高级算法设计与分析

张量网络

夏盟佶
Xia, Mingji

中科院软件所
计算机科学国家重点实验室

2016.6

张量网络

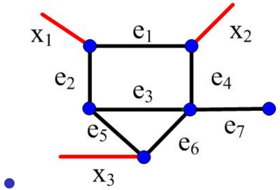


Figure: 图$G(V,E\cup X)$

- 边是变量，点是函数，点v被赋予函数F_v。
- E中边有两个顶点。X中边有一个顶点。$E=\{e_j|j=1,\ldots,m\}$。
- 定义一个以$X=\{x_1,x_2,\ldots,x_n\}\in D^n$为输入的函数$F_G$。

$$F_G(x_1,x_2,\ldots,x_n)=\sum_{e_j\in D}\prod_{v\in V}F_v(e_{v,1},e_{v,2},\ldots,e_{v,d_v})$$

- X外部边，E内部边。
- 无外部边时，是个0元函数，也定义了$2^0=1$个值。

函数的矩阵形式

- 设F是一个$n+m$元函数，$x_1,\ldots,x_n,y_1,\ldots,y_m$是它的输入。对应$d^n \times d^m$的矩阵$M=(M_{x_1\cdots x_n,y_1\cdots y_m})$，其中

$$M_{x_1\cdots x_n,y_1\cdots y_m}=F(x_1,\ldots,x_n,y_1,\ldots,y_m)。$$

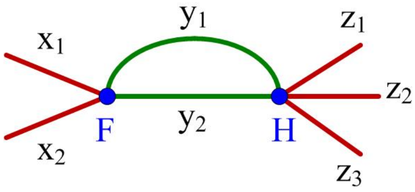
  若取$m=0$（$n=0$），就成了列（行）向量。

函数的矩阵形式

- 设F是一个$n+m$元函数，$x_1,\ldots,x_n,y_1,\ldots,y_m$是它的输入。对应$d^n\times d^m$的矩阵$M=(M_{x_1\cdots x_n,y_1\cdots y_m})$，其中
$$M_{x_1\cdots x_n,y_1\cdots y_m}=F(x_1,\ldots,x_n,y_1,\ldots,y_m)。$$
若取$m=0$（$n=0$），就成了列（行）向量。
- 如下张量网络的函数可用矩阵乘法表示：

函数的矩阵形式

- 设F是一个$n+m$元函数，$x_1,\ldots,x_n,y_1,\ldots,y_m$是它的输入。对应$d^n\times d^m$的矩阵$M=(M_{x_1\cdots x_n,y_1\cdots y_m})$，其中
$$M_{x_1\cdots x_n,y_1\cdots y_m}=F(x_1,\ldots,x_n,y_1,\ldots,y_m)。$$
若取$m=0$（$n=0$），就成了列（行）向量。
- 如下张量网络的函数可用矩阵乘法表示：



-

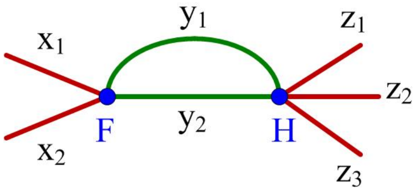
$$(F_{x_1x_2,y_1y_2})(H_{y_1y_2,z_1z_2z_3})$$

函数的矩阵形式

- 设F是一个$n+m$元函数，$x_1,\ldots,x_n,y_1,\ldots,y_m$是它的输入。对应$d^n\times d^m$的矩阵$M=(M_{x_1\cdots x_n,y_1\cdots y_m})$，其中

$$M_{x_1\cdots x_n,y_1\cdots y_m}=F(x_1,\ldots,x_n,y_1,\ldots,y_m)。$$

  若取$m=0$（$n=0$），就成了列（行）向量。
- 如下张量网络的函数可用矩阵乘法表示：



-

$$(F_{x_1x_2,y_1y_2})(H_{y_1y_2,z_1z_2z_3})$$

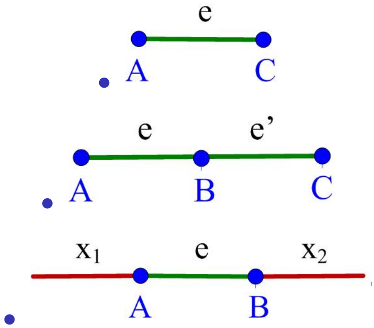
- 遵循行（列）标的变量边画在左（右）边。矩阵转置后怎么画？

张量网络例子:向量矩阵乘法

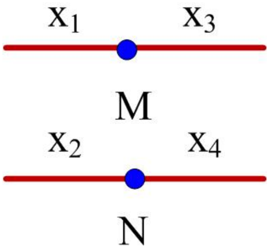
$AC = \sum_{e \in [d]} A_e C_e$

$ABC = \sum_{e,e' \in [d]} A_e B_{ee'} C_{e'}$

$AB_{x_1,x_2} = \sum_{e \in [d]} A_{x_1 e} B_{e x_2}$

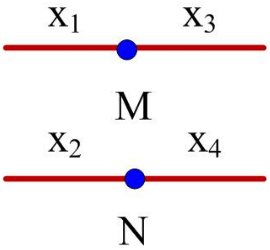


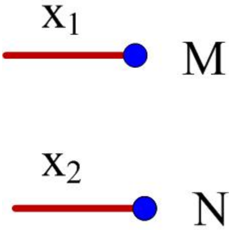
张量积



$$(M \otimes N)_{x_1 x_2, x_3 x_4} = M_{x_1, x_3} N_{x_2, x_4}$$

张量积



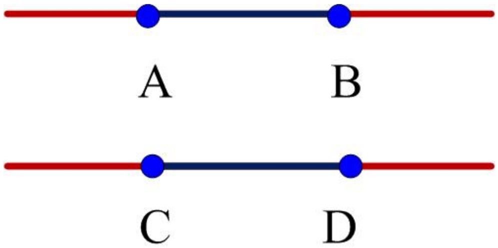


$$(M\otimes N)_{x_1x_2,x_3x_4} = M_{x_1,x_3}N_{x_2,x_4}$$

$$(M \otimes N)_{x_1,x_2} = M_{x_1}N_{x_2}$$

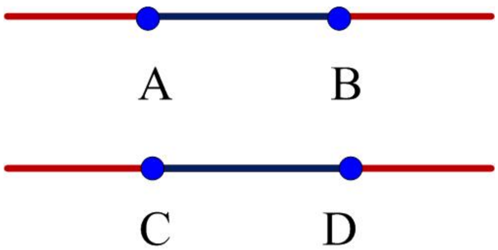
（记法$M^{\otimes 3} = M \otimes M \otimes M$）。

张量积与矩阵乘法



除了定义外，怎样用矩阵运算写出此张量网络的函数？

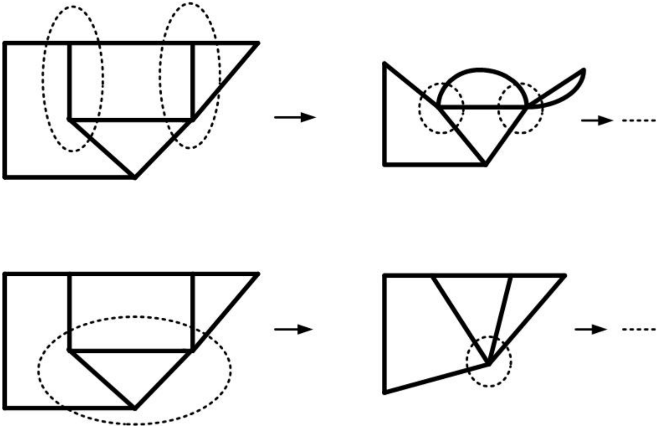
张量积与矩阵乘法



除了定义外，怎样用矩阵运算写出此张量网络的函数？

$$(A \cdot B) \otimes (C \cdot D) \qquad \text{or} \qquad (A \otimes C) \cdot (B \otimes D)$$

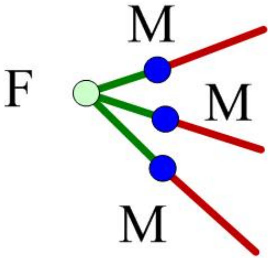
定义的不同嵌套次序，结果总相同



称之为张量网络的结合律。

矩阵乘法和张量积的联合使用例子

$F(M^{\otimes 3})$



零元函数的张量积

$$(M \otimes N) = MN$$

推论

一个无外部边的张量网络的值，是它各个连通分支的值的乘积。

Proof.

1、定义直接证明。

2、先用张量网络的结合律，把每个连通分支缩成点，然后零元函数的张量积。 □

布尔变量对称函数的表示

- F是对称函数，当且仅当对任意的置换π，任意$x_1, \ldots, x_n$，

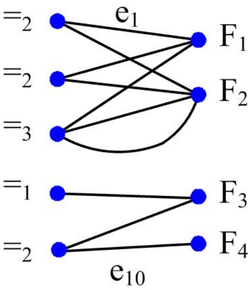
$$F(x_1, \ldots, x_n) = F(x_{\pi(1)}, \ldots, x_{\pi(n)})$$

- 布尔变量$x_j \in \{0, 1\}$。
- F值取决于输入中有多少个0和1。
- f_j 表示输入中有j个1时的F值，$j = 0, 1, \ldots, n$。
- 记

$$F = [f_0, f_1, \ldots, f_n]$$

#CSP问题与张量网络

- $=_k$表示k元相等关系函数。它要求所有输入变量的值相同。
- 布尔变量时，“$=_k$”$= [1, 0, \ldots, 0, 1]$。
- 对一个#CSP问题实例的图稍加改造，得到如下张量网络。

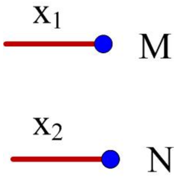
- 

- 实例的答案就是这个张量网络的值。
- 不连通时，可先计算每个连通分支。（product type的算法）

图同态数目问题的一个易解类

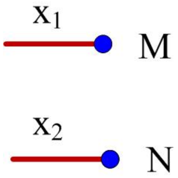
- 图H同态数目问题，问输入图G到H的同态映射数目。
- 就是一个二元函数H定义的#CSP问题。
- 如果二元函数H的矩阵形式的秩小于等于1，有多项式时间算法。
  （值域非负实数时，假设H联通，这是二分定理的一个易解类。）

H秩为1时的算法



- $(M \otimes N)_{x_1, x_2} = M_{x_1} N_{x_2}$ or MN'

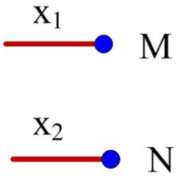
H秩为1时的算法



$$(M \otimes N)_{x_1, x_2} = M_{x_1} N_{x_2} \text{ or } MN'$$

- 因为H秩为1，设$H = ab'$。

H秩为1时的算法



$$(M \otimes N)_{x_1,x_2} = M_{x_1} N_{x_2} \text{ or } MN'$$

- 因为H秩为1，设$H = ab'$。

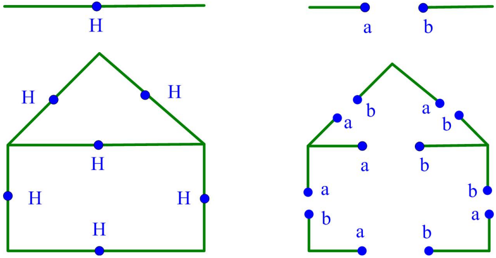


Figure: 作为输入的张量网络的两种等价形式

计算一个星

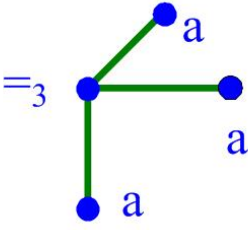


Figure: 每个连通分支是一个星

- 中心点函数是相等函数，只有两个赋值可以满足它。

图同态数目问题的另一个易解类简介

- H是一个偶图，其邻接矩阵是

$$\begin{pmatrix} \mathbf{0} & A \\ B & \mathbf{0} \end{pmatrix}$$

- A和B的秩是1。
- 算法：
  - 如果G不是偶图，不存在从G到H的同态映射。
  - 如果G是偶图，G左顶点集合映射到H的左顶点集合或者右顶点集合。
    两种情况之后的计算，都和H秩为1时的算法相同。

迹

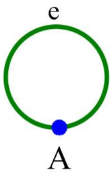


Figure: $\sum_e A_{e,e}$

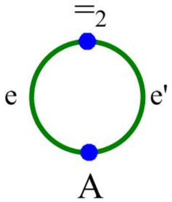


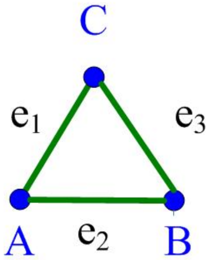
Figure: $\sum_{e,e'} A(e,e')\text{“}=_2\text{”}(e',e)$

迹

- $$\mathrm{trace}(ABC) = \sum_{e_1, e_2, e_3 \in [d]} A_{e_1 e_2} B_{e_2 e_3} C_{e_3 e_1}$$

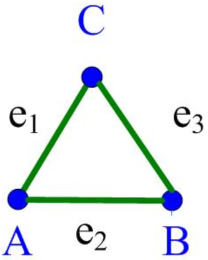
迹

- $$\text{trace}(ABC) = \sum_{e_1,e_2,e_3\in[d]} A_{e_1e_2}B_{e_2e_3}C_{e_3e_1}$$

- 

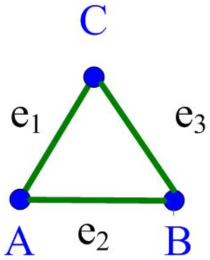
迹

- $$\mathrm{trace}(ABC) = \sum_{e_1,e_2,e_3\in[d]} A_{e_1e_2}B_{e_2e_3}C_{e_3e_1}$$

- 

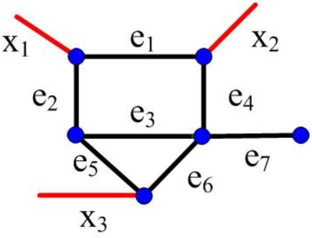
- 相同的张量网络图，
  不同的画法可表示$\mathrm{trace}(BCA)$和$\mathrm{trace}(C'B'A')$。

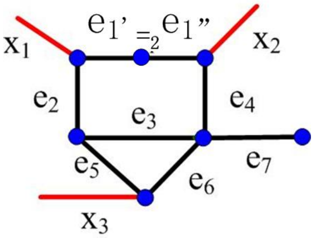
- $$\text{trace}(ABC) = \sum_{e_1,e_2,e_3\in[d]} A_{e_1e_2}B_{e_2e_3}C_{e_3e_1}$$

- 

- 相同的张量网络图，
  不同的画法可表示$\text{trace}(BCA)$和$\text{trace}(C'B'A')$。
- 量子物理里用到partial trace。

边与二元相等函数





一条边实际上是一个出现两次的变量，
在其两个端点的函数中各出现一次，
也等价于一个二元相等函数“$=_2$”。

不带外部边的张量网络与计数问题

- 一个函数都来自一个集合$\mathcal{F}$的张量网络求值问题，记为$\#\mathcal{F}$。（叫做Holant问题，或者Read-twice #CSP问题）
- #CSP中，变量可以使用多次。
  Holant中，变量（一条边）只能用两次。

不带外部边的张量网络与计数问题

- 一个函数都来自一个集合$\mathcal{F}$的张量网络求值问题，记为$\#\mathcal{F}$。（叫做Holant问题，或者Read-twice #CSP问题）
- #CSP中，变量可以使用多次。
  Holant中，变量（一条边）只能用两次。

取定一个函数集合$\mathcal{F}$。

- $\#\mathcal{F}$的实例也是$\#\text{CSP}(\mathcal{F})$的实例。
- 前者可归约到后者。所以，如果前者#P难，后者#P难；如果后者有算法，前者有算法。

不带外部边的张量网络与计数问题

- 一个函数都来自一个集合$\mathcal{F}$的张量网络求值问题，记为$\#\mathcal{F}$。（叫做Holant问题，或者Read-twice #CSP问题）
- #CSP中，变量可以使用多次。
  Holant中，变量（一条边）只能用两次。

取定一个函数集合$\mathcal{F}$。

- $\#\mathcal{F}$的实例也是$\#\text{CSP}(\mathcal{F})$的实例。
- 前者可归约到后者。所以，如果前者#P难，后者#P难；如果后者有算法，前者有算法。

$\mathcal{F}$作为参数，定义两个问题集合：
Holant问题类$\{\#\mathcal{F}\}$，和#CSP问题类$\{\#CSP(\mathcal{F})\}$

- 后者是前者的子集。
- $\#\text{CSP}(\mathcal{F}) = \#\mathcal{F} \cup \{=_1, =_2, \ldots, =_k, \ldots,\}$。

计数问题的偶图输入

- #CSP问题和Holant问题的输入，都可以用偶图张量网络表示。
- #CSP($\mathcal{F}$)的实例，表示成偶图张量网络，左侧顶点的函数来自$\{=_j | j \in \mathbb{N}\}$，右侧顶点的函数都来自$\mathcal{F}$。
  $\#\{=_j | j \in \mathbb{N}\} | \mathcal{F}$
- #$\mathcal{F}$的实例，表示成偶图张量网络，左侧顶点的函数是$=_2$，右侧顶点的函数都来自$\mathcal{F}$。
  $\#\{=_2\} | \mathcal{F}$
- 不同计数问题类的要求不一样，例如，天生自带（免费使用）的函数不同，图的要求（平面图、偶图等）不同。

构件与张量网络

计数问题的偶图输入

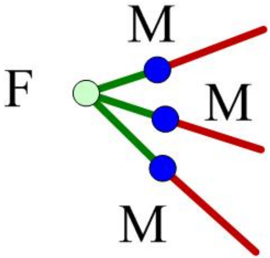
- #CSP问题和Holant问题的输入，都可以用偶图张量网络表示。
- #CSP($\mathcal{F}$)的实例，表示成偶图张量网络，左侧顶点的函数来自$\{=_j |j \in \mathbb{N}\}$，右侧顶点的函数都来自$\mathcal{F}$。
  $\#\{=_j |j \in \mathbb{N}\}|\mathcal{F}$
- #$\mathcal{F}$的实例，表示成偶图张量网络，左侧顶点的函数是$=_2$，右侧顶点的函数都来自$\mathcal{F}$。
  $\#\{=_2\}|\mathcal{F}$
- 不同计数问题类的要求不一样，例如，天生自带（免费使用）的函数不同，图的要求（平面图、偶图等）不同。

构件与张量网络

- 最朴素的计数问题$A \le B$归约方法，就是构造B中的构件（Gadget,即张量网络）模拟A中的函数。原因是结合律。
- 平行的，把计数问题换成判定问题，张量网络中的$\sum\prod$换成$\bigvee\bigwedge$，也有归约，也有结合律。

回顾一个张量网络

$F(M^{\otimes 3})$

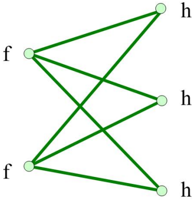


全息归约：Holant定理

$AB = AEB = AMM^{-1}B = (AM)(M^{-1}B)$。($E$是单位阵)

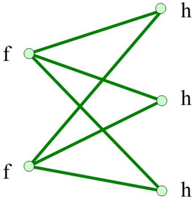
全息归约：Holant定理

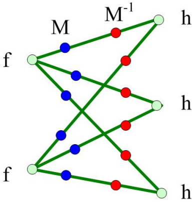
$AB = AEB = AMM^{-1}B = (AM)(M^{-1}B)$。($E$是单位阵)



全息归约：Holant定理

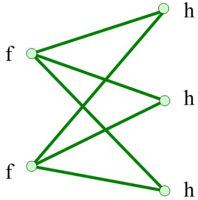
$AB = AEB = AMM^{-1}B = (AM)(M^{-1}B)$。($E$是单位阵)

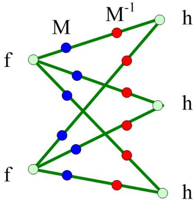


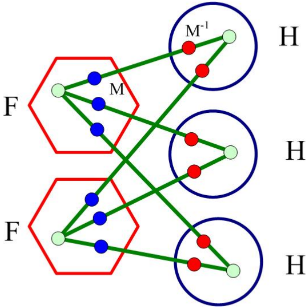


全息归约：Holant定理

$AB = AEB = AMM^{-1}B = (AM)(M^{-1}B)$。($E$是单位阵)

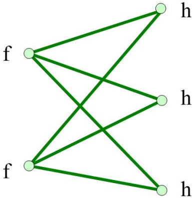


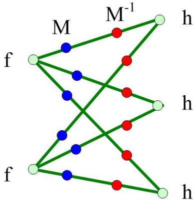


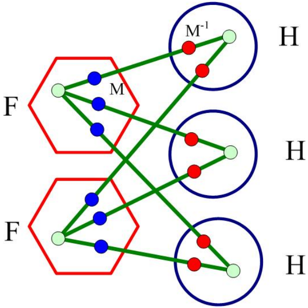


全息归约：Holant定理

$AB = AEB = AMM^{-1}B = (AM)(M^{-1}B)$。($E$是单位阵)







定理 (Valiant 2004)

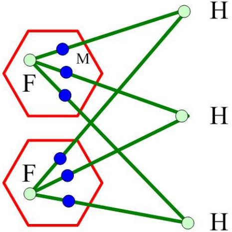
$\#\{F\}|\{G\}$和$\#\{f\}|\{g\}$在相同的图上的值相等。其中，

$$F = fM^{\otimes 3},$$

$$(M^{-1})^{\otimes 2}h = H。$$

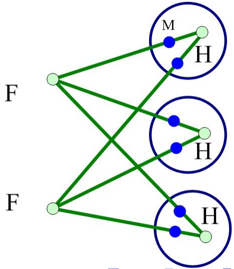
全息归约另一个一般形式

类比$(AB)C = A(BC)$。



定理

$\#\{FM^{\otimes 3}\}|\{H\}$和$\#\{F\}|\{M^{\otimes 2}H\}$值相同。



回顾——用图证明代数运算律

- 矩阵乘法和张量积的结合律，它们之间的分配律。
- 迹与矩阵乘法的律。
- 秩为1的矩阵，列向量乘行向量。（用于解释图同态易解类）
- 全息归约。（张量网络中的基变换）

- Holan与#CSP问题
- 简单情形的图同态二分定理易解情况

参考文献

- Matthew Cook, Networks of Relations, Ph.D Thesis 2005. (判定问题)
- http://arxiv.org/abs/1603.03039
- http://arxiv.org/abs/1306.2164
- https://simons.berkeley.edu/workshops/qhc2014-3 (workshop "Tensor Networks and Simulations", in simons institute for the theory of computing)