Aprendizado Automático de Sum-Product Networks (SPNs)

Aluno: Renato Lui Geh (Bacharelado em Ciência da Computação)

Orientador: Denis Deratani Mauá (IME-USP)

1 Introdução

O objetivo deste projeto de Iniciação Científica é utilizar Aprendizado de Máquina para aprender automaticamente a estrutura de um modelo probabilístico denominado Sum-Product Network (SPN).

Modelos probabilísticos baseados em grafos (PGM) têm como objetivo representar distribuições de probabilidade de forma compacta.

Para extrair conhecimento de um modelo probabilístico, computa-se inferência. Inferência na maioria dos modelos gráficos é intratável, já que o número de termos na distribuição é exponencial.

Existem modelos gráficos que possuem inferência tratável, porém a maioria não consegue representar de forma compacta e geral uma distribuição. A maioria dos PGMs solucionam o problema da intractabilidade computando a inferência aproximada.

Sum-Product Networks são PGMs que, quando completas e consistentes, computam a inferência exata e em tempo tratável. Adicionalmente, SPNs se mostraram mais gerais que outros modelos que computam inferência em tempo tratável.[PD11]

Como aprendizado de uma SPN depende da inferência, a intractabilidade do aprendizado depende da intractabilidade da inferência.

2 Definição

Esta seção faz uma breve descrição de Sum-Product Networks, assim como explica quais são os desafios e problemas relacionados à SPNs.

2.1 Modelos Gráficos Probabilísticos (PGM)

Um modelo probabilístico tem como objetivo representar uma distribuição de probabilidade de forma compacta. PGMs são utilizadas em *data mining*, previsão de eventos, classificação, entre outros.

Para fazer previsões, deduzir eventos ou extrair novo conhecimento, faz-se inferência na distribuição de probabilidade. Inferência pode ser vista como uma probabilidade de um certo evento ocorrer dados os eventos observados:

$$P(X = x_1, \dots, x_n | \mathbf{e} = e_1, \dots, e_q)$$

O conjunto $X = \{x_1, \ldots, x_n\}$ representa a *query*, ou seja, o evento que deve ser inferido. O conjunto $\mathbf{e} = \{e_1, \ldots, e_q\}$ é dito a evidência, as variáveis que já são conhecidas. Probabilidades do tipo $P(a|b)$ são chamadas de probabilidades posteriores.

Para representar uma distribuição de probabilidade, é preciso tomar as probabilidades de cada possível evento. Por exemplo, sejam duas variáveis A e B cujos possíveis valores são Booleanos, e B depende de A, então pode-se montar as seguintes tabelas:

	$P(A)$	$P(B)$
1	$P(A=1) = 0.4 = \theta_a$	$P(B=1) = 0.8 = \theta_b$
0	$P(A=0) = 0.6 = \theta_{\overline{a}}$	$P(B=0) = 0.2 = \theta_{\overline{b}}$

	$A=0$	$A=1$
$B=1$	$P(B=1\|A=0) = 0.3 = \theta_{b\|\overline{a}}$	$P(B=1\|A=1) = 0.4 = \theta_{b\|a}$
$B=0$	$P(B=0\|A=0) = 0.7 = \theta_{\overline{b}\|\overline{a}}$	$P(B=0\|A=1) = 0.6 = \theta_{\overline{b}\|a}$

Pela definição de probabilidade condicional,

$$P(X,Y) = P(X)P(Y|X)$$

chega-se à distribuição de probabilidade das variáveis A e B:

A	B	$P(A,B)$
0	0	$\theta_{\overline{a}}\theta_{\overline{b}\|\overline{a}}$
0	1	$\theta_{\overline{a}}\theta_{b\|\overline{a}}$
1	0	$\theta_a\theta_{\overline{b}\|a}$
1	1	$\theta_a\theta_{a\|b}$

É fácil notar que dado um número n de variáveis, as possíveis probabilidades são n^2. $O(n^2)$ é intratável, e portanto é necessário representar distribuições de forma compacta e computar inferência em tempo tratável.

Redes Bayesianas conseguem representar distribuições de probabilidade, porém inferência no caso geral é intratável, e portanto recorrem a inferência aproximada. Outros modelos, como mixture models e árvores thin junction possuem inferência tratável, porém são limitadas a certas distribuições. Portanto, um modelo cuja inferência seja tratável e que possa representar uma grande variedade de distribuições é desejável.

2.2 Sum-Product Networks

Em 2011, foi proposta[PD11] uma nova classe de PGM chamada Sum-Product Network.

SPNs, quando completas e consistentes, têm inferência exata e tratável. Representam compactamente uma distribuição de probabilidade e se mostraram mais gerais do que outros modelos com inferência tratável.

SPNs representam uma distribuição de probabilidade por meio de uma função polinomial denominada *network polynomial* [Dar09; PD11]. Tomando o exemplo da subseção passada, a network polynomial resultante é dada por:

$$f = \lambda_a\lambda_b\theta_a\theta_{b|a} + \lambda_a\lambda_{\overline{b}}\theta_a\theta_{\overline{b}|a} + \lambda_{\overline{a}}\lambda_b\theta_{\overline{a}}\theta_{b|\overline{a}} + \lambda_{\overline{a}}\lambda_{\overline{b}}\theta_{\overline{a}}\theta_{\overline{b}|\overline{a}}$$

Onde λ_a é o indicador da variável A e toma valores consistentes à evidência dada.

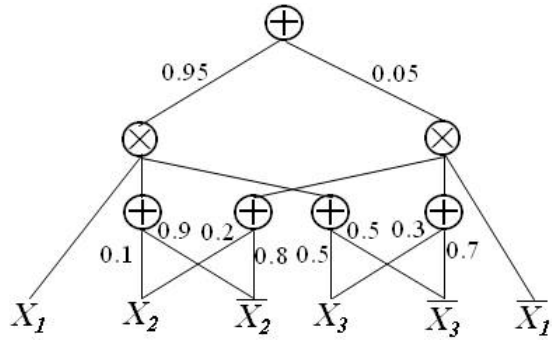

Figura 1: Uma SPN com variáveis Booleanas, onde $x_1, \ldots, x_d$ e $\overline{x}_1, \ldots, \overline{x}_d$ são folhas e o resto dos nós são somas ou produtos.[PD11]

Uma SPN é representada por um dígrafo acíclico onde as folhas são distribuições mono-variáveis, ou seja, representam uma das variáveis; e cada camada de nó interno é alternado entre um nó soma e nó produto. Toda aresta que parte de um nó qualquer para um nó soma possue um peso não negativo.

O valor de uma SPN é o valor do nó raíz. O valor de um nó variável é a distribuição da própria variável. O valor de um nó soma é a soma ponderada dos nós filhos dela. O valor de um nó produto é o produto dos filhos.

Estudos em aprendizado de SPNs mostraram grande potencial. Em [PD11], Poon e Domingos mostraram, por meio de experimentos, que SPNs tiveram melhor desempenho que outros modelos. Nesse trabalho, o aprendizado foi realizado com uma estrutura já existente (e portanto menos flexível), apenas aprendendo os pesos. Em [GD13], Gens e Domingos mostraram que, por meio do aprendizado da estrutura e dos pesos, é possível gerar uma SPN ainda mais geral.

3 Aplicações

SPNs obtiveram resultados impressionantes em muitos conjuntos de dados[Dep], tais como:

- Reconstrução de imagens.
- Classificação.
- Reconhecimento de atividade.
- Logs click-through.
- Sequências de ácido nucleico.
- Filtragem colaborativa.

4 Objetivos

Neste projeto de Iniciação Científica, o aluno irá estudar os seguintes tópicos:

- Propriedades e estrutura de uma Sum-Product Network.
- Inferência em SPNs.
- Aprendizado:
  - Dos pesos de uma SPN.[PD11]
  - Da estrutura de uma SPN.[GD13]

- Por busca gulosa.[DV15]
- Por clustering de variáveis.[DV12]
- Por SPNs bayesianas não-paramétricas.[LWZ14]

5 Cronograma

O aluno deverá reservar 10 horas por semana para estudos relacionados ao projeto. Além disso, o aluno irá escrever relatórios semanais do que foi estudado na semana. Os relatórios estarão disponíveis em:

`http://www.ime.usp.br/~renatolg/spn/doc/reports/`

Tanto os relatórios quanto as implementações estarão disponíveis pelo repositório do projeto:

`https://github.com/RenatoGeh/spn/`

Referências

[Dar09] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. 1st Edition. Cambridge University Press, 2009.

[Dep] University of Washington Department of Computer Science. *Sum-Product Networks*. URL: `http://spn.cs.washington.edu/index.shtml`.

[DV12] 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).

[DV15] Aaron Dennis e Dan Ventura. "Greedy Structure Search for Sum-Product Networks". Em: *International Joint Conference on Artificial Intelligence* 24 (2015).

[GD13] Robert Gens e Pedro Domingos. "Learning the Structure of Sum-Product Networks". Em: *International Conference on Machine Learning* 30 (2013).

[LWZ14] Sang-Woo Lee, Christopher Watkins e Byoung-Tak Zhang. "Non-Parametric Bayesian Sum-Product Networks". Em: *Workshop on Learning Tractable Probabilistic Models* (2014).

[PD11] Hoifung Poon e Pedro Domingos. "Sum-Product Networks: A New Deep Architecture". Em: *Uncertainty in Artificial Intelligence* 27 (2011).