ESTUDO SOBRE SUM-PRODUCT NETWORKS E APRENDIZAGEM PROFUNDA

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

RELATÓRIO 2

RENATO LUI GEH
NUSP: 8536030

RESUMO. Sum-Product Networks (SPNs) são modelos probablísticos baseados em grafo que representam uma distribuição de probabilidade por meio de uma função multilinear dos parâmetros de uma Rede Bayesiana. Neste relatório, definiremos uma SPN recursivamente e mostraremos algumas propriedades interessantes da rede. Em seguida mostraremos como executar aprendizagem estrutural de uma SPN de forma que consigamos criar uma estrutura profunda a partir dos dados observados.

1. DEFINIÇÕES

No relatório anterior, definimos uma Sum-Product Network (SPN) como um DAG com nós soma, produto e indicadores. Vamos transcrever a definição abaixo.

Definição 1.1. *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.*

Considere um nó arbitrário i de uma SPN, então a SPN representada por i é uma SPN e é denotada por $S_i(\cdot)$.

Proposição 1.1. *Seja um nó arbitrário i de uma SPN S, então S_i é uma sub-SPN que tem nó raíz em i.*

Demonstração. Considere o caso base em que i é um nó indicador. Um nó indicador é uma distribuição de probabilidade monovariável. Portanto i é uma distribuição de probabilidade e pode ser representada por uma SPN, que no caso possue apenas um nó.

Se i é um nó soma, então o valor de i é $v_i = \sum_{j \in Ch(i)} w_{ij} v_j$. A soma de várias distribuições de probabilidade é uma distribuição de probabilidade. Portanto um nó soma é representável por uma SPN.

Caso i seja um nó produto, então o valor de i é $v_i = \sum_{j \in Ch(i)} v_j$. A multiplicação de distribuições de probabilidade é bem definida e é uma distribuição de probabilidade. Um nó produto é uma SPN. □

O escopo de uma SPN S é o conjunto união de todos os escopos de seus filhos. Denotaremos por $Sc(S_i)$ o escopo da SPN S_i.

2. Propriedades

Vamos abordar duas propriedades que são essenciais para definirmos uma SPN que computa de forma eficiente probabilidades marginais.

Definição 2.1 (Completude)**.** *Uma SPN S é completa se e somente se, para todo nó soma i, o escopo de i é igual par-a-par $Sc(S_i) = Sc(S_j)$ ao escopo de cada filho $j \in Ch(i)$.*

Em outras palavras, se tivermos duas distribuições de probabilidade $p_1(\mathbf{X})$ e $p_2(\mathbf{X})$, onde seus escopos são $\mathbf{X}$, uma SPN S que representa a soma de p_1 e p_2 é completa sse $Sc(S) = Sc(p_1) = Sc(p_2) = \mathbf{X}$.

Definição 2.2 (Consistência)**.** *Uma SPN S é consistente se e somente se, para todo nó produto i, nenhum filho de i tem valor diferente dos outros filhos.*

A propriedade de consistência garante que não haja contradição entre as variáveis indicadoras, ou seja, o valor de uma variável indicadora não pode ter dois valores simultâneos.

Definição 2.3 (Validade)**.** *Uma SPN S é válida se S é consistente e completa.*

Note que a volta não é necessariamente verdade. De fato existem SPNs válidas que não são consistentes nem completas. No entanto, se uma SPN é completa e consistente, então todas as suas sub-SPNs são válidas. A validade de uma SPN garante que inferência seja computada em tempo eficiente e que seu resultado seja exato. Uma SPN que não seja válida computa uma probabilidade aproximada.

Definição 2.4 (Decomponibilidade)**.** *Uma SPN S é decomponível se e somente se, para cada par $c_1, c_2 \in Ch(i)$ para qualquer i nó produto em S, $Sc(c_1) \cap Sc(c_2) = \emptyset$.*

Ou seja, todo filho de um nó produto tem escopos diferentes entre si. Uma SPN decomponível é uma SPN consistente, no entanto consistência não implica em decomponibilidade.

3. Uma definição alternativa

Em [GD13], foi proposto uma nova definição de SPNs. Esta definição, é mais forte que a definição dada em [PD11] (cuja Definição 1.1 se baseia), já que, como veremos, assume já que a distribuição possa ser representada por uma SPN completa e decomponível. Esta definição, de mais "alto nível", nos permite mudar o foco de nossos estudos, distanciando-nos da área de Representação e voltando nossos interesses ao tópico de aprendizado e inferência.

Definição 3.1. *Uma SPN tem uma definição recursiva. Definimos que uma SPN S_i pode ser apenas:*

(i) Uma distribuição monovariável $p(\mathbf{X})$ ou;
(ii) Um nó soma tal que $S_i = \sum_{j \in Ch(i)} w_{ij} v_j$ onde para cada filho $j, k \in Ch(i)$, $Sc(S_j) = Sc(S_k)$ ou;
(iii) Um nó produto tal que $S_i = \prod_{j \in Ch(i)} v_j$ onde para cada filho $j, k \in Ch(i)$, $Sc(S_j) \cap Sc(S_k) = \emptyset$.

Note que em (ii), estamos assumindo completude em nós somas. Da mesma forma, regra (iii) impõe decomponibilidade a nós produtos. Com relação a (i), restringimos apenas que $p(\mathbf{X})$ seja tratável, ou seja, supomos que exista uma Rede Bayesiana ótima que compute $p(\mathbf{X})$ de forma tratável (sub-exponencial). Por exemplo, uma distribuição gaussiana.

4. SPNs profundas e rasas

Em [DB11] Delalleau e Bengio mostraram que SPNs profundas possuem uma maior representabilidade do que redes rasas. Por representabilidade queremos dizer que existe uma SPN que possa, de forma tratável, representar a distribuição desejada. Mais formalmente, considere uma SPN S, uma distribuição suficientemente complexa p que é representada por S e m a dimensão do escopo de p. Definiremos uma rede rasa como uma SPN com três camadas: uma camada de entrada, uma camada oculta e uma camada de saída. A camada de entrada é composta de variáveis indicadoras que representam uma distribuição monovariável. A camada de saída é o nó raíz que representa o valor desejado. A camada oculta contém nós somas produtos. Suponha que p requer um número exponencialmente grande de arestas para se representar por S. Então, existe uma SPN com $n > 3$ camadas ocultas que representa a mesma distribuição p com um número polinomial de arestas.

5. Classificação por Naive Bayes

Um modelo Naive Bayes é uma subclasse de Redes Bayesianas onde cada nó atributo tem um mesmo pai. Neste modelo, temos um nó classe que representa a classificação que desejamos encontrar, enquanto que cada nó atributo é uma característica que influencia no resultado da classificação. Como cada nó atributo

possue uma única aresta com origem no nó classe, cada atributo é d-separado par-a-par. Esta suposição, apesar de restritiva, obtém resultados bons na prática. Esse relaxamento de dependência entre os atributos permite aprendizado rápido, simples e fácil.

Pelo Teorema da Fatorização, a distribuição de probabilidade conjunta de um modelo Naive Bayes com n atributos é computada como $\Pr(C, A_1, \ldots, A_n) = \Pr(C) \prod_{i=1}^{n} \Pr(A_i|C)$, onde C é a variável classe e $\mathbf{A} = \{A_1, \ldots, A_n\}$ é o conjunto de atributos. Para acharmos a explicação mais provável para uma certa instanciação $\mathbf{a} = \{a_1, \ldots, a_n\}$, maximizamos a instanciação da classe $\arg\max_c \Pr(C = c|A_1 = a_1, \ldots, A_n = a_n)$. A probabilidade posteriori pode ser encontrada marginalizando sob C e $\mathbf{A}$: $\Pr(C = c|A_1 = a_1, \ldots, A_n = a_n) = \Pr(C = c, A_1 = a_1, \ldots, A_n = a_n)/\Pr(A_1 = a_1, \ldots, A_n = a_n)$.

Aprendizado em Naive Bayes pode ser feita por uma estimação da máxima verossimilhança (MLE). Para a variável classe, $\Pr(C = c) = \frac{N[C=c]}{N}$ onde $N[X = x]$ é a contagem de cada dado em que houve a ocorrência da variável X instanciada a x e N é o número de contagens totais. Para cada atributo i, o MLE é dado por $\Pr(A_i = a_i|C = c) = \frac{N[A_i=a_i, C=c]}{N[C=c]}$.

Quando há poucos dados, a variância das probabilidades pode tornar-se demasiadamente grande. Para suavizarmos os resultados, podemos aplicar um fator de suavização somando alguma constante aos termos de cada atributo.

6. Um algoritmo de aprendizado estrutural

Nesta seção descreveremos simplificadamente um método de se aprender a estrutura de uma SPN a partir dos dados. Este método pode ser lido em detalhes em [GD13].

Para este algoritmo, supomos que estamos construindo um classificador baseado em Naive Bayes, ou seja, as características para classificação são independentes entre si.

No Algoritmo 1, descrevemos como aprender a estrutura de uma SPN. O algoritmo procura buscar independências entre as variáveis dos dados. Se houver uma possível partição nas variáveis, ou seja, existe uma independência entre uma partição do conjunto de variáveis e o complemento dela, podemos então criar um nó produto cujos filhos são as sub-SPNs cujos escopos são as partições mas que possuem as mesmas instanciações. Note que obedecemos a propriedade de decomponibilidade, já que como cada atributo é independente par-a-par, teremos escopos diferentes para cada filho de um nó produto. Para acharmos independências, podemos recorrer a um teste estatístico sobre os dados. Queremos minimizar os erros do tipo II (falso negativo), já que preferimos minimizar casos em o teste diz que as partições são independentes quando não são. Para isso podemos controlar os erros do tipo I (falso positivo), onde o teste diz que não são independentes mas na realidade os eventos são de fato independentes.

Algoritmo 1 LearnSPN

Input Conjunto $\mathbf{X}$ de variáveis, conjunto $\mathbf{I}$ de instâncias
Output Uma SPN resultante do aprendizado estrutural

if $|\mathbf{X}| = 1$ **then**
    Retorna uma distribuição monovariável de $\mathbf{X}$
end if
Tente dividir as variáveis $\mathbf{X}$ em duas partições $\mathbf{X}_1$ e $\mathbf{X}_2$ onde $\mathbf{X}_1$ é (aproximadamente) independente de $\mathbf{X}_2$
if dá para dividir **then**
    return $\prod_{i=1}^{2}$ LearnSPN($\mathbf{X}_i$, $\mathbf{I}$)
else
    Divida as instâncias $\mathbf{I}$ em partições $\mathbf{I}_1$ e $\mathbf{I}_2$ tal que $\mathbf{I}_1$ e $\mathbf{I}_2$ sejam o mais similares possíveis.
    return $\sum_{i=1}^{2} \frac{|\mathbf{I}_i|}{|\mathbf{I}|}$ LearnSPN($\mathbf{X}$, $\mathbf{I}_i$)
end if

Se não foi possível criar um nó produto, então resta o caso em que podemos criar um nó soma. Os nós somas são divididos de tal forma que as instâncias sejam mais similares possíveis. Isso possibilita que a SPN não fique desbalanceada, e que as camadas de nós sejam o mais "profundas" possíveis.

Quando chegamos ao caso base onde temos apenas uma variável em $\mathbf{X}$, retornamos uma distribuição monovariável como nó folha.

Para podermos visualizar mais graficamente, considere uma matriz de dados onde as linhas representam as instanciações e as colunas as variáveis. Acharmos duas partições $\mathbf{X}_1$, $\mathbf{X}_2$ é graficamente equivalente a dividir a matriz em duas submatrizes verticalmente. Podemos então computar, recursivamente, uma sub-SPN usando uma das sub-matrizes e a outra com o seu complementar. Analogamente, a divisão de instâncias semelhantes $\mathbf{I}_1$, $\mathbf{I}_2$ equivale a uma divisão horizontal da matriz. Quando temos uma submatriz de tamanho $1 \times n$, temos uma distribuição monovariável, que no caso equivale a um vetor das valorações de uma única variável.

Referências

[DB11] Olivier Delalleau e Yoshua Bengio. "Shallow vs. Deep Sum-Product Networks". Em: *Advances in Neural Information Processing Systems 24 (NIPS 1011)* (2011).

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

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