Online Structure Learning for Feed-Forward and Recurrent Sum-Product Networks

Agastya Kalra, Abdullah Rashwan, Wilson Hsu, Pascal Poupart
University of Waterloo, Waterloo AI Institute, Vector Institute
agastya.kalra@gmail.com, {arashwan,wwhsu,ppoupart}@uwaterloo.ca

Prashant Doshi
University of Georgia
pdoshi@cs.uga.edu

George Trimponias
Huawei Noah's Ark Lab
g.trimponias@Huawei.com

Motivation

Feature engineering replaced by architecture engineering
- But architecture design is an art (trial and error)
- Need automated way to learn structure

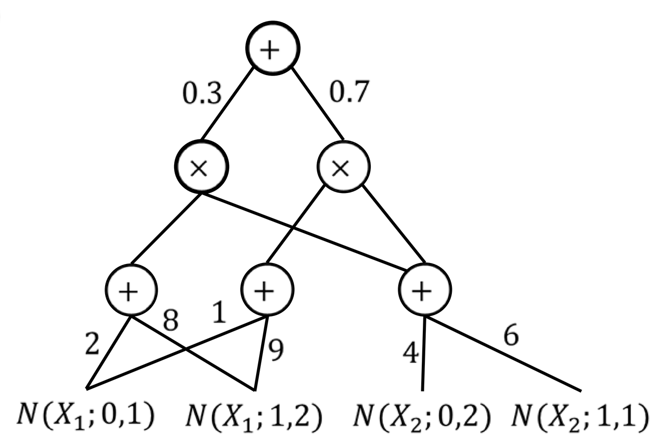
Contribution: online structure learning algorithm for Recurrent and Feed-forward SPNs

Sum Product Network

Leaves: base distributions (e.g., Gaussians)
Interior nodes: sums and products
Edges:
- Unweighted below product nodes
- Non-negative weights below sum nodes

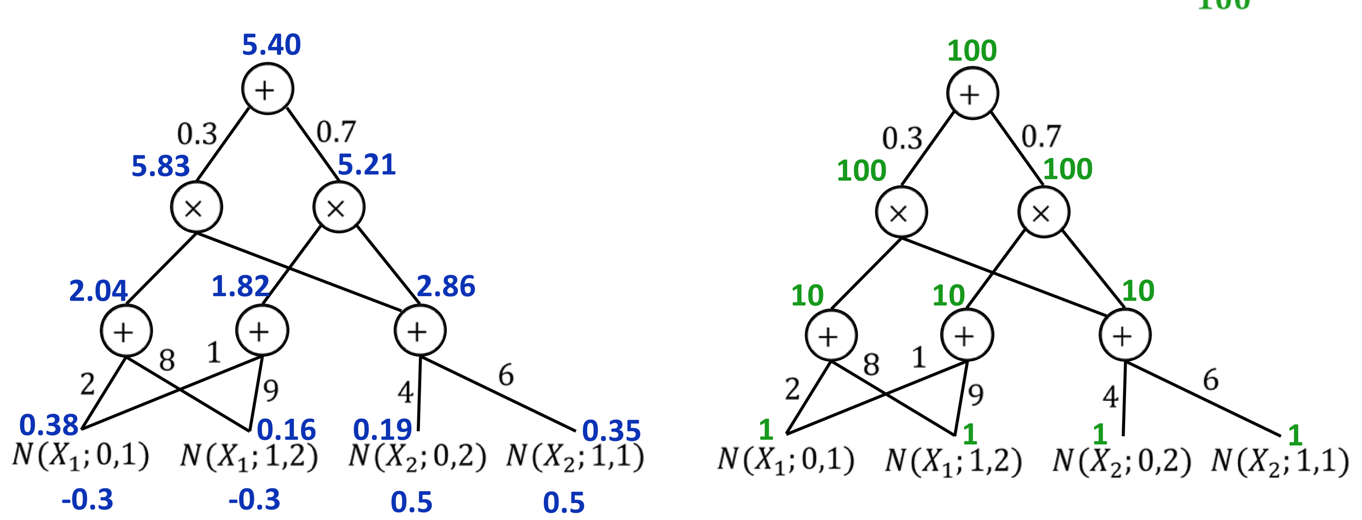
Evaluation

$$f_n(\boldsymbol{X}=\boldsymbol{x}) = \begin{cases} pdf(X_n = x_n) & \text{if } isLeaf(n) \\ \sum_i w_i f_{child_i(n)}(\boldsymbol{x}) & \text{if } isSum(n) \\ \prod_i f_{child_i(n)}(\boldsymbol{x}) & \text{if } isProduct(n) \end{cases}$$



Probabilistic Inference

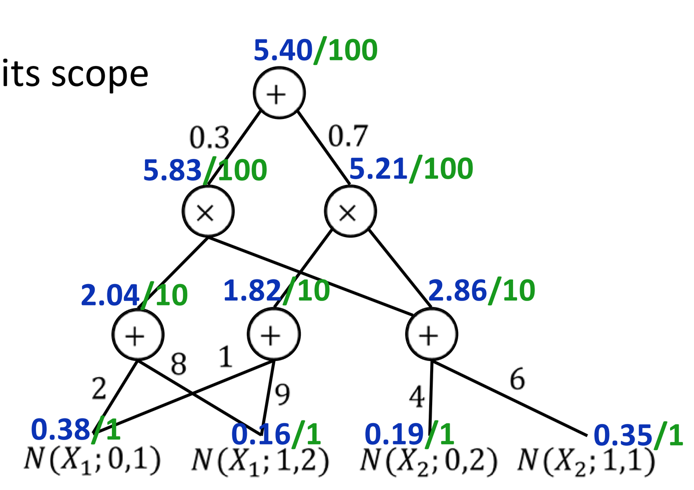
SPN represents a joint distribution, e.g., $pdf(X_1 = -0.3, X_2 = 0.5) = \frac{5.40}{100}$



Semantics

Each node computes a probability over its scope

Scope of a node: set of variables in sub-SPN rooted at that node
Decomposable product node: children with disjoint scopes
Complete/smooth sum node: children with identical scopes

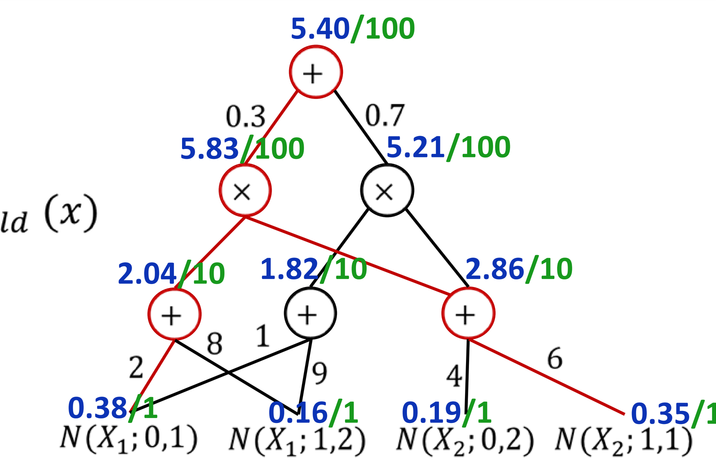


decomposability + completeness → **valid** distribution, **linear** inference

Online Parameter Update (for each data point x)

1) Determine most likely subtree
   Product node: follow all children
   Sum node: follow child with highest likelihood
   $$child^* \leftarrow argmax_{child}\ pdf_{child}(x)$$

2) Update parameters of most likely subtree
   All nodes: $n \leftarrow n+1$ where $w_{sum,child} \leftarrow \frac{n_{child}}{n_{sum}}$
   Leaves: mean $\mu_i' \leftarrow \frac{1}{n+1}(n\mu_i + x_i)$
   covariance $\Sigma_{ij}' \leftarrow \frac{1}{n}[n\Sigma_{ij} + (x_i - \mu_i)(x_j - \mu_j)] - (\mu_i' - \mu_i)(\mu_j' - \mu_j)$



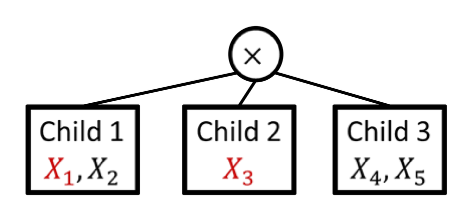
Theorem: the parameter update procedure is guaranteed to increase the likelihood of the last data point

Online Structure Learning with Running Average Update (oSLRAU)

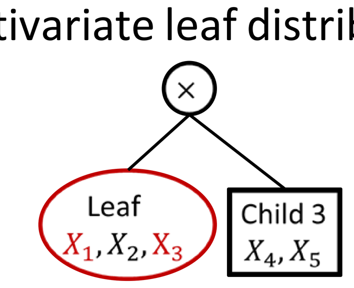
At each product node, monitor covariance in the scope of the product node

If the correlation of two variables exceeds a threshold, introduce correlation

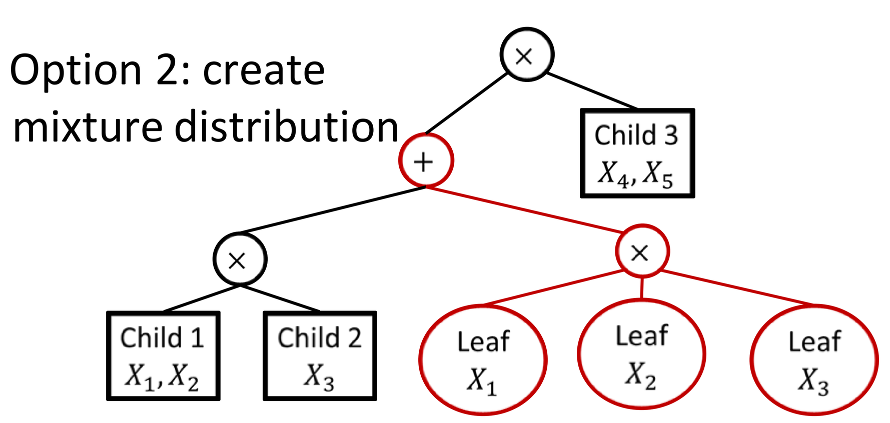
e.g. $correlation(X_1, X_3) > threshold$



Option 1: create multivariate leaf distribution

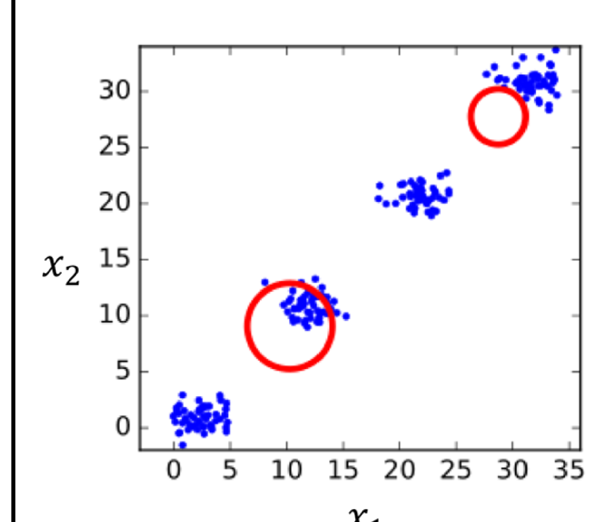


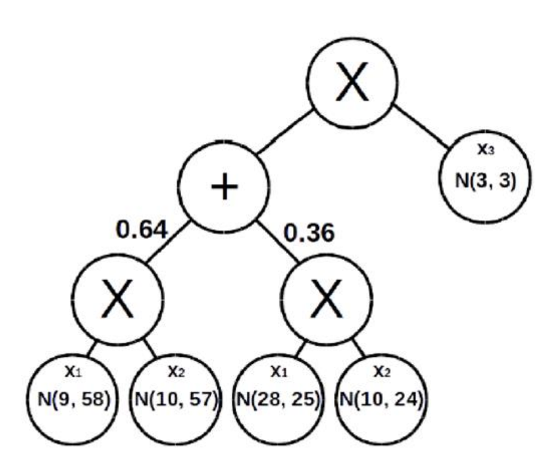
Option 2: create mixture distribution



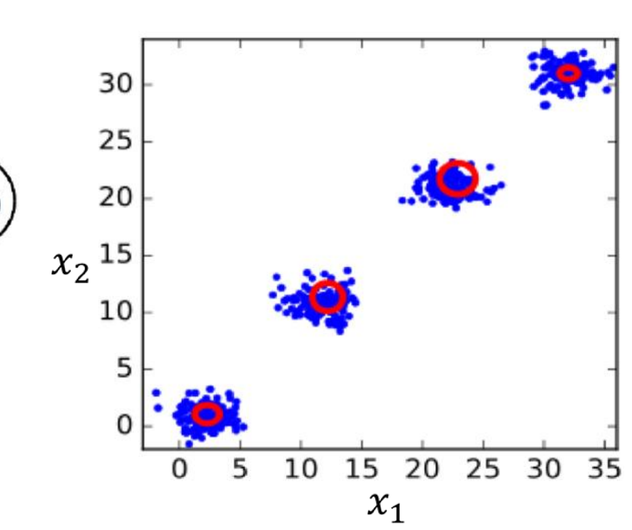
Proof of Concept

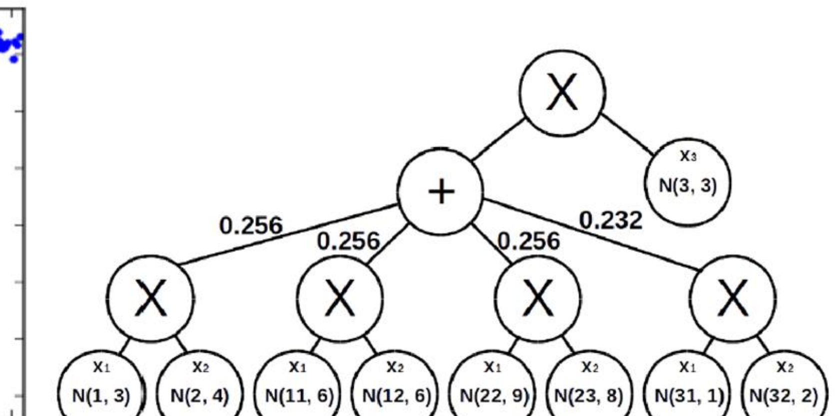
Structure learned after 200 data points





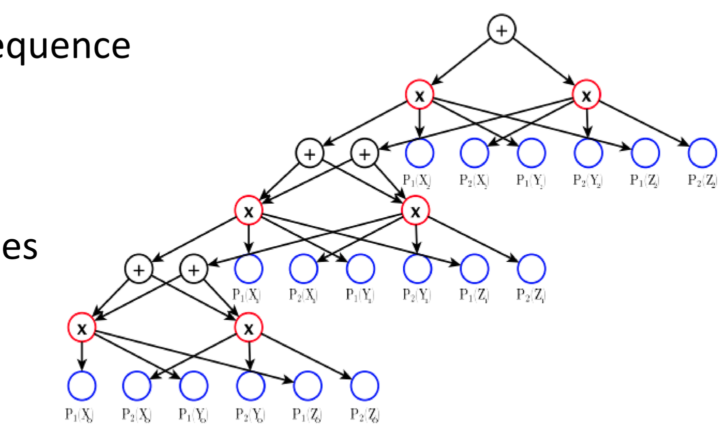
Structure learned after 500 data points





Recurrent SPNs (stacked copies of a template network)

1) Unroll network with as many template copies as length of data sequence
2) Share parameters across all template copies
3) Online parameter update: same as for feedforward networks
4) Online structure update:
   a) relabel scope of input interface nodes to binary hidden variables that allow scopes in different template copies to be the same
   b) detect correlations and update structure across all template copies in the same way



Experiments

Large Continuous Datasets: average log likelihood comparison

Datasets	Random	ILSPN	oSLRAU	RealNVP Online	RealNVP Offline
Voxforge	-33.9 ± 0.3	—	**-29.6** ± 0.0	-169.0 ± 0.6	-168.2 ± 0.8
Power	-2.83 ± 0.13	**-1.85 ± 0.02**	-2.46 ± 0.11	-18.70 ± 0.19	-17.85 ± 0.22
Network	-5.34 ± 0.03	-4.71 ± 0.16	**-4.27** ± 0.04	-10.80 ± 0.02	-7.89 ± 0.05
GasSen	-114 ± 2	—	**-102** ± 4	-748 ± 99	-443 ± 64
MSD	-538.8 ± 0.7	—	-531.4 ± 0.3	**-362.4** ± 0.4	-257.1 ± 2.03
GasSenH	-21.5 ± 1.3	-182.3 ± 4.5	**-15.6** ± 1.2	-44.5 ± 0.1	44.2 ± 0.1

oSLRAU is better than
- Incremental LearnSPN (ILSPN)
- Real Non-Volume Preserving (RealNVP)
- Random Structures for gaussian SPNs.

Recurrent Datasets: average log likelihood comparison

Dataset (#i,length,#oVars)	hillValley (600,100,1)	eegEye (14970,14,1)	libras (350,90,1)	JapanVowels (270,16,12)	ozLevel (2170,24,2)
HMM	286 ± 6.9	22.9 ± 1.8	-116.5 ± 2.2	-275 ± 13	-34.6 ± 0.3
RNN	205 ± 23	15.2 ± 3.9	-92.9 ± 12.9	-257 ± 35	**-15.3** ± 0.8
RSPN+S&S	296 ± 16.1	25.9 ± 2.1	-93.5 ± 7.2	-241 ± 12	-34.4 ± 0.4
RSPN+oSLRAU	**299.5** ± 18	**36.9** ± 1.4	**-83.5** ± 5.4	**-231** ± 12	-30.1 ± 0.4

RSPN + oSLRAU is much faster and more accurate than
- RSPN + Search and Score.

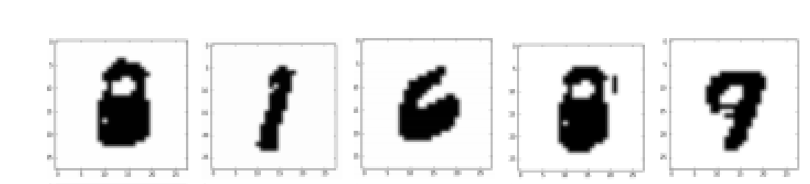
oSLRAU is the new state of the art for online structure learning in both recurrent and regular SPNs

Pruning: Removing nodes that haven't been updated in a certain timeframe

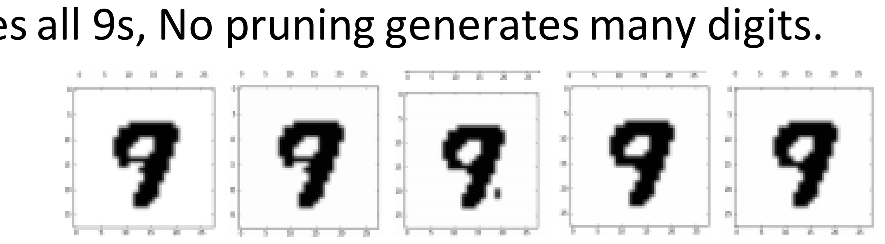
Table 2: Large datasets: comparison of oSLRAU with and without periodic pruning.

	log-likelihood		time (sec)		SPN size (# nodes)	
Dataset	no pruning	pruning	no pruning	pruning	no pruning	pruning
Power	-2.46 ± 0.11	**-2.40** ± 0.18	183	39	23360	5330
Network	-4.27 ± 0.02	**-4.20** ± 0.09	14	12	7214	5739
GasSen	**-102** ± 4	-130 ± 3	351	276	5057	1749
MSD	-527.7 ± 0.28	**-526.8** ± 0.27	74	72	1442	1395
GasSenH	**-15.6** ± 1.2	-17.7 ± 1.58	12	10	920	467

Adapting to Non-Stationary datasets -- Trained on MNIST sorted from 0-9, then generated samples from the distribution. Pruning generates all 9s, No pruning generates many digits.



Without Pruning



With Pruning

Conclusion

Contributions:
- New online structure learning algorithm for both feed forward and recurrent SPNs
- Code available: **github.com/kalraa/spnz-sl**

Future work
- Reduce complexity w.r.t. # of features from quadratic to linear
- Discriminative structure learning