Sum-Product Networks

Advances in Sum-Product Networks for Probabilistic Modelling

Martin Trapp

July 4, 2019

TU Graz

Austrian Research Institute for Artificial Intelligence
OFAI

Sum-Product Networks

Motivation

Bayesian networks?

- Compact specification of the joint distribution
- Exact inference is intractable in the general case

Mixture models?

- GMMs are universal densities approximators in theory
- Exact inference is tractable in GMMs

Motivation

Bayesian networks?

- Compact specification of the joint distribution
- Exact inference is intractable in the general case

Mixture models?

- GMMs are universal densities approximators in theory
- Exact inference is tractable in GMMs

Sum-Product Networks

- Universal densities approximators
- Compact specification of the joint distribution
- Exact inference is tractable if the network is valid

Sum-Product Networks

Sum-Product Network (SPN) [Poon 2011]

An SPN is a $\boxed{\text{rooted directed acyclic graph}}$ consisting of:

leaf

$$p_\mathrm{L}(x_{n,d} \mid \theta_\mathrm{L})$$

sum nodes

$$\sum_{\mathrm{C} \in \mathbf{ch}(\mathrm{S})} w_{\mathrm{S},\mathrm{C}}\, p_\mathrm{C}(\mathbf{x}_n \mid \cdot)$$

product nodes

$$\prod_{\mathrm{C} \in \mathbf{ch}(\mathrm{P})} p_\mathrm{C}(\mathbf{x}_{n,\mathbf{sc}(\mathrm{C})} \mid \cdot)$$

Weights $w_{\mathrm{S},\mathrm{C}}$ are $\boxed{\text{non-negative}}$ and $\sum_{\mathrm{C} \in \mathbf{ch}(\mathrm{S})} w_{\mathrm{S},\mathrm{C}} = 1 \quad , \forall \mathrm{S} \in \mathcal{S}$

Interpretations of SPNs

SPNs can be seen as:

- A generalisation of mixture models *(If the SPN is shallow)*
- Deep structured mixture models [Zhao 2016]

$$\mathrm{f}(\mathbf{x}_\mathrm{n} \mid \theta) = \sum_{k=1}^{\kappa} \prod_{w_{S,C} \in \mathcal{T}_k} w_{S,C} \prod_{\mathrm{L} \in \mathcal{T}_k} p(\mathbf{x}_n \mid \theta_\mathrm{L}) \tag{1}$$

- Multi-linear feed-forward neural networks with non-linear inputs

Validity Conditions

Completeness
For all sum nodes, all children have the $\boxed{\text{same scope}}$.

Validity Conditions

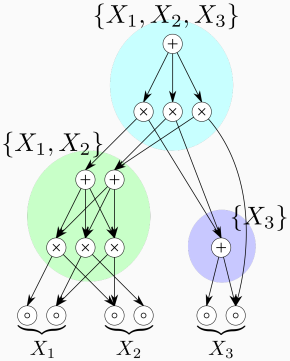
Completeness
For all sum nodes, all children have the $\boxed{\text{same scope}}$.

Decomposability
For all product nodes, the children's $\boxed{\text{scopes are disjoint}}$.

Validity Conditions

Completeness
For all sum nodes, all children have the $\boxed{\text{same scope}}$.

Decomposability
For all product nodes, the children's $\boxed{\text{scopes are disjoint}}$.



Learning Sum-Product Networks

Parameter Learning in SPNs

Generative Learning [Poon 2011, Peharz 2016]

$$\mathcal{L}(\theta \mid \mathcal{X}) = \sum_{n=1}^{N} \log \mathsf{f}(\mathbf{x}_\mathsf{n} \mid \theta) - \log \mathsf{f}(* \mid \theta)\,, \quad \mathbf{x}_n \in \mathbb{R}^D \tag{2}$$

Parameter Learning in SPNs

Generative Learning [Poon 2011, Peharz 2016]

$$\mathcal{L}(\theta \mid \mathcal{X}) = \sum_{n=1}^{N} \log \mathrm{f}(\mathbf{x}_\mathrm{n} \mid \theta) - \log \mathrm{f}(* \mid \theta)\,, \quad \mathbf{x}_n \in \mathbb{R}^D \tag{2}$$

Supervised Learning [Gens 2012]

$$\mathcal{L}(\theta, \boldsymbol{y} \mid \mathcal{X}) = \sum_{n=1}^{N} \log \mathrm{f}(\mathbf{x}_\mathrm{n}, \mathrm{y}_\mathrm{n} \mid \theta) - \log \mathrm{f}(\mathbf{x}_\mathrm{n} \mid \theta)\,, \quad \mathbf{x}_n \in \mathbb{R}^D,\ y_n \in \mathbb{R} \tag{3}$$

Parameter Learning in SPNs

Generative Learning [Poon 2011, Peharz 2016]

$$\mathcal{L}(\theta \mid \mathcal{X}) = \sum_{n=1}^{N} \log \mathrm{f}(\mathbf{x}_n \mid \theta) - \log \mathrm{f}(* \mid \theta)\,, \quad \mathbf{x}_n \in \mathbb{R}^D \tag{2}$$

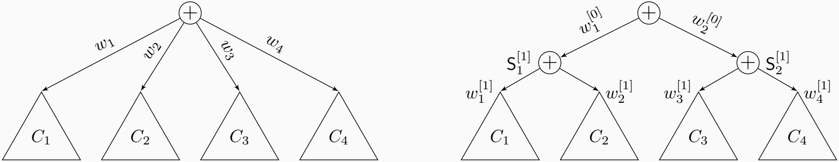
Supervised Learning [Gens 2012]

$$\mathcal{L}(\theta, \boldsymbol{y} \mid \mathcal{X}) = \sum_{n=1}^{N} \log \mathrm{f}(\mathbf{x}_n, \mathrm{y}_n \mid \theta) - \log \mathrm{f}(\mathbf{x}_n \mid \theta)\,, \quad \mathbf{x}_n \in \mathbb{R}^D,\ y_n \in \mathbb{R} \tag{3}$$

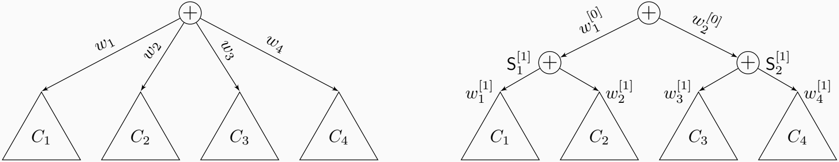
(Safe) Semi-Supervised Learning [Trapp 2017]

$$\mathrm{CPLE} = \operatorname*{argmax}_{\theta \in \Theta} \operatorname*{arg\,min}_{\boldsymbol{q} \in \Delta^M_{K-1}} \mathcal{L}(\theta, \boldsymbol{y}, \boldsymbol{q} \mid \mathcal{X}, \mathcal{U}) - \mathcal{L}(\theta^+, \boldsymbol{y}, \boldsymbol{q} \mid \mathcal{X}, \mathcal{U}) \tag{4}$$

Overparameterization in SPNs [Trapp 2019]



Overparameterization in SPNs [Trapp 2019]



$$\begin{aligned} w_k^{(t+1)} &= w_\gamma^{[0](t+1)} \cdot w_k^{[1](t+1)} \\ &\approx w_k^{(t)} + \eta (w_\gamma^{[0](t)})^2 \nabla_{w_k^{(t)}} + \eta \nabla_{w_\gamma^{[0](t)}} (w_\gamma^{[0](t)})^{-1} w_k^{(t)} \qquad (5) \\ &= w_k^{(t)} + \rho^{(t)} \nabla_{w_k^{(t)}} + \lambda^{(t)} w_k^{(t)} \qquad (6) \end{aligned}$$

If η is small and all weights are initialised close to zero.

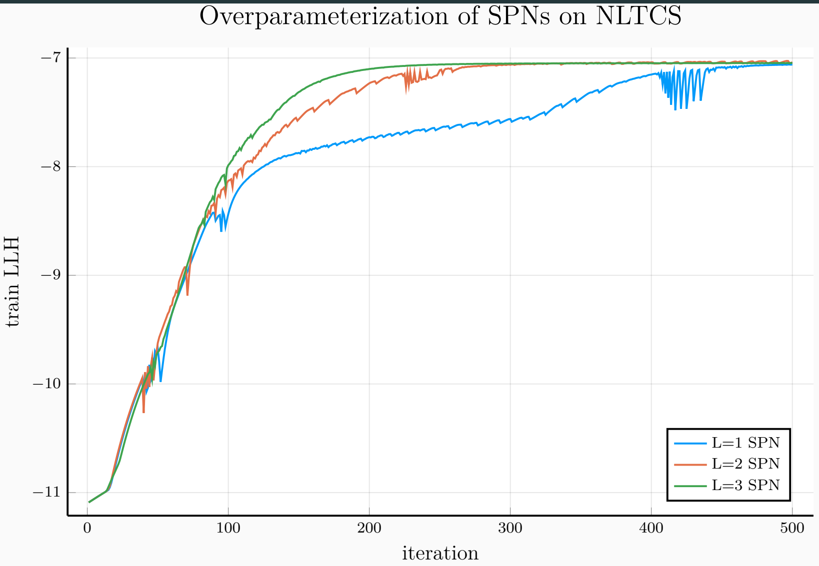
Overparameterization in SPNs [Trapp 2019]

$$w_k^{(t)} \approx w_k^{(t)} + \rho^{(t)} \nabla_{w_k^{(t)}} + \left[\sum_{l=0}^{L-1} \eta \nabla_{w_{\phi(k,l)}^{[l]}} (w_{\phi(k,l)}^{[l]})^{-1} \right] w_k^{(t)} \tag{7}$$

$$= w_k^{(t)} + \rho^{(t)} \nabla_{w_k^{(t)}} + \sum_{\tau=1}^{t-1} \mu^{(t,\tau)} \nabla_{w_k^{(\tau)}} \tag{8}$$

Gradient-based optimisation of any deep tree-structured sum-product network with small (fixed) learning rate and near zero initialisation of the weights is equivalent to gradient-based optimisation with adaptive and time-varying learning rate and momentum term.

Overparameterization in SPNs [Trapp 2019]



Structure Learning in SPNs

Tailored to Images

- Poon & Gens presented a structure learning approaches tailored to images which recursively sub-divide the image into sub-regions. [Poon 2011, Gens 2012]

Structure Learning in SPNs

Tailored to Images

- Poon & Gens presented a structure learning approaches tailored to images which recursively sub-divide the image into sub-regions. [Poon 2011, Gens 2012]

Heuristic Structure Learning

- [Gens 2013] proposed a heuristic data driven approach which got adapted by many other authors later on. See `https://github.com/arranger1044/awesome-spn`

Structure Learning in SPNs

Tailored to Images

- Poon & Gens presented a structure learning approaches tailored to images which recursively sub-divide the image into sub-regions. [Poon 2011, Gens 2012]

Heuristic Structure Learning

- [Gens 2013] proposed a heuristic data driven approach which got adapted by many other authors later on. See `https://github.com/arranger1044/awesome-spn`

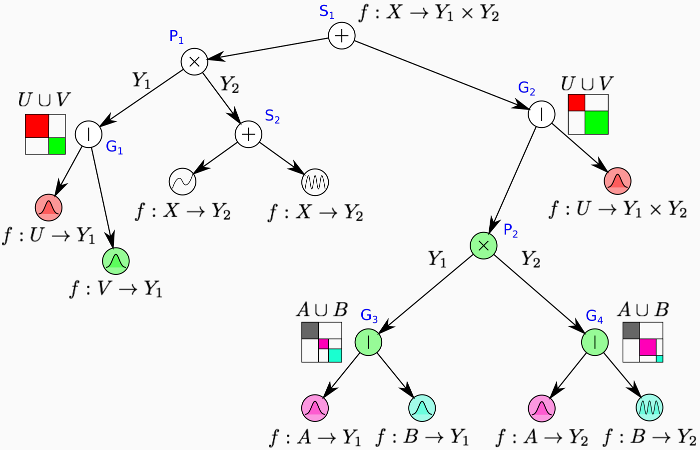
Recent Approaches

- Random tensorized SPNs. [Peharz 2019]
- Bayesian structure and parameter learning. [Trapp 2019 (ArXiv)]

Applications

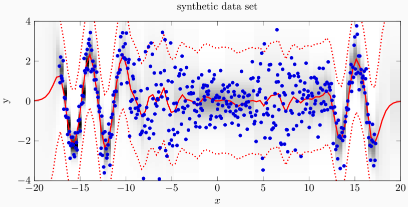
SPNs for Efficient Non-Linear Non-Parametric Regression

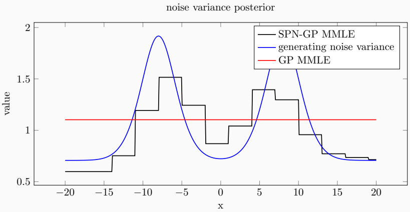
An SPN over GPs (SPN-GP) [Trapp (2018)] is a hierarchically structured mixture model which recursively combines mixtures over SPN-GPs or GPs with sub-divisions of the input space.



Experiments on non-stationary data

Left figure shows the density region of an SPN-GP and a full GP on non-stationary data with conditional heteroscedastic noise. The right figure shows the estimated variance hyper-parameter of the noise model for a GP and an SPN-GP.

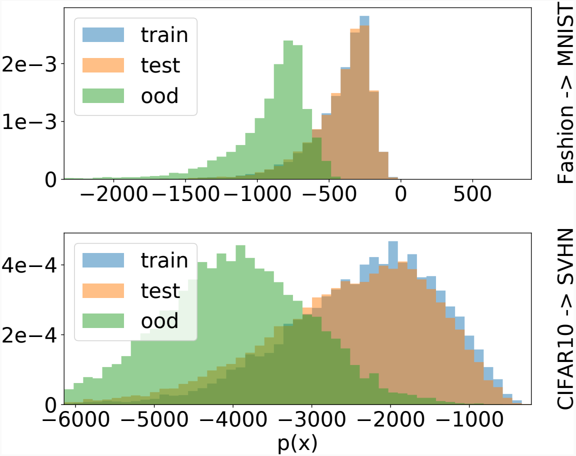




SPNs for Out-Of-Domain Detection

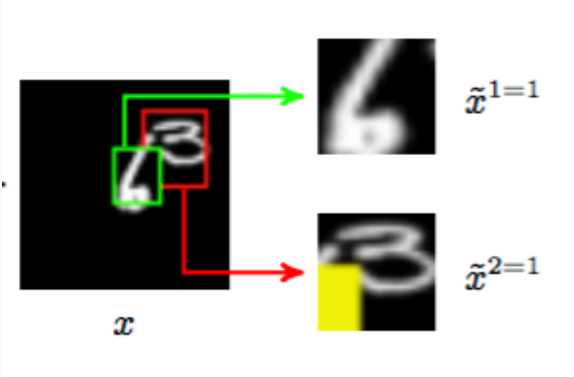
RAT-SPNs know what they don’t know

Histograms of the log-likelihoods of RAT-SPNs [Peharz 2019] on the native data set (blue: train, orange: test) and out-of-domain (ood) data set (green).



SPNs for Unsupervised Scene Understanding

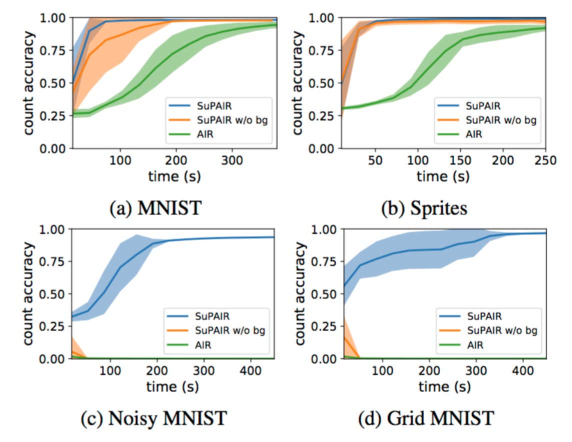
Faster Attend-Infer-Repeat with SPNs [Stelzner 2019]
SuPAIR integrates SPNs as a subroutine into the Attend-Infer-Repeat (AIR) framework. SuPAIR learns an order of magnitude faster than AIR, treats object occlusions in a consistent manner and allows for the inclusion of a background noise model.



SPNs for Unsupervised Scene Understanding

Faster Attend-Infer-Repeat with SPNs [Stelzner 2019]

Learning progress of SuPAIR and AIR on the four datasets.



Thank you for your attention!

References

K. Stelzner, R. Peharz and K. Kersting (2019) Faster Attend-Infer-Repeat with Tractable Probabilistic Models. Proceedings of ICML.

R. Peharz, A. Vergari, K. Stelzner, A. Molina, M. Trapp, X. Shao, K. Kersting, Z. Ghahramani (2019) Random Sum-Product Networks: A Simple and Effective Approach to Probabilistic Deep Learning. Proceedings of UAI.

M. Trapp, R. Peharz H. Ge, F. Pernkopf and Z. Ghahramani (2019) Bayesian Learning of Sum-Product Networks. ArXiv.

M. Trapp, R. Peharz and F. Pernkopf (2019) Optimisation of Overparametrized Sum-Product Networks. 3rd Tractable Probabilistic Models at ICML.

M. Trapp, R. Peharz, F. Pernkopf and C. E. Rasmussen (2018) Learning Deep Mixtures of Gaussian Process Experts Using Sum-Product Networks. ArXiv.

M. Trapp, T. Madl, R. Peharz, F. Pernkopf and R. Trappl (2017) Safe Semi-Supervised Learning of Sum-Product Networks. Proceedings of UAI.

H. Zhao, P. Poupart and G. Gordon (2016) A Unified Approach for Learning the Parameters of Sum-Product Networks. Proceedings of NeurIPS.

R. Peharz, R. Gens, F. Pernkopf, P. Domingos (2016) On the Latent Variable Interpretation in Sum-Product Networks. TPAMI 39 (10).

R. Gens and P. Domingos (2013) Learning the Structure Sum-Product Networks: A New Deep Architecture. Proceedings of ICML.

R. Gens and P. Domingos (2012) Discriminative learning of Sum-Product Networks. Proceedings of NeurIPS.

H. Poon and P. Domingos (2011) Sum-Product Networks: A New Deep Architecture. Proceedings of UAI.