

Universidade de São Paulo
Escola Politécnica

Tiago Madeira

**Redes soma-produto
e sua relação com redes bayesianas**

PCS5708 – Técnicas de raciocínio probabilístico em inteligência artificial
Prof. Dr. Paulo Sergio Cugnasca

São Paulo
Maio de 2018

Resumo

Redes soma-produto são modelos profundos tratáveis para inferência probabilística. Neste trabalho, desenvolvido durante a disciplina de *Técnicas de Raciocínio Probabilístico em Inteligência Artificial*, estudamos sobre redes soma-produto, sua semântica probabilística e sua relação com redes bayesianas. Implementamos um algoritmo para construir uma rede bayesiana a partir de uma rede soma-produto em tempo linear no tamanho da rede. Nesta monografia, apresentamos os fundamentos teóricos e os resultados obtidos.

Palavras-chave: redes soma-produto, redes bayesianas, aprendizagem profunda, modelos probabilísticos baseados em grafos.

Abstract

Sum-product networks are tractable deep models for probabilistic inference. In this work, developed during the course in *Técnicas de Raciocínio Probabilístico em Inteligência Artificial*, we studied about sum-product networks, its probabilistic semantic and its relationship with Bayesian networks. We implemented an algorithm to convert a sum-product network into a Bayesian network in linear time in the size of the network. In this monograph, we present the theoretical foundations and the obtained results.

Keywords: sum-product networks, Bayesian networks, deep learning, probabilistic graphical models.

Sumário

1	Introdução	1
2	Fundamentos	3
2.1	Redes bayesianas	3
2.2	Redes soma-produto	4
2.3	Diagramas de decisão algébrica	7
3	De redes soma-produto para redes bayesianas	9
3.1	Algoritmo de Zhao <i>et al.</i>	9
3.1.1	Construção da estrutura	10
3.1.2	Distribuições de probabilidade condicional	10
3.2	Implementação	12
3.3	Exemplos	14
3.4	Conclusões	14
4	Considerações finais	17

Capítulo 1

Introdução

Redes soma-produto [10] (*SPN*, do inglês *Sum-Product Network*) são modelos profundos tratáveis para inferência probabilística. Embora uma rede soma-produto possa ser vista como uma rede neural com funções de ativação logarítmicas e exponenciais [4], ela é mais flexível por permitir aprendizagem não supervisionada e inferência de probabilidades condicionais no lugar de uma função fixa como saída. Por outro lado, diferente de modelos probabilísticos baseados em grafos como redes bayesianas e redes de Markov [5], as redes soma-produto são robustas e permitem inferência exata em tempo linear no tamanho da rede.

Desde que foram propostas, em 2011, as redes soma-produto têm apresentado resultados muito bons para diversas aplicações, como modelagem de linguagem [3], modelagem de sinais de fala [9], reconhecimento de atividade [1], classificação e reconstrução de imagens [10], entre outras. Além disso, se mostrou que elas podem aprender de forma *online* [6] e distribuída [12], características que demonstram escalabilidade e utilidade em situações com grande volume de dados.

Em 2015, um artigo de Zhao *et al.* [13] apresentou uma conexão teórica entre redes soma-produto e redes bayesianas (*BN*, do inglês *Bayesian Networks*),

junto com algoritmo linear para realizar a conversão de um modelo para o outro. O que essa conexão pode nos dizer sobre o conhecimento probabilístico e as relações de dependência codificadas nas redes soma-produto?

Isso nos levou a estudar redes soma-produto e sua relação com redes bayesianas, assim como implementar o algoritmo de Zhao *et al.* [13] para construir uma rede bayesiana a partir de uma rede soma-produto. O segundo foi motivado também por não termos encontrado tal implementação disponível na Internet durante o desenvolvimento deste trabalho.

Assim como pode-se situar as redes soma-produto na interseção entre modelos de aproximação de função como redes neurais profundas e modelos probabilísticos baseados em grafos, este trabalho — essencialmente teórico — pode ser situado na interseção entre Ciência da Computação e a disciplina de *Técnicas de Raciocínio Probabilístico em Inteligência Artificial*. Ele parte da tentativa de compreender o raciocínio probabilístico codificado num modelo profundo usado em aprendizagem de máquina.

Capítulo 2

Fundamentos

Neste capítulo, apresentamos definições fundamentais sobre redes bayesianas, redes soma-produto e diagramas de decisão algébrica, que o leitor deve conhecer para compreender o trabalho.

Assumimos que o leitor conhece noções básicas de conjuntos, grafos e que está familiarizado com teoria de probabilidades discretas [5].

2.1 Redes bayesianas

Redes bayesianas são modelos probabilísticos baseados em grafo que representam distribuições de probabilidade conjunta e são usados para raciocinar em situações com incerteza. Formalmente, ela é definida [7] da seguinte forma:

Definição 1 (rede bayesiana). Seja $N = \{1, \dots, n\}$ e seja $X = \{X_i : i \in N\}$ um conjunto de variáveis aleatórias X_i tomando valores em conjuntos finitos \mathcal{X}_i . Uma **rede bayesiana** é uma tripla (X, G, θ) , onde $G = (V, E)$ é um DAG (que chamamos de **estrutura** da rede bayesiana) cujos vértices correspondem a variáveis em X e $\theta = \{\theta_i(x_i, x_{\pi_i})\}$ é um conjunto de parâmetros numéricos especificando valores de probabilidade condicional $\theta_i(x_i, x_{\pi_i}) = P(x_i | x_{\pi_i})$ para todo vértice $i \in V$, valor $x_i \in \mathcal{X}_i$ e atribuição

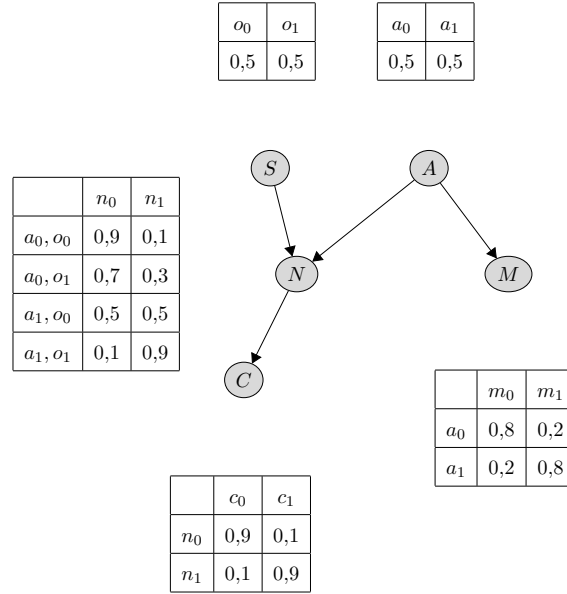


Figura 2.1: Exemplo de rede bayesiana com distribuições de probabilidade condicional.

x_{π_i} para os pais π_i de X_i (em G).

Um exemplo de rede bayesiana é mostrado na figura 2.1. Intuitivamente, as arestas codificam dependências entre as variáveis.

Redes bayesianas são geralmente usadas para computar a probabilidade de alguma hipótese depois que alguma evidência é observada. Por exemplo, uma rede bayesiana pode representar as relações de probabilidade entre doenças e sintomas. Observados alguns sintomas, a rede pode ser usada para computar a probabilidade da presença das doenças. Chamamos esse processo de **inferência**.

2.2 Redes soma-produto

Uma **rede soma-produto** [10] é um grafo acíclico dirigido enraizado no qual as folhas são indicadores de variáveis ou distribuições univariadas, os

nós internos são somas ou produtos, e as arestas que saem de nós do tipo soma são ponderadas.

Um exemplo de rede soma-produto é mostrado na figura 2.2 (a).

Para calcular uma probabilidade numa rede soma-produto, basta preencher suas folhas com os valores das variáveis na inferência que desejamos fazer — por exemplo, se desejamos calcular $P(X_1, \bar{X}_2)$ devemos colocar 1 como valor das folhas que representam X_1 e \bar{X}_2 e 0 como valor das folhas que representam \bar{X}_1 e X_2 — e caminhar na rede de baixo pra cima realizando as somas e produtos indicadas em cada nó. Num nó soma v com filhos $v_i \in Ch(v)$, calcula-se $\sum_{i=1}^{|Ch(v)|} w_i v_i$; num nó produto v com filhos $v_i \in Ch(v)$ calcula-se $\prod_{i=1}^{|Ch(v)|} v_i$. Tal processo é exemplificado na figura 2.2 (b).

O resultado não está normalizado. Para normalizá-lo, basta marginalizar todas as variáveis. Se realizarmos esse processo na rede da figura 2.2 (a), chegaremos ao valor 3500. Portanto, $P(X_1, \bar{X}_2) = \frac{1776}{3500}$.

Para marginalizar uma variável — marginalizar X_2 para calcular $P(X_1)$, por exemplo — basta considerar que tanto X_2 como \bar{X}_2 valem 1. Para calcular uma probabilidade condicional como $P(\bar{X}_2|X_1)$, basta percorrer a rede duas vezes calculando $P(X_1, \bar{X}_2)$ e $P(X_1)$, e depois calcular a probabilidade condicional com base nelas:

$$P(\bar{X}_2|X_1) = \frac{P(X_1, \bar{X}_2)}{P(X_1)}$$

Numa rede soma-produto, definimos o **escopo** de um nó como o conjunto de variáveis presentes na sub-rede enraizada nele. Formalmente [13],

Definição 2 (escopo). Seja v é um nó. Se v é uma folha indicando uma distribuição sobre uma variável X , então $\text{escopo}(v) = \{X\}$. Caso contrário, $\text{escopo}(v) = \cup_{w \in Ch(v)} \text{escopo}(w)$, onde $Ch(v)$ é o conjunto dos filhos de v na rede soma-produto.

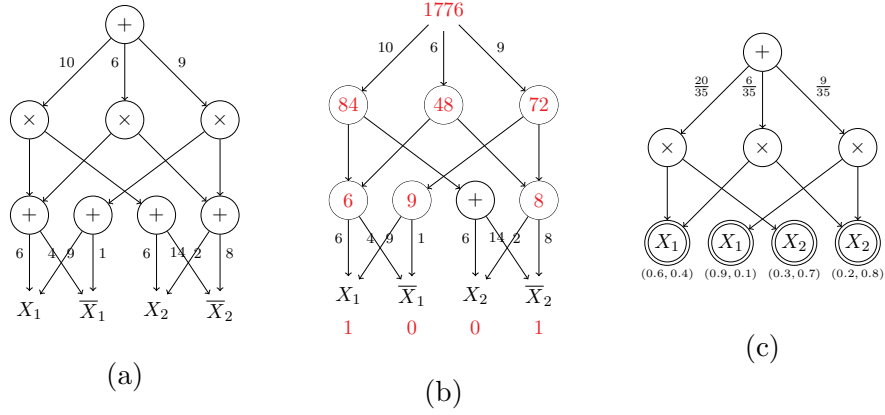


Figura 2.2: (a) Uma rede soma-produto \mathcal{S} . (b) Exemplo de cálculo de $P(X_1, \bar{X}_2)$ (sem normalização) na rede soma-produto \mathcal{S} . (c) A rede soma-produto \mathcal{S} convertida para a forma normal.

A partir da definição de escopo, Poon e Domingos [10] definem outras propriedades importantes sobre redes soma-produto:

Definição 3 (SPN completa). Uma rede soma-produto é **completa** se cada nó do tipo soma tem filhos com o mesmo escopo.

Definição 4 (SPN consistente). Uma rede soma-produto é **consistente** se nenhuma variável aparece negada num filho de um nó produto e não-negada em outro.

Definição 5 (SPN decomponível). Uma rede soma-produto é **decomponível** se para todo nó produto v , a interseção dos seus filhos dois a dois é vazia, isso é, $\text{escopo}(v_i) \cap \text{escopo}(v_j) = \emptyset$ para todos $v_i, v_j \in Ch(v)$ com $i \neq j$.

É fácil ver que se uma SPN é decomponível então ela é consistente.

Definição 6 (SPN válida). Uma rede soma-produto é **válida** se ela define uma distribuição de probabilidade (não necessariamente normalizada).

Poon e Domingos provaram [10] que se uma rede soma-produto é completa e consistente, então ela é válida.

Com base nas propriedades acima, podemos agora definir uma rede soma-produto normal [13]:

Definição 7 (forma normal). Uma SPN é dita **normal** se:

1. Ela é completa e decomponível.
2. Os pesos das arestas saindo de cada nó da SPN são não-negativos e somam 1. Isso é, para todo nó soma v com filhos $v_i \in Ch(v)$ e arestas com peso w_i conectando v e v_i , vale $w_i \geq 0$ e $\sum_{i=1}^{|Ch(v)|} w_i = 1$.
3. Todo nó terminal da SPN é uma distribuição univariada sobre uma variável booleana e o tamanho do escopo de um nó soma é pelo menos 2 (nós soma cujo escopo é 1 são reduzidos em nós terminais).

Zhao *et al.* [13] provam que para toda SPN completa e consistente \mathcal{S} existe uma SPN normal \mathcal{S}' tal que $P_{\mathcal{S}}(\cdot) = P_{\mathcal{S}'}(\cdot)$ e $|\mathcal{S}'| = O(|\mathcal{S}|^2)$.

Um exemplo de uma SPN normal construída de uma SPN geral é mostrado na figura 2.2 (c).

2.3 Diagramas de decisão algébrica

Diagramas de decisão algébrica (*ADD*, do inglês *Algebraic Decision Diagram*) são semelhantes a árvores de decisão, mas são mais compactos porque exploram subgrafos isomórficos. Formalmente são definidos como segue [2]:

Definição 8 (diagrama de decisão algébrica). Um **diagrama de decisão algébrica** é uma representação num grafo acíclico dirigido enraizado de uma função real com entrada de variáveis booleanas: $f : \{0, 1\}^N \rightarrow \mathbb{R}$. Há dois tipos de nós num ADD: nós terminais, cujo grau de saída é 0, são associados com valores reais. Nós internos, cujo grau de saída é 2, são associados com variáveis booleanas X_n , $n \in [1, N]$.

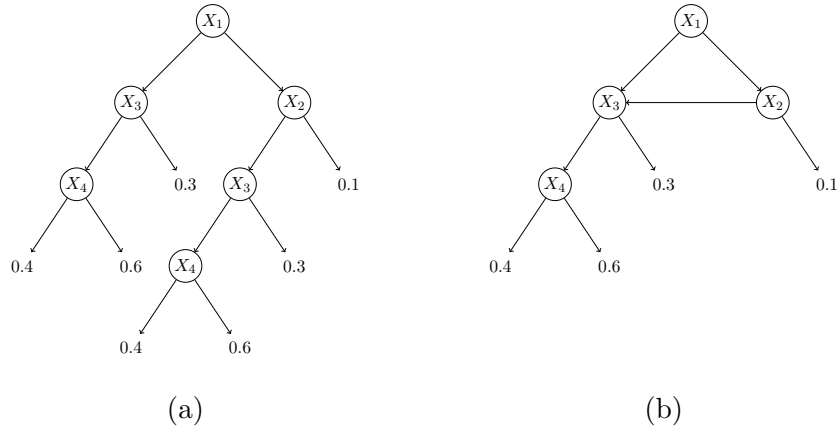


Figura 2.3: **(a)** Representação de uma função de variáveis binárias com árvore de decisão. **(b)** Representação da mesma função com diagrama de decisão algébrica.

A figura 2.3 mostra uma árvore de decisão e uma ADD equivalente.

Zhao *et al.* [13] estendem a definição original de diagrama de decisão algébrica permitindo que ele represente não apenas funções de variáveis booleanas, mas qualquer função de variáveis discretas com um domínio finito. Quando falamos em diagramas de decisão algébrica nesta monografia estamos nos referindo a essa extensão.

Capítulo 3

De redes soma-produto para redes bayesianas

Enquanto as redes bayesianas são visualizações gráficas de dependências diretas e redes de Markov são visualizações gráficas de correlações, as redes soma-produto são visualizações gráficas de computações [11].

Desde a introdução de SPNs no trabalho de Poon e Domingos [10], esteve claro que as SPNs e BNs são igualmente expressivas no sentido que elas podem representar qualquer distribuição conjunta sobre variáveis discretas. Entretanto, apenas em 2015 foi provado, por Zhao *et al.* [13], que toda rede soma-produto pode ser convertida numa rede bayesiana em complexidade linear de tempo e espaço no tamanho da SPN.

Tal artigo apresenta um algoritmo para realizar essa conversão, que foi estudado e implementado neste trabalho.

3.1 Algoritmo de Zhao *et al.*

Dada uma SPN normal \mathcal{S} sobre variáveis booleanas $X_{1:N}$, o algoritmo de Zhao *et al.* [13] retorna uma rede bayesiana \mathcal{B} que representa a mesma distribuição com $|\mathcal{B}| = O(N|\mathcal{S}|)$. Pode-se explicar o algoritmo mais facilmente

separando-o em duas etapas:

1. Construção da estrutura da rede bayesiana;
2. Computação das distribuições de probabilidade condicional (*CPD*, do inglês *conditional probability distribution*).

3.1.1 Construção da estrutura

A rede bayesiana resultante terá dois tipos de variáveis: observáveis e ocultas. As variáveis observáveis correspondem a variáveis de \mathcal{S} , enquanto as variáveis ocultas correspondem a nós soma de \mathcal{S} .

Para construir a estrutura da BN, portanto, deve-se criar nós para cada variável de \mathcal{S} e para cada nó soma de \mathcal{S} . Os nós da BN associados aos nós soma da SPN tem arestas direcionadas para todas as variáveis do seu escopo.

Logo, a rede bayesiana terá uma estrutura bipartida. As variáveis ocultas terão apenas grau de saída, enquanto as variáveis observáveis terão apenas grau de entrada. A figura 3.1 (a) mostra como fica a estrutura da rede bayesiana que corresponde a conversão da SPN normal vista na figura 2.2 (c).

3.1.2 Distribuições de probabilidade condicional

Apresentaremos primeiro as distribuições de probabilidade condicional das variáveis ocultas e depois das variáveis observáveis.

Se H_v é a variável oculta correspondente ao nó soma v de \mathcal{S} e l é o grau de saída de v , como \mathcal{S} é normal, temos $\sum_{i=1}^l w_i = 1$ e $w_i \geq 0 \quad \forall i$. Isso sugere tomar $P(H_v = i) = w_i$.

Agora fixemos uma variável observável X . Sejam H_{v_1}, \dots, H_{v_k} as variáveis ocultas que apontam para ela na rede bayesiana. Precisamos definir $P(X|H_{v_1} =$

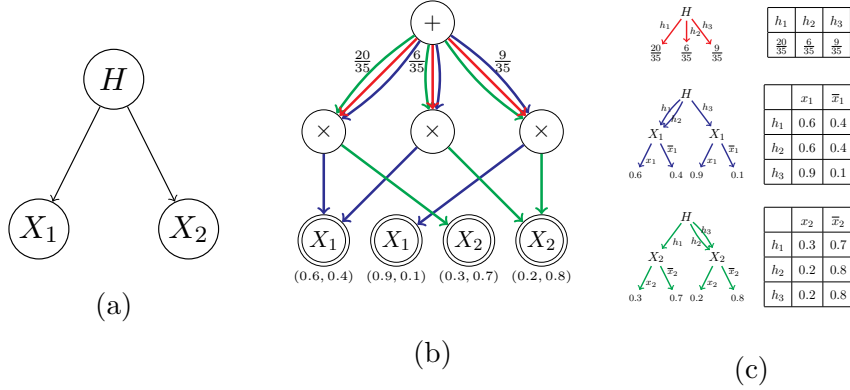


Figura 3.1: **(a)** Estrutura da rede bayesiana correspondente à SPN normal vista na figura 2.2 (c). **(b)** Sub-SPNs induzidas usadas para construir os diagramas de decisão algébrica. **(c)** Distribuições de probabilidade condicional calculados para a rede bayesiana convertida (diagramas de decisão algébrica e tabelas equivalentes).

$v_1^*, \dots, H_{v_m} = v_m^*$) para cada combinação de v_1^*, \dots, v_m^* (valores que as variáveis ocultas H_{v_1}, \dots, H_{v_m} podem assumir).

A ideia para fazer isso vem da Proposição 1 do artigo [13]:

Proposição 1. Seja p um nó produto em \mathcal{S} com l filhos. Sejam p_1, \dots, p_l os filhos de p . Sejam v_1, \dots, v_k os nós soma no caminho da raiz de \mathcal{S} até p . Então

$$P(X_{|\text{escopo}(p)} | H_{v_1} = v_1^*, \dots, H_{v_k} = v_k^*) = \prod_{i=1}^l P(X_{|\text{escopo}(p_i)} | H_{v_1} = v_1^*, \dots, H_{v_k} = v_k^*) \quad (3.1)$$

onde $H_v = v^*$ significa que o nó soma v seleciona seu v^* -ésimo filho e $X_{|A}$ denota a restrição de X pelo conjunto A .

Como a SPN é decomponível, cada nó produto tem filhos com escopos disjuntos. Para cada variável observável X , construímos um ADD extraíndo

de \mathcal{S} a sub-SPN induzida por X e contraindo todos seus nós produto. Per-corremos o ADD para encontrar $P(X|H_{v_1} = v_1^*, \dots, H_{v_m} = v_m^*)$.

A figura 3.1 (b) mostra as sub-SPNs induzidas usadas para criar os diagramas de decisão algébrica da rede bayesiana na figura 3.1 (a). A figura 3.1 (c) mostra os ADDs resultantes e as tabelas de CPDs equivalentes a esses ADDs.

Zhao *et al.* [13] apresenta uma prova para o algoritmo por indução na altura de \mathcal{S} .

3.2 Implementação

A implementação desse algoritmo foi realizada na linguagem Go¹. *Go* é uma linguagem de código aberto criada em 2007 e apoiada pelo *Google*. Ela é compilada e usa tipagem estática como o C, mas possui recursos avançados como *garbage collection* e bom suporte a programação concorrente. Escolhemos *Go* porque ela oferece um bom balanço entre a agilidade de escrita de código e a eficiência computacional; tem sistemas de pacotes (`go get`), testes (`go test`) e documentação (*GoDoc*²) padronizados facilitando que os códigos sejam testados e reutilizados; e ajuda a gerar código limpo e padronizado: indentação, espaçamento e outros detalhes de estilo são automatizados pela ferramenta `gofmt`.

Foram desenvolvidos quatro pacotes:

- **add**: estrutura de dados que representa diagramas de decisão algébrica
- **bn**: estrutura de dados que representa redes bayesianas com CPDs representados por ADDs
- **spn**: estrutura de dados que representa redes soma-produto

¹*The Go Programming Language*: <https://golang.org/>

²*GoDoc*: <https://godoc.org/>

- zhao: implementação do algoritmo

Eles foram disponibilizados publicamente em <https://github.com/tmadeira/spnbn>. Um exemplo de uso do algoritmo pode ser visto em `examples/a/a.go`. Tal código cria a rede soma-produto S vista na figura 2.2 (c) e usa o algoritmo chamando `zhao.BuildBN(S)`:

```

1 func makeSPN(x1, x2 int) spn.Node {
2     S := spn.NewSumNode()
3     a, b, c := spn.NewProductNode(), spn.NewProductNode(), spn.
        NewProductNode()
4     d, e := spn.NewTerminalNode(x1, 0.6), spn.TerminalNode(x1,
        0.9)
5     f, g := spn.NewTerminalNode(x2, 0.3), spn.NewTerminalNode(x2,
        0.2)
6
7     a.AddChild(d)
8     a.AddChild(f)
9     b.AddChild(d)
10    b.AddChild(g)
11    c.AddChild(e)
12    c.AddChild(g)
13    S.AddChild(a, 20.0/35.0)
14    S.AddChild(b, 6.0/35.0)
15    S.AddChild(c, 9.0/35.0)
16
17    return S
18 }
19
20 func main() {
21     S := makeSPN(1, 2)
22     B := zhao.BuildBN(S)

```

3.3 Exemplos

A figura 3.2 (a) mostra uma SPN \mathcal{T} construída a partir da duplicação da SPN apresentada na figura 2.2 (c) e da adição de um nó produto que aponta para a rede original e para sua cópia.

Como se pode perceber, a rede bayesiana resultante da conversão usando o algoritmo implementado neste trabalho, apresentada na figura 3.2 (b), é um grafo desconexo, separado em dois subgrafos bipartidos exatamente iguais ao visto na figura 3.1.

Já que não são adicionados nós soma, não há nenhuma nova variável oculta na rede bayesiana resultante. Como não há nenhum nó soma que tenha como escopo uma variável entre X_1, X_2 e outra entre X_3, X_4 o grafo acaba ficando desconexo. Como se pode esperar, portanto, a adição de um nó produto na raiz não tem muita utilidade no sentido de mudar o entendimento da semântica probabilística da estrutura.

Esse exemplo pode ser visto no arquivo `examples/b/b.go` disponibilizado junto com a implementação do algoritmo.

Um segundo exemplo foi construído duplicando-se a SPN do exemplo anterior e adicionando-se um nó soma que aponta para tal SPN e sua cópia. Tal SPN e a estrutura da rede bayesiana construída a partir da sua conversão são mostradas na figura 3.3.

Com a adição do nó soma, aparece uma nova variável oculta na rede bayesiana que aponta para todas as variáveis observáveis do exemplo.

3.4 Conclusões

O artigo de Zhao *et al.* [13] mostra que embora a rede bayesiana resultante da conversão tenha estrutura simples (bipartida), é possível relacionar a profundidade de uma SPN com a *treewidth* da rede bayesiana convertida. Isso

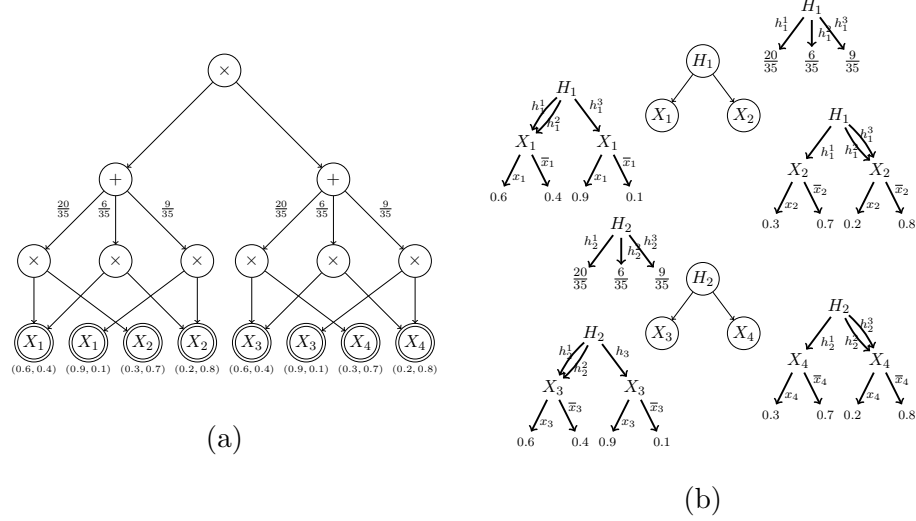


Figura 3.2: **(a)** SPN \mathcal{T} construída duplicando a SPN da figura 2.2 (c) e adicionando um nó produto que aponta para a SPN original e sua cópia. **(b)** Conversão da SPN \mathcal{T} em rede bayesiana usando algoritmo de Zhao *et al.*

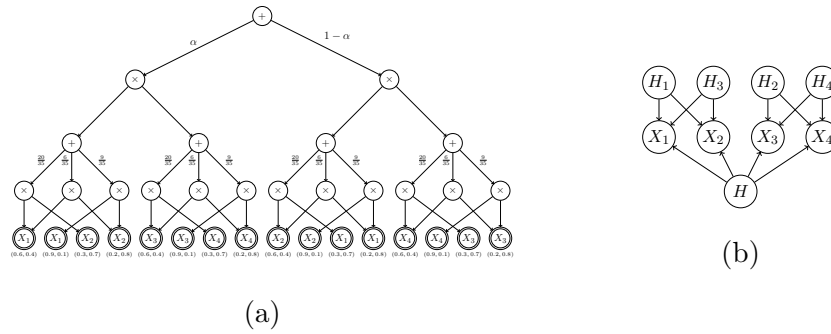


Figura 3.3: **(a)** SPN \mathcal{U} construída duplicando a SPN \mathcal{T} da figura 3.2 (a) e adicionando um nó soma que aponta para a SPN original e sua cópia. **(b)** Estrutura da rede bayesiana construída por meio da conversão da SPN \mathcal{U} usando algoritmo de Zhao *et al.*

indica que mais camadas numa rede soma-produto implicam num *treewidth* maior para a rede bayesiana e portanto uma SPN com mais camadas pode representar distribuições mais complexas.

Além disso, ele esclarece que pode haver outras técnicas para converter uma SPN numa BN com uma representação mais compacta e um *treewidth* menor. De fato, ele não prova que seu algoritmo é a única forma de fazê-lo.

No mais, um resultado importante desse trabalho é que a partir dele pode-se usar algoritmos para aprender estrutura e parâmetros de redes soma-produto para aprender redes bayesianas com diagramas de decisão algébrica.

Capítulo 4

Considerações finais

Redes soma-produto são um modelo relativamente novo e bastante promissor. Ainda há muitas questões teóricas em aberto, como mostrou Peharz em 2015 [8].

Estudar o artigo de Zhao *et al.* [13] nos permitiu ter mais compreensão sobre a relação entre redes soma-produto e redes bayesianas. Sabe-se que redes soma-produto e redes bayesianas podem codificar as mesmas distribuições de probabilidade, assim como sabe-se que é bem mais fácil relacionar o tamanho de redes soma-produto com o seu tempo de inferência do que fazer o mesmo com redes bayesianas.

De fato, se definirmos os conceitos de **compactibilidade** como “uso de espaço polinomial no número de variáveis” e **tratabilidade** como “tempo de inferência polinomial no número de variáveis”, a partir do resultado de Zhao *et al.* [13], Poupart [11] conclui que o conjunto das CPDs representadas por redes soma-produto compactas é igual ao conjunto das CPDs representadas por redes soma-produto tratáveis e do conjunto das CPDs representadas por redes bayesianas tratáveis. Porém, redes bayesianas compactas não necessariamente são tratáveis. Essas relações podem ser vistas na figura 4.1.

Embora a implementação do algoritmo de Zhao *et al.* [13] — resultado material deste trabalho — possa não ter muita aplicação prática neste mo-

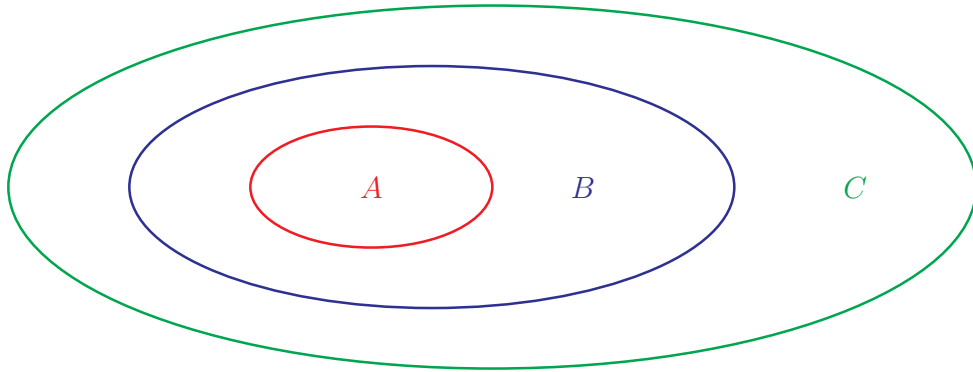


Figura 4.1: *A*: SPNs tratáveis = SPNs compactas = BNs tratáveis. *B*: BNs compactas. *C*: SPNs gerais = BNs gerais.

mento, serviu para facilitar a compreensão do método e para se refletir sobre como representar redes soma-produto e redes bayesianas com diagramas de decisão algébrica utilizando estruturas de dados computacionais.

Referências Bibliográficas

- [1] Mohamed R. Amer and Sinisa Todorovic. Sum product networks for activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.
- [2] R.I. Bahar, E.A. Frohm, C.M. Gaona, G.D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Algebraic decision diagrams and their applications. In *Proceedings of 1993 International Conference on Computer Aided Design (ICCAD)*, pages 188–191. IEEE Comput. Soc. Press, 1993.
- [3] Wei-Chen Cheng, Stanley Kok, Hoai Vu Pham, Hai Leong Chieu, Kian Ming, and A Chai. Language Modeling with Sum-Product Networks. 2014.
- [4] Wilson Hsu, Agastya Kalra, and Pascal Poupart. Online Structure Learning for Sum-product Networks with Gaussian Leaves. *Iclr*, pages 1–10, 2017.
- [5] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. 2009.
- [6] Sang-Woo Lee, Min-Oh Heo, and Byoung-Tak Zhang. Online Incremental Structure Learning of Sum-Product Networks. *LNCS*, 8227:220–227, 2013.

- [7] Siqi Nie, Denis Deratani Maua, Cassio Polpo de Campos, Qiang Ji, Denis Deratani Mauá, Cassio Polpo de Campos, and Qiang Ji. Advances in Learning Bayesian Networks of Bounded Treewidth. *Advances in Neural Information Processing Systems 27*, 2014.
- [8] Robert Peharz. *Foundations of Sum-Product Networks for Probabilistic Modeling*. PhD thesis, Graz University of Technology, 2015.
- [9] Robert Peharz, Georg Kapeller, Pejman Mowlae, and Franz Pernkopf. Modeling speech with sum-product networks: Application to bandwidth extension. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2014.
- [10] Hoifung Poon and Pedro Domingos. Sum-Product Networks: A New Deep Architecture. 2012.
- [11] Pascal Poupart. Guest Lecture in STAT946 - Deep Learning (University of Waterloo) on October 17th, 2017.
- [12] Abdullah Rashwan, Han Zhao, and Pascal Poupart. Online and Distributed Bayesian Moment Matching for Parameter Learning in Sum-Product Networks. 2016.
- [13] Han Zhao, Mazen Melibari, and Pascal Poupart. On the Relationship between Sum-Product Networks and Bayesian Networks. 2015.