Tractable probabilistic models for hierarchical data

Tomáš Pevný

Czech Technical University in Prague

February 1, 2024

Problem statement

We assume a set of samples $\{x_i | x_i \in \mathcal{X}\}_{i=1}^{n}$.

We want to fit a model $p(x|\theta)$by maximizing likelihood

$$\arg\max_{\theta} \sum_{i} \log p(x_i|\theta)$$

such that

- $p(x|\theta)$ is a valid probability distribution
- $p(x|\theta)$ is tractable,

What is tractability?

The model $p(x)$ is tractable with respect to $f \in \mathcal{F}$ if the integral

$$\int_{\mathcal{X}} f(x)p(x|\theta)dx,$$

can by computed in polynomial time with respect to the size of $p(x|\theta)$.

Tractability

Fully factorized

NaiveBayes *AndOrGraphs* *PDGs*

Trees *PSDDs* *CNets* *LTMs* *SPNs* *NADEs*

Thin Junction Trees *ACs* *MADEs* *MAFs* *VAEs*

Polytrees *FVSBNs* *TACs* *IAFs* *NAFs* *RAEs*

Mixtures *BNs* *NICE* *FGs* *GANs*

RealNVP *MNs*

Used with the permission of Guy Van den Broeck.

Sum-Product networks

- in **leaf node** $\mathsf{L}(x)$
$$p(x) = p(x|\theta_{\mathsf{L}})$$
- in **sum node** $\mathsf{S}(x)$
$$p(x) = \sum_{\mathsf{N} \in \mathbf{ch}(\mathsf{S})} w_{\mathsf{N}} p_{\mathsf{N}}(x_{\psi(\mathsf{N})})$$
- in **product node** $\mathsf{P}(x)$
$$p(x) = \prod_{\mathsf{N} \in \mathbf{ch}(\mathsf{P})} p_{\mathsf{N}}(\mathsf{N}(x_{\psi(\mathsf{N})})),$$

$\psi(\mathsf{N})$ is a scoping function.

Special cases of SPNs

- Mixture model

$$p(\vec{x}) = \sum_{i=1}^{n} w_i p_i(x), \quad \sum_{i=1}^{n} w_i = 1$$

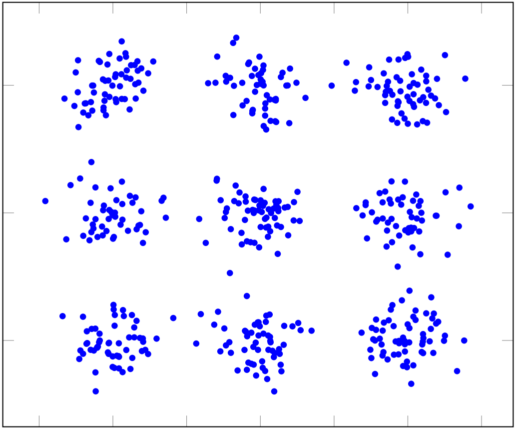
- Product of marginals (naive bayes)

$$p(\vec{x}) = \prod_{i=1}^{n} p_i(x_i)$$

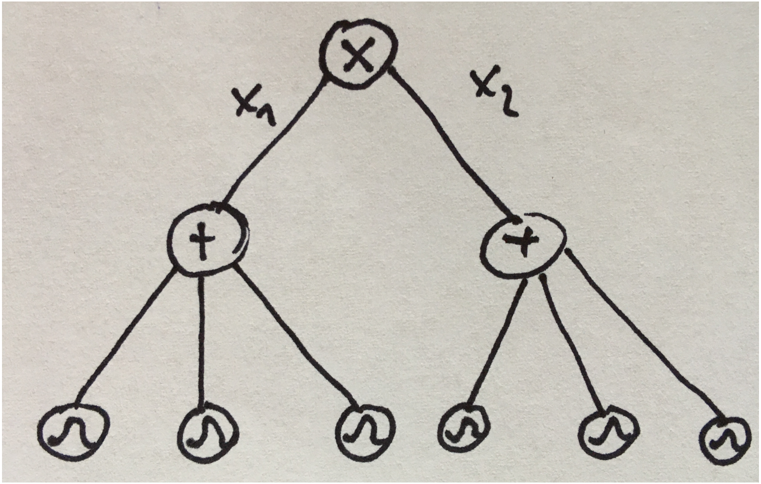
- Diagonal mixtures

$$p(\vec{x}) = \sum_{i=1}^{n} w_i \prod_{j=1}^{n} p_{ij}(x_j)$$

Are SPNs useful?



Are SPNs useful?



Scoping function

Scoping function $\psi(\mathsf{N})$ is used to specify a subspace on which the distribution of a given node is defined.

Example

Let, $\mathcal{X} = \mathbb{R}^n$, but the node L operates just on features (x_1, x_5, x_6), the scope function $\psi(\mathsf{L}) = (x_1, x_5, x_6)$.

Tractability

Tractability of SPNs

Sum-Product networks are tractable

- if they are *smooth* and *decomposable*
- and for query f holds

$$f(x) = \prod_{\psi_u \in \cup\{\psi(\mathsf{L})\}} f_u(\psi_u),$$

where

- $\cup\{\psi(\mathsf{L})\}$ is the set of all possible scopes of leafnodes
- probability distribution of leafnodes are tractable

Smoothing and Decomposability

- **Smootheness:** All childs of sumnode S has to have the same scope as S, i.e.

$$\psi(a_1) = \psi(a_2)\ \forall a_1, a_2 \in \mathbf{ch}[\mathrm{S}],$$

  and weights $w_i \geq 0$ and $\sum_i wi = 1$.
- **Decomposability:** Scopes od childs of productnode P are pairwise disjoint

$$\bigcap_{a \in \mathbf{ch}[\mathrm{P}]} \psi(a) = \emptyset,$$

  but complete $\psi(\mathrm{P}) = \bigcup_{a \in \mathbf{ch}[\mathrm{P}]} \psi(a)$.
- The scope of rootnode R is over all features, $\psi(\mathrm{R}) = [P]$.

Recursive computation of integrals

For *smooth* and *decomposable* SPNs,
the integral $I = \int f(x)p(x)dx$ can be computed recursively as follows:

$$I_u = \begin{cases} \sum_{c \in \mathbf{ch}(u)} w_{u,c} I_c, & \text{for } u \in \mathsf{S}, \\ \prod_{c \in \mathbf{ch}(u)} I_c, & \text{for } u \in \mathsf{P}, \\ \int f_u(\psi_u) p_u(\psi_u) d\psi_u, & \text{for } u \in \mathsf{L}, \end{cases}$$

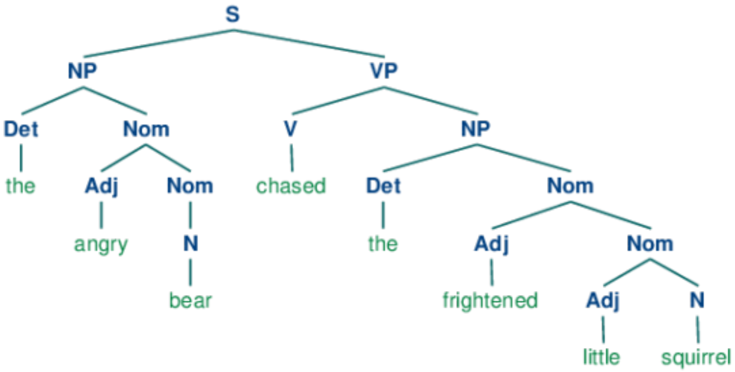
Hierarchically-Structured trees

also called Hierarchical Multi-Instance Learning

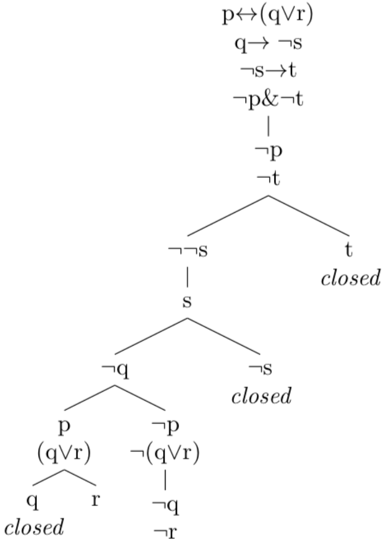
Motivation


```
{
  services: [
  {
    port: 22,
    protocol: tcp
  },
  {
    port: 4070,
    protocol: tcp
  },
  {
    port: 4071,
    protocol: tcp
  },
  {
    port: 5353,
    protocol: udp
  }],
  device_id: 8bb8971c-5983-4baa-9753-f0ac21faf162,
  ip: 192.168.1.80,
  mac: ac:63:be:a5:50:43,
  mdns_services: [_workstation._tcp.local., _ssh._tcp.local., _sftp-ssh._tcp.local.]
}
```

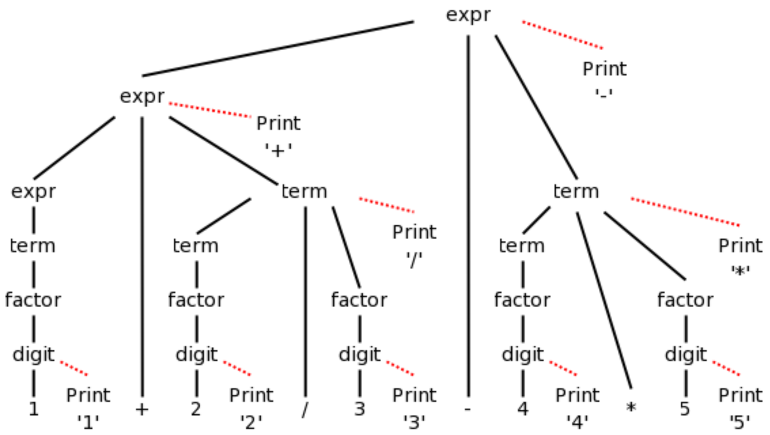
Motivation — semantic tree



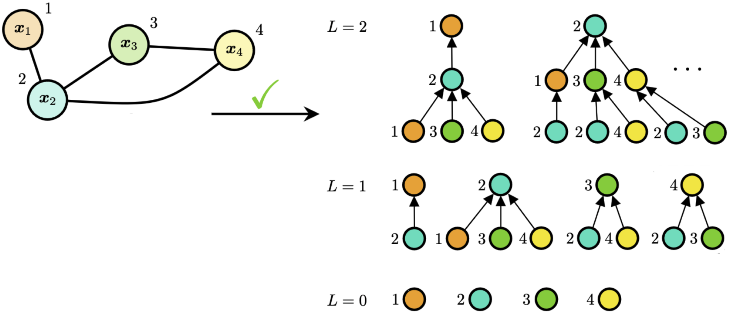
Motivation — logic



Motivation — abstract syntax tree



Motivation — general graph



Errica, Federico, and Mathias Niepert. "Tractable Probabilistic Graph Representation Learning with Graph-Induced Sum-Product Networks.", 2023.

Types of nodes

```
{
  services: [
  {
    port: 22,
    protocol: tcp
  },
  {
    port: 4070,
    protocol: tcp
  },
  {
    port: 4071,
    protocol: tcp
  },
  {
    port: 5353,
    protocol: udp
  }],
  device_id: 8bb8971c-5983-4baa-9753-f0ac21faf162,
  ip: 192.168.1.80,
  mac: ac:63:be:a5:50:43,
  mdns_services: [_workstation._tcp.local., _ssh._tcp.local., _sftp-ssh._tcp.local.]
}
```

- Dictionaries
- Lists (sets)
- Leafs

Set node

- Set node computes a probability density over a hyper-space

$$\bar{\mathcal{X}} = \cup_{m=0}^{\infty} \underbrace{\mathcal{X} \times \ldots \times \mathcal{X}}_{m}$$

- probability density of a *set node* B is computed as

$$p_{\mathrm{B}}(x) = p(m) c^m m! p(x_1, \ldots, x_m)$$

where

  - $p(m)$ is a cardinality distribution
  - $p(x)$ is a probability distribution on $\mathcal{X}$
  - c is a constant for unit normalization

Why the factorial?

- We need probability distribution on sets $\{x_1, \ldots, x_m\}$, but $p(x_1, \ldots, x_m)$ is a probability distribution on Cartesian space.
- We define distribution on sets as

$$p(\{x_1, \ldots, x_m\}) = \sum_{\pi \in \text{perm}} p(x_{\pi(1)}, \ldots, x_{\pi(m)})$$

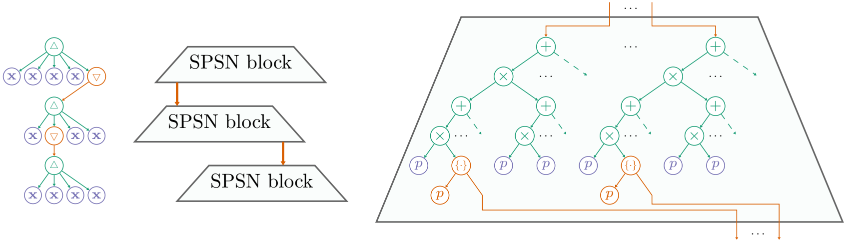
- If p is *exchengeable*, it can be simplified to

$$p(\{x_1, \ldots, x_m\}) = m! p^{\text{sym}}(x_1, \ldots, x_m)$$

- *cluster model*

$$p(\{x_1, \ldots, x_m\}) = m! \prod_{i=1}^{m} p(x_i)$$

Construction of Sum-Product-Set network



(a) Schema (b) SPSN (c) SPSN block

Recursive computation of integrals

For *smooth* and *decomposable* SPSNs,
the integral $\int f(x)p(x)dx$ can be computed recursively as follows:

$$I_u = \begin{cases} \sum_{k=0}^{\infty} p(k) \prod_{i=1}^{k} I_i, & \text{for } u \in \mathsf{B}, \\ \sum_{c \in \mathbf{ch}(u)} w_{u,c} I_c, & \text{for } u \in \mathsf{S}, \\ \prod_{c \in \mathbf{ch}(u)} I_c, & \text{for } u \in \mathsf{P}, \\ \int f_u(\psi_u) p_u(\psi_u) d\psi_u, & \text{for } u \in \mathsf{L}, \end{cases}$$

Experimental comparison

dataset	MLP	GRU	LSTM	HMIL	SPSN
chess	**0.41±0.03**	**0.41±0.05**	0.34±0.04	0.39±0.02	0.39±0.03
citeseer	0.69±0.02	0.74±0.01	0.74±0.02	**0.75±0.01**	**0.75±0.01**
cora	0.75±0.03	**0.86±0.01**	0.84±0.01	0.85±0.00	**0.86±0.01**
genes	0.99±0.01	**1.00±0.01**	0.98±0.01	**1.00±0.01**	0.95±0.01
hepatitis	0.86±0.02	**0.88±0.01**	0.87±0.03	**0.88±0.02**	**0.88±0.02**
mutagenesis	**0.84±0.02**	0.83±0.02	0.82±0.04	0.83±0.00	**0.84±0.02**
uwcse	0.84±0.02	**0.87±0.03**	0.85±0.02	0.86±0.03	0.84±0.02
webkp	0.77±0.02	**0.82±0.01**	0.81±0.02	**0.82±0.01**	0.81±0.02
rank	3.62	**1.62**	3.88	**1.62**	2.38

Conclusion

- We have extended SPNs to a class of HS-tres (HMIL)
- It is the first generative model for this class of problems.
- Seems to deliver similar accuracy to non-tractable models.
- Is it useful?