ESTUDO SOBRE SUM-PRODUCT NETWORKS E APRENDIZAGEM PROFUNDA

PTC2669 - INTRODUÇÃO A INTELIGÊNCIA COMPUTACIONAL
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA - USP

RELATÓRIO 1

RENATO LUI GEH
NUSP: 8536030

Resumo. Na área de Indecisão Probabilística, buscamos representar conhecimento por meio de distribuições de probabilidade. Modelos probabilísticos baseados em grafos são usados para realizar modelagem e raciocínio de forma compacta e tratável, já que distribuições de probabilidade têm tamanho exponencial no número de variáveis. No entanto, efetuar inferência exata na maioria destes modelos é uma tarefa NP-difícil [Coo88], e portanto dependemos de técnicas de inferência aproximada, o que resulta em aprendizado aproximado, já que aprendizado utiliza inferência como subrotina. Apesar de existirem modelos onde inferência exata e tratável é possível, as distribuições que estes modelos conseguem representar é limitada.

Sum-Product Networks (SPNs) são modelos em que inferência é exata e tratável. Além disso, SPNs são mais gerais que outros modelos exatos no sentido que SPNs conseguem representar mais distribuições. SPNs diferenciam-se de outros modelos probabilísticos por não terem uma sintaxe explícita quanto as probabilidades como Redes Bayesianas ou Redes de Markov. Sua estrutura assemelha-se a de uma rede neural, e portanto a derivação do algoritmo de *backpropagation* é natural e direto. Além disso, sua arquitetura é natural a implementação de camadas ocultas, e portanto a *deep learning*.

Neste relatório, vamos enumerar os problemas encontrados com os chamados modelos probabilísticos baseados em grafo clássicos como motivação a elaboração de SPNs. Em seguida vamos mostrar de forma superficial a estrutura de uma SPN e fazer comparações com redes neurais artificiais.

1. Introdução

Considere uma distribuição de probabilidade conjunta $\Pr(\mathbf{X} = \{X_1, \ldots, X_n\})$. Se enumerarmos cada probabilidade de cada instanciação de $\mathbf{X}$, teremos uma tabela da forma

Se o número de variáveis de uma distribuição é n, e $p = \max |Val(X_i)|$ para um i que maximiza p, onde $Val(X)$ é o domínio da variável aleatória X_i (ou seja, o conjunto de possíveis valores que X pode tomar), então o número de termos na distribuição é $\mathcal{O}(p^n)$.

$\mathbf{x}^{(i)}$	X_1	$\cdots$	X_n	$\Pr(x_1, \ldots, x_n)$
$\mathbf{x}^{(1)}$	$x_1^{(1)}$	$\cdots$	$x_n^{(1)}$	$\Pr(\mathbf{x}^{(1)})$
$\mathbf{x}^{(2)}$	$x_1^{(2)}$	$\cdots$	$x_n^{(2)}$	$\Pr(\mathbf{x}^{(2)})$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
$\mathbf{x}^{(n)}$	$x_1^{(n)}$	$\cdots$	$x_n^{(n)}$	$\Pr(\mathbf{x}^{(n)})$

TABELA 1 O número de termos de uma distribuição de probabilidade é exponencial.

Exemplo 1.0.1. *Considere um dado não-viesado de seis faces. O conjunto de variáveis aleatórias* $\mathbf{X}$ *é*

X *se o número da face é par*

Y *se o número da face é múltiplo de três*

Neste caso temos que $|Val(X)| = |Val(Y)| = 2$*, já que podemos ter dois possíveis resultados: 1 se verdadeiro e 0 se falso. Queremos saber a distribuição de probabilidade conjunta* $\Pr(X, Y)$ *Para representarmos esta distribuição, precisaríamos de*

$\mathbf{x}$	X	Y	$\Pr(x, y)$
$\{x = 1, y = 1\}$	1	1	1/6
$\{x = 1, y = 0\}$	1	0	1/2
$\{x = 0, y = 1\}$	0	1	2/6
$\{x = 0, y = 0\}$	0	0	1/6

TABELA 2 O número de termos desta distribuição é 2^2.

um tamanho exponencial na memória. Além disso, para acharmos as probabilidades conjuntas (ou condicionais) precisaríamos de tempo exponencial no número de variáveis.

É fácil ver que precisamos um jeito mais compacto de se representar distribuições de probabilidade. Para isso utilizamos modelos probabilísticos baseados em grafo.

2. Modelos Probabilísticos Baseados em Grafo

Um Modelo Probabilístico baseado em Grafo (PGM) é um grafo que busca representar uma distribuição de probabilidade de forma compacta, para que possamos computar marginais e probabilidades a posteriori de forma de forma eficiente e aprender os parâmetros e estrutura de forma precisa.

Vamos adotar a nomenclatura dada em [Peh15] e referirmos aos PGMs clássicos (CPGMs) como o conjunto de modelos como Redes Bayesianas e Redes de Markov, ou seja, modelos em que a estrutura é graficamente explícita quanto as distribuições de probabilidade.

Nesta seção vamos definir de forma sucinta Redes Bayesianas e mostrar o porquê da necessidade de um modelo como SPNs.

2.1. *Redes Bayesianas*

Definição 2.1.1. *Uma Rede Bayesiana $\mathcal{N}$ é uma tupla $\mathcal{N} = (\Omega, \mathcal{F}, \Pr, G)$, onde Ω é o espaço de possibilidades, $\mathcal{F}$ é uma álgebra sobre Ω, $\Pr$ é uma função de probabilidade e $G = (\mathbf{X}, E)$ é um grafo onde $\mathbf{X}$ é o conjunto de variáveis de $\mathcal{N}$ e E é o conjunto de arestas. Cada variável aleatória $X_i \in \mathbf{X}$ representa uma probabilidade condicional $\Pr(X_i|Pa(X_i))$. Uma Rede Bayesiana é uma representação para a distribuição de probabilidade conjunta*

$$\Pr(\mathbf{X} = \{X_1, \ldots, X_n\}) = \prod_{X \in \mathbf{X}} \Pr(X|Pa(X)). \tag{2.1.1}$$

A equação 2.1.1 é o resultado do Teorema da Fatorização. Considere a seguinte Rede Bayesiana como exemplo, tirado de [Dar09].

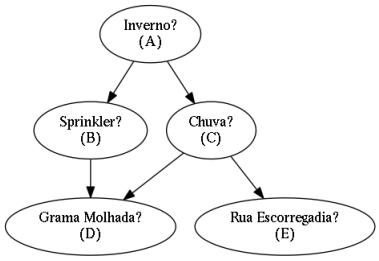

FIGURA 1. Uma Rede Bayesiana com cinco variáveis booleanas.

A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|B,C}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

TABELA 3 As CPTs da Rede Bayesiana da Figura 1.

As tabelas da Rede Bayesiana mostram as probabilidades de cada evento dados os eventos de dependência. Tais tabelas são chamadas de Tabelas de Probabilidade Condicional (CPTs) por serem as probabilidades condicionais de cada nó. Denotaremos por $\Theta_{X|Pa(X)}$ a CPT do nó X e por $\theta_{x\sim X|v\sim Pa(X)}$ um parâmetro, ou seja, uma instanciação de X dada uma instanciação de $Pa(X)$. Por exemplo, $\theta_{b|\overline{a}} = .8$ indica que a instanciação $b = true$ e $a = false$ da CPT $\Theta_{B|A}$ é a probabilidade 0.8.

Exemplo 2.1.1. *Considere que tenhamos a Rede Bayesiana da Figura 1 e que desejamos extrair conhecimento a partir do nosso modelo. Por exemplo, se soubermos que a rua da casa em que vivemos está molhada, teremos mais cuidado ao dirigirmos nela. Ou que se a grama de nosso jardim estiver molhada, não a cortaremos nesse dia. Para isso devemos considerar alguns outros eventos que têm direta relação com as predições que desejamos inferir. Afinal, se soubermos que hoje choveu, temos certeza que a grama ou rua estará molhada. No entanto, mesmo que hoje não tenha chovido, é possível que o sistema de sprinklers que instalamos tenha acionado, e portanto a grama estará molhada mas a rua não. Além disso, se estivermos no inverno, teremos uma maior chance de chuva e provavelmente acionaremos os sprinklers menos vezes devido a isso. Para modelarmos todas estas interações, podemos usar uma Rede Bayesiana. Note que há eventos que provavelmente impactariam em nosso sistema, porém decidimos não toma-los em conta para simplificarmos o nosso exemplo. Note que existe uma aresta entre um par de nó na Rede Bayesiana se e somente se existe uma relação direta entre os eventos. Também note que todo ancestral de um nó tem uma relação (mesmo que indireta) com o nó em questão.*

2.2. *Inferência Exata em Redes Bayesianas*

Acharmos a probabilidade de um certo evento é possível por meio do Teorema da Fatorização, que apesar de não termos enunciado, tem como resultado a Equação 2.1.1. O Teorema da Fatorização tem uma relação de se e somente se com a Propriedade Local de Markov (LMP), que define independências com todas as variáveis não vizinhas dadas as variáveis vizinhas. Podemos então facilmente derivar 2.1.1 por meio da LMP e da Regra da Cadeia de Teoria de Probabilidade. Vamos relembrar a Equação 2.1.1:

$$\Pr(\mathbf{X}) = \prod_{X \in \mathbf{X}} \Pr(X|Pa(X))$$

Aplicando esta equação à Rede Bayesiana da Figura 1, temos

$$\Pr(A, B, C, D, E) = \Pr(A)\Pr(B|A)\Pr(C|A)\Pr(D|B, C)\Pr(E|C)$$

A partir da definição de probabilidade condicional, podemos achar qualquer probabilidade de algum conjunto de eventos $\mathbf{Y}$ dado uma evidência $\mathbf{E}$ dividindo as probabilidades conjuntas

$$\Pr(\mathbf{Y}|\mathbf{E}) = \frac{\Pr(\mathbf{Y}, \mathbf{E})}{\Pr(\mathbf{E})}$$

Para acharmos as probabilidades conjuntas $\Pr(\mathbf{Y}, \mathbf{E})$ e $\Pr(\mathbf{E})$, precisamos aplicar o Teorema da Fatorização para os eventos relevantes e somar todas as probabilidades que não temos uma valoração (i.e. instanciação) exata. Suponha que $\mathbf{Y} \subseteq \mathbf{X}$ e que $\mathbf{Y} \cap \mathbf{E} = \emptyset$ e $\mathbf{Y} \cup \mathbf{E} \subset \mathbf{X}$. Temos que

$$\Pr(\mathbf{Y}, \mathbf{E}) = \sum_{v \in \mathbf{X} \setminus (\mathbf{Y} \cup \mathbf{E})} \Pr(v, y, e)$$
$$\Pr(\mathbf{E}) = \sum_{v \in \mathbf{X} \setminus \mathbf{E}} \Pr(v, e)$$

onde e e y são as instanciações de $\mathbf{Y}$ e $\mathbf{E}$ respectivamente. Note que como v, y, e formam uma instanciação completa (em que todas as variáveis estão presentes) podemos aplicar o Teorema da Fatorização e achar a probabilidade correspondente. É fácil ver que estas operações de soma e produto são $\mathcal{O}(\exp(n))$. De fato, computar a probabilidade exata de uma Rede Bayesiana no caso geral é NP-difícil. Não só isso como a redução do problema de se computar a probabilidade exata de uma RB é análoga ao do problema de NP-completude de SAT. Se conseguirmos provar que o problema de se resolver uma sentença booleana é reduzível a um problema com solução eficiente, teremos uma prova equivalente a de P vs NP [KBG10]. Por exemplo, computar a seguinte sentença proposicional tal que o resultado seja verdadeiro para uma instanciação das variáveis da sentença é, acredita-se, impossível em tempo subexponencial.

$$A \vee (B \wedge (C \vee (D \wedge E \vee F)))$$

Portanto, crê-se que o problema de inferência exata em RBs é insolúvel em tempo eficiente.

Exemplo 2.2.1. *Retomando o Exemplo 2.1.1, suponha que desejamos saber a probabilidade de a grama estar molhada dado que sabemos que hoje choveu e que não estamos no inverno.*

$$\Pr(D = true|C = true, A = false) = \frac{\Pr(D = true, C = true, A = false)}{\Pr(C = true, A = false)}$$

Usaremos a notação $X = true \equiv X = 1$ e $X = false \equiv X = 0$. Para cada probabilidade conjunta na equação anterior temos

$$\begin{aligned}\Pr(A=0,C=1,D=1) &= \sum_{b,e} \Pr(A=0,b,C=1,D=1,e) = \\ &= \Pr(A=0,B=0,C=1,D=1,E=0)+ \\ &+\cdots+ \\ &+\Pr(A=0,B=1,C=1,D=1,E=1)\end{aligned}$$

$$\begin{aligned}\Pr(C=1,A=0) &= \sum_{a,b,d,e} \Pr(A=0,b,C=1,d,e) = \\ &= \Pr(A=0,B=0,C=1,D=0,E=0)+ \\ &+\cdots+ \\ &+\Pr(A=0,B=1,C=1,D=1,E=1)\end{aligned}$$

A probabilidade resultante da divisão das probabilidades conjuntas é o resultado que desejamos saber.

2.3. *Inferência aproximada em Redes Bayesianas*

Inferência aproximada pode ser feita de diversos modos. No entanto, por sua própria definição, o resultado da probabilidade possue um erro que tende a aumentar a cada iteração. Não somente isso, mas como aprendizado utiliza inferência, o aprendizado torna-se também aproximado.

Existem muitos métodos para inferência aproximada.

(1) Amostragem estocástica:
   (a) Lógica,
   (b) Por importância de verossimilhança,
   (c) Amostragem de Gibbs;
(2) Propagação de crença;
(3) Algoritmo soma-produto;
(4) entre outros.

Não discutiremos os algoritmos neste relatório.

3. Sum-Product Networks

Sabemos que inferência exata em modelos probabilísticos baseados em grafo clássicos é NP-difícil e portanto intratável no caso geral. Nossa motivação é formularmos um modelo onde inferência exata seja eficiente. Existem de fato modelos em que computar a probabilidade conjunta ou condicional exata é sub-exponencial, contudo tais modelos se mostraram muito limitados quanto a sua generalidade ao representar distribuições.

Inferência e aprendizado em Redes Neurais (RNs) são eficientes, no sentido que são subexponenciais, quando realizamos o algoritmo de retropropagação em suas estruturas. Seja pelo gradiente da função ou por EM (expectation-maximization), a complexidade da retropropagação é linear no tamanho do conjunto de arestas da rede.

Sum-Product Networks (SPNs) são PGMs pois representam uma distribuição de probabilidade. Ao mesmo tempo, SPNs podem ser vistos como uma RN devido a sua estrutura profunda com camadas ocultas e operações de soma e produto nos nós. Além disso, inferência é linear no número de arestas da SPN e portanto aprendizado é tratável. Além disso, se uma SPN obedecer a uma certa propriedade, inferência por retropropagação será sempre exata e eficiente. Assim como RNs, o aprendizado profundo de sua estrutura melhora os resultados obtidos, tornando SPNs um modelo atraente para a área de Inteligência Artificial e principalmente para Incerteza e aprendizado probabilístico.

Antes de discutirmos sobre SPNs, vamos primeiro esclarecer alguns assuntos importantes para o entendimento de SPNs. Iremos tratar do chamado *network polynomial*, um polinômio em função das variáveis de uma distribuição de probabilidade que representa a função que desejamos representar pela SPN. Em seguida trataremos da descrição da estrutura de uma SPN e em seguida daremos uma rápida explicação de como se computar inferência exata em sua estrutura pelo algoritmo de retropropagação. Finalmente, discutiremos brevemente sobre aprendizagem.

3.1. *Network polynomial*

Sabemos que uma Rede Bayesiana pode ser representada por CPTs, e que cada CPT é uma variável da rede. Considere a rede da Figura 2 como exemplo.

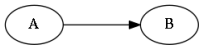

FIGURA 2. Uma simples Rede Bayesiana $A \to B$.

A	Θ_A
true	.7
false	.3

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.6
false	false	.4

Definição 3.1.1. *O polinômio da rede (*network polynomial*) é a função da soma de todas as instanciações da distribuição conjunta de uma Rede Bayesiana multiplicadas com as variáveis indicadoras de cada variável.*

$$f(\mathbf{X}) = \sum_{\mathbf{x} \sim \mathbf{X}} \lambda_{\mathbf{x}} \theta_{\mathbf{x}|v \sim Pa(\mathbf{x})}$$

Uma variável indicadora é 1 *se a variável é consistente com a instanciação e 0 caso contrário. Caso a variável não seja instanciada, a variável indicadora é* 1.

Exemplo 3.1.1. *Denotaremos a se* $A = true$ *e* $\overline{a}$ *se* $A = false$*. O polinômio da rede da Figura 2 é:*

$$f(A,B) = \lambda_a\lambda_b\theta_a\theta_{b|a} + \lambda_{\overline{a}}\lambda_b\theta_{\overline{a}}\theta_{b|\overline{a}} + \lambda_a\lambda_{\overline{b}}\theta_a\theta_{\overline{b}|a} + \lambda_{\overline{a}}\lambda_{\overline{b}}\theta_{\overline{a}}\theta_{\overline{b}|\overline{a}}$$

Se tivermos a e $\overline{b}$*, as variáveis indicadoras são* $\lambda_a = \lambda_{\overline{b}} = 1$ *e* $\lambda_b = \lambda_{\overline{a}} = 0$ *e a probabilidade* $\Pr(a,\overline{b})$ *é*

$$\Pr(a,\overline{b}) = f(a,\overline{b}) = \theta_a\theta_{\overline{b}|a}$$

Se desejassemos a probabilidade $\Pr(b)$*, as variáveis indicadoras teriam valores* $\lambda_a = \lambda_{\overline{a}} = \lambda_b = 1$ *e* $\lambda_{\overline{b}} = 0$*.*

3.2. *Estrutura de uma Sum-Product Network*

Definição 3.2.1. *Uma SPN S é um DAG com três tipos de nós: soma, produto e indicadores. Todo nó indicador é uma folha. Todo nó soma tem pais produto, e todo nó produto tem pais soma. Toda aresta com destino a um nó soma tem uma aresta com um peso associado. O valor de um nó soma i é* $\sum_{j\in Ch(i)} w_{ij}v_j$ *e o valor de um nó produto i é* $\prod_{j\in Ch(i)} v_j$*, onde* $Ch(i)$ *é o conjunto de filhos de i,* v_i *é o valor do nó i e* w_{ij} *é o peso associado a aresta* $i \to j$*. Uma SPN representa um* network polynomial *de uma distribuição de probabilidade, e os indicadores da função são as folhas da SPN. O valor de uma SPN é o valor do nó raíz.*

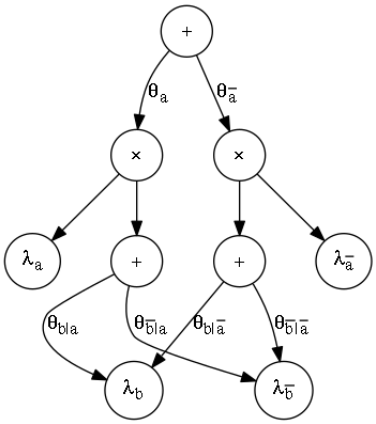

FIGURA 3. Estrutura da SPN que representa a *network polynomial* da Figura 2.

Ao contrário dos CPGMs que possuem uma estrutura que explicitamente representa a distribuição de probabilidade, SPNs tem sua representatividade implícita por uma função polinomial, assim como Redes Neurais.

3.3. *Inferência por Retropropagação*

Devido a sua estrutura semelhante a de Redes Neurais, podemos computar inferência exata por meio do método de retropropagação, assim como em RNs.

Para computarmos a probabilidade marginal $\Pr(\mathbf{Y})$, utilizamos as variáveis indicadoras de forma consistente com a instanciação $\mathbf{Y}$. Por exemplo, seja $\mathbf{Y} = \{X_1 = true\}$ e sejam as variáveis da rede $\mathbf{X} = \{X_1, X_2\}$. Então queremos a probabilidade $\Pr(X_1 = true)$. Portanto, as variáveis indicadores serão $\lambda_{X_1} = 1$, $\lambda_{\overline{X}_1} = 0$, $\lambda_{X_2} = \lambda_{\overline{X}_2} = 1$.

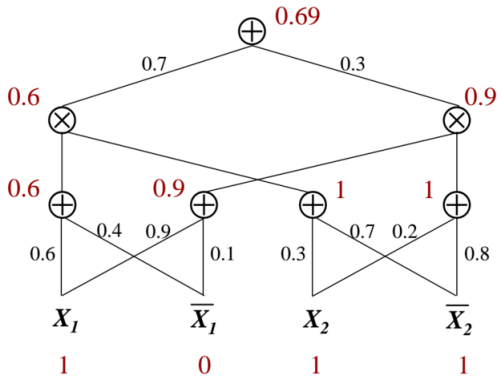

FIGURA 4. O valor da SPN é o valor do nó raíz. Neste caso, o valor da SPN $S = \Pr(X_1 = true) = 0.69$.

Para acharmos a probabilidade correspondente, fixamos os valores das folhas (variáveis indicadoras) para seus valores consistentes com a nossa instanciação e em seguida computamos o valor da raíz recursivamente. Probabilidades posteriores podem então serem computadas pela definição de probabildade condicional após achadas as probabilidades marginais.

3.4. *Aprendizado*

Aprendizado em SPNs é dividido em duas classes: paramétrico [PD11] e estrutural [GD13].

Na primeira classe, temos uma estrutura fixa com uma quantidade de camadas ocultas razoavelmente grande e desejamos aprender os pesos das arestas da SPN. Assim como em Redes Neurais, é possível utilizarmos o gradiente para encontrarmos um máximo local. Outro possível método é utilizarmos EM (expectation-maximization).

Na segunda classe, desejamos criar uma estrutura com um grande número de camadas ocultas. Podemos arranjar as camadas por meio da identificação de independências entre as variáveis.

4. Planejamento

Planeja-se estudar os seguintes tópicos:

(1) *Background* teórico (PGMs clássicos, probabilidade)
(2) Inferência em SPNs
    (a) Função de partição
    (b) Marginais
    (c) MAP
    (d) MPE
(3) Aprendizagem em SPNs
    (a) Paramétrica
    (b) Estrutural

Por ser um tópico bem recente e avançado, a maior parte do estudo será teórico.

Referências

[Coo88] Gregory F. Cooper. "The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks". Em: (1988).

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

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

[KBG10] J. H.P. Kwisthout, Hans L. Bodlaender e L. C. van der Gaag. "The Necessity of Bounded Treewidth for Efficient Inference in Bayesian Networks". Em: *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference in Artificial Intelligence* (2010).

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

[Peh15] Robert Peharz. "Foundations of Sum-Product Networks for Probabilistic Modeling". Tese de doutorado. Graz University of Technology, 2015.