Sum-Product Networks

STAT946 Deep Learning

Guest Lecture by Pascal Poupart
University of Waterloo

October 17, 2017

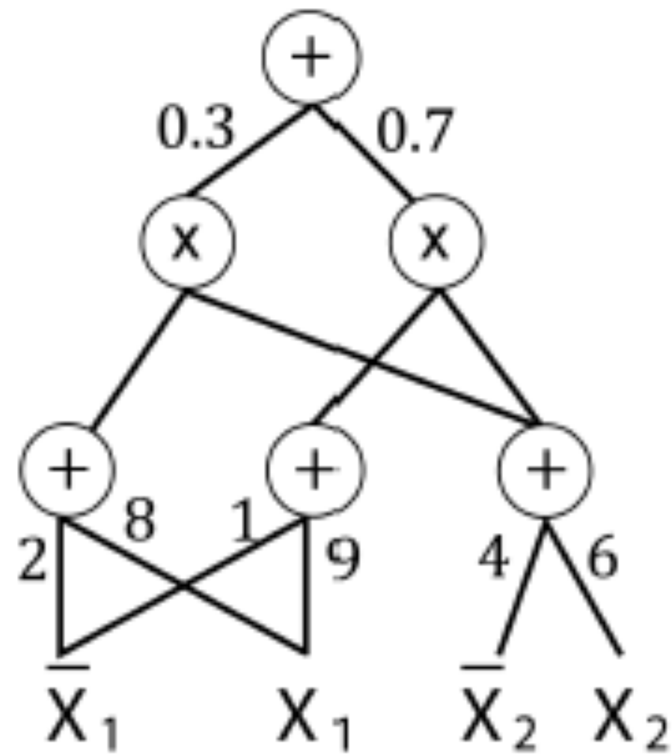
Outline

- Introduction
  - What is a Sum-Product Network?
  - Inference
  - Applications

- In more depth
  - Relationship to Bayesian networks
  - Parameter estimation
  - Online and distributed estimation
  - Structure estimation

What is a Sum-Product Network?

- Poon and Domingos, UAI 2011
- Acyclic directed graph of sums and products
- Leaves can be indicator variables or univariate distributions



Two Views

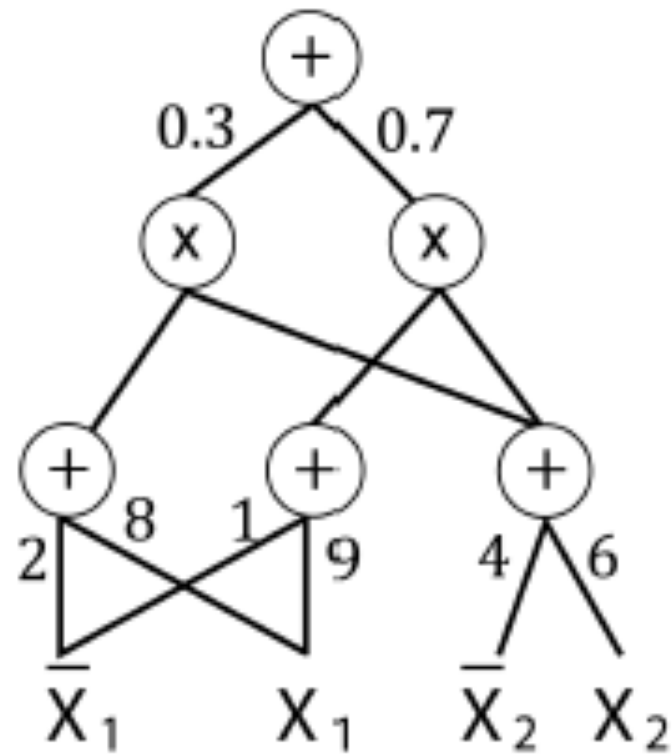
Deep neural network with clear semantics

Tractable probabilistic graphical model

Deep Neural Network View

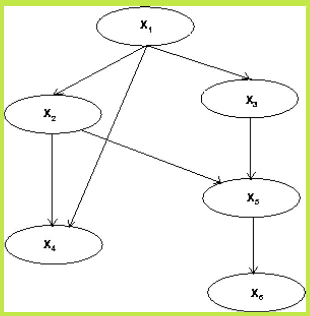
- Specific type of neural network
  - Sum node: $\log(\sum_i w_i input_i)$
  - Product node: $\exp(\sum_i input_i)$

- Advantages:
  - Clear semantics
  - Generative model
  - Efficient training
  - Structure estimation



Probabilistic Graphical Models

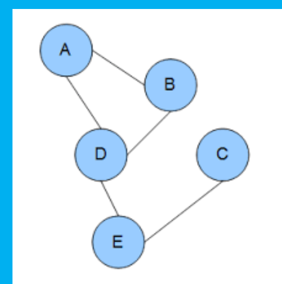
Bayesian Network



Graphical view of direct dependencies

Inference
#P: intractable

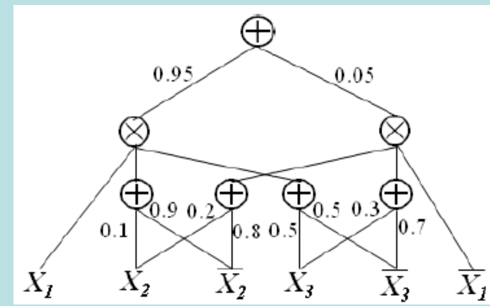
Markov Network



Graphical view of correlations

Inference
#P: intractable

Sum-Product Network



Graphical view of computation

Inference
P: tractable

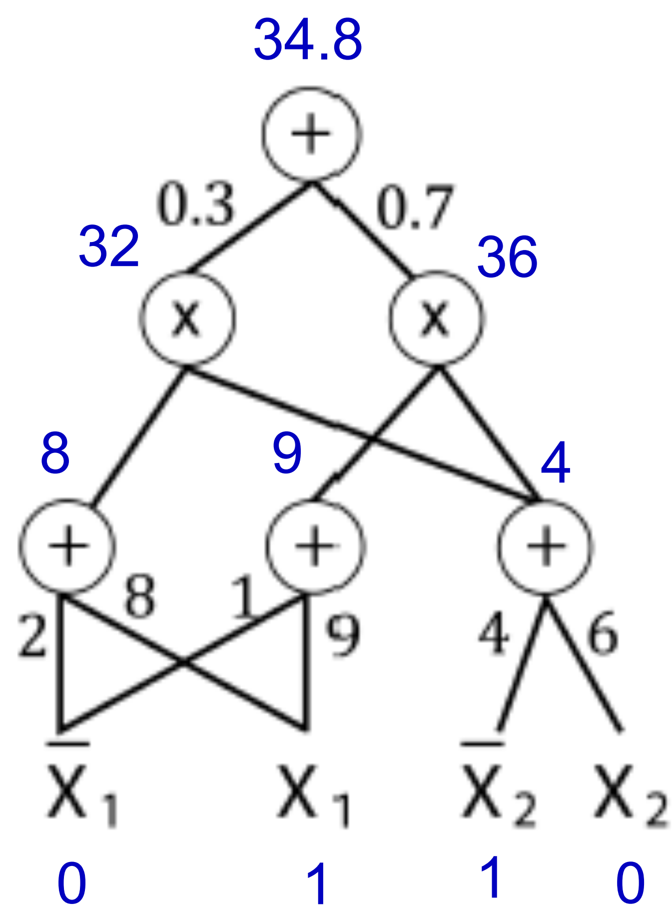
Probabilistic Inference

- SPN represents a joint distribution over a set of random variables

- Example:

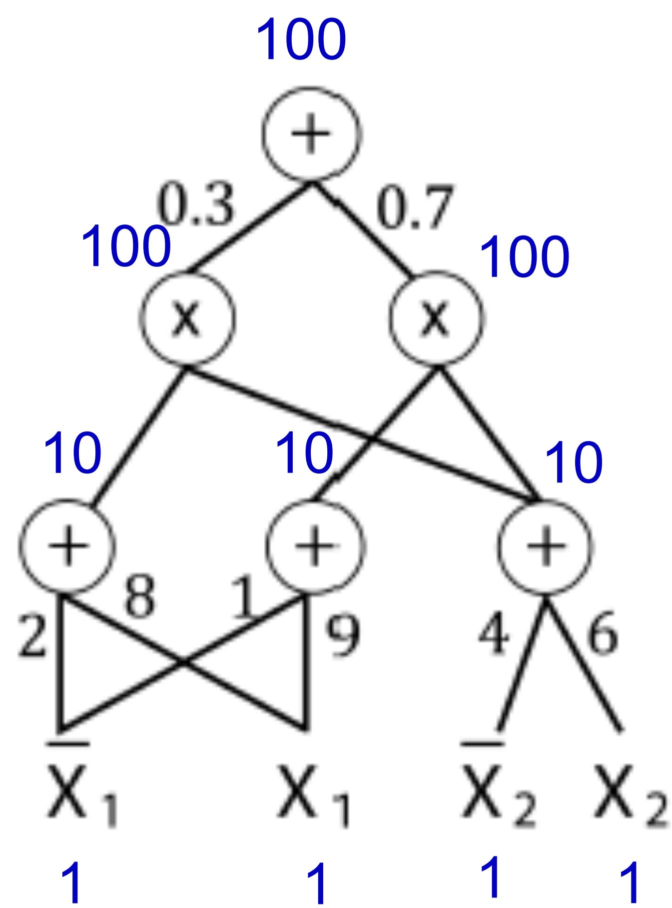
  $\Pr(X_1 = true, X_2 = false)$

  $= \frac{34.8}{}$



Probabilistic Inference

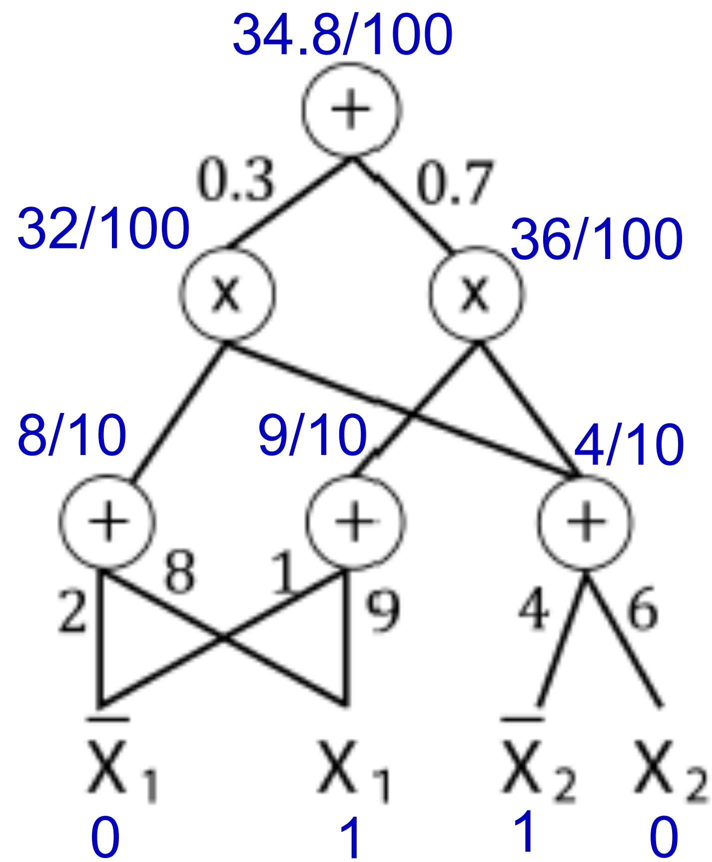
- SPN represents a joint distribution over a set of random variables
- Example:
  $\Pr(X_1 = true, X_2 = false)$
  $= \frac{34.8}{100}$
- **Linear complexity!**



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

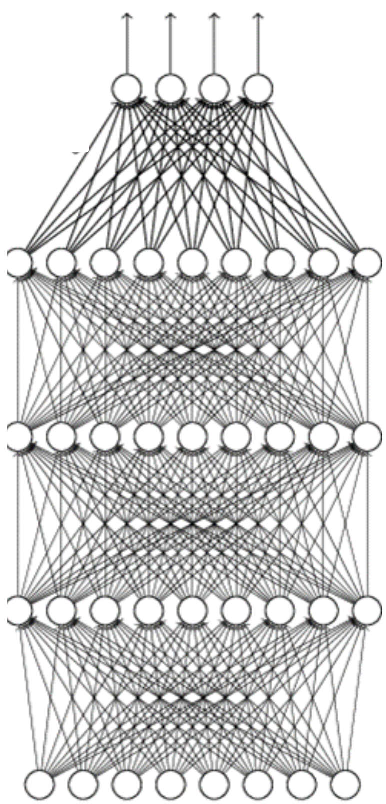


decomposability + completeness → **valid** distribution **linear** inference

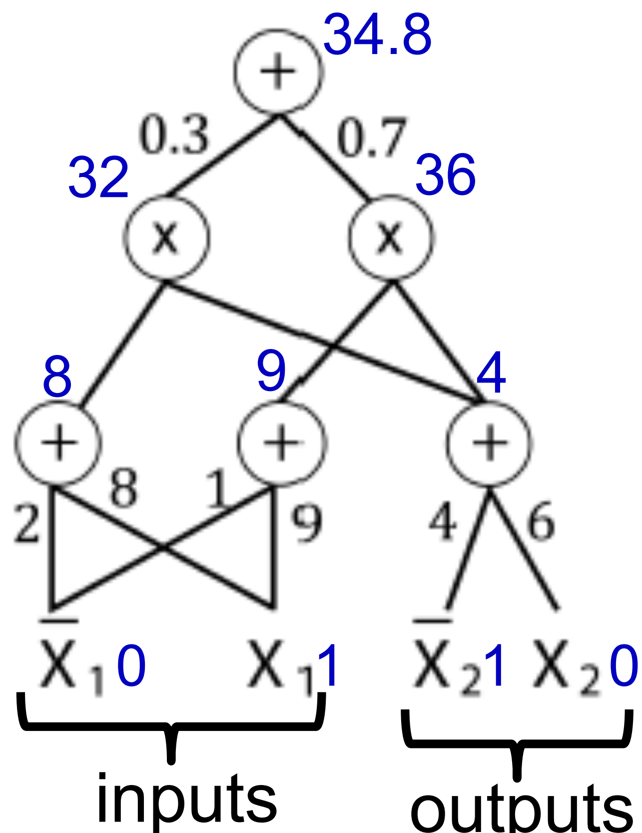
Queries

Most Neural Nets

outputs=f(inputs)



Sum-Product Networks

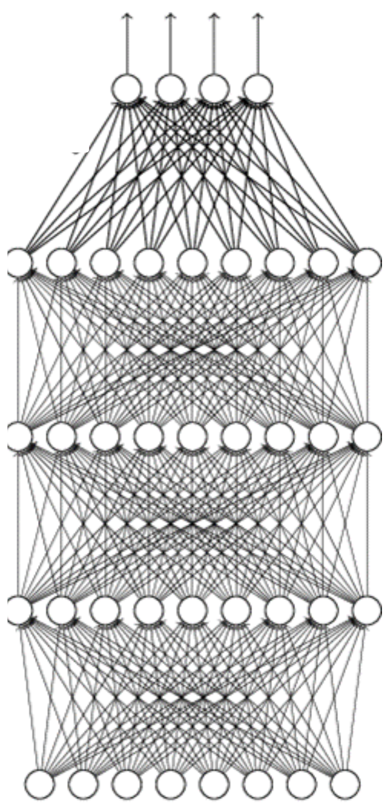


$$\Pr(X_2 = F | X_1 = T) = \frac{\Pr(X_2 = F, X_1 = T)}{\Pr(X_1 = T)} = \underline{34.8\quad}$$

Queries

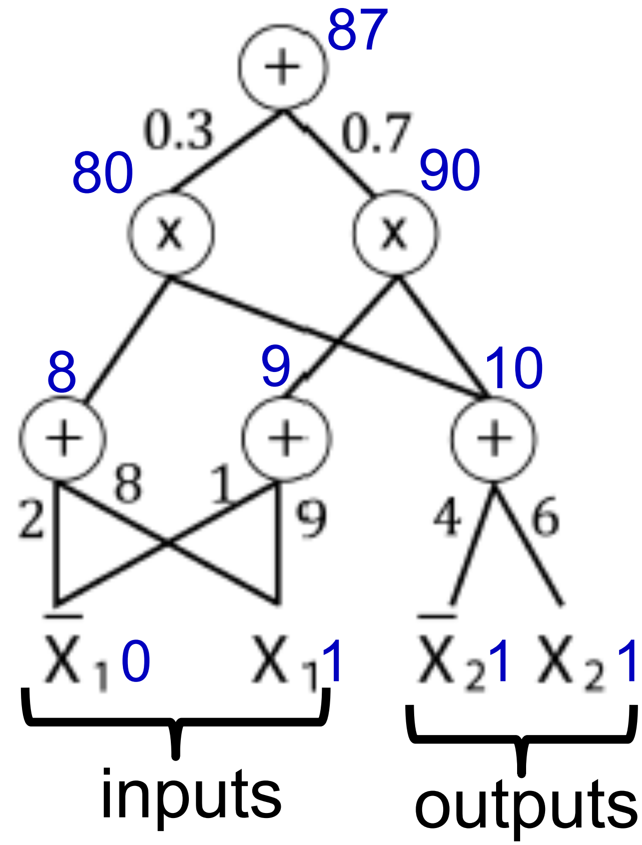
Most Neural Nets

outputs=f(inputs)



inputs

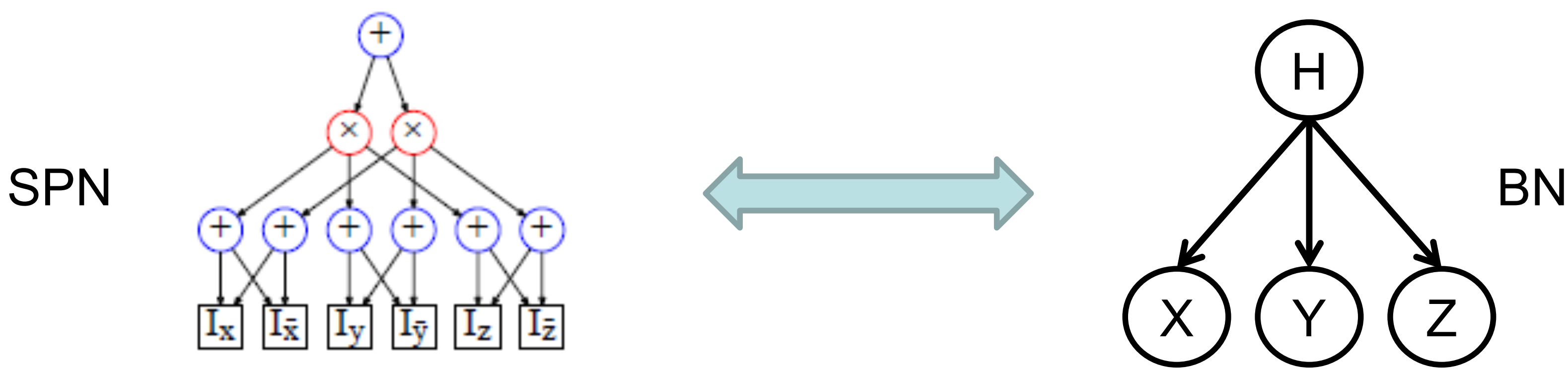
Sum-Product Networks



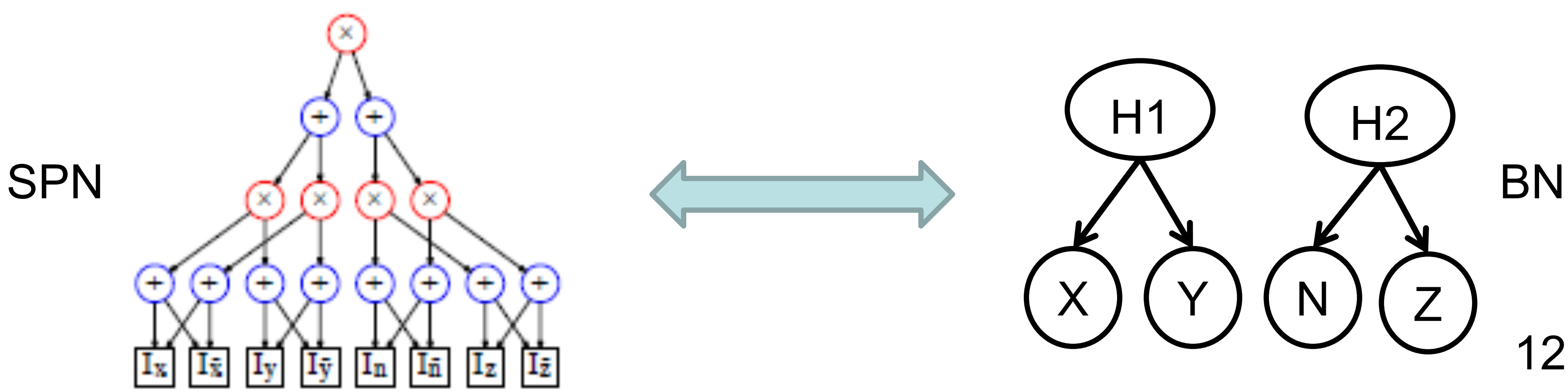
$$\Pr(X_2 = F | X_1 = T) = \frac{\Pr(X_2 = F, X_1 = T)}{\Pr(X_1 = T)} = \frac{34.8}{87}$$

Relationship with other PGMs

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



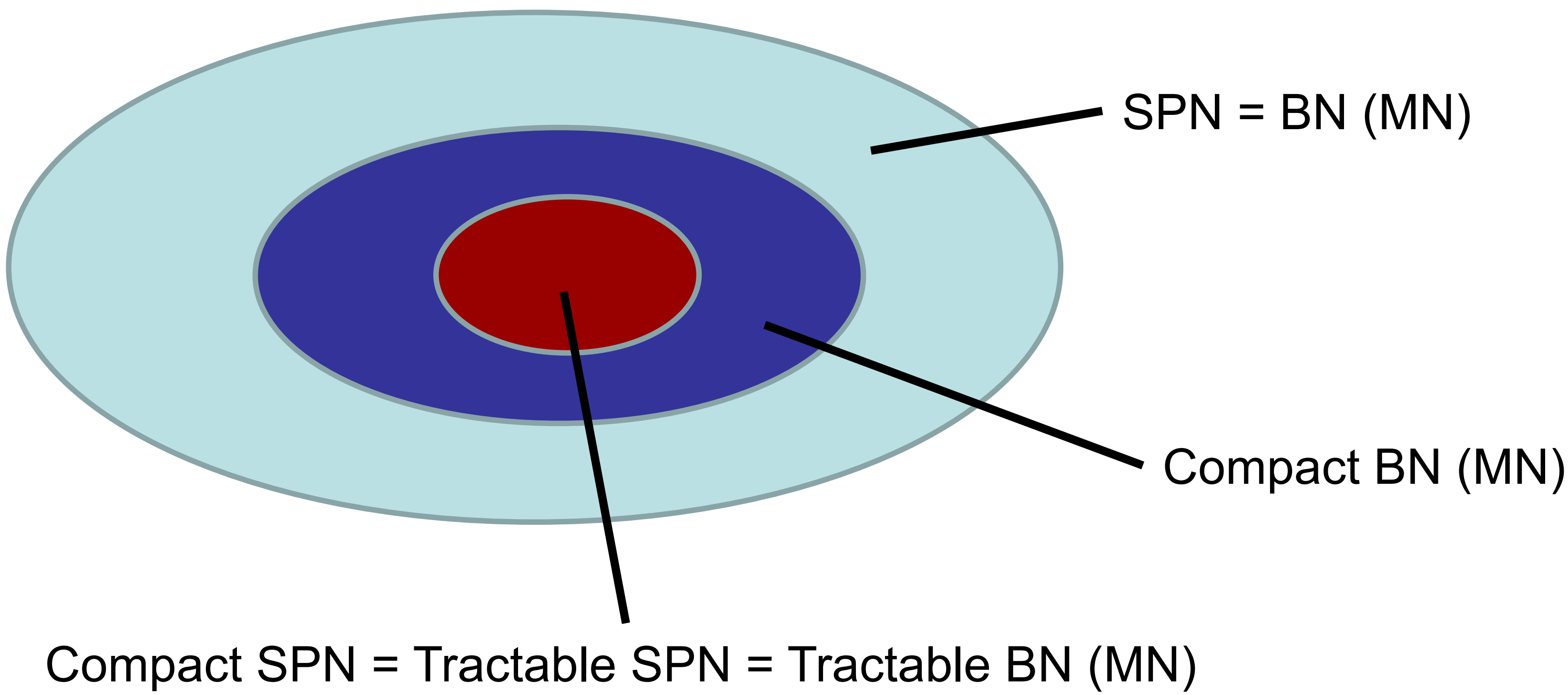
- Product of Naïve Bayes models



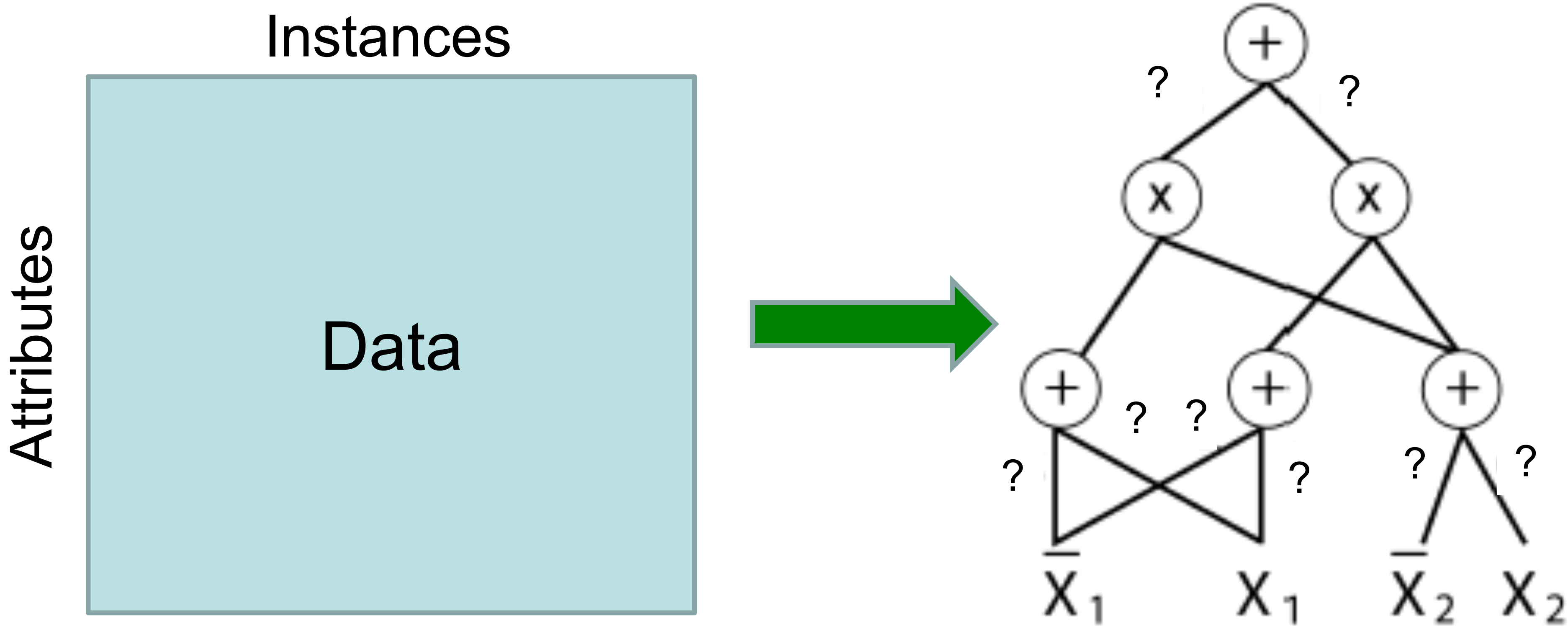
Relationship with other PGMs

Probability distributions

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



Parameter Estimation



Maximum likelihood: Stochastic gradient descent (SGD) (Poon & Domingos, 2011), expectation maximization (EM) (Perharz, 2015), signomial programming (Zhao & Poupart, 2016)

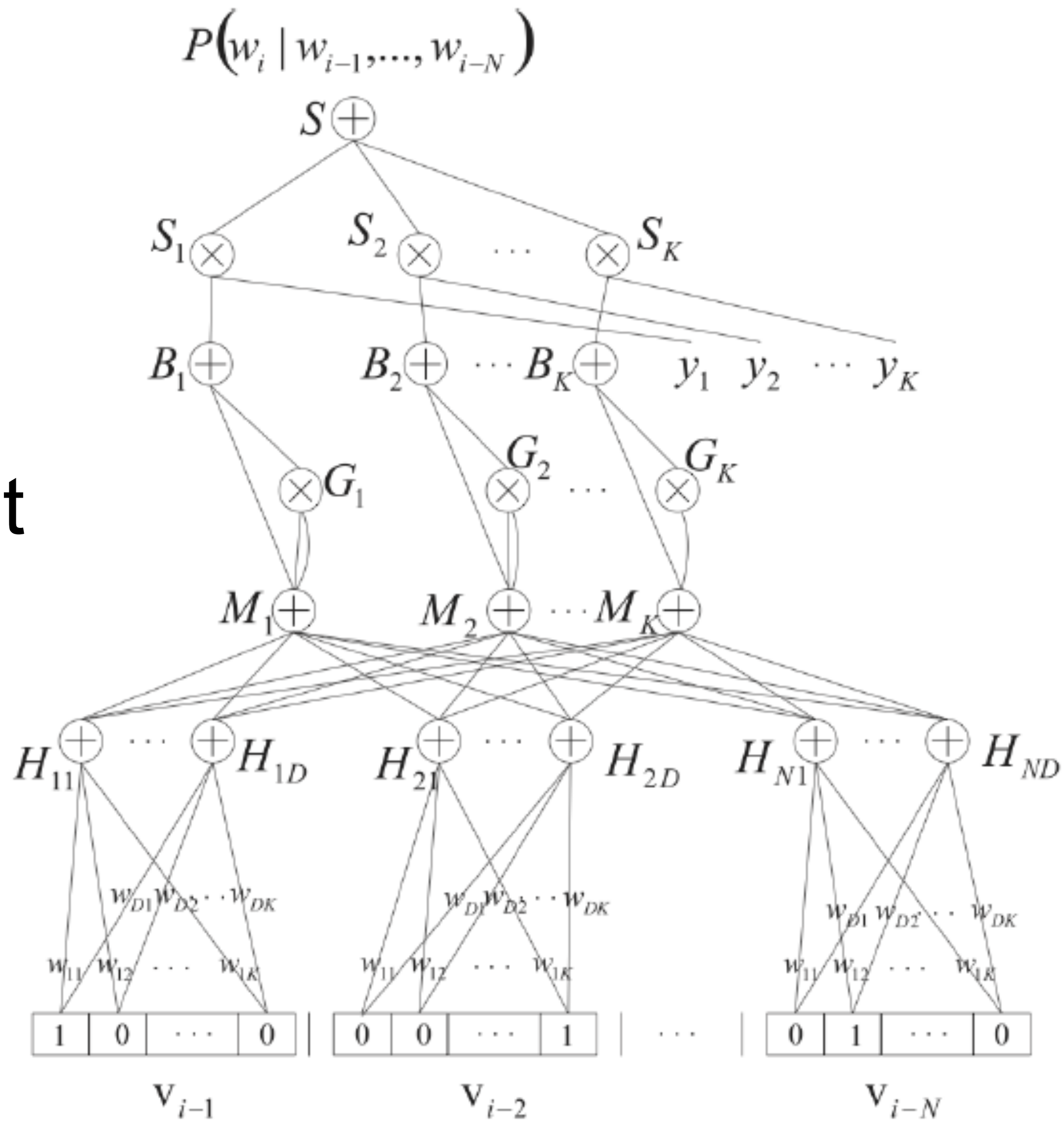
Bayesian learning: Bayesian Moment Matching (BMM) (Rashwan et al., 2015), Collapsed Variational Inference (Zhao et al., 2016)

Applications

- Image completion (Poon, Domingos; 2011)
- Activity recognition (Amer, Todorovic; 2012)
- **Language modeling (Cheng et al.; 2014)**
- Speech modeling (Perhaz et al.; 2014)
- Mobile robotics (Pronobis, Rao; 2016)

Language Model

- An SPN-based n-gram model
- Fixed structure
- Discriminative weight estimation by gradient descent



Results

- From Cheng et al. 2014

Table 1: Perplexity scores (PPL) of different language models.

Model	Individual PPL	+KN5
TrainingSetFrequency	528.4	
KN5 [3]	141.2	
Log-bilinear model [4]	144.5	115.2
Feedforward neural network [5]	140.2	116.7
Syntactical neural network [8]	131.3	110.0
RNN [6]	124.7	105.7
LDA-augmented RNN [9]	113.7	98.3
SPN-3	**104.2**	**82.0**
SPN-4	**107.6**	**82.4**
SPN-4'	**100.0**	**80.6**

Maximum Log-Likelihood

- Objective: $w^* = argmax_{w \in R_+} \log \Pr(data|w)$
  $= argmax_{w \in R_+} \sum_x \log \Pr(x|w)$

where $\Pr(x|w) = \frac{f\left(e(x)|w\right)}{f(\mathbf{1}|w)} = \frac{\sum_{tree \in e(x)} \prod_{ij \in tree} w_{ij}}{\sum_{tree \in 1} \prod_{ij \in tree} w_{ij}}$

- Non-convex optimization

$$\max_w \sum_x \log \sum_{tree \in e(x)} \prod_{ij \in tree} w_{ij} - \log \sum_{tree \in 1} \prod_{ij \in tree} w_{ij}$$

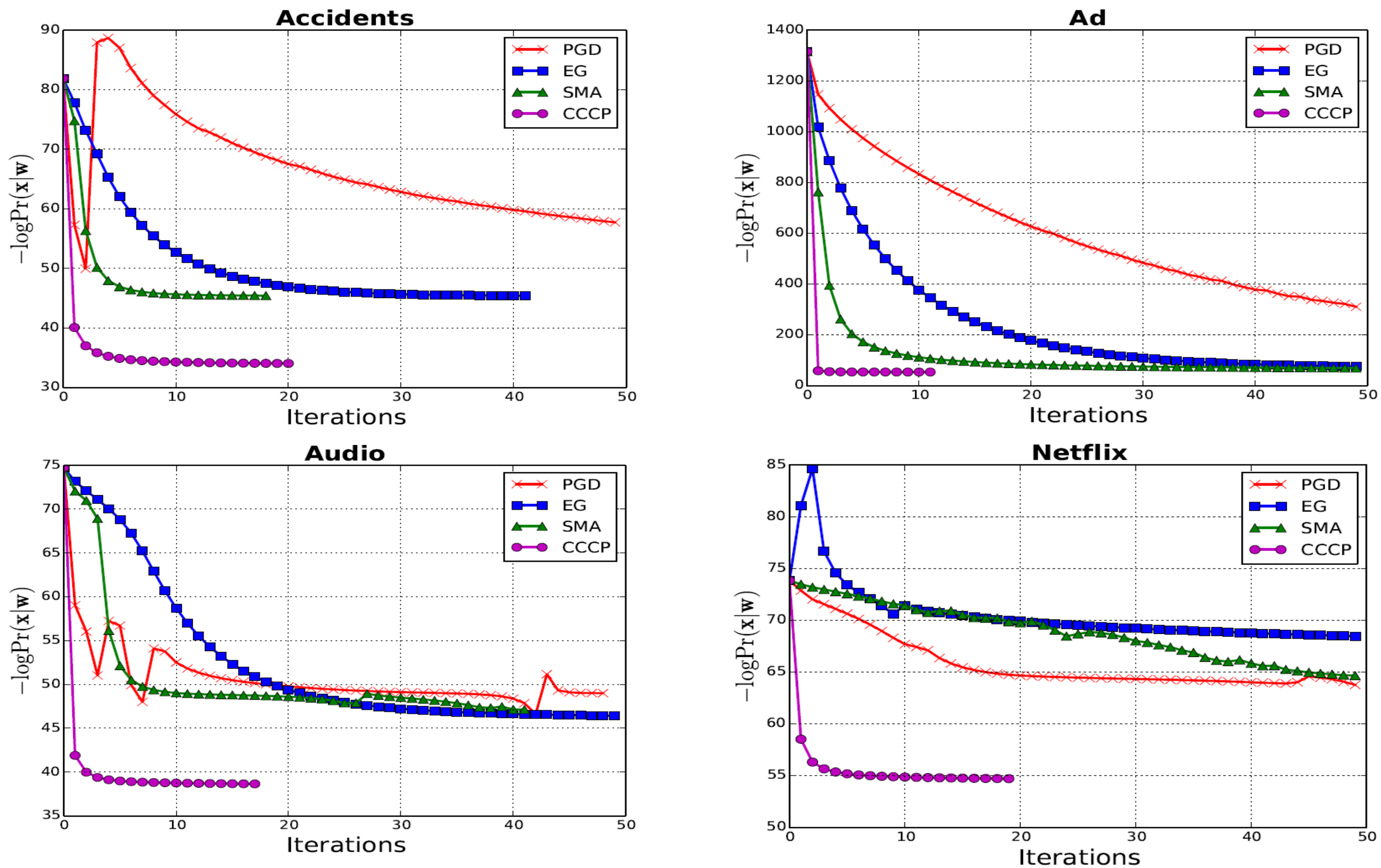
s.t. $w_{ij} \geq 0 \quad \forall ij$

Summary

Algo	Var	Update	Approximation
PGD	w	additive	linear
	$w_{ij}^{k+1} \leftarrow projection\left(w_{ij}^{k} + \gamma\left[\frac{\partial \log f(e(x)\|w)}{\partial w_{ij}} - \frac{\partial \log f(\mathbf{1}\|w)}{\partial w_{ij}}\right]\right)$		
EG	w	multiplicative	linear
	$w_{ij}^{k+1} \leftarrow w_{ij}^{k}\, exp\left(\gamma\left[\frac{\partial \log f(e(x)\|w)}{\partial w_{ij}} - \frac{\partial \log f(\mathbf{1}\|w)}{\partial w_{ij}}\right]\right)$		
SMA	$\log w$	multiplicative	monomial
	$w_{ij}^{k+1} \leftarrow w_{ij}^{k} \exp\left(\gamma\left[\frac{\partial \log f(e(x)\|w)}{\partial \log w_{ij}} - \frac{\partial \log f(\mathbf{1}\|w)}{\partial \log w_{ij}}\right]\right)$		
CCCP (EM)	$\log w$	multiplicative	Concave lower bound
	$w_{ij}^{k+1} \propto w_{ij}^{k} \frac{f_{v_j}(x\|w^k)}{f(x\|w^k)} \frac{\partial f(x\|w^k)}{\partial f_{v_i}(x\|w^k)}$		

Results

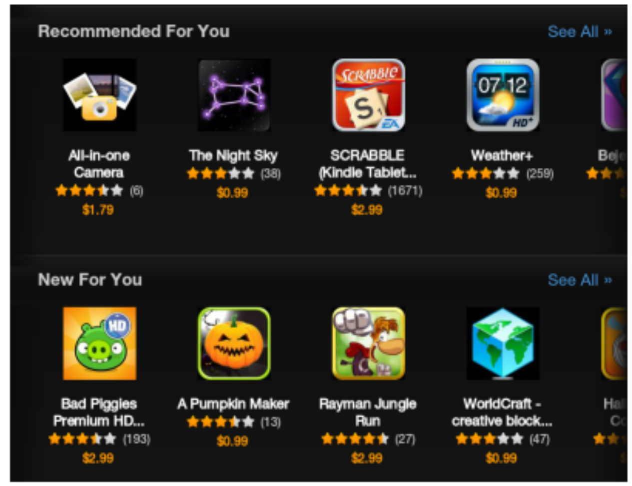
- Zhao, Poupart et al. (NIPS 2016)



Streaming Data

Traffic classification

App recommendation



- **Challenge:** update model after each data vector
- **Solution:** online learning for SPNs

Scalability

- Online: process data sequentially once only
- Distributed: process subsets of data on different computers

- Mini-batches: SGD, online EG, online EM
- Problems: loss of information due to mini-batches, how to adjust learning rate?

- Can we do better?

Thomas Bayes

Bayesian Learning

- Bayes' theorem (1764)

$$\Pr(\theta|X_{1:n}) \propto \Pr(\theta)\Pr(X_1|\theta)\Pr(X_2|\theta)\ldots\Pr(X_n|\theta)$$

- Broderick et al. (2013): facilitates
  - **Online learning (streaming data)**

$$\Pr(\theta|X_{1:n}) \propto \Pr(\theta)\Pr(X_1|\theta)\Pr(X_2|\theta)\ldots\Pr(X_n|\theta)$$

  - **Distributed computation**

$$\underbrace{\Pr(\theta)\Pr(X_1|\theta)}_{\text{core \#1}}\underbrace{\Pr(X_2|\theta)\Pr(X_3|\theta)}_{\text{core \#2}}\underbrace{\Pr(X_4|\theta)\Pr(X_5|\theta)}_{\text{core \#3}}$$

Exact Bayesian Learning

- Assume a normal SPN where the weights $w_{i\cdot}$ of each sum node i form a discrete distribution.

- Prior: $\Pr(w) = \prod_{i\cdot} Dir(w_{i\cdot}|\alpha_{i\cdot})$

  where $Dir(w_{i\cdot}|\alpha_{i\cdot}) \propto \prod_j \left(w_{ij}\right)^{\alpha_{ij}}$
- Likelihood: $\Pr(x|w) = f(e(x)|w) = \sum_{tree \in e(x)} \prod_{ij \in tree} w_{ij}$

- Posterior: $\sum_k c_k \prod_i Dir(w_{i\cdot}|\alpha_{i\cdot}^{(k)})$

  Exponentially large mixture of Dirichlets

Karl Pearson

Method of Moments (1894)

- Estimate model parameters by matching a subset of moments (i.e., mean and variance)

- Performance guarantees
  - Break through: First provably consistent estimation algorithm for several mixture models
    - HMMs: Hsu, Kakade, Zhang (2008)
    - MoGs: Moitra, Valiant (2010), Belkin, Sinha (2010)
    - LDA: Anandkumar, Foster, Hsu, Kakade, Liu (2012)

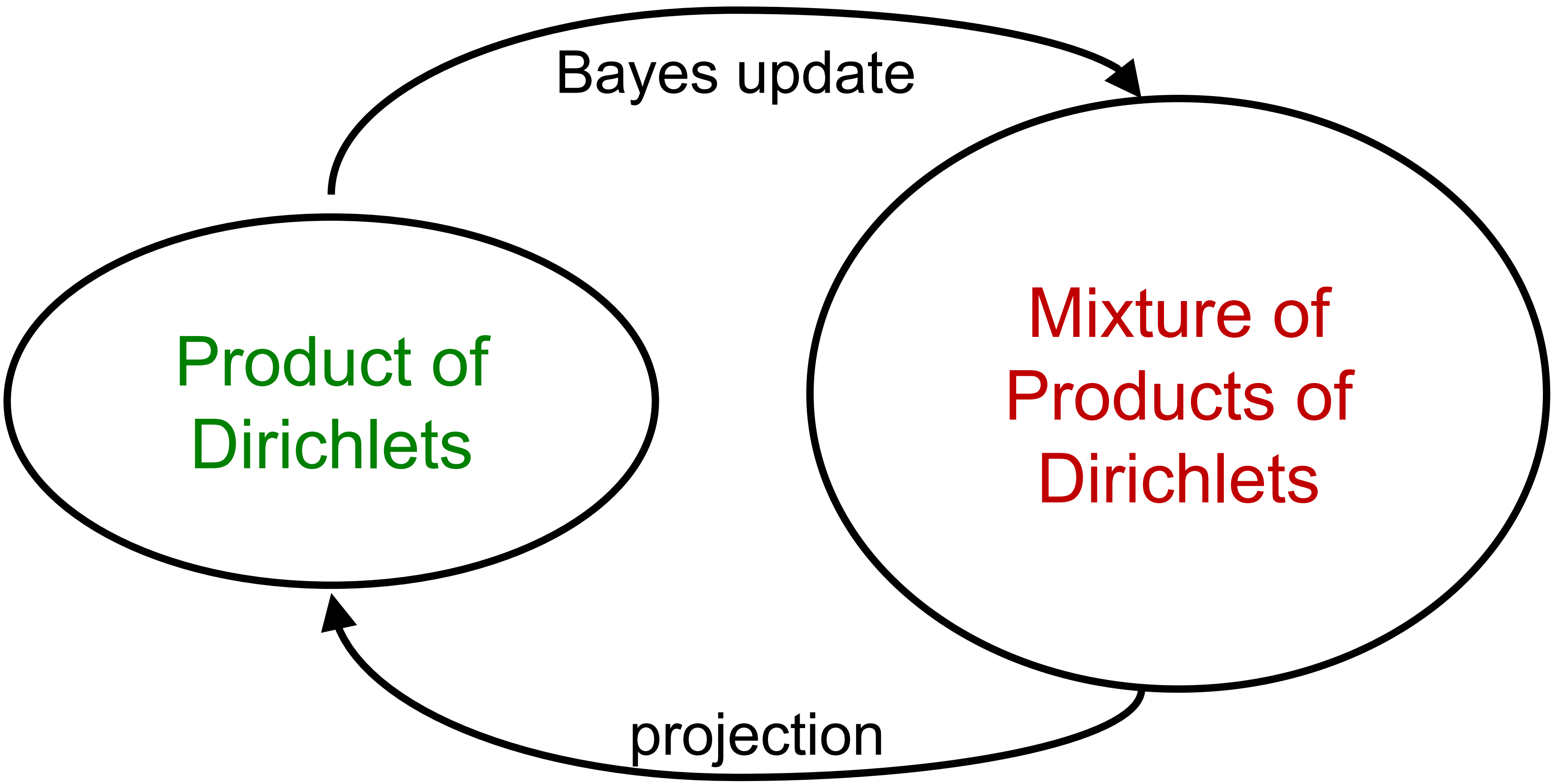
Bayesian Moment Matching for Sum Product Networks

Bayesian Learning
+
Method of Moments

Online, distributed and **tractable** algorithm for **SPNs**

Approximate **mixture of products of Dirichlets**
by a **single product of Dirichlets**
that **matches first and second order moments**

Bayesian Moment Matching



Results (benchmarks)

- Rashwan, Zhao, Poupart (AISTATS 2016)

Dataset	Var#	LearnSPN	oBMM	SGD	oEM	oEG
NLTCS	16	-6.11	**-6.07**	↓-8.76	↓-6.31	↓-6.85
MSNBC	17	-6.11	**-6.03**	↓-6.81	↓-6.64	↓-6.74
KDD	64	-2.18	**-2.14**	↓-44.53	↓-2.20	↓-2.34
PLANTS	69	-12.98	**-15.14**	↓-21.50	↓-17.68	↓-33.47
AUDIO	100	-40.50	**-40.7**	↓-49.35	↓-42.55	↓-46.31
JESTER	100	-53.48	**-53.86**	↓-63.89	↓-54.26	↓-59.48
NETFLIX	100	-57.33	**-57.99**	↓-64.27	↓-59.35	↓-64.48
ACCIDENTS	111	-30.04	**-42.66**	↓-53.69	-43.54	↓-45.59
RETAIL	135	-11.04	**-11.42**	↓-97.11	↓-11.42	↓-14.94
PUMSB-STAR	163	-24.78	**-45.27**	↓-128.48	↓-46.54	↓-51.84
DNA	180	-82.52	**-99.61**	↓-100.70	↓-100.10	↓-105.25
KOSAREK	190	-10.99	**-11.22**	↓-34.64	↓-11.87	↓-17.71
MSWEB	294	-10.25	**-11.33**	↓-59.63	↓-11.36	↓-20.69
BOOK	500	-35.89	**-35.55**	↓-249.28	↓-36.13	↓-42.95
MOVIE	500	-52.49	**-59.50**	↓-227.05	↓-64.76	↓-84.82
WEBKB	839	-158.20	**-165.57**	↓-338.01	↓-169.64	↓-179.34
REUTERS	889	-85.07	**-108.01**	↓-407.96	-108.10	↓-108.42
NEWSGROUP	910	-155.93	**-158.01**	↓-312.12	↓-160.41	↓-167.89
BBC	1058	-250.69	-275.43	↓-462.96	**-274.82**	↓-276.97
AD	1556	-19.73	**-63.81**	↓-638.43	↓-63.83	↓-64.11

Results (Large Datasets)

Rashwan, Zhao, Poupart (AISTATS 2016)

- Log likelihood

Dataset	Var#	LearnSPN	oBMM	oDMM	SGD	oEM	oEG
KOS	6906	-444.55	**-422.19**	-437.30	-3492.9	-538.21	-657.13
NIPS	12419	-	**-1691.87**	-1709.04	-7411.20	-1756.06	-3134.59
ENRON	28102	-	**-518.842**	-522.45	-13961.40	-554.97	-14193.90
NYTIMES	102660	-	-1503.65	-1559.39	-43153.60	**-1189.39**	-6318.71

oBMM and oDMM outperform other algos on 3 (out of 4) problems

- Time (minutes)

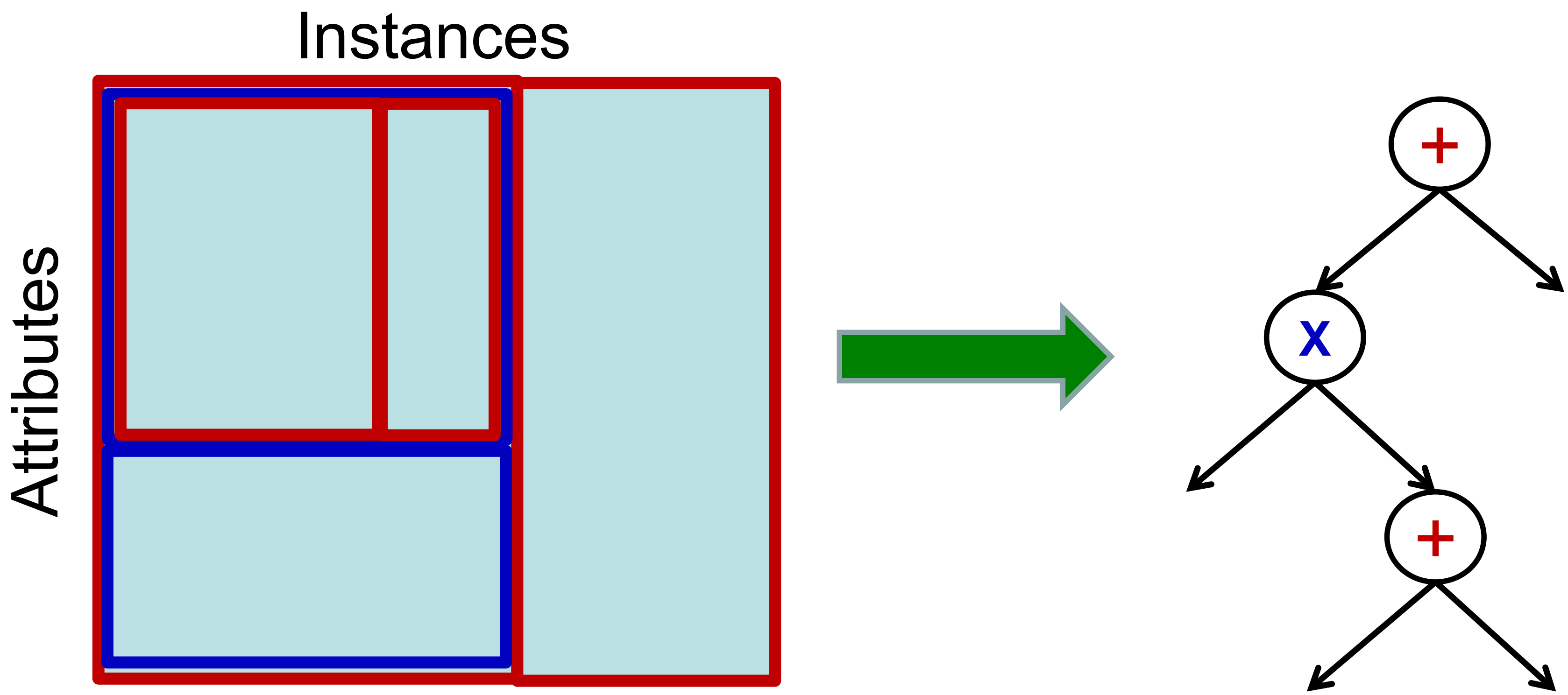
Dataset	Var#	LearnSPN	oBMM	oDMM	SGD	oEM	oEG
KOS	6906	1439.11	89.40	**8.66**	162.98	59.49	155.34
NIPS	12419	-	139.50	**9.43**	180.25	64.62	178.35
ENRON	28102	-	2018.05	**580.63**	876.18	694.17	883.12
NYTIMES	102660	-	12091.7	**1643.60**	5626.33	5540.40	6895.00

oDMM is significantly faster

Structure Estimation

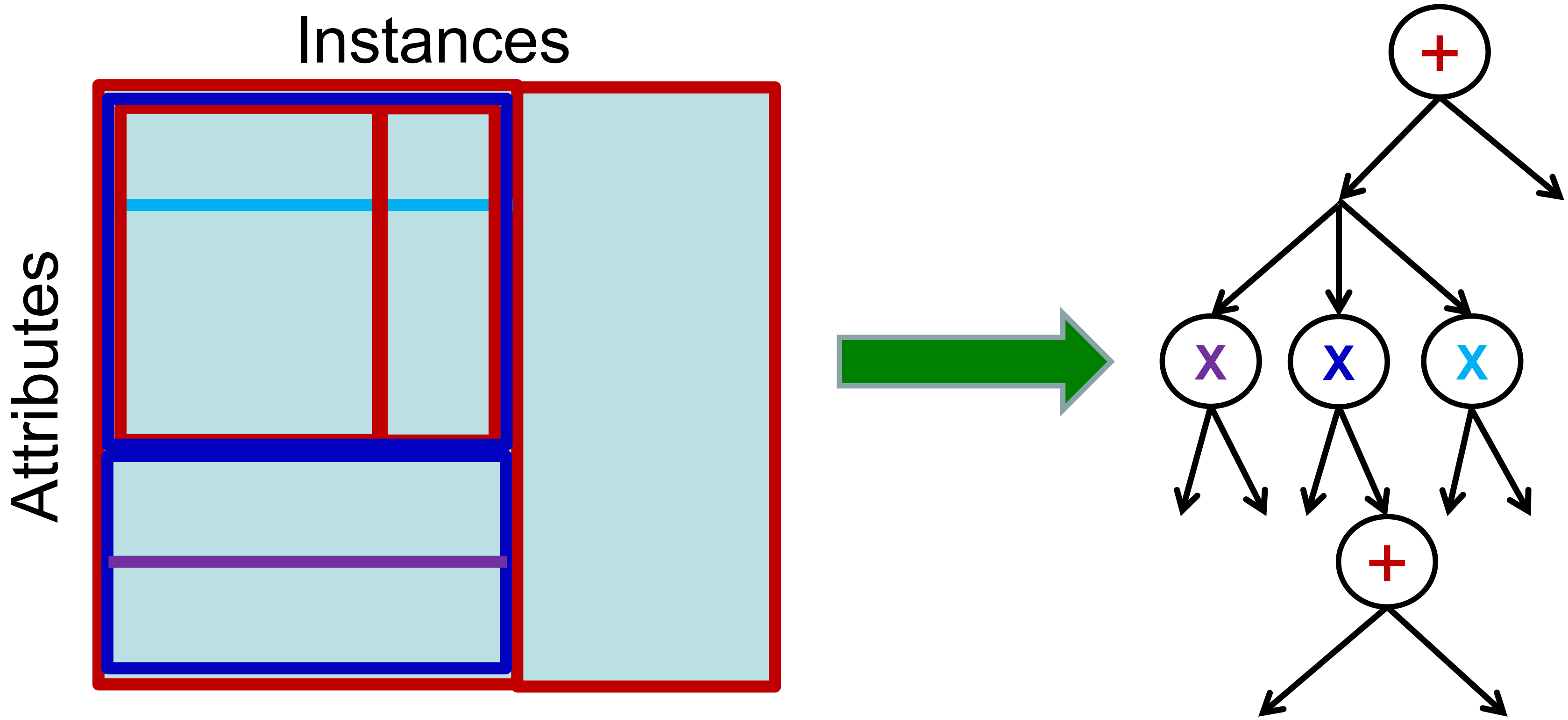
- What is the most popular technique to estimate the structure of a deep neural network?
- Parameter estimation:
  - Gradient descent
- Structure estimation:
  - Graduate student descent
- State-of-the-art: evolutionary techniques, hyperparameter search

Structure Estimation in SPNs



- LearnSPN (Gens & Domingos, 2013): alternate between
  - Data clustering: sum nodes
  - Variable partition (independence testing): product nodes

Improved Structure Estimation



- Prometheus (Jaini, Ghose et al, 2017): alternate between
  - Data clustering: sum nodes
  - **Multiple** variable partitions: product nodes

Results (log likelihood)

- From Jaini, Ghose and Poupart (2017)

Discrete Datasets				
Data set	Learn-SPN	ID-SPN	CCCP	Prometheus
NLTCS	-6.10 ↓	-6.05↓	-6.03↓	**-6.01**
MSNBC	-6.11 ↓	-6.05	-6.05	**-6.04**
KDD	-2.23 ↓	-2.15↓	-2.13	**-2.13**
Plants	-12.95↓	**-12.55↑**	-12.87↓	-12.81
Audio	-40.51↓	-39.82	-40.02↓	**-39.80**
Jester	-53.45↓	-52.91↓	-52.88↓	**-52.80**
Netflix	-57.38↓	-56.55	-56.78↓	**-56.47**
Accidents	-29.07↓	**-27.23↑**	-27.70	-27.91
Retail	-11.14 ↓	-10.94↓	-10.92↓	**-10.87**
Pumsbstar	-24.58 ↓	-22.55	-24.23↓	**-22.75**
DNA	-85.24↓	-84.69↓	-84.92↓	**-84.45**
Kosarek	-11.06↓	-10.61	-10.88↓	**-10.59**
MSWeb	-10.27↓	**-9.80**	-9.97↓	-9.86
Book	-36.25↓	-34.44	-35.01↓	**-34.40**
Movie	-52.82↓	-51.55↓	-52.56↓	**-51.49**
WebKB	-158.54↓	**-153.3↑**	-157.49↓	-155.21
Reuters	-85.98↓	**-84.39**	-84.63	-84.59
Newsgroup	-156.61↓	**-151.6↑**	-153.20↓	-154.17
BBC	-249.79↓	-252.60↓	-248.60	**-248.5**
AD	-27.41↓	-40.01↓	-27.20↓	**-23.96**

Continuous Datasets				
Data set (Attributes)	SRBMs	oSLRAU	oBMM	Prometheus
Abalone (8)	-2.28↓	-1.12↓	-1.21↓	**-0.85**
CA (22)	-4.95↓	17.10↓	-1.78↓	**27.82**
Quake (4)	-2.38↓	-1.86↓	-3.84↓	**-1.50**
Sensorless(48)	-26.91↓	54.82↓	1.58↓	**62.03**
Banknote(4)	-2.76↓	-2.04↓	-4.81↓	**-1.96**
Flowsize (3)	-0.79↓	14.78↓	4.80↓	**18.03**
Kinematics(8)	**-5.55↑**	-11.15↓	-11.2↓	-11.12

Continuous Datasets			
Data set	iSPT	GMM	Prometheus
Iris	-3.744↓	-3.943↓	**-1.06**
Old Faithful	-1.700↓	-1.737↓	**-1.48**
Chemical Diabetes	-2.879↓	-3.022↓	**-2.59**

MNIST dataset

DSPN-SVD	SPN-SVD	SPN-Gens	ID-SPN	Prometheus
97.6%	85%	81.8%	84.4%	**98.1%**

Conclusion

- Sum-Product Networks
  - Deep architecture with clear semantics
  - Tractable probabilistic graphical model
- Related work
  - Decision SPNs (Melibari et al., AAAI-2016)
  - Dynamic (recurrent) SPNs (Melibari et al., PGM-2016)
- Future work:
  - PyTorch library for SPNs
  - SPNs for conversational agents