



**Università degli Studi di Bari**

Dipartimento di Informatica



**LACAM**

Machine Learning

# Learning Sum-Product Networks

Nicola Di Mauro   Antonio Vergari

*September 2016*

# The need for SPN

Sum-Product Networks (SPNs) are a type of probabilistic model<sup>1</sup>

- ▶ for Probabilistic Graphical Models (PGMs) there exist multi-purpose inference tools
  - ▶ the computational effort scales unproportional to the complexity of the graph
  - ▶ solution: using approximate inference

---

<sup>1</sup> H. Poon and P. Domingos, *Sum-Product Network: a New Deep Architecture*, UAI 2011

# The need for SPNs

*Why should you work on SPNs?*

- ▶ exact tractable inference
- ▶ NN for which structure learning is easy

SPNs represent probability distributions and a corresponding exact inference machine for the represented distribution at the same time

# Representation

How do we represent the world?

How do we represent the world in a computer?

How do we represent the world in a neural network?

How do we represent the world in a robot?

How do we represent the world in a game?

How do we represent the world in a language model?

How do we represent the world in a recommendation system?

How do we represent the world in a search engine?

How do we represent the world in a social network?

How do we represent the world in a video game?

How do we represent the world in a self-driving car?

How do we represent the world in a virtual reality environment?

How do we represent the world in a video game?

How do we represent the world in a self-driving car?

How do we represent the world in a virtual reality environment?

How do we represent the world in a social network?

How do we represent the world in a search engine?

How do we represent the world in a recommendation system?

How do we represent the world in a language model?

How do we represent the world in a game?

How do we represent the world in a robot?

How do we represent the world in a neural network?

How do we represent the world in a computer?

How do we represent the world?

# Density estimation

# (Different kinds of) Inference

Different kinds of queries:

- ▶  $p(\mathbf{X})$  (evidence)
- ▶  $p(\mathbf{E}), \mathbf{E} \subset \mathbf{X}$  (marginals)
- ▶  $p(\mathbf{Q}|\mathbf{E}), \mathbf{Q}, \mathbf{E} \subset \mathbf{X}, \mathbf{Q} \cap \mathbf{E} = \emptyset$  (conditionals)
- ▶  $\arg \max_{\mathbf{q} \sim \mathbf{Q}} p(\mathbf{q}|\mathbf{E})$  (MPE assignment)
- ▶ complex queries

# Tractable Probabilistic Models

# Sum-Product Networks



# Scopes

- **Global Scope**
  - Variables defined outside any function or block
  - Accessible from anywhere in the program
  - Example: `global_var = 10`
- **Local Scope**
  - Variables defined inside a function or block
  - Only accessible within that function or block
  - Example: `def my_func(): local_var = 5`
- **Module Scope**
  - Variables defined within a module but outside functions
  - Accessible within the module and other modules that import it
  - Example: `module_var = 20` inside a module
- **Class Scope**
  - Variables defined within a class
  - Accessible within the class and its instances
  - Example: `class MyClass: class_var = 30`
- **Function Scope**
  - Variables defined within a function
  - Accessible only within the function
  - Example: `def my_func(): func_var = 40`
- **Block Scope**
  - Variables defined within a specific block of code (e.g., loops, conditionals)
  - Accessible only within that block
  - Example: `if condition: block_var = 50`

# Structural Properties

# Inference

- **Bayesian Inference** (aka **Bayesian Statistics**)
  - **Bayesian Inference** is a statistical approach that uses Bayes' theorem to update the probability of a hypothesis as more evidence or data is observed.
  - It is a probabilistic framework for making inferences about unknown parameters based on observed data.

- **Bayesian Inference** is a probabilistic framework for making inferences about unknown parameters based on observed data.
- It is a probabilistic framework for making inferences about unknown parameters based on observed data.

- **Bayesian Inference** is a probabilistic framework for making inferences about unknown parameters based on observed data.
- It is a probabilistic framework for making inferences about unknown parameters based on observed data.

- **Bayesian Inference** is a probabilistic framework for making inferences about unknown parameters based on observed data.
- It is a probabilistic framework for making inferences about unknown parameters based on observed data.

- **Bayesian Inference** is a probabilistic framework for making inferences about unknown parameters based on observed data.
- It is a probabilistic framework for making inferences about unknown parameters based on observed data.

- **Bayesian Inference** is a probabilistic framework for making inferences about unknown parameters based on observed data.
- It is a probabilistic framework for making inferences about unknown parameters based on observed data.

# Complete evidence

# Marginal inference

# MPE inference

# Interpretation

- **Interpretation** is the process of understanding the meaning of a text or a situation.
- It involves analyzing the context, the language used, and the intentions of the author or speaker.
- Interpretation is a subjective process, as different people may have different interpretations of the same text or situation.

- Interpretation is a key skill in many fields, including literature, history, and social sciences.
- It is also an important part of everyday life, as we constantly interpret the actions and words of others.
- Interpretation is a process that involves both rational and emotional factors.

- Interpretation is a process that involves both the individual and the social context.
- It is a process that is constantly evolving, as new interpretations are constantly being developed.
- Interpretation is a process that is essential for understanding the world around us.

- Interpretation is a process that is essential for understanding the world around us.
- It is a process that is constantly evolving, as new interpretations are constantly being developed.
- Interpretation is a process that is essential for understanding the world around us.

- Interpretation is a process that is essential for understanding the world around us.
- It is a process that is constantly evolving, as new interpretations are constantly being developed.
- Interpretation is a process that is essential for understanding the world around us.

- Interpretation is a process that is essential for understanding the world around us.
- It is a process that is constantly evolving, as new interpretations are constantly being developed.
- Interpretation is a process that is essential for understanding the world around us.

# Interpretation

- ▶ probabilistic model
- ▶ deep feedforward neural network



# Network Polynomials

# Arithmetic Circuits

Differences with ACs:

- ▶ probabilistic semantics
  - ▶ learning
  - ▶ sampling
- ▶ no shared weights

# SPNs as NNs (I)

SPNs are a particular kind of ***labelled constrained and fully probabilistic*** neural networks.

**Labelled:** each neuron is associated a *scope*

**Constrained:** completeness and decomposability determine network topology.

**Fully probabilistic:** each valid sub-SPN is still a valid-SPN.

SPNs provide a direct encoding of the input space into a deep architecture → ***visualizing representations*** (back) into the ***input space***.

## SPNs as NNs (II)

A classic MLP hidden layer computes the function:

$$h(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

SPNs can be reframed as *DAGs* of MLPs, each sum layer computing:

$$\mathbf{S}(\mathbf{x}) = \log(\mathbf{W}\mathbf{x})$$

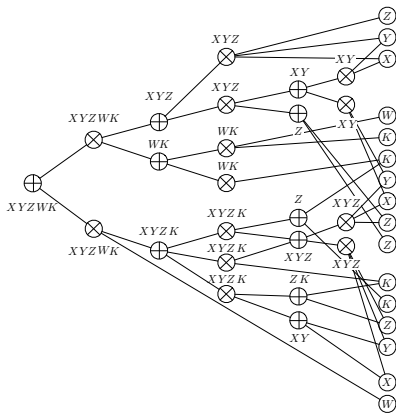
and product layers computing:

$$\mathbf{S}(\mathbf{x}) = \exp(\mathbf{P}\mathbf{x})$$

where  $\mathbf{W} \in \mathbb{R}_+^{s \times r}$  and  $\mathbf{P} \in \{0, 1\}^{s \times r}$  are the weight matrices:

$$\mathbf{W}_{(ij)} = \begin{cases} w_{ij} & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases} \quad \mathbf{P}_{(ij)} = \begin{cases} 1 & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases}$$

# SPNs as NNs (III)



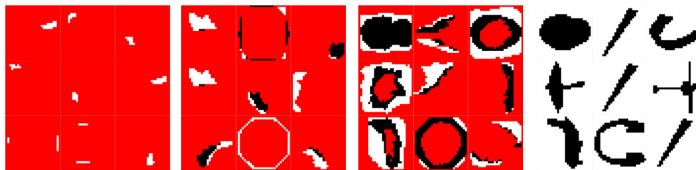
## SPNs as NNs (IV): filters

Learned features as images maximizing neuron activations []:

$$\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}, \|\mathbf{x}\|=\gamma} h_{ij}(\mathbf{x}; \boldsymbol{\theta}).$$

With SPNs, joint solution as an MPE assignment for all nodes (linear time):

$$\mathbf{x}_{|\text{sc}(n)}^* = \operatorname{argmax}_{\mathbf{x}} S_n(\mathbf{x}_{|\text{sc}(n)}; \mathbf{w})$$



→ *scope length* ( $|\text{sc}(n)|$ ) correlates with feature abstraction level

# SPNs as BNs I

Zhao and Poupart

# SPNs as BNs II

Peharz



# Myths about SPNs

**SPNs are PGMs.** false

**SPNs are convolutional NNs.** false

SPNs

# Learning

- **Learning** is the process of acquiring new information or skills through experience, study, or teaching.
- It involves the **acquisition** of knowledge and the **modification** of behavior based on that knowledge.
- Learning can occur through **direct experience**, **observation**, or **instruction**.

- The process of learning is often **gradual** and **ongoing**, as individuals continue to acquire new information and skills throughout their lives.
- Learning is a **fundamental** aspect of human development and is essential for **growth** and **progress**.
- It is a **dynamic** process that is influenced by a variety of factors, including **motivation**, **environment**, and **teaching methods**.

- Learning is a **complex** process that involves the **integration** of new information with existing knowledge and skills.
- It is a **continuous** process that is **essential** for **personal** and **professional** growth.
- Learning is a **fundamental** aspect of human development and is essential for **growth** and **progress**.

- Learning is a **dynamic** process that is influenced by a variety of factors, including **motivation**, **environment**, and **teaching methods**.
- It is a **continuous** process that is **essential** for **personal** and **professional** growth.
- Learning is a **fundamental** aspect of human development and is essential for **growth** and **progress**.

- Learning is a **dynamic** process that is influenced by a variety of factors, including **motivation**, **environment**, and **teaching methods**.
- It is a **continuous** process that is **essential** for **personal** and **professional** growth.
- Learning is a **fundamental** aspect of human development and is essential for **growth** and **progress**.

- Learning is a **dynamic** process that is influenced by a variety of factors, including **motivation**, **environment**, and **teaching methods**.
- It is a **continuous** process that is **essential** for **personal** and **professional** growth.
- Learning is a **fundamental** aspect of human development and is essential for **growth** and **progress**.

# Structure Learning

Structure matters

Alternatives:

- ▶ handcrafted structure, then weight learning []
- ▶ random structures, then weight learning []
- ▶ learned from data

# Why Structure Quality Matters

Tractable inference is guaranteed *if the network size is polynomial* in # vars.

Smaller networks, faster inference (comparing network sizes is better than comparing inference times).

*Deeper* networks are possibly *more expressively efficient* [Martens2014, Zhao2015 ].

Structural simplicity as a bias: overcomplex networks may not generalize well.

Structure quality desiderata: **smaller** but **accurate**, **deeper** but not wider, SPNs.

# LearnSPN (I)

Build a tree-like SPN by recursively split the data matrix:

- ▶ splitting columns into pairs by a greedy **G Test** based procedure with threshold  $\rho$ :

$$G(X_i, X_j) = 2 \sum_{x_i \sim X_i} \sum_{x_j \sim X_j} c(x_i, x_j) \cdot \log \frac{c(x_i, x_j) \cdot |T|}{c(x_i)c(x_j)}$$

- ▶ clustering instances into  $|C|$  sets with **online Hard-EM** with cluster penalty  $\lambda$ :

$$Pr(\mathbf{X}) = \sum_{C_i \in \mathbf{C}} \prod_{X_j \in \mathbf{X}} Pr(X_j | C_i) Pr(C_i)$$

weights are estimated as cluster proportions

- ▶ if there are less than  $m$  instances, put a **naive factorization** over leaves
- ▶ each univariate distribution get **ML estimation** smoothed by  $\alpha$

Hyperparameter space:  $\{\rho, \lambda, m, \alpha\}$ .

# LearnSPN (II)

	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$
1					
2					
3					
4					
5					
6					
7					
8					

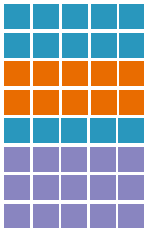
## LearnSPN (II)

$X_1$   $X_2$   $X_3$   $X_4$   $X_5$

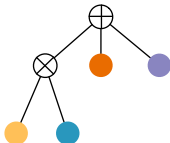
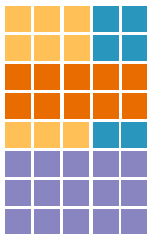


## LearnSPN (II)

$X_1$   $X_2$   $X_3$   $X_4$   $X_5$



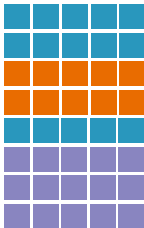
$X_1$   $X_2$   $X_3$   $X_4$   $X_5$





# LearnSPN (II)

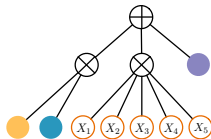
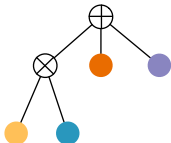
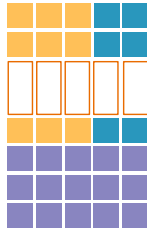
$X_1$   $X_2$   $X_3$   $X_4$   $X_5$



$X_1$   $X_2$   $X_3$   $X_4$   $X_5$



$X_1$   $X_2$   $X_3$   $X_4$   $X_5$



## LearnSPN (III)

LearSPN performs two interleaved ***greedy hierarchical*** divisive ***clustering*** processes (co-clustering on the data matrix).

Fast and simple. But both processes never look back and are committed to the choices they take.

Online EM does not need to specify the number of clusters  $k$  in advance. But overcomplex structures are learned by exploding the number of sum node children.

Tractable leaf estimation. But naive factorization independence assumptions may be too strong.

ML estimations are effective. But they are not robust to noise, they can overfit the training set easily.

# LearnSPN

LearnSPN: A Self-Supervised

Learning Framework for

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

Learning SPN

# LearnSPN-b

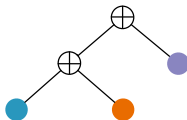
Observation: each clustering process benefits from the other one  
improvements/highly suffers from other's mistakes.

Idea: slowing down the processes by limiting the number of nodes to split into.  
SPN-B, variant of LearnSPN that uses EM for mixture modeling with  $k = 2$   
to cluster rows.

No need for  $\lambda$  anymore.

Objectives:

- ▶ not committing to complex structures too early
- ▶ same expressive power as LearnSPN
- ▶ reducing node out fan increases the depth
- ▶ same accuracy, smaller networks



# LearnSPN-b: depth VS size

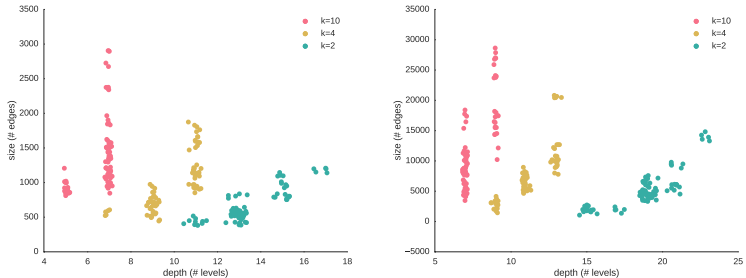


Figure : Comparing network sizes and depths while varying the max number of sum node children splits ( $k \in \{10, 4, 2\}$ ). Each dot is an experiment in the grid search hyperparameter space performed by SPN-B on NLTCs (left) and Plants (right).

# **New Tendencies in Structure Learning**

Pruning and compressing

# Parameter Learning

# Hard/Soft Parameter Learning



# Bayesian Parameter Learning

# Parameter Learning VS LearnSPN

Collapsed Variational Inference is useless : D

# Representation Learning

# Extracting Embeddings

Problem extracting embeddings

# **Supervised classification**

# Filtering Embeddings

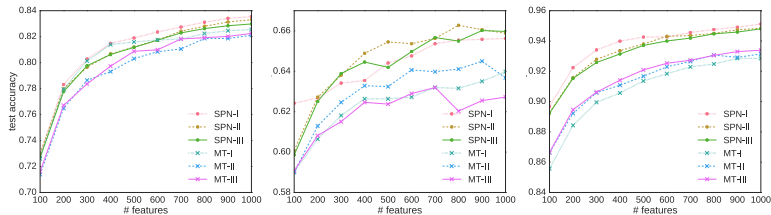
Filtering embeddings by:

- ▶ node type
- ▶ scope
- ▶ scope length

# Random Marginal Queries

Generate embeddings by asking several random queries to a black box density estimator.

Eg. marginals:  $e_j^i = P_\theta(\mathbf{Q}_j = \mathbf{x}_{\mathbf{Q}_j}^i)$ , according to estimator  $\theta$  where  $\mathbf{Q}_j \subseteq \mathbf{X}, j = \dots, k$ .



# Encoding/Decoding Embeddings

MPN as autoencoders<sup>2</sup>.

---

<sup>2</sup>Vergari et al. Encoding and Decoding Representations with Sum-Product Networks, 2016, to appear



# Applications

• **Modeling** (e.g., **Bayesian networks**)

• **Classification** (e.g., **Naïve Bayes**)

• **Regression** (e.g., **Bayesian linear regression**)

• **Decision trees** (e.g., **Bayesian decision trees**)

• **Hidden Markov models** (e.g., **Bayesian HMMs**)

• **Markov decision processes** (e.g., **Bayesian MDPs**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

• **Bayesian networks** (e.g., **Bayesian networks**)

# **Applications I: computer vision**

## **Applications II: language modeling**

## **Applications III: activity recognition**

## **Applications IV: speech**

## **Trends & What to do next**



# awesome-spn

A curated and structured list of resources about SPNs<sup>3</sup>.

<https://github.com/arranger1044/awesome-spn>

---

<sup>3</sup>Inspired by the SPN page <http://spn.cs.washington.edu/> at the Washington University