Strudel: Learning Structured-Decomposable Probabilistic Circuits

Meihua Dang
University of California, Los Angeles

Antonio Vergari
University of California, Los Angeles

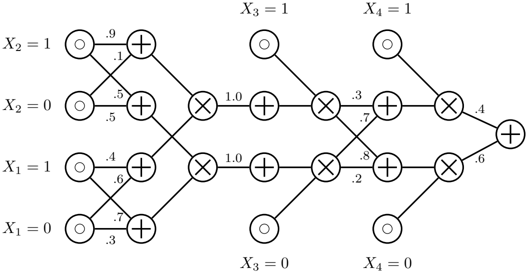
Guy Van den Broeck
University of California, Los Angeles

*September 23th, 2020 - **Probabilistic Graphical Models (PGM)***

Learning Structured-Decomposable ***Probabilistic Circuits***

Probabilistic circuits (PCs) are ***tractable probabilistic models***

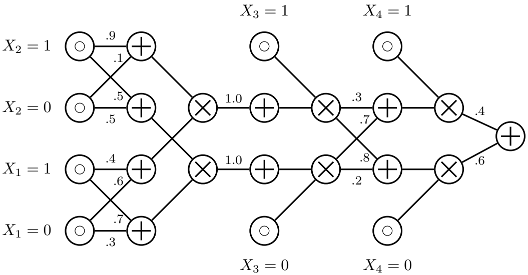
$\Longrightarrow$ ***exact*** *and* ***efficient*** *inference!*



Learning Structured-Decomposable **Probabilistic Circuits**

Probabilistic circuits (PCs) are ***tractable probabilistic models***

$\Longrightarrow$ ***exact*** *and* ***efficient*** *inference!*



known under many different names: ***SPNs***, ***PSDDs***, ***CNets***,...

$\Longrightarrow$ *checkout Guy's tutorial tomorrow!*

Learning ***Structured-Decomposable*** *Probabilistic Circuits*

To answer different classes of ***tractable inferences***, PCs are required to have certain ***structure properties*** [1]

	EVI	*MAR*	*CON*	*MAP*
smooth		✓	✓	
decomposable		✓	✓	✓
deterministic				✓

[1] Probabilistic circuits tutorial @ ECML-PKDD available at `https://www.youtube.com/watch?v=2RAG5-L9R70`

Learning Structured-Decomposable Probabilistic Circuits

To answer different classes of ***tractable inferences***, PCs are required to have certain ***structure properties*** [1]

	EVI	*MAR*	*CON*	*MAP*	*KLD*	*Multiply*	*Expected Predictions*
smooth		✓	✓				
decomposable		✓	✓	✓			
deterministic				✓			

[1] Probabilistic circuits tutorial @ ECML-PKDD available at `https://www.youtube.com/watch?v=2RAG5-L9R70`

Learning ***Structured-Decomposable*** *Probabilistic Circuits*

To answer different classes of ***tractable inferences***, PCs are required to have certain ***structure properties*** [1]

	EVI	MAR	CON	MAP	KLD	Multiply	Expected Predictions
smooth		✓	✓		✓	✓	✓
decomposable		✓	✓	✓	✓	✓	✓
deterministic				✓	✓		
struct.-decomp.					✓	✓	✓

For advanced queries...we need ***structured-decomposable*** PCs!

[1] Probabilistic circuits tutorial @ ECML-PKDD available at `https://www.youtube.com/watch?v=2RAG5-L9R70`

Learning *Structured-Decomposable Probabilistic Circuits*

1. Initial structure
2. Generate candidate improvements
3. Score candidates $\Rightarrow$ *expensive!*
4. go to 2.

LearnPSDD *[Liang et al. 2017]*

Learning *Structured-Decomposable Probabilistic Circuits*

1. Initial structure
2. Generate candidate improvements
3. Score candidates

   $\Longrightarrow$ *expensive!*
4. go to 2.

LearnPSDD *[Liang et al. 2017]*

1. Better initial structure
2. Generate single candidate

   $\Longrightarrow$ *fast heuristics!*
3. No scoring!
4. go to 2.

Strudel

Learning *Structured-Decomposable Probabilistic Circuits*

1. Initial structure
2. Generate candidate improvements
3. Score candidates $\Longrightarrow$ *expensive!*
4. go to 2.

LearnPSDD *[Liang et al. 2017]*

1. Better initial structure
2. Generate single candidate $\Longrightarrow$ *fast heuristics!*
3. No scoring!
4. go to 2.

Strudel

faster single structure learning $\longrightarrow$ fast mixture models!

Outline

Probabilistic Circuits

Scalable Learning Algorithm

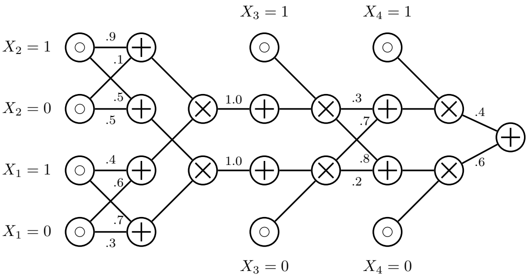
- Structured Decomposability and Learning Algorithm
- Scale More with Mixtures

Advanced Queries

Conclusions

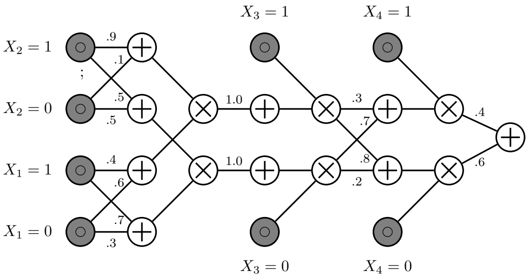
Probabilistic Circuits (PCs)

PCs encode joint distributions via computational graphs



Probabilistic Circuits (PCs)

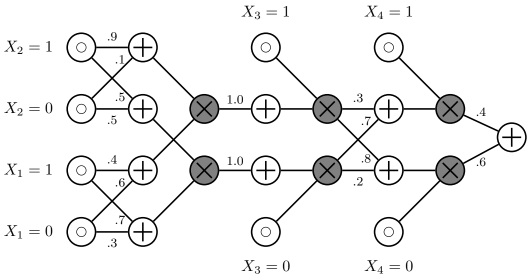
PCs encode joint distributions via computational graphs



Input nodes are tractable distributions, *e.g., indicator functions* $\mathrm{p}(X_i = 1) = [X_i = 1]$

Probabilistic Circuits (PCs)

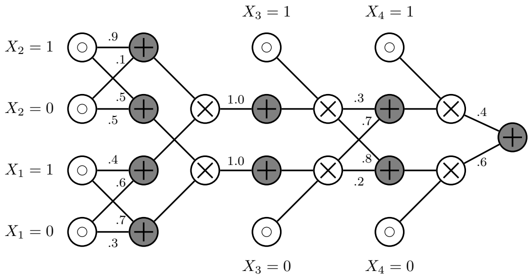
PCs encode joint distributions via computational graphs



Product nodes are factorizations $\prod_{c \in \mathsf{in}(n)} \mathrm{p}_c(\mathbf{x})$

Probabilistic Circuits (PCs)

PCs encode joint distributions via computational graphs

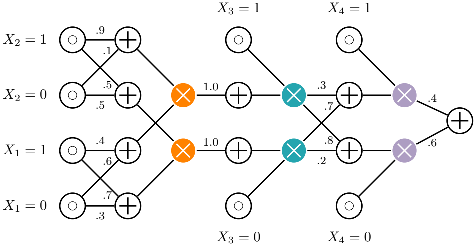


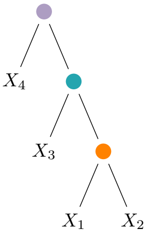
Sum nodes are mixture models $\sum_{c \in \mathsf{in}(n)} \theta_{n,c}\, \mathrm{p}_c(\mathbf{x})$

- Probabilistic Circuits
- Scalable Learning Algorithm
    - Structured Decomposability and Learning Algorithm
    - Scale More with Mixtures
- Advanced Queries
- Conclusions

Structured-Decomposable PCs

Every product node over the same set of variables, ***decomposes in the same way...***





...and all allowed ways to decompose are encoded in a ***vtree***

Initialization

How to enforce structured-decomposability?

Start from some ***simple but expressive*** distribution, then ***compile*** it into a PC with a vtree

Q: which distribution?

A: Use a ***Chow-Liu tree*** as the “best” initial PC

- ***easy*** compilation to structured-decomposable PCs
- ***best tree model*** according to KLD *[Chow et al. 1968]*

Structure refinement

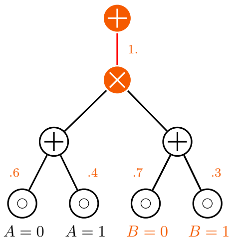
How to improve the current structure?

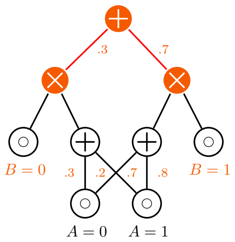
Q: How to build a ***more expressive PC***...while ***preserving properties***?

Structure refinement

How to improve the current structure?

Q: How to build a ***more expressive PC***...while ***preserving properties***?





A: by applying the ***split operator***!

Structure refinement

How to split?

How to pick an ***edge*** to split?

$\Longrightarrow$ *take the edge with most samples "flowing" through it*

How to pick a ***variable*** to condition on?

$\Longrightarrow$ *take the variable with the strongest dependencies among others*

Strudel: accurate PCs

DATASET	LEARNPSDD	STRUDEL	SELSPN	SEASPN
NLTCS	-6.03	**-6.06**	-6.03↓	-6.07↑
MSNBC	-6.04	-6.05	-6.04↓	-6.06↑
KDD	-2.17	-2.17	-2.16↓	-2.16↓
PLANTS	-13.49	-13.72	-12.97↓	-13.12↓
AUDIO	-41.51	-42.26	-41.23↓	-40.13↓
JESTER	-54.63	-55.30	-54.38↓	-53.08↓
NETFLIX	-58.53	-58.68	-57.98↓	-56.91↓
ACCIDENTS	-28.291	-29.46	-26.88↓	-30.02↓
RETAIL	-10.92	**-10.90**	-10.88↓	-10.97↑
PUMSB-STAR	-25.40	**-25.28**	-22.66↓	-28.69↓
DNA	-83.02	-87.10	-80.44↓	-81.76↓
KOSAREK	-10.99	**-10.98**	-10.85↓	-11.00↓
MSWEB	-9.93	-10.19	-9.93↓	-10.25↓
BOOK	-36.06	**-35.77**	-36.01↓	-34.91↓
EACHMOVIE	-55.41	-59.47	-55.73↓	-53.28↓
WEBKB	-161.42	**-160.50**	-158.52↓	-157.88↓
ROUTERS-52	-93.30	**-92.38**	-88.48↓	-86.38↓
20NEWS-GRP	-160.43	**-160.77**	-158.68↓	-153.63↓
BBC	-260.24	**-258.96**	-259.35↑	-253.13↓
AD	-20.13	**-16.52**	-16.94↑	-16.77↑

Now we have good single model

Fit the data better and scale more: ***mixture models***

Probabilistic Circuits

Scalable Learning Algorithm

Structured Decomposability and Learning Algorithm

Scale More with Mixtures

Advanced Queries

Conclusions

Strudel: Mixtures of PCs

Difficulties:

- **Structured-decomposable**

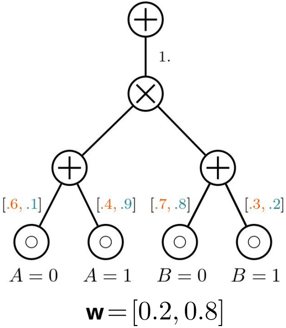
  each component being struct.-decomp....conform to the ***same vtree***

- **Scalable**

  memory, evaluation, learning...

Strudel: Mixtures of PCs

Solution : learn the mixtures ***sharing the same structure***, but having different parameters



- Simple learning and memory efficiency
- Computational efficiency by ***circuit flows***!

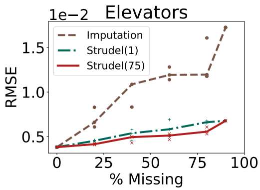
  $\text{logsumexp}(f_{\mathcal{M}}(\mathbf{x})^T \cdot \log(\boldsymbol{\Theta}) + \log(\boldsymbol{w}))$

  E.g., #component 100; #paras each 10,000; N 100,000 : (100,000, 10,000) (10,000, 100) + (1, 100)*

  $\Longrightarrow$ *stop by at poster session :)*

Advanced Queries

Compute the ***expected predictions*** *[Khosravi et al. 2019]*



⇒ ***single*** *probabilistic circuits* ***perform better*** *than the baseline, the* ***mixtures*** *also help further* ***reduce the error***

Conclusions

Probabilistic Circuits can answer advanced queries

$\Rightarrow$ ***structured-decomposable*** *PCs*

We propose a simple and scalable structure learning algorithm

Scale learning:

- Fast heuristic
- Cheap mixtures of circuits sharing the structures
- ***Circuit flows***

References I

- Chow, C and C Liu (1968). “Approximating discrete probability distributions with dependence trees”. In: *IEEE Transactions on Information Theory* 14.3, pp. 462–467.
- Liang, Yitao, Jessa Bekker, and Guy Van den Broeck (2017). “Learning the Structure of Probabilistic Sentential Decision Diagrams”. In: *UAI*.
- Khosravi, Pasha et al. (2019). “Tractable Computation of the Moments of Predictive Models”. In: *Proceedings of NeurIPS*.

Technical Details: Determinism and Circuit Flows

Determinism : for every sum node, at most one of its input is non-zero

Circuit flows : encodes which parameters are activated by different input configurations

Benefits:

- Closed-form MLE
- Fast inference $f_{\mathcal{C}}(\mathbf{x})^T \cdot \log(\boldsymbol{\theta})$

$$[\,0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 0\ \ \ldots] \cdot$$

$$\log([.4\ .6\ .3\ .7\ .8\ .2\ \ldots])$$

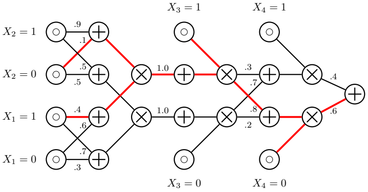


Figure: For input configuration $X_1\!=\!1, X_2\!=\!0, X_3\!=\!1, X_4\!=\!0$, red colors indicate all the **active** edges