



## Motivação

Um dos maiores problemas na área de Aprendizado Computacional em Inteligência Artificial é a questão da intratabilidade da inferência, e portanto aprendizado, em amostras muito grandes, já que a complexidade na maioria dos Modelos Gráficos Probabilísticos (PGM) é exponencial. Apesar de existirem modelos onde a inferência é, de fato, tratável, elas possuem limitações quanto à compactibilidade de suas representações.

Em 2011[PD11], Pedro Domingos e Hoifung Poon introduziram um novo tipo de modelo probabilístico cuja inferência é sempre tratável e ainda assim é mais flexível que muitos outros modelos. Por meio de experimentos também comprovou-se que tanto inferência quanto aprendizado foram mais rápidos e precisos que outras redes profundas.

O objetivo desse estudo é aprender a definição, estrutura e propriedades de Sum-Product Networks e em seguida estudar os vários tipos de aprendizado que podemos efetuar neste modelo.

## Sum-Product Networks

Uma Sum-Product Network (SPN) com variáveis  $x_1, \dots, x_d$  é um grafo enraizado, direcionado e acíclico (DAG) cujas folhas são indicadores  $x_1, \dots, x_d$  e  $\bar{x}_1, \dots, \bar{x}_d$  e cujos nós internos são nós somas ou produtos. Toda aresta  $ij$  onde  $i$  tem origem em um nó soma tem um peso  $w_{ij} \geq 0$  associado. O valor de um nó  $i$  é  $v_i$ . O valor de um nó soma  $i$  é  $\sum_{j \in Ch(i)} w_{ij} v_j$ . O valor de um nó produto  $i$  é  $\prod_{j \in Ch(i)} v_j$ .  $Ch(i)$  é o conjunto de nós filhos de  $i$ . O valor de um nó folha é o valor da própria variável. O valor de uma SPN  $S$  é o valor de sua raiz.

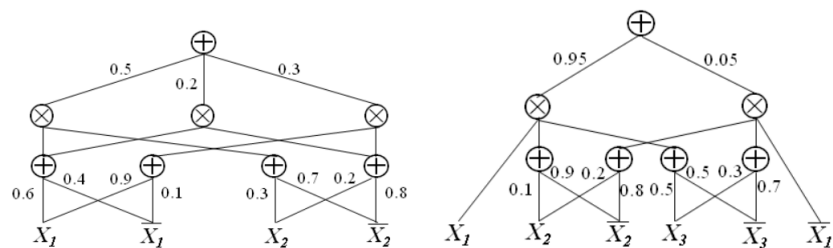


Figura : A esquerda uma SPN implementando uma naive Bayes mixture model. A direita uma SPN implementando uma junction tree. Fonte: Poon e Domingos[PD11].

### Definição

Uma SPN  $S$  é válida sse  $S(e) = \Phi_S(e)$  para toda evidência  $e$ , onde  $\Phi_S$  é a distribuição de probabilidade não-normalizada da SPN  $S$ .

### Definição

Uma SPN é completa sse todos os filhos do mesmo nó soma tem mesmo escopo.

### Definição

Uma SPN é consistente sse nenhuma variável aparece negada em um filho de um nó produto e não-negada em outro.

### Teorema

Uma SPN é válida se ela é completa e consistente.

SPNs válidas são desejáveis pois uma SPN válida computa a probabilidade de evidência em tempo linear em seu tamanho, além de completude e consistência permitirem que a inferência da SPN seja garantidamente eficiente.

## Aprendizado

Podemos gerar uma SPN por aprendizado criando uma SPN densa inicialmente e em seguida aprendermos os pesos. Domingos e Poon[PD11] sugerem um algoritmo que cria uma SPN inicial e em seguida aprende os pesos por Gradient Descent ou Expectation-Maximization (EM) que é mostrada na seção Algoritmo de Aprendizado. No entanto, há outros jeitos de se aprender uma SPN.

Gens e Domingos[GD13] desenvolveram um método de aprendizado que explora dependência e independência dos dados do conjunto de treino para melhorar a flexibilidade e custo de aprendizado de uma SPN. O método proposto usa a expressividade da estrutura de SPNs para alcançar resultados superiores aos experimentos realizados anteriormente.

Outros métodos de aprendizado incluem buscas gulosas[DV15], clustering de variáveis[DV12] e o uso de Non-Parametric Bayesian Sum-Product Networks[LWZ14].

## Algoritmo de Aprendizado

```
Input: Conjunto  $D$  de instâncias sobre variáveis  $X$ .
Output: Uma SPN com estrutura e parâmetros construídos por aprendizado.
/* Cria uma SPN inicial que seja válida. */
 $S \leftarrow \text{GenerateDenseSPN}(X)$ ;
InitializeWeights( $S$ );
repeat
  forall the  $d \in D$  do
    /* Atualiza pesos por Gradient Descent ou EM. */
    UpdateWeights( $S$ , Inference( $S$ ,  $d$ ));
  end
until convergência;
/* Apara arestas com peso  $w_{ij} = 0$  e nós não-raiz sem pais. */
 $S \leftarrow \text{PruneZeroWeights}(S)$ ;
return  $S$ 
```

## Experimentos

Os experimentos mostrados a seguir foram extraídos a partir da implementação do algoritmo mostrado na seção anterior e mostram os resultados do código [DP] implementado por Domingos e Poon e citados em [PD11].

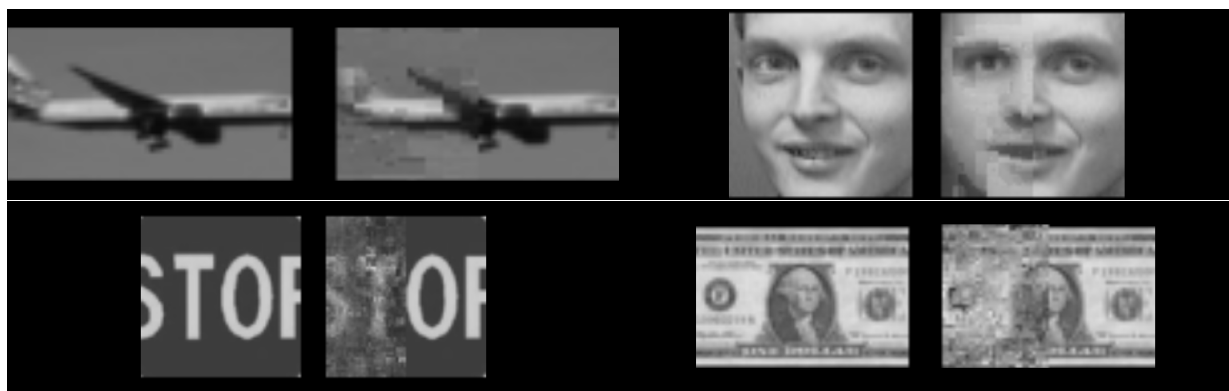


Figura : A saída do algoritmo consiste na compleção do lado esquerdo das imagens a partir de um conjunto de treino. Para cada par de imagens, a imagem da esquerda é a original, enquanto que a direita tem a metade esquerda completada pela SPN e a outra metade igual a da original como evidência.

Arquitetura	Rostos	Motos	Carros
SPN	0.99	0.99	0.98
CDBN	0.95	0.81	0.87

Tabela : Comparação entre SPNs e CDBNs (Convolutional Deep Belief Networks) em classificação (reconhecimento) de imagens.

Pode-se ver que os resultados das SPNs são muito promissores e, dado que o algoritmo produzido por Domingos e Poon não toma muita vantagem da expressividade da estrutura local de SPNs, é fácil notar que ainda há muito espaço para melhorias.

## Trabalhos futuros

Pretende-se estudar a implementação do método de aprendizado proposto por Poon e Domingos[PD11] e explorar mais a fundo as propriedades de uma SPN, principalmente o uso de estrutura local para tornar o aprendizado mais rápido e preciso.

Em seguida planeja-se estudar outros tipos de aprendizado em SPNs, como o introduzido por Gens e Domingos[GD13], buscas gulosas e clustering por Dennis e Ventura[DV12, DV15] e Non-Parametric Bayesian Sum-Product Networks[LWZ14].

## Referências

- Pedro Domingos and Hoifung Poon. Sum-product networks: A new deep architecture (code). URL: <http://spn.cs.washington.edu/spn/>.
- Aaron Dennis and Dan Ventura. Learning the architecture of sum-product networks using clustering on variables. *Advances in Neural Information Processing Systems*, 25, 2012.
- Aaron Dennis and Dan Ventura. Greedy structure search for sum-product networks. *International Joint Conference on Artificial Intelligence*, 24, 2015.
- Robert Gens and Pedro Domingos. Learning the structure of sum-product networks. *International Conference on Machine Learning*, 30, 2013.
- Sang-Woo Lee, Christopher Watkins, and Byoung-Tak Zhang. Non-parametric bayesian sum-product networks. *Workshop on Learning Tractable Probabilistic Models*, 2014.
- Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. *Uncertainty in Artificial Intelligence*, 27, 2011.