

# Estudo sobre Sum-Product Networks e Aprendizagem Profunda

Renato Lui Geh

Instituto de Matemática e Estatística  
Universidade de São Paulo

8 de junho de 2016



INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
UNIVERSIDADE DE SÃO PAULO

# Índice

- 1 Definição
- 2 Propriedades
- 3 Uma definição alternativa
- 4 Classificação por Naive Bayes
- 5 Aprendizado estrutural de SPNs
- 6 Referências e Bibliografia

# Relembrando...

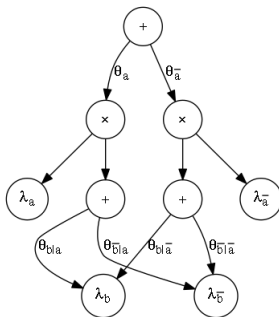
## Definição

Uma SPN  $S$  é um DAG com três tipos de nós: soma, produto e indicadores. Todo nó indicador é uma folha. Todo nó soma tem pais produto, e todo nó produto tem pais soma. Toda aresta com destino a um nó soma tem uma aresta com um peso associado. O valor de um nó soma  $i$  é  $\sum_{j \in Ch(i)} w_{ij} v_j$  e o valor de um nó produto  $i$  é  $\prod_{j \in Ch(i)} v_j$ , onde  $Ch(i)$  é o conjunto de filhos de  $i$ ,  $v_i$  é o valor do nó  $i$  e  $w_{ij}$  é o peso associado a aresta  $i \rightarrow j$ . Uma SPN representa um *network polynomial* de uma distribuição de probabilidade, e os indicadores da função são as folhas da SPN. O valor de uma SPN é o valor do nó raiz.

---

[PD11]

# Relembrando...



# Sub-SPNs

## Proposição

*Seja um nó arbitrário  $i$  de uma SPN  $S$ , então  $S_i$  é uma sub-SPN que tem nó raiz em  $i$ .*

# Prova sub-SPNs

## Demonstração.

Considere o caso base em que  $i$  é um nó indicador. Um nó indicador é uma distribuição de probabilidade monovariável. Portanto  $i$  é uma distribuição de probabilidade e pode ser representada por uma SPN, que no caso possui apenas um nó.

Se  $i$  é um nó soma, então o valor de  $i$  é  $v_i = \sum_{j \in Ch(i)} w_{ij} v_j$ . A soma de várias distribuições de probabilidade é uma distribuição de probabilidade. Portanto um nó soma é representável por uma SPN.

Caso  $i$  seja um nó produto, então o valor de  $i$  é  $v_i = \sum_{j \in Ch(i)} v_j$ . A multiplicação de distribuições de probabilidade é bem definida e é uma distribuição de probabilidade. Um nó produto é uma SPN. □

# Completude

## Definição (Completude)

Uma SPN  $S$  é completa se e somente se, para todo nó soma  $i$ , o escopo de  $i$  é igual par-a-par  $Sc(S_i) = Sc(S_j)$  ao escopo de cada filho  $j \in Ch(i)$ .

# Consistência

## Definição (Consistência)

Uma SPN  $S$  é consistente se e somente se, para todo nó produto  $i$ , nenhum filho de  $i$  tem valor diferente dos outros filhos.



# Validade

## Definição (Validade)

Uma SPN  $S$  é válida se  $S$  é consistente e completa.

# Decomponibilidade

## Definição (Decomponibilidade)

Uma SPN  $S$  é decomponível se e somente se, para cada par  $c_1, c_2 \in Ch(i)$  para qualquer  $i$  nó produto em  $S$ ,  $Sc(c_1) \cap Sc(c_2) = \emptyset$ .

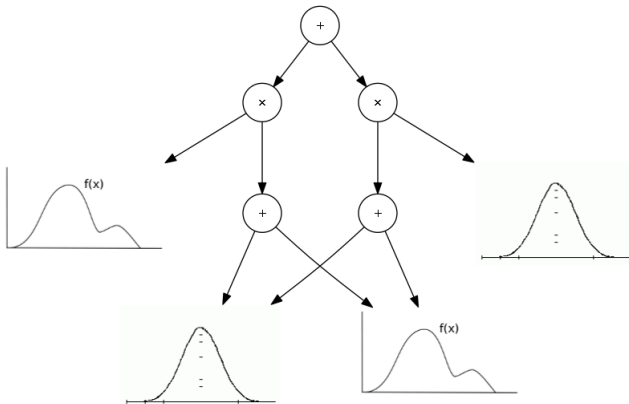
# Uma definição alternativa

## Definição

Uma SPN tem uma definição recursiva. Definimos que uma SPN  $S_i$  pode ser apenas:

- (i) Uma distribuição monovariável  $p(\mathbf{X})$  ou;
- (ii) Um nó soma tal que  $S_i = \sum_{j \in Ch(i)} w_{ij} v_j$  onde para cada filho  $j, k \in Ch(i)$ ,  $Sc(S_j) = Sc(S_k)$  ou;
- (iii) Um nó produto tal que  $S_i = \prod_{j \in Ch(i)} v_j$  onde para cada filho  $j, k \in Ch(i)$ ,  $Sc(S_j) \cap Sc(S_k) = \emptyset$ .

# Uma definição alternativa

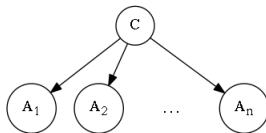


# Classificação por Naive Bayes

$C$  : variável classe

$\mathbf{A} = \{A_1, \dots, A_n\}$  : variáveis atributos

$A_i \perp A_j \equiv A_i \perp_d A_j$ , para  $1 \leq i, j \leq n$  e  $i \neq j$



# Classificação por Naive Bayes

Pelo Teorema da Fatorização:

$$\Pr(C, A_1, \dots, A_n) = \Pr(C) \prod_{i=1}^n \Pr(A_i | C)$$

Classificação resume-se a encontrar um máximo  $c$ :

$$\arg \max_c \left( \Pr(C = c | A_1 = a_1, \dots, A_k = a_k) = \frac{\Pr(C = c, A_1 = a_1, \dots, A_k = a_k)}{\Pr(A_1 = a_1, \dots, A_k = a_k)} \right)$$

# Aprendizado de Naive Bayes por MLE

MLE (Maximum Likelihood Estimation)  $\equiv$  Máxima verossimilhança

Aprender uma Naive Bayes:

**Variável classe**  $\Pr(C = c) = \frac{N[C=c]}{N}$

**$i$ -ésimo atributo**  $\Pr(A_i = a_i | C = c) = \frac{N[A_i=a_i, C=c]}{N[C=c]}$

# Aprendizado estrutural de SPNs

---

## Algoritmo 1 LearnSPN [GD13]

---

**Input** Conjunto  $\mathbf{X}$  de variáveis, conjunto  $\mathbf{I}$  de instâncias

**Output** Uma SPN resultante do aprendizado estrutural

```
1: if  $|\mathbf{X}| = 1$  then
2:   Retorna uma distribuição monovariável de  $\mathbf{X}$ 
3: end if
4: Tente dividir as variáveis  $\mathbf{X}$  em duas partições  $\mathbf{X}_1$  e  $\mathbf{X}_2$  onde  $\mathbf{X}_1$  é
   (aproximadamente) independente de  $\mathbf{X}_2$ 
5: if dá para dividir then
6:   return  $\prod_{i=1}^2 \text{LearnSPN}(\mathbf{X}_i, \mathbf{I})$ 
7: else
8:   Divida as instâncias  $\mathbf{I}$  em partições  $\mathbf{I}_1$  e  $\mathbf{I}_2$  tal que  $\mathbf{I}_1$  e  $\mathbf{I}_2$  sejam o
   mais similares possíveis.
9:   return  $\sum_{i=1}^2 \frac{|\mathbf{I}_i|}{|\mathbf{I}|} \text{LearnSPN}(\mathbf{X}, \mathbf{I}_i)$ 
10: end if
```

---



# Referências e Bibliografia I



Robert Gens e Pedro Domingos. “Learning the Structure of Sum-Product Networks”. Em: *International Conference on Machine Learning* 30 (2013).



Hoifung Poon e Pedro Domingos. “Sum-Product Networks: A New Deep Architecture”. Em: *Uncertainty in Artificial Intelligence* 27 (2011).