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. O conteúdo presente nesse relatório é baseado fortemente nos dois papers acima.

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 intractabilidade 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 1. *Sejam $x \in \mathcal{X}$ um vetor d-dimensional representando uma instância de d variáveis, ϕ_k uma função potential[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:*

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

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 2. *Seja ϕ_k uma função potential, dizemos que a função partição é*

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

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

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 $P(x_1)P(x_2|x_1)x_1x_2+P(x_1)P(\overline{x}_2|x_1)x_1\overline{x}_2+P(\overline{x}_1)P(x_2|\overline{x}_1)\overline{x}_1x_2+P(\overline{x}_1)P(\overline{x}_2|\overline{x}_1)\overline{x}_1\overline{x}_2$.

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 polynomial usando Sum-Product Networks.

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

Definição 3. *Uma Sum-Product Network (SPN) sob variáveis $x_1, ..., x_d$ é um grafo enraizado, direcionado e acíclico (DAG) cujas folhas são indicadores $x_1, ..., x_d$ e $\overline{x}_1, ..., \overline{x}_d$ 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 raíz.*

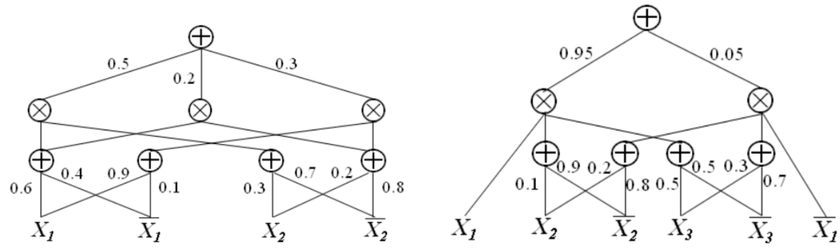

Figura 1: A esquerda uma SPN implementando uma naive Bayes mixture model. A direita uma SPN implementando uma junction tree. [P. Domingos, H. Poon][6]

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 $x_1, ..., x_d$ e $\overline{x}_1, ..., \overline{x}_d$ como $S(x_1, ..., x_d, \overline{x}_1, ..., \overline{x}_d)$. Um estado x é completo quando para cada variável X_i, seus indicadores nunca são iguais, ou seja, $x_i = 1$ e $\overline{x}_i = 0$ ou $x_i = 0$ e $\overline{x}_i = 1$. 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 $S_n(.)$. 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[2] 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 $\overline{x}_i$ 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 4. *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:

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

E a network polynomial será a expansão de S, ou seja, $(0.5 \times 0.6 \times 0.3 + 0.2 \times 0.6 \times 0.2 + 0.3 \times 0.9 \times 0.2)x_1 x_2 + ...$ 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, $S(*) = S(1, 1, 1, 1)$.

2.3 Propriedades

Nesta subseção, vamos definir a validade de uma SPN, assim como completude e consistência. Em seguida vamos provar que uma SPN é válida se é completa e consistente. Depois vamos definir representabilidade, complexidade da função partição, decomponibilidade de uma SPN e finalmente provaremos que a função partição, probabilidade de evidência e o estado MAP[2] de uma SPN podem ser computados em tempo linear no número de arestas da SPN.

Definição 5. *Uma Sum-Product Network S é válida sse $S(e) = \Phi_S(e)$ para toda evidência e.*

A Definição 5 diz que a SPN sempre computa a probabilidade de evidência corretamente, então a SPN é válida. Em particular, se uma SPN S é válida, então $S(*) = Z_S$. Uma SPN válida computa a probabilidade de evidência em tempo linear no número de arestas, como provaremos mais a frente.

Já que uma SPN válida é linear, é preferível aprendermos apenas SPNs válidas. Vamos ver algumas outras condições para a validade de uma SPN, mas para isso precisamos definir completude e consistência.

Definição 6. *Uma Sum-Product Network é* completa *sse todos os filhos do mesmo nó soma tem mesmo escopo.*

Definição 7. *Uma Sum-Product Network é* consistente *sse nenhuma variável aparece negada em um filho de um nó produto e não-negada em outro.*

A partir disso podemos construir o seguinte teorema de P. Domingos e H. Poon[6].

Teorema 1. *Uma Sum-Product Network é válida se ela é completa e consistente.*

Prova.

Toda SPN S pode ser representada por uma network polynomial $\sum_k s_k \prod_k(...)$, onde $\prod_k(...)$ é um monômio das variáveis indicadores e $s_k \geq 0$ é seu coeficiente (a soma dos produtos dos parâmetros que tenham os respectivos indicadores). Chamamos essa representação de *expansão* da SPN, já que ela é obtida aplicando a distributiva de baixo-para-cima em todos os nós produtos da SPN. Ao aplicarmos a distributiva, tratamos cada folha x_i como a soma expandida $1x_i + 0\overline{x}_i$ e $\overline{x}_i$ como $0x_i + 1\overline{x}_i$.

Uma SPN é valida se sua expansão é a própria network polynomial, ou seja, se os monômios da expansão e os estados x são correspondentes. Isso quer dizer que:

1. Cada monômio é não-zero em exatamente um estado.
2. Cada estado tem exatamente um monômio não-zero.

Pela condição 2, $S(x)$ é igual ao coeficiente s_x do monômio não-zero e portanto $\Phi_s(e) = \sum_{x \in e} S(x) = \sum_{x \in e} s_x = \sum_k s_k n_k(e)$, onde $n_k(e)$ é o número de estados x consistentes com e que tenham $\sum_k(x) = 1$.

Pela condição 1, $n_k = 1$ se o estado x que tenha $\prod_k(x) = 1$ é consistente com a evidência. Senão $n_k = 0$ e portanto $\Phi_s(e) = \sum_{k:\prod_k(e)=1} s_k = S(e)$ e a SPN é válida.

Por indução, podemos provar, começando pelas folhas até a raíz, que, se a SPN é completa e consistente, então sua expansão é sua network polynomial.

O caso folha é trivialmente verdade, pois se o nó é folha, então ele tem apenas uma variável em seu escopo e portanto é completo. Pela mesma hipótese, não se pode ter uma variável negada e não-negada já que existe apenas um indicador, tornando-a consistente. Portanto, o caso folha é completo e consistente e, se computarmos sua SPN $S(l) = v_l x_l$, podemos ver que S é igual a sua network polynomial.

Consideremos agora apenas os nós internos. Assumimos apenas nós internos com dois filhos, mas a extensão para o caso geral é imediata. Seja n^0 um nó interno arbitrário com filhos n^1 e n^2. Teremos as seguintes notações:

- V^i o escopo do nó n^i,
- x^i o estado do escopo V^i,
- S^i a expansão do subgrafo enraizado em n^i,
- $\Phi^i(x^i)$ a probabilidade não-normalizada de x_i sob S_i.

Pela hipótese de indução, $S^1 = \sum_{x^1} \Phi^1(x^1) \prod(x^1)$ e $S^2 = \sum_{x^2} \Phi^2(x^2) \prod(x^2)$.

Se n^0 é um nó soma, então $S^0 = w_{01} \sum_{x^1} \Phi^1(x^1) \prod(x^1) + w_{02} \sum_{x^2} \Phi^2(x^2) \prod(x^2)$. Se $V^1 \neq V^2$, então cada estado em V^1 (ou V^2) corresponde a múltiplos estados de $V^0 = V^1 \cup V^2$, e portanto cada monômio de V^1 (e V^2) é não-zero em mais de um estado de V^0, quebrando a correspondência entre monômios de S^0 e estados de V^0. Mas se a SPN é completa, então $V^0 = V^1 = V^2$ e seus estados são correspondentes. Portanto, pela indução de hipótese os monômios de V^1 e V^2 são também correspondentes e $S^0 = \sum_{x^0} (w_{01}\Phi^1(x^0) + w_{02}\Phi^2(x^0)) \prod(x^0)$, ou seja, a expansão de S^0 é sua network polynomial.

Se n^0 é um nó produto, então $S^0 = (\sum_{x^1} \Phi^1(x^1) \prod(x^1))(\sum_{x^2} \Phi^2(x^2) \prod(x^2))$. Se $V^1 \cap V^2 = \emptyset$, então segue-se imediatamente que a expansão de V^0 é a network polynomial. No caso mais geral, sejam $V^{12} = V^1 \cup V^2$, $V^{1--} = V^1 \setminus V^2$ e $V^{2--} = V^2 \setminus V^1$, e sejam x^{12}, x^{1--} e x^{2--} seus estados correspondentes. Mas já que $\Phi^1(x^1)$ é não-zero em exatamente um estado x^1 e similarmente para $\Phi^2(x^2)$, então cada monômio no produto de S^1 e S^2 é não-zero no máximo em um único estado de $V^0 = V^1 \cup V^2$. Se a SPN não é consistente, então pelo menos um monômio no produto contém indicadores tanto positivos quanto negativos, x_i e $\overline{x}_i$, de uma variável. Como nenhum monômio na network polynomial contém tanto x_i e $\overline{x}_i$, então a expansão de S^0 não é igual a sua network polynomial. Para garantirmos que cada monômio em S^0 seja não-zero em pelo menos um estado de V^0 então, para cada par $(\prod(x^{1--}, x^{12}), \prod(x^{12}, x^{2--}))$, deve existir um estado $x^0 = (x^{1--}, x^{12}, x^{2--})$ onde ambos $\prod(x^{1--}, x^{12})$ e $\prod(x^{12}, x^{2--})$ são iguais a 1, e portanto os indicadores sobre $x^1 2$ em ambos os monômios devem ser consistentes. Como, pela hipótese de indução, eles especificam completamente x^{12}, então eles devem ser os mesmos nos dois monômios. Portanto todos os monômios $(\prod(x^{1--}, x^{12}), \prod(x^{12}, x^{2--}))$ devem ter mesmo x^{12} indicadores, ou seja, a SPN deve ser consistente. □

Completude e consistência não são necessárias para validade de uma SPN. Por exemplo, a rede $S(x_1, x_2, \overline{x}_2, \overline{x}_2) = \frac{1}{2}x_1\overline{x}_2 + \frac{1}{2}x_1$ é incompleta e inconsistênte, mas satisfaz $\Phi_S(e) = \sum_{x \in e} S(x)$ para toda evidência e, e pela definição de validade S é válida. No entanto, completude e consistência são necessárias para afirmar que toda subrede de S é válida. Tal hipótese pode ser provada por contradição. Seja S um nó que viola ou a completude ou a consistência da rede. Pode-se mostrar que S não é válida, já que ela ou subestima o valor da soma se é incompleta ou superestima se ela é inconsistente.

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).