

Uma Introdução a Sum-Product Networks

Relatório semana 9 - MAC0215 (Atividade Curricular em Pesquisa)
Aluno: Renato Lui Geh (Bacharelado em Ciência da Computação)
Orientador: Denis Deratani Mauá

1 ATIVIDADES REALIZADAS NA SEMANA

Durante a semana foram lidos as seguintes partes dos papers abaixo:

- *Learning the Structure of Sum-Product Networks*, [R. Gens, P. Domingos] [5]
 - Introduction
 - Sum-Product Networks
- *Sum-Product Networks: A New Deep Architecture*, [H. Poon, P. Domingos] [6]
 - Introduction
 - Sum-Product Networks
 - Sum-Product Networks and other models

2 DEFINIÇÃO DAS ATIVIDADES

Os tópicos mencionados na seção anterior referem-se à definição de uma Sum-Product Network e citam algumas semelhanças com outros modelos probabilísticos assim como suas diferenças.

Neste relatório vamos definir o que são Sum-Product Networks de uma forma mais didática e vamos supor que o leitor tenha conhecimento prévio de todo conteúdo coberto nos relatórios anteriores. Após termos definido Sum-Product Networks, vamos ver algumas propriedades e teoremas relacionados e em seguida vamos comparar, de forma sucinta, com outros modelos probabilísticos.

Vamos separar esta seção nos seguintes tópicos:

1. Introdução
 - 1.1. Distribuição normalizada de produtos de factors
 - 1.2. Função de partição
2. Definição
3. Propriedades
4. Comparação

2.1 INTRODUÇÃO

Um dos maiores problemas com modelos gráficos é a intratabilidade da inferência e aprendizado da estrutura. Inferência é sempre exponencial no pior caso, e como aprendizado usa inferência, a complexidade continua intratável. Além do mais, a amostragem necessária para aprendizado preciso é também exponencial no pior dos casos no tamanho do escopo. De fato existem modelos gráficos onde a inferência é tratável, no entanto elas são limitadas quanto às representações de distribuições de forma compacta.

Vamos mostrar que Sum-Product Networks (SPN), um novo tipo de arquitetura profunda, permite que computemos a função partição, a probabilidade de evidência e o estado MAP[2] com complexidade linear no número de arestas da SPN. Também vamos definir validade de uma SPN assim como completude e consistência. Depois vamos mostrar outras definições assim como alguns teoremas derivados dessas propriedades.

Antes de começarmos a definir Sum-Product Networks, precisamos antes explicar o que é uma distribuição normalizada de produtos de factors e definir uma função de partição.

2.1.1 Distribuição normalizada de produtos de factors

O objetivo de modelos gráficos probabilísticos é representar distribuições de forma compacta. Podemos representar tais distribuições como um produto normalizado dos factors[3] envolvidos. Tal representação é um jeito compacto de se representar as CPTs envolvidas.

Definição. *Sejam $x \in \mathcal{X}$ um vetor d -dimensional representando uma instância de d variáveis, ϕ_k uma função potencial[3] do subconjunto $x_{\{k\}}$ de variáveis (ou seja, seu escopo[1]) e Z a função partição que veremos mais a frente. Representamos distribuições compactamente como o seguinte produto normalizado:*

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}}) \quad (1)$$

A representação acima é dita normalizada pois queremos representa-la como uma probabilidade, ou seja, um número real no intervalo $[0, 1]$. Como pode-se notar, dividimos o produtório por Z , a chamada função partição. De fato, como veremos a seguir, a função partição normaliza o produto dos factors.

2.1.2 Função de partição

Dizemos a função partição uma função que toma como argumentos todos os estados das variáveis e retorna a soma de todos os produtórios de todos os factors de cada estado.

Definição. *Seja ϕ_k uma função potencial, dizemos que a função partição é*

$$Z = \sum_{x \in \mathcal{X}} \prod_k \phi_k(x_{\{k\}}) \quad (2)$$

Portanto, é fácil notar que $\frac{1}{Z} \prod_k \phi_k(x_{\{k\}})$ é uma normalização por Z , já que Z é a soma de todos os possíveis resultados do produtório, e portanto será sempre maior ou igual ao valor do produtório normalizado, levando a $0 \leq P(X = x) \leq 1$, assumindo-se que $\phi_i \geq 0$.

No caso de $Z = 1$, então temos o caso trivial onde o produtório dos factors dada uma instância já está dentro do intervalo $[0, 1]$.

Uma das dificuldades de se computar inferência em modelos gráficos é a intractabilidade de Z , já que Z é a soma de um número exponencial de termos. Como todas as marginals[4] são somas de subconjuntos desses termos, computa-las é igualmente intratável. No entanto, se acharmos uma maneira eficiente de computar Z , então também podemos computar as marginals eficientemente. Mas Z é computado apenas com somas e produtos, e pode ser eficientemente computado se aplicarmos a distributiva em Z de tal forma que envolvamos um número polinomial de somas e produtos.

2.2 DEFINIÇÃO

Assim como em [P. Domingos, H. Poon][6], vamos introduzir Sum-Product Networks com variáveis Booleanas. Mais para frente veremos que para variáveis discretas ou contínuas o processo é similar.

Antes de definirmos SPNs, vamos introduzir algumas notações:

- A negação de X_i é representada por \bar{X}_i .
- A função indicadora[2] $[.]$ tem valor 1 se seu argumento é *true* e 0 caso contrário.
- Abreviaremos $[X_i]$ por x_i e $[\bar{X}_i]$ por \bar{x}_i .

Seja $\Phi(x) \geq 0$ uma distribuição de probabilidade não-normalizada. A network polynomial [2] de $\Phi(x)$ é $\sum_x \Phi(x)\Pi(x)$, onde $\Pi(x)$ é o produto dos indicadores que tenham valor 1 no estado x . Lembrando o que vimos nos relatórios anteriores, a network polynomial da Rede Bayesiana $X_1 \rightarrow X_2$ é $P(x_1)P(x_2|x_1)x_1x_2 + P(x_1)P(\bar{x}_2|x_1)x_1\bar{x}_2 + P(\bar{x}_1)P(x_2|\bar{x}_1)\bar{x}_1x_2 + P(\bar{x}_1)P(\bar{x}_2|\bar{x}_1)\bar{x}_1\bar{x}_2$.

A função partição é o valor da network polynomial quando todos os indicadores são 1. Para qualquer evidência e , computar $P(e) = \Phi(e)/Z$ é linear no tamanho da network polynomial, que por sua vez tem tamanho exponencial em número de variáveis. No entanto, podemos representar e avaliar a network polynomial em tempo e espaço polinomial usando Sum-Product Networks.

Vamos a seguir ver a definição de SPNs dada por P. Domingos e H. Poon[6].

Definição. Uma Sum-Product Network (SPN) sob 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 somas e produtos. Cada aresta (i, j) com origem em um nó soma i tem um peso não-negativo w_{ij} . O valor de um nó produto é o produto dos valores de seus filhos. O valor de um nó soma é $\sum_{j \in Ch(i)} w_{ij}v_j$, onde $Ch(i)$ são os filhos de i e v_j é o valor do nó j . O valor da SPN é o valor de sua raiz.

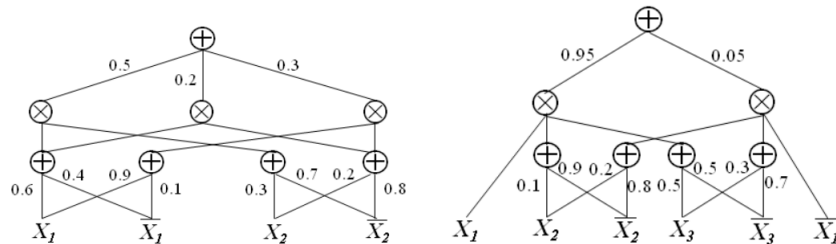


Figura 1: A esquerda uma SPN implementando uma naive Bayes mixture model. A direita uma SPN implementando uma junction tree. [P. Domingos, H. Poon][6]

Vamos assumir sem perda de generalidade que nós somas e nós produtos estão organizados de tal forma que somas são sempre alternadas com produtos e vice-versa. Ou seja, se i é um nó soma, então $\forall x \in Ch(i)$, x é ou um nó produto ou uma folha (e portanto uma variável). Analogamente, se i é um nó produto, então $\forall x \in Ch(i)$, x é ou um nó soma ou folha.

Denotamos a Sum-Product Network S como uma função das variáveis indicadoras x_1, \dots, x_d e $\bar{x}_1, \dots, \bar{x}_d$ como $S(x_1, \dots, x_d, \bar{x}_1, \dots, \bar{x}_d)$. Um estado x é completo quando para cada variável X_i , seus indicadores nunca são iguais, ou seja, $x_i = 1$ e $\bar{x}_i = 0$ ou $x_i = 0$ e $\bar{x}_i = 1$. Se queremos S em função de um estado x completo, então representamos como $S(x)$. Se os indicadores especificam uma evidência[1] e , então abreviamos como $S(e)$. No caso de todos os indicadores serem valorados em 1, então dizemos $S(*)$.

A subrede enraizada em um nó n em uma SPN é uma SPN, e é representada como $S_n(\cdot)$. Os valores de $S(x), \forall x \in \mathcal{X}$ define uma distribuição de probabilidade não-normalizada sob \mathcal{X} . A probabilidade não-normalizada da evidência e é $\Phi_S(e) = \sum_{x \in e} S(x)$, onde a soma tem seus estados consistentes com e . A função partição da distribuição definida por $S(x)$ é $Z_S = \sum_{x \in \mathcal{X}} S(x)$. O escopo[1] de uma SPN S é o conjunto de variáveis que aparecem em S . Uma variável X_i é negada em S se \bar{x}_i é uma folha de S e não-negada se x_i é uma folha de S .

A partir disso podemos construir a definição encontrada em [P. Domingos, R. Gens] [5].

Definição. Definimos recursivamente uma Sum-Product Network (SPN):

1. Uma distribuição monovariável tratável é uma SPN.
2. Um produto de SPNs com escopos disjuntos é uma SPN.
3. Uma soma ponderada de SPNs com mesmo escopo é uma SPN se todos os pesos são não-negativos.
4. Nada mais é uma SPN.

Exemplificando com a Figura 1, a SPN S será:

$$\begin{aligned} S(x_1, x_2, \bar{x}_1, \bar{x}_2) = & 0.5(0.6x_1 + 0.4\bar{x}_1)(0.3x_2 + 0.7\bar{x}_2) + \\ & + 0.2(0.6x_1 + 0.4\bar{x}_1)(0.2x_2 + 0.8\bar{x}_2) + \\ & + 0.3(0.9x_1 + 0.1\bar{x}_1)(0.2x_2 + 0.8\bar{x}_2) \end{aligned} \quad (3)$$

E a network polynomial será a expansão de S , ou seja, $(0.5 \times 0.6 \times 0.3 + 0.2 \times 0.6 \times 0.2 + 0.3 \times 0.9 \times 0.2)x_1x_2 + \dots$. Se um estado completo $x = \{X_1 = 1, X_2 = 0\}$, então $S(x) = S(1, 0, 0, 1)$. Se a evidência $e = X_1 = 1$, então $S(e) = S(1, 1, 0, 1)$. E, por fim, $S(*) = S(1, 1, 1, 1)$.

REFERÊNCIAS

- [1] Renato Lui Geh. *Aprendizado Automático de Sum-Product Networks (SPN)*. 2015. URL: <http://www.ime.usp.br/~renatolg/mac0215/doc/project/relatorio.pdf>.
- [2] Renato Lui Geh. *Modeling and Reasoning with Bayesian Networks: Compiling Bayesian Networks*. 1. 2015. URL: <http://www.ime.usp.br/~renatolg/mac0215/doc/reports/week1/relatorio.pdf>.
- [3] Renato Lui Geh. *Modeling and Reasoning with Bayesian Networks: Inference by Variable Elimination 6.1-6.5*. 2. 2015. URL: <http://www.ime.usp.br/~renatolg/mac0215/doc/reports/week2/relatorio.pdf>.
- [4] Renato Lui Geh. *Modeling and Reasoning with Bayesian Networks: Inference by Variable Elimination 6.6-6.9*. 3. 2015. URL: <http://www.ime.usp.br/~renatolg/mac0215/doc/reports/week5/relatorio.pdf>.
- [5] Robert Gens e Pedro Domingos. “Learning the Structure of Sum-Product Networks”. Em: *International Conference on Machine Learning* 30 (2013).
- [6] Hoifung Poon e Pedro Domingos. “Sum-Product Networks: A New Deep Architecture”. Em: *Uncertainty in Artificial Intelligence* 27 (2011).