Estudo sobre Sum-Product Networks e Aprendizagem Profunda

Renato Lui Geh

Instituto de Matemática e Estatística
Universidade de São Paulo

29 de junho de 2016

Índice

Outros modelos

- Redes Bayesianas
- Redes de Markov
- Máquinas restritas de Boltzmann

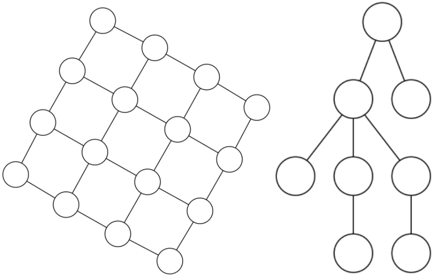
Representam uma distribuição de probabilidade de forma compacta. Mas. . .

Inferência intratável ou aproximada!

Aprendizado difícil!

Largura de árvore

Largura de árvore grande $\Rightarrow$ inferência intratável.



SPNs computam inferência exata e tratável mesmo quando a largura de árvore é grande.

Arquiteturas profundas

DBNs: Deep Belief Networks [HS06]

CDBNs: Convolutional Deep Belief Networks [Lee+09]

DBMs: Deep Boltzmann Machines [SH10]

Arquiteturas profundas são mais interessantes [Ben09]. Mas. . .

Inferência mais difícil ainda!

Aprendizado mais difícil também!

Representatividade

1. Modelos tratáveis existentes

vs

2. Modelos baseados em grafos clássicos

vs

3. Redes Soma-Produto (Sum-Product Networks)

1 e 3 ambos tem inferência e aprendizado tratáveis.

$$3 >_g 1$$

Definição

Definição

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

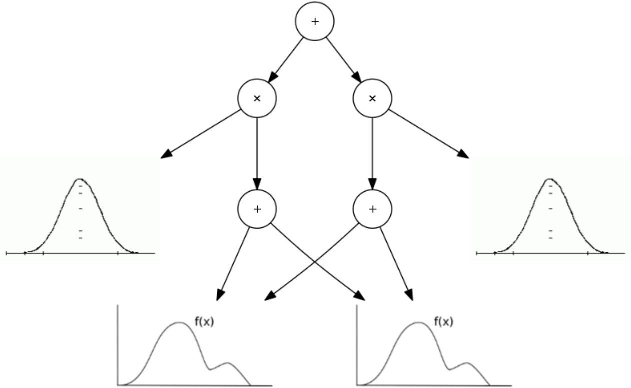
(i) Uma distribuição monovariável $p(\mathbf{X})$ ou;

(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 $j, k \in Ch(i)$, $Sc(S_j) \cap Sc(S_k) = \emptyset$.

[GD13]

Definição



Uma visão mais intuitiva

Nós internos: variáveis latentes – camadas ocultas:

- $\oplus$**:** mistura de distribuições – "semelhança entre instâncias";
- $\otimes$**:** independência entre variáveis.

Nós folhas: Valoração/instanciação das variáveis.

Uma visão mais intuitiva

Nós internos: variáveis latentes – camadas ocultas:

- $\oplus$: mistura de distribuições – "semelhança entre instâncias";
- $\otimes$: independência entre variáveis.

Nós folhas: Valoração/instanciação das variáveis.

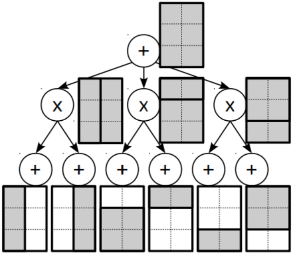
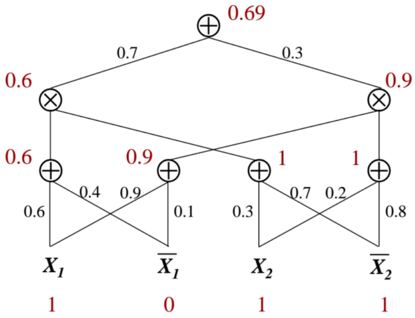


Figura: *Learning the Structure of Sum-Product Networks Using Clustering on Variables*, Dennis e Ventura [DV12]

Inferência



$$\lambda_{X_1} = 1, \lambda_{\overline{X}_1} = 0, \lambda_{X_2} = 1, \lambda_{\overline{X}_2} = 1$$

$$S = \Pr(X_1 = true) = f(x_1) = 0.69$$

Aprendizado

Queremos aprender uma SPN profunda.

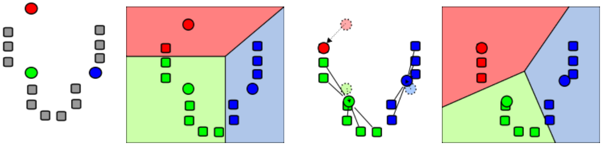
Lembrando...

$+$ semelhanças nos dados

$\times$ independência entre variáveis

Nós somas

Queremos achar *clusters* de instâncias semelhantes: k-means clustering.



Cada cluster $i = 0..k-1$ será um filho de um nó soma.

Note: todo filho tem mesmo escopo pois todo cluster tem mesmo escopo.

Nós produtos

Queremos achar conjuntos independentes de variáveis: grafo de independência.

Definição

Um grafo de independência é um grafo $G = (X, E)$ com variáveis X como vértices e arestas $E = \{e_{ij} : X_i \text{—} X_j \iff X \not\perp Y\}$.

Como testar independências par-a-par? Teste de independência por Chi-Quadrado ou por Entropia.

Nós produtos

Proposição

Seja um grafo de independência $G = (X, E)$, os k-subgrafos $H_0, \dots, H_k \subseteq G$ desconexos tem escopos independentes par-a-par.

Cada subgrafo H_i será um filho de um nó produto.

Note: todo filho tem escopo disjunto de outro filho pois todo filho é independente par-a-par de seus irmãos.

Nós folhas

Queremos uma distribuição monovariável: contagem de frequências quando $|X| = 1$.

Considere um vetor p onde cada índice $i = 0, \ldots, n$ é uma possível valoração distinta e única de uma variável X e que X pode tomar até n valorações diferentes. Dizemos que p é então uma distribuição sobre X. Se p obedece os axiomas da Teoria de Probabilidade, então p é uma distribuição de probabilidade monovariável com domínio discreto.

Seja um vetor v de tamanho k onde cada v_i é uma valoração de X. Então v é um vetor de frequência de X.

Então p será:

$$p_i = \sum_{v_j = i} \frac{1}{k} = \Pr(X = i)$$

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

```
if |X| = 1 then
    Retorna uma distribuição monovariável de X
end if
Tente dividir as variáveis X em q partições X_1, ..., X_q onde X_i é (aproximadamente) independente de todo X_j para i ≠ j.
if dá para dividir then
    return ∏_{i=1}^{q} LearnSPN(X_i, I)
else
    Divida as instâncias I em partições I_1, ..., I_k tal que I_i seja uma coleção de instâncias mais similares possíveis entre si.
    return ∑_{i=1}^{k} (|I_i| / |I|) LearnSPN(X, I_i)
end if
```

- **if** $|\mathbf{X}| = 1$ **then**
  - Retorna uma distribuição monovariável de $\mathbf{X}$
- **end if**
- Tente dividir as variáveis $\mathbf{X}$ em q partições $\mathbf{X}_1, \ldots, \mathbf{X}_q$ onde $\mathbf{X}_i$ é (aproximadamente) independente de todo $\mathbf{X}_j$ para $i \neq j$.
- **if** dá para dividir **then**
  - **return** $\prod_{i=1}^{q}$ LearnSPN($\mathbf{X}_i$, $\mathbf{I}$)
- **else**
  - Divida as instâncias $\mathbf{I}$ em partições $\mathbf{I}_1, \ldots, \mathbf{I}_k$ tal que $\mathbf{I}_i$ seja uma coleção de instâncias mais similares possíveis entre si.
  - **return** $\sum_{i=1}^{k} \frac{|\mathbf{I}_i|}{|\mathbf{I}|}$ LearnSPN($\mathbf{X}$, $\mathbf{I}_i$)
- **end if**

Código

Se der tempo, agora vou mostrar o código. $\ddot\smile$

Referências e Bibliografia I

Yoshua Bengio. "Learning Deep Architectures for AI". Em: *Foundations and Trends in Machine Learning* (2009).

Aaron Dennis e Dan Ventura. "Learning the Architecture of Sum-Product Networks Using Clustering on Variables". Em: *Advances in Neural Information Processing Systems* 25 (2012).

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

Geoffrey E. Hinton e Ruslan R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". Em: *Science Magazine* (2006).

Referências e Bibliografia II

Honglak Lee et al. "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representarions". Em: *International Conference on Machine Learning (ICML 2009)* (2009).

Ruslan R. Salakhutdinov e Geoffrey E. Hinton. "Deep Boltzmann Machines". Em: *International Conference on Artificial Intelligence and Statistics (AISTATS)* (2010).