Dynamic Sum-Product Networks for Tractable Inference on Sequence Data

International Conference on Probabilistic Graphical Models, Lugano, Switzerland

Mazen Melibari
Pascal Poupart

Prashant Doshi

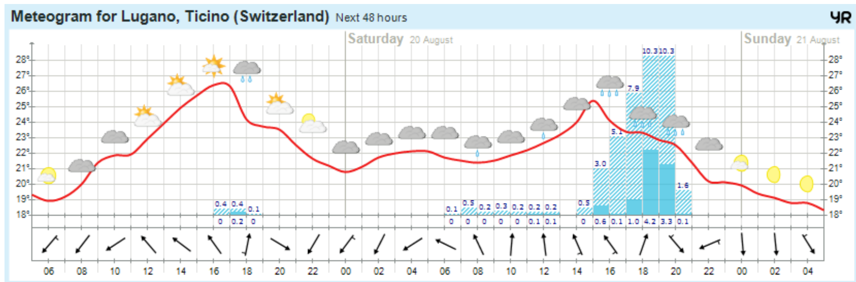
George Trimponias

Sequence Data

There was, and remains, a daring and a bigness to Mailer, derived from his preference for being knocked off balance instead of dug in. Among American writers of his day, he was alone in thinking that a trip to the moon, even one funded by the military-industrial complex of the country that he sometimes called Cancer Gulch, might be worth a book.

Sentence modeling: sequence of words

Activity recognition: measurement sequence

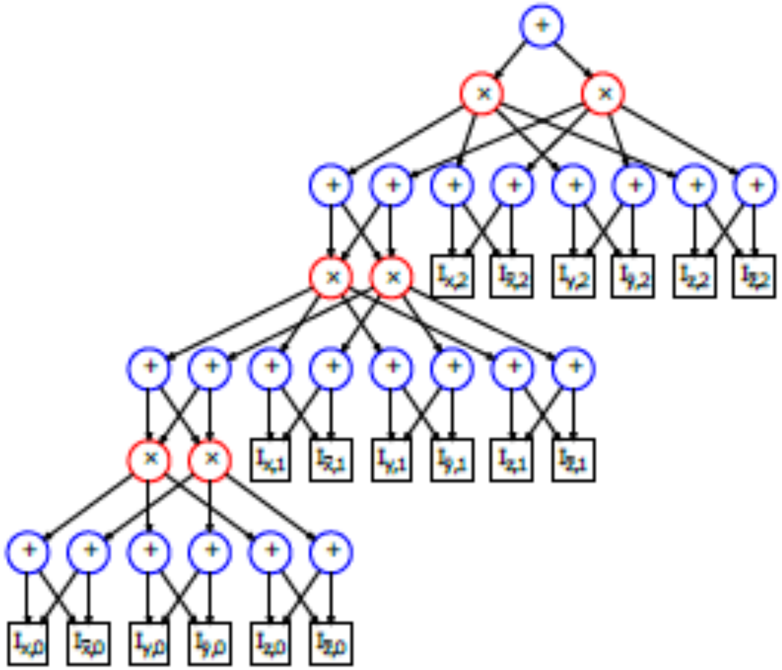


Weather prediction: time-series data

- Models: Dynamic Bayesian Networks and Dynamic Markov Random Fields
- Problem: inference exponential in # of variables per slice

Solution

- **Directly learn** Dynamic Sum Product Network (DSPN) (equivalent to Dynamic AC, Brandherm & Jameson 2004)



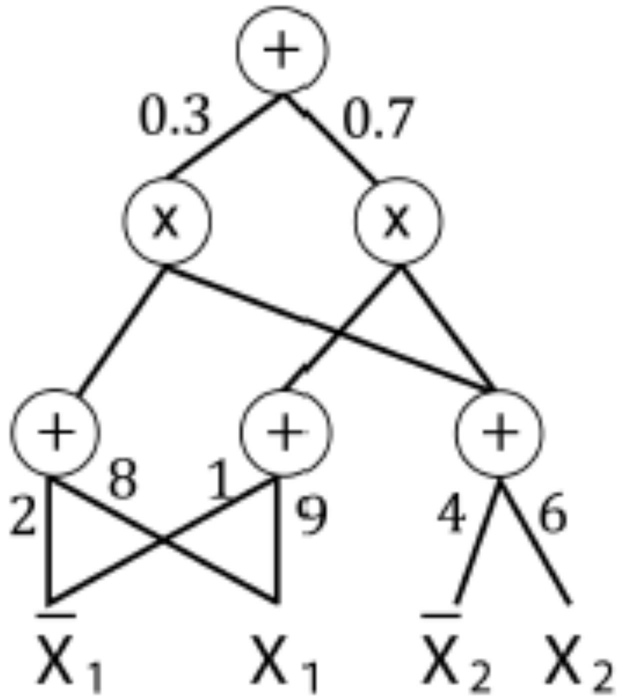
- Inference: **linear in the size of the network**

Outline

- Background: Sum product networks
- Dynamic Sum-Product Networks
  - Template networks
  - Invariance property
  - Structure and parameter learning
- Experiments
  - Comparison with HMMs, DBNs, RNNs
- Conclusion and future work

Sum-Product Network

- Defined by Poon & Domingos 2011 (equivalent to ACs, Darwiche 2003)

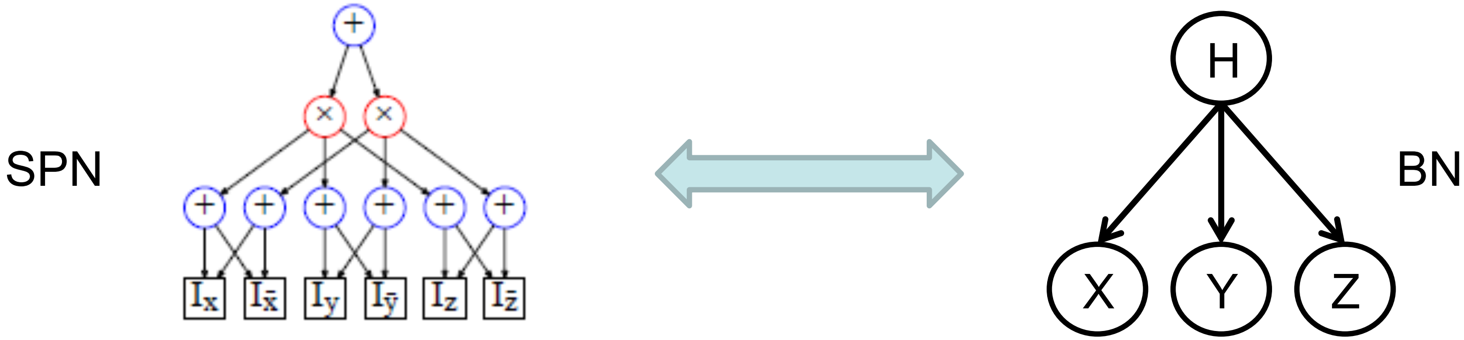


- Scope of a node: set of variables in sub-SPN rooted at that node
- Decomposable product node: children with disjoint scopes
- Complete/smooth sum node: children with identical scopes

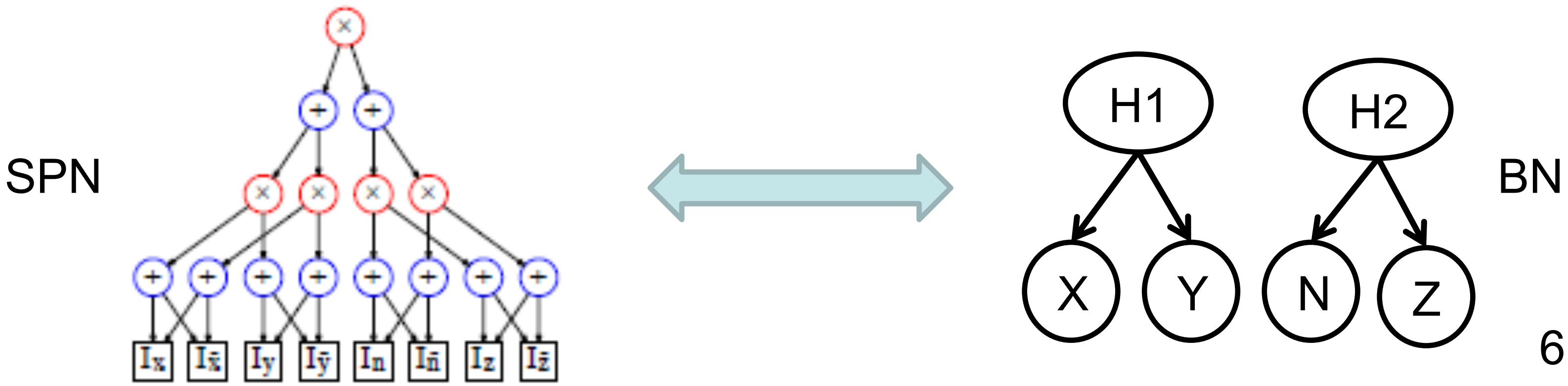
decomposability + completeness → valid distribution, linear inference

Relationship with Bayes Nets

- Any SPN can be converted into a Bayes net without any exponential blow up (Zhao, Melibari, Poupart, ICML-15)
- Naïve Bayes model



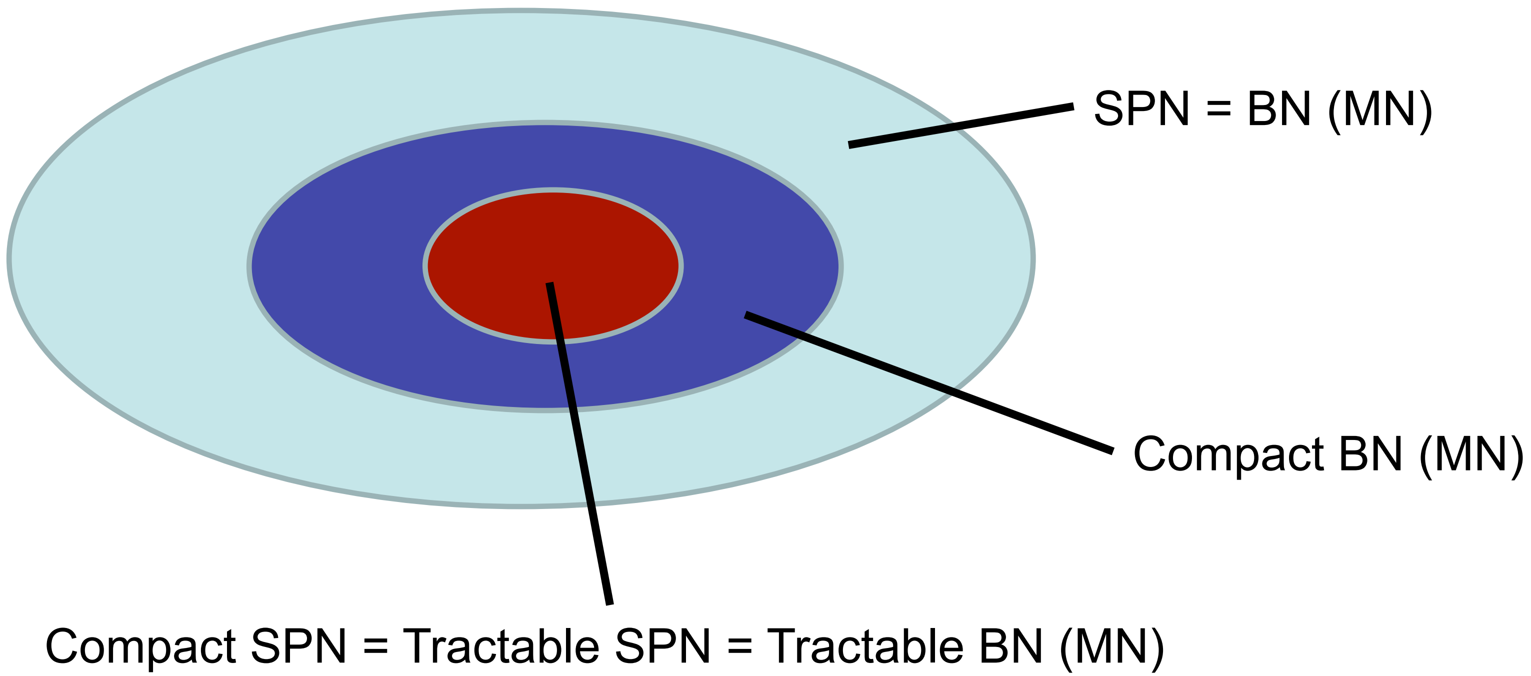
- Product of Naïve Bayes models



Relationship with other PGMs

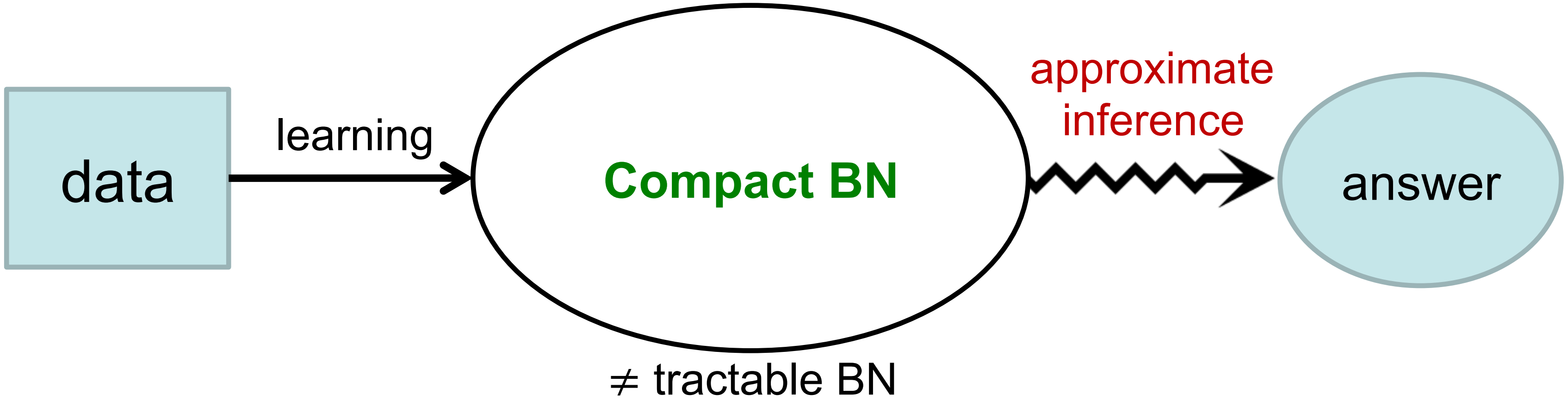
Probability distributions

- **Compact:** space is polynomial in # of variables
- **Tractable:** inference time is polynomial in # of variables

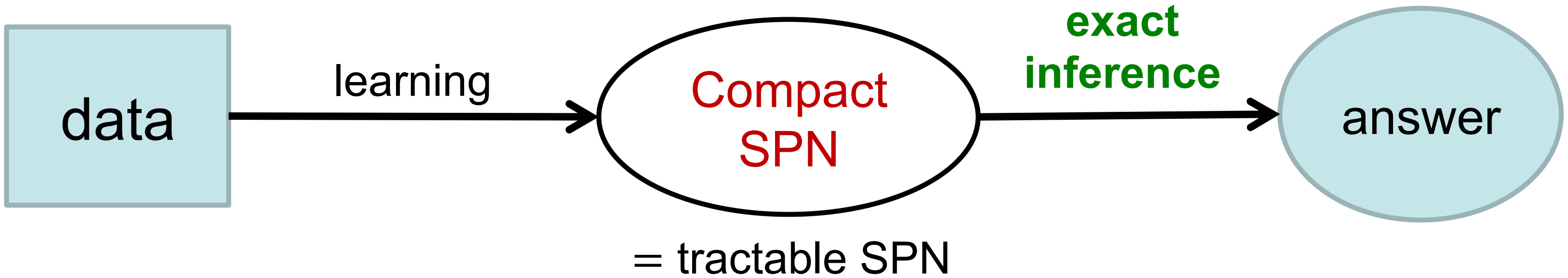


Practical Implications

Traditional approach



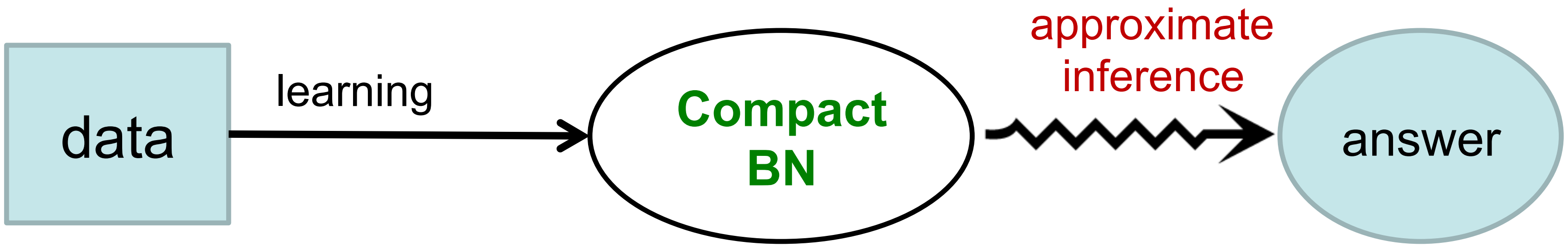
New approach



Tradeoff: hypothesis space vs inference accuracy

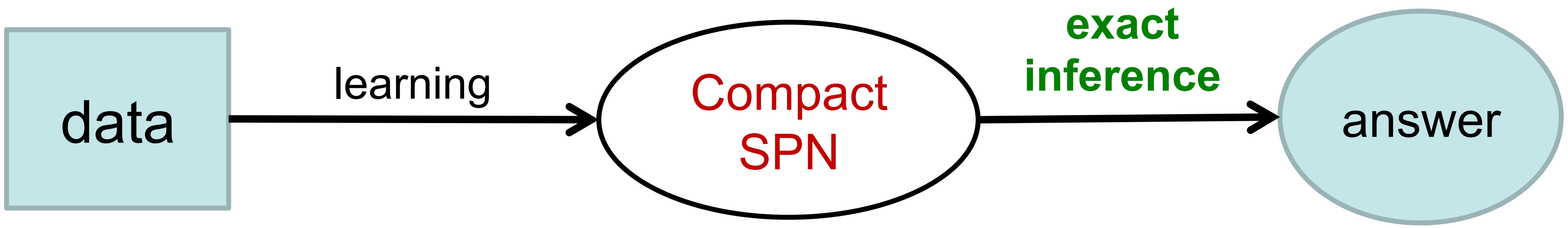
Practical Implications

Traditional approach



$\neq$ tractable BN

New approach



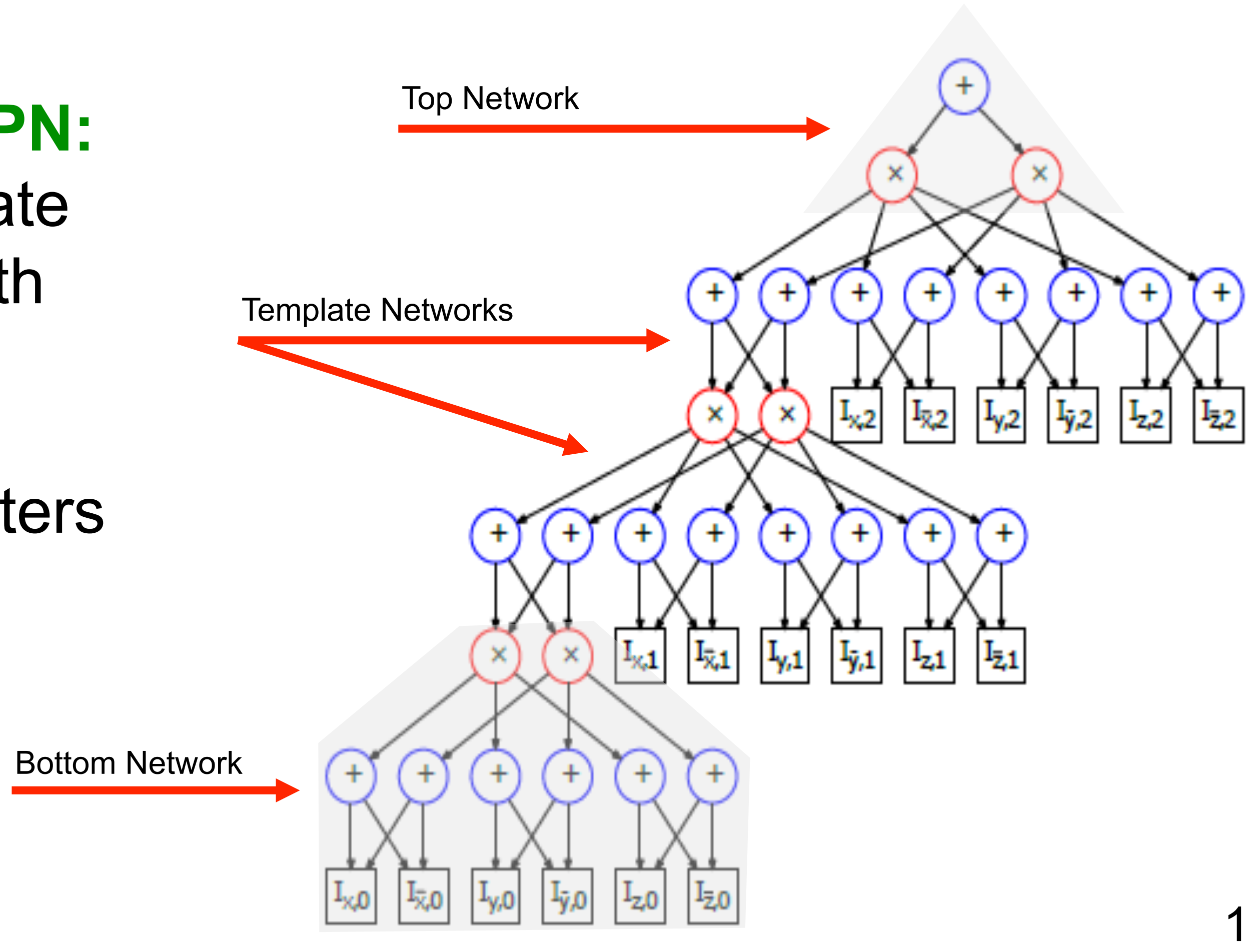
$=$ tractable SPN

Size of search space determined by learning algo

Sequence Data

- How can we train an SPN with data sequences of varying length?

- **Dynamic SPN:** stack template networks with identical structure and parameters



Invariance

Invariance: a template network is invariant when:

- For all pairs of interface nodes i, j
  - Scopes are identical or disjoint
  - Scope relationships in input and output interfaces are the same
- All interior and output nodes are complete and decomposable

Completeness and Decomposability

Theorem: If

a. the bottom network is complete and decomposable,
b. the scopes of all pairs of output interface nodes of the bottom network are either identical or disjoint,
c. the scopes of the output interface nodes of the bottom network can be used to assign scopes to the input interface nodes of the template and top networks in such a way that the template network is invariant and the top network is complete and decomposable,

then the **DSPN is complete and decomposable**

Structure Learning

Anytime search-and-score framework

Input: data, variables $X_1, \ldots, X_n$

Output: $templateNet$

$templateNet \leftarrow initialStructure(data, X_1, \ldots, X_n)$

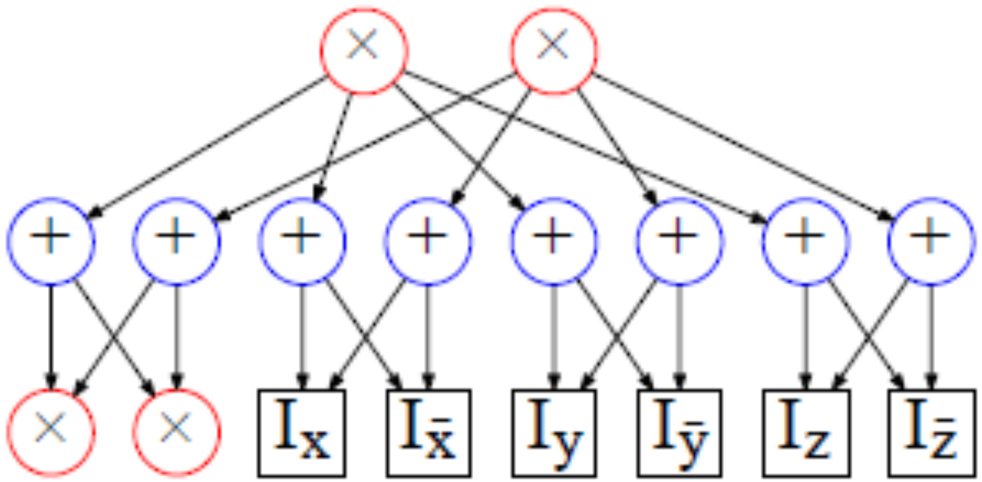
Repeat

$templateNet \leftarrow neighbour(templateNet, data)$

Until stopping criterion is met

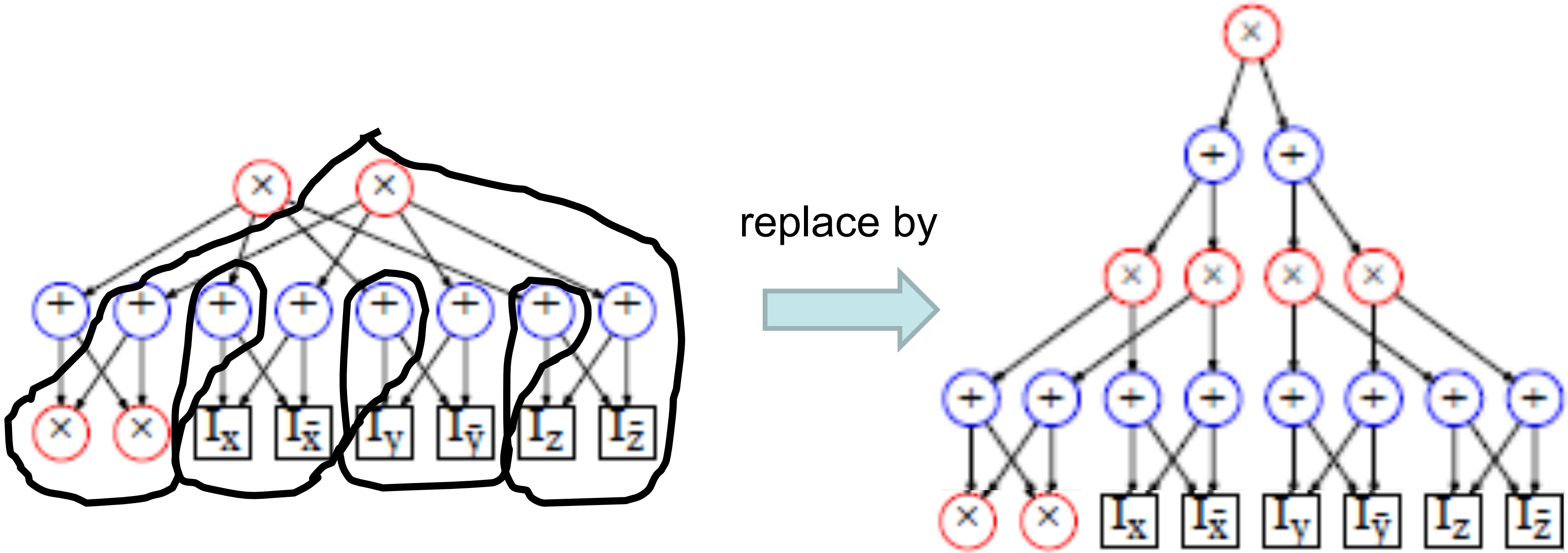
Initial Structure

- Factorized model of univariate distributions



Neighbour generation

- Replace sub-SPN rooted at a product node by a product of Naïve Bayes modes



Scoring

- For each neighbour SPN
  - Estimate weights by Expectation-Maximization
  - Score neignbour SPN based on data likelihood

- Retain neighbour with highest score

- **Theorem:** search algorithm preserves template invariance, which ensures a valid distribution

Results (Simulated Data)

Negative log-likelihood and standard error (10-fold cross validation)

Dataset (#i, length, #oVars)	HMM-Samples (100, 100, 1)	Water (100, 100, 4)	BAT (100, 100, 10)
True model	62.2 ± 0.8	249.6 ± 1.0	628.2 ± 2.0
LearnSPN	65.4 ± 0.7	270.4 ± 0.9	684.4 ± 1.3
DSPN	$\mathbf{62.5} \pm 0.7$	$\mathbf{252.4} \pm 0.9$	$\mathbf{641.6} \pm 1.1$

DSPN close to through model

Results (Real Data)

Negative log-likelihood and standard error (10-fold cross validation)

Dataset (#i,length,#oVars)	ozLevel (2533,24,2)	PenDigits (10992,16,7)	ArabicDigits (8800,40,13)	JapanVowels (640,16,12)	ViconPhysic (200,3026,27)
HMM	56.7 ± 1.1	74.2 ± 0.1	327.5 ± 0.4	94.3 ± 0.3	40862 ± 369
HMM-SPN	49.8 ± 0.9	67.7 ± 0.6	305.8 ± 1.8	89.8 ± 1.2	38410 ± 440
RNN	$\mathbf{16.2} \pm 0.7$	68.7 ± 1.3	303.6 ± 6.4	78.8 ± 2.3	57217 ± 873
Search-Score DBN	40.2 ± 4.7	67.3 ± 2.3	263.7 ± 4.6	75.6 ± 2.5	-
Reveal DBN	52.4 ± 2.5	74.4 ± 0.2	260.2 ± 1.0	71.3 ± 1.2	-
DSPN	33.0 ± 1.0	$\mathbf{63.5} \pm 0.3$	$\mathbf{257.9} \pm 0.5$	$\mathbf{68.8} \pm 0.3$	$\mathbf{36385} \pm 682$

DSPN outperfoms other algos (except RNN for 1 prob)

Learning and Inference Time

Dataset	Learning Time (Seconds)				Inference Time (Seconds)			
	Reveal	Per Iteration			Reveal	RNN	SS DBN	DSPN
		RNN	SS DBN	DSPN				
ozLevel	952	56	108	54	6.3	0.1	15.6	0.1
PenDigits	3,977	558	1,463	475	15.0	0.2	30.7	0.1
ArabicDigits	16,549	2572	14,911	2,909	53.6	2.5	465.8	2.9
JapaneseVowls	516	55	363	51	15.2	0.2	69.2	0.5
ViconPhysical	-	4705	-	6734	-	2274	-	1825

DSPN and RNN: linear learning and inference time

Reveal and SS DBN: exponential time complexity

Conclusion

- Dynamic Sum-Product Networks
  - Formalization and learning algorithm
  - Tractable probabilistic graphical model

- Future work
  - DSPNs with continuous variables
  - Decision DSPNs and Reinforcement learning