Automatic Bayesian Density Analysis

Antonio Vergari
Max-Plank-Institute IS

Alejandro Molina
TU Darmstadt

Robert Peharz
University of Cambridge

Zoubin Ghahramani
University of Cambridge
Uber AI Labs

Kristian Kersting
TU Darmstadt

Isabel Valera
Max-Plank-Institute IS

*31st January 2019 - **AAAI19*** **Honolulu**

In a nutshell

or why should you care about SPNs

A short tutorial on

***Sum-Product Networks* (SPNs)** *[Poon and Domingos 2011]*

an appealing deep and ***tractable density estimator*** allowing ***exact inference***

Plus, SPNs can be ***interpreted*** as ***hierarchical latent variable*** (LV) probabilistic models, ***Multi-layer perceptrons***, and special kind of ***Arithmetic Circuits*** and even ***Bayesian Networks***

Briefly reviewing representation capabilities, ***inference*** and ***learning*** schemes with SPNs

Ultimately touching some ***applications***, extensions and research ideas

Notation

$X, Y, Z, \ldots$	random variables (RVs)
$\mathsf{val}(X)$	support of RV X
$x \sim X$	x is drawn/distributed according to X or $x \in \mathsf{val}(X)$
$\mathbf{X} = (X_1, X_2, \ldots, X_n)$	ordered set of RVs
$\mathbf{x} \sim \mathbf{X}, \mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$	multivariate sample from $\mathbf{X}$
$\mathbf{x}_{\mid \mathbf{Y}}$	restriction of sample $\mathbf{x}$ to RVs $\mathbf{Y} \subset \mathbf{X}$
$p_{\mathbf{X}}, p$	PMF or PDF over RVs $\mathbf{X}$
$p_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y})$, $p(\mathbf{x}, \mathbf{y})$	joint distribution over $\mathbf{X} \cup \mathbf{Y}$
$p_{\mathbf{Y}\mid\mathbf{x}}(\mathbf{y}\mid\mathbf{x})$, $p(\mathbf{Y}\mid\mathbf{x})$	conditional probability distribution

Density estimation

Unsupervisedly learning an estimator for the joint probability distribution $p(\mathbf{X})$ from a set of i.i.d. samples $\mathcal{D} = \{\mathbf{x}^i\}_{i=1}^m$ over random variables (RVs) $\mathbf{X} = \{X_1, \ldots, X_n\}$[3]

Given such an estimator, one uses it to ***answer probabilistic queries*** about configurations on $\mathbf{X}$, i.e. to do ***inference***.

$\Longrightarrow$ *most ML task can be reframed as probabilistic inference!*

The **inherent trade-off** in density estimation: balancing

- the ***representation expressiveness*** of the model to learn
- the ***cost of performing inference*** on it
- and the ***cost of learning*** such a model

(Different kinds of) Inference

- complete evidence (EVI) $p(\mathbf{X}=\mathbf{x})$
- marginals (MAR) $p(\mathbf{E}=\mathbf{e}), \quad \mathbf{E}\subset\mathbf{X}$
- conditionals (CON) $p(\mathbf{Q}|\mathbf{E}), \quad \mathbf{Q},\mathbf{E}\subset\mathbf{X}, \mathbf{Q}\cap\mathbf{E}=\emptyset$
- Most Probable Explanation (MPE) $\arg\max_{\mathbf{q}\sim\mathbf{Q}} p(\mathbf{q}|\mathbf{E}), \quad \mathbf{Q}\cup\mathbf{E}=\mathbf{X}, \mathbf{Q}\cap\mathbf{E}=\emptyset$
- Maximum A Posteriori (MAP) $\arg\max_{\mathbf{q}\sim\mathbf{Q}} \sum_{\mathbf{h}\sim\mathbf{H}} p(\mathbf{q},\mathbf{h}|\mathbf{E})$
  $\mathbf{Q}\cup\mathbf{H}\cup\mathbf{E}=\mathbf{X}, \mathbf{Q}\cap\mathbf{H}=\emptyset, \mathbf{Q}\cap\mathbf{E}=\emptyset, \mathbf{H}\cap\mathbf{E}=\emptyset$
- partition function computation $Z=\sum_{\mathbf{x}}\phi(\mathbf{x})$
- sampling (SAM): generate independent samples from p

We strive for ***exact*** inference, computable in ***tractable*** time, i.e. polynomial in $|\mathbf{X}|$

SPNs: exact and tractable inferences

Let $\mathbf{S}^{\oplus}$ (resp. $\mathbf{S}^{\otimes}$) be the set of all sum (resp. product) nodes in an SPN S, then

- S is ***complete*** iff $\forall n \in \mathbf{S}^{\oplus}, \forall c_1, c_2 \in \mathsf{ch}(n) : \mathsf{sc}(c_1) = \mathsf{sc}(c_2)$
- S is ***decomposable*** iff $\forall n \in \mathbf{S}^{\otimes}, \forall c_1, c_2 \in \mathsf{ch}(n) : \mathsf{sc}(c_1) \cap \mathsf{sc}(c_2) = \emptyset$

If S is complete and decomposable, it is ***valid***, and it exactly computes, in **time linear w.r.t. to its size** $|S|$[4][5]:

$\Rightarrow$ *caveat: $|S|$ shall be polynomial in $|\mathbf{X}|$...*

- complete evidence $p(\mathbf{X} = \mathbf{x})$
- marginals $p(\mathbf{Q} = \mathbf{q})$, conditionals $p(\mathbf{Q} = \mathbf{q}|\mathbf{e})$
- partition function $\mathbf{Z}$

An SPN S is ***selective*** [6], iff $\forall \mathbf{x}^i \sim \mathbf{X}, \forall n \in \mathbf{S}^{\oplus} : |\{c \,|\, c \in \mathsf{ch}(n) : S_c(\mathbf{x}^i) > 0\}| \leq 1$

$\Rightarrow$ |S| ***MPE inference***, ***assignments*** in time linear to $|S|$[7]

Trivia: Interpreting SPNs

An SPN encodes a ***multi-linear function*** in a compact data structure [8]

$\Longrightarrow$ *a giant (network) polynomial over* $\mathbf{X}$*!*

SPNs are **not PGMs**! They are ***computational graphs***

$\Longrightarrow$ *equivalent to Arithmetic Circuits for finite discrete domains* [9]

SPNs are ***hierarchical LV probabilistic*** models

$\Longrightarrow$ *one can think of them as the* ***deep version of mixture models***

SPNs are peculiar ***feedforward neural networks***

$\Longrightarrow$ *reparameterizable as* ***sparse***, ***constrained***, ***fully-probabilistic*** *MLPs* [10]

References I

- Choi, Arthur and Adnan Darwiche (2017). “On Relaxing Determinism in Arithmetic Circuits”. In: *Proceedings of ICML*, pp. 825–833.
- Darwiche, Adnan (2009). *Modeling and Reasoning with Bayesian Networks*. Cambridge.
- Peharz, Robert, Robert Gens, and Pedro Domingos (2014). “Learning Selective Sum-Product Networks”. In: *Workshop on Learning Tractable Probabilistic Models*. LTPM.
- Poon, Hoifung and Pedro Domingos (2011). “Sum-Product Networks: a New Deep Architecture”. In: *UAI 2011*.
- Rooshenas, Amirmohammad and Daniel Lowd (2014). “Learning Sum-Product Networks with Direct and Indirect Variable Interactions”. In: *Proceedings of ICML 2014*.
- Vergari, Antonio, Nicola Di Mauro, and Floriana Esposito (2016). “Visualizing and Understanding Sum-Product Networks”. In: *preprint arXiv*. URL: `https://arxiv.org/abs/1608.08266`.

out