted sums**.



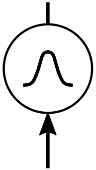
Probabilistic Circuits — Leaf Distributions



Probabilistic Circuits — Leaf Distributions

Arbitrary probability function (pdf, pmf, mixed) over some set of random variables $\mathbf{X}$.

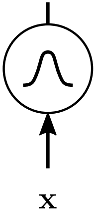
Should facilitate tractable inference routines, e.g. marginalization, conditioning, MAP, ...



Probabilistic Circuits — Leaf Distributions

Arbitrary probability function (pdf, pmf, mixed) over some set of random variables $\mathbf{X}$.

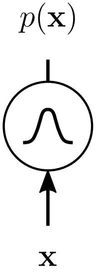
Should facilitate tractable inference routines, e.g. marginalization, conditioning, MAP, ...



Probabilistic Circuits — Leaf Distributions

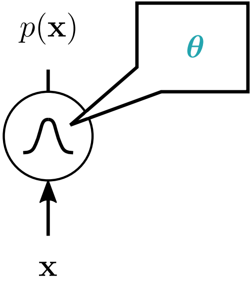
Arbitrary probability function (pdf, pmf, mixed) over some set of random variables $\mathbf{X}$.

Should facilitate tractable inference routines, e.g. marginalization, conditioning, MAP, ...



Probabilistic Circuits — Leaf Distributions

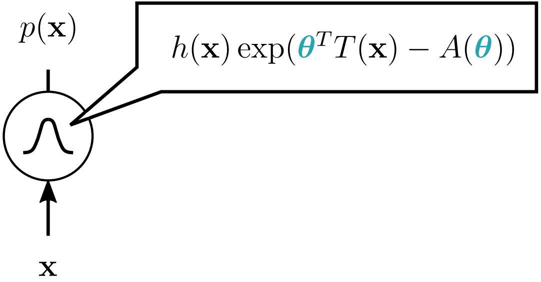
Arbitrary probability function (pdf, pmf, mixed) over some set of random variables $\mathbf{X}$.
Should facilitate tractable inference routines, e.g. marginalization, conditioning, MAP, ...



Probabilistic Circuits — Leaf Distributions

Arbitrary probability function (pdf, pmf, mixed) over some set of random variables $\mathbf{X}$.

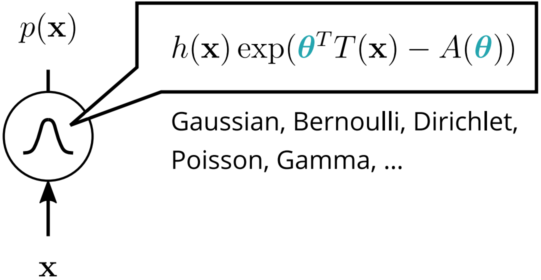
Should facilitate tractable inference routines, e.g. marginalization, conditioning, MAP, ...



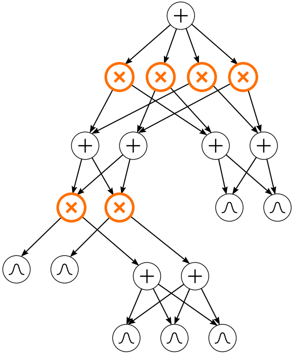
Probabilistic Circuits — Leaf Distributions

Arbitrary probability function (pdf, pmf, mixed) over some set of random variables $\mathbf{X}$.

Should facilitate tractable inference routines, e.g. marginalization, conditioning, MAP, ...

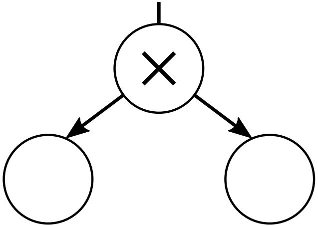


Probabilistic Circuits — Products



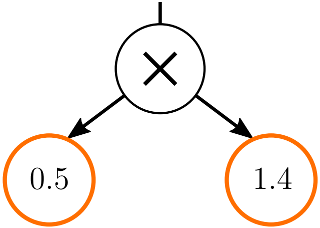
Probabilistic Circuits — Products

Simply product units



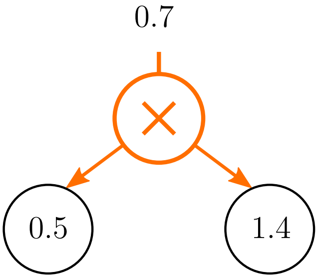
Probabilistic Circuits — Products

Simply product units

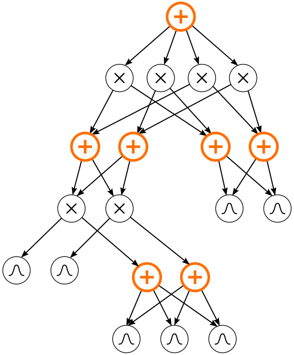


Probabilistic Circuits — Products

Simply product units

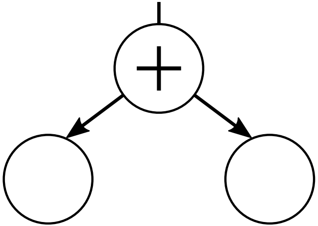


Probabilistic Circuits — Sums



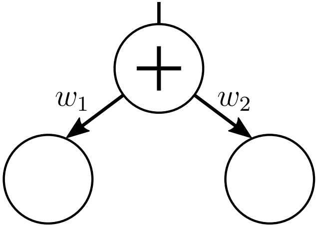
Probabilistic Circuits — Sums

Weighted sums



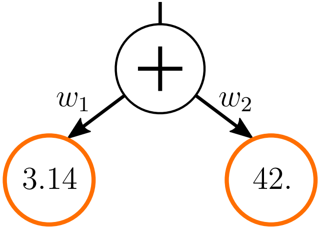
Probabilistic Circuits — Sums

Weighted sums



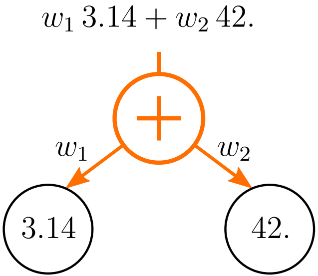
Probabilistic Circuits — Sums

Weighted sums



Probabilistic Circuits — Sums

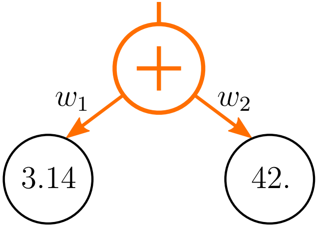
Weighted sums



Probabilistic Circuits — Sums

Weighted sums

$$w_1\, 3.14 + w_2\, 42.$$

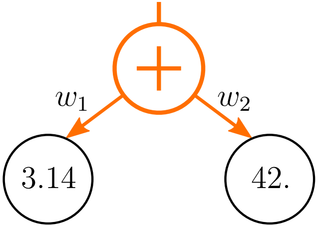


$$w_k \geq 0$$

Probabilistic Circuits — Sums

Weighted sums

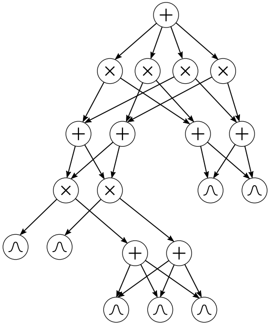
$$w_1\, 3.14 + w_2\, 42.$$



$$w_k \geq 0$$
$$\sum_k w_k = 1$$

Probabilistic Circuits

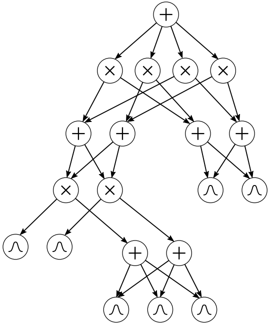
Computational graph containing distributions, products, and weighted sums.



Probabilistic Circuits

Computational graph containing distributions, products, and weighted sums.

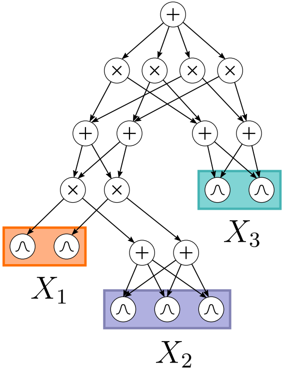
Plus: Structural properties!



Probabilistic Circuits

Computational graph containing distributions, products, and weighted sums.

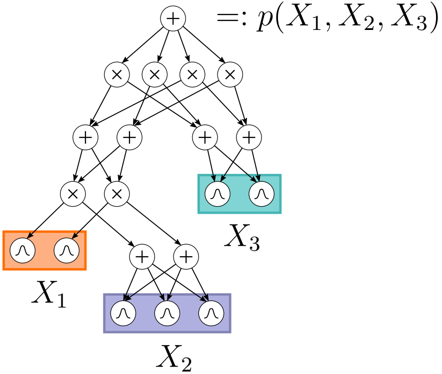
Plus: Structural properties!



Probabilistic Circuits

Computational graph containing distributions, products, and weighted sums.

Plus: Structural properties!



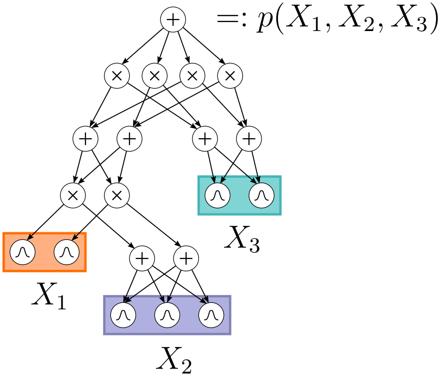
Probabilistic Circuits

Computational graph containing distributions, products, and weighted sums.

Plus: Structural properties!

Smoothness

sum children have same scope



Probabilistic Circuits

Computational graph containing distributions, products, and weighted sums.

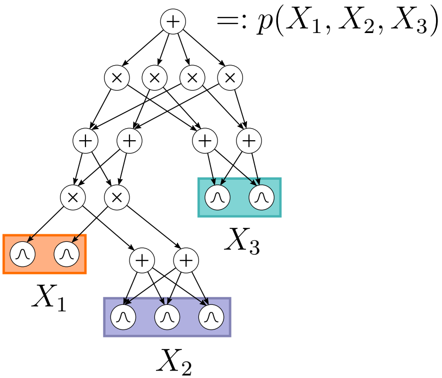
Plus: Structural properties!

Smoothness

sum children have same scope

Decomposability

product children have disjoint scope



Probabilistic Circuits — Inference

Example: Marginalization and Conditioning

$$\mathbf{X} = \mathbf{X}_q \cup \mathbf{X}_m \cup \mathbf{X}_e$$

$$p(\mathbf{X}_q \,|\, \mathbf{x}_e) = \frac{\int p(\mathbf{X}_q, \mathbf{x}'_m, \mathbf{x}_e)\mathrm{d}\mathbf{x}'_m}{\int\int p(\mathbf{x}'_q, \mathbf{x}'_m, \mathbf{x}_e)\mathrm{d}\mathbf{x}'_q\mathrm{d}\mathbf{x}'_m}$$

Probabilistic Circuits — Inference

Example: Marginalization and Conditioning

$$\mathbf{X} = \mathbf{X}_q \cup \mathbf{X}_m \cup \mathbf{X}_e$$

$$p(\mathbf{X}_q \mid \mathbf{x}_e) = \frac{\int p(\mathbf{X}_q, \mathbf{x}'_m, \mathbf{x}_e) \mathrm{d}\mathbf{x}'_m}{\int \int p(\mathbf{x}'_q, \mathbf{x}'_m, \mathbf{x}_e) \mathrm{d}\mathbf{x}'_q \mathrm{d}\mathbf{x}'_m}$$

Smoothness and decomposability $\Rightarrow$ Single bottom up pass!

Probabilistic Circuits — Inference

Example: Marginalization and Conditioning

$$\mathbf{X} = \mathbf{X}_q \cup \mathbf{X}_m \cup \mathbf{X}_e$$

$$p(\mathbf{X}_q \,|\, \mathbf{x}_e) = \frac{\int p(\mathbf{X}_q, \mathbf{x}'_m, \mathbf{x}_e) \mathrm{d}\mathbf{x}'_m}{\int \int p(\mathbf{x}'_q, \mathbf{x}'_m, \mathbf{x}_e) \mathrm{d}\mathbf{x}'_q \mathrm{d}\mathbf{x}'_m}$$

Smoothness and decomposability $\Rightarrow$ Single bottom up pass!

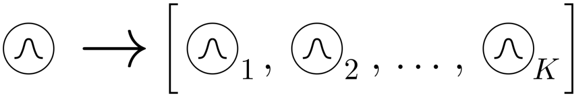
Check out our AAAI tutorial on Probabilistic Circuits!

Upcoming tutorials at ECAI, ECML/PKDD, IJCAI!

The Problem

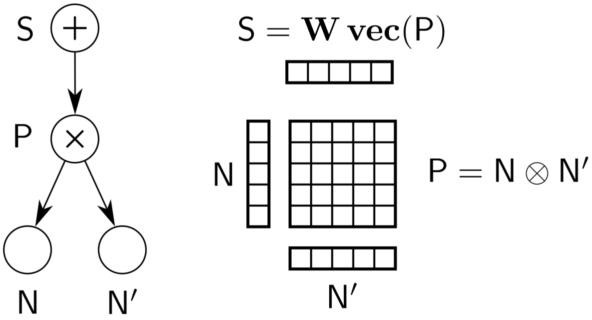
Einsum Networks

Step I – Vectorize Nodes

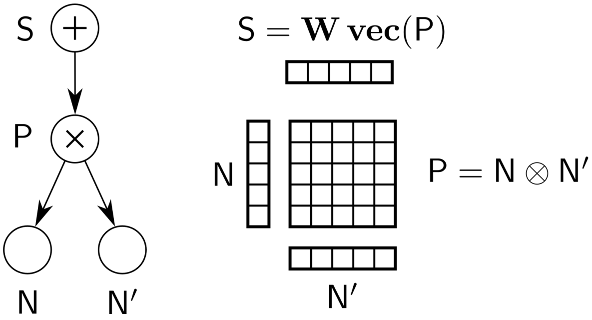


$$\oplus \longrightarrow \left[\oplus_1, \oplus_2, \dots, \oplus_K \right]$$

Step II – The Basic Einsum Operation

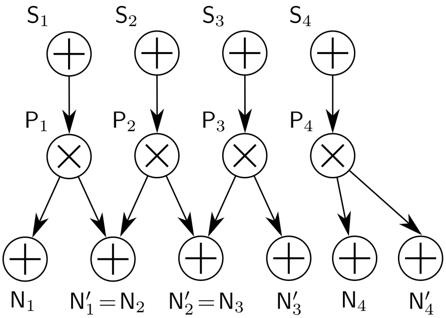


Step II – The Basic Einsum Operation

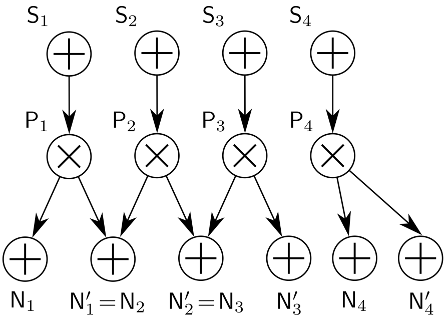


$\mathsf{S}_k = W_{kij} \mathsf{N}_i \mathsf{N}'_j$ single `einsum`-operation

Step III – Einsum Layers



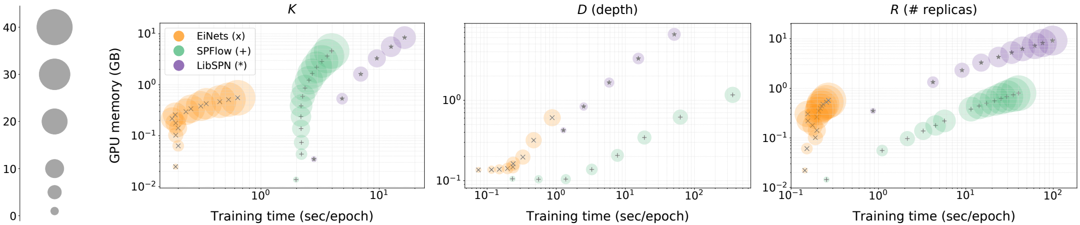
Step III – Einsum Layers



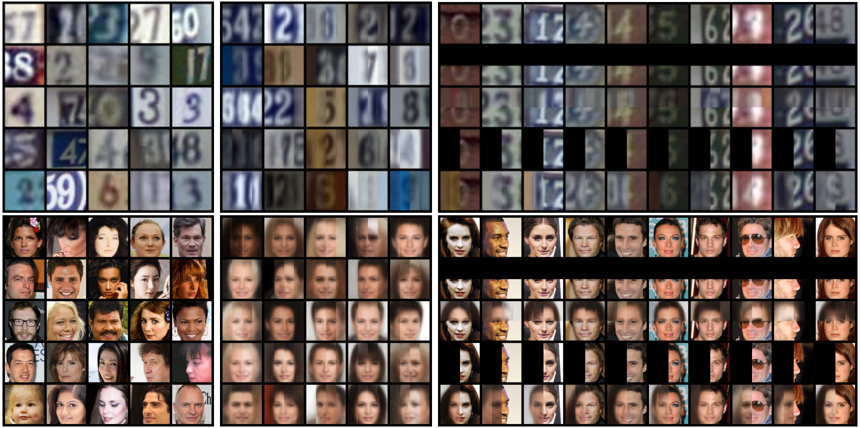
$\mathbf{S}_{lk} = \mathbf{W}_{lkij}\mathbf{N}_{li}\mathbf{N}'_{lj}$ single `einsum`-operation

Results

Runtime and Memory Comparison



Generative Image Models



Conclusion

- PCs: intersection of classical graphical models and neural networks.
- Crucial advantage: many exact inference routines.
- But, they used to be painful to scale.
- In this paper, we made a big step to close the gap. More to come!

`https://github.com/cambridge-mlg/EinsumNetworks`
`https://github.com/SPFlow/SPFlow`