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

2019

张量网络

- 图$G(V,E)$的每个点v被赋予一个d_v元函数F_v，d_v是v的度。
- 有限集D表示F_v的一个自变量的定义域。
- 边集合$E=\{e_1,\ldots,e_m\}$。
- 张量网络的值定义为：

$$\sum_{e_1,\ldots,e_m\in D}\prod_{v\in V}F_v(e_{v,1},e_{v,2},\ldots,e_{v,d_v})$$

其中$e_{v,1},e_{v,2},\ldots,e_{v,d_v}$表示$v$的$d_v$条边。

张量网络

- 图$G(V,E)$的每个点v被赋予一个d_v元函数F_v，d_v是v的度。
- 有限集D表示F_v的一个自变量的定义域。
- 边集合$E=\{e_1,\ldots,e_m\}$。
- 张量网络的值定义为：

$$\sum_{e_1,\ldots,e_m\in D}\prod_{v\in V}F_v(e_{v,1},e_{v,2},\ldots,e_{v,d_v})$$

其中$e_{v,1},e_{v,2},\ldots,e_{v,d_v}$表示$v$的$d_v$条边。

- 其他表示:

$$\sum_{\sigma:E\to D}\prod_{v\in V}F_v(\sigma(e_{v,1}),\sigma(e_{v,2}),\ldots,\sigma(e_{v,d_v}))$$

$$\sum_{\sigma:E\to D}\prod_{v\in V}F_v(\sigma|_{Neighbor(v)})$$

张量网络

- 边是变量，点是函数，点v被赋予函数F_v。
- X是外部边集合，E是内部边集合。
- 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$F_G$。

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

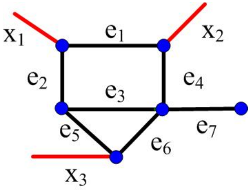


Figure: 图$G(V, E \cup X)$

张量网络

- 第二定义更一般：无外部边时，是个0元函数，定义式也给了$2^0 = 1$个值，就是之前的封闭张量网络的值。

张量网络

- 第二定义更一般：无外部边时，是个0元函数，定义式也给了$2^0 = 1$个值，就是之前的封闭张量网络的值。
- 定义域D只是个起标记作用的符号的集合，
例如，D取$[3] = \{0, 1, 2\}$或$\{a, b, c\}$或$\{R, B, G\}$都一样。

张量网络

- 第二定义更一般：无外部边时，是个0元函数，定义式也给了$2^0 = 1$个值，就是之前的封闭张量网络的值。
- 定义域D只是个起标记作用的符号的集合，
  例如，D取$[3] = \{0, 1, 2\}$或$\{a, b, c\}$或$\{R, B, G\}$都一样。
- 张量网络是多种定义的部分本质：

张量网络

- 第二定义更一般：无外部边时，是个0元函数，定义式也给了$2^0 = 1$个值，就是之前的封闭张量网络的值。
- 定义域D只是个起标记作用的符号的集合，例如，D取$[3] = \{0, 1, 2\}$或$\{a, b, c\}$或$\{R, B, G\}$都一样。
- 张量网络是多种定义的部分本质：
  - 因子图、布尔线路，是其特殊情况。

张量网络

- 第二定义更一般：无外部边时，是个0元函数，定义式也给了$2^0 = 1$个值，就是之前的封闭张量网络的值。
- 定义域D只是个起标记作用的符号的集合，例如，D取$[3] = \{0, 1, 2\}$或$\{a, b, c\}$或$\{R, B, G\}$都一样。
- 张量网络是多种定义的部分本质：
  - 因子图、布尔线路，是其特殊情况。
  - 量子线路要加入测量等概念和约束。

张量网络

- 第二定义更一般：无外部边时，是个0元函数，定义式也给了$2^0 = 1$个值，就是之前的封闭张量网络的值。
- 定义域D只是个起标记作用的符号的集合，例如，D取$[3] = \{0, 1, 2\}$或$\{a, b, c\}$或$\{R, B, G\}$都一样。
- 张量网络是多种定义的部分本质：
  - 因子图、布尔线路，是其特殊情况。
  - 量子线路要加入测量等概念和约束。
  - 贝叶斯网络、Gibbs分布、爱因斯坦标记（求和约定）等。

张量网络

- 第二定义更一般：无外部边时，是个0元函数，定义式也给了$2^0 = 1$个值，就是之前的封闭张量网络的值。
- 定义域D只是个起标记作用的符号的集合，
例如，D取$[3] = \{0,1,2\}$或$\{a,b,c\}$或$\{R,B,G\}$都一样。
- 张量网络是多种定义的部分本质：
    - 因子图、布尔线路，是其特殊情况。
    - 量子线路要加入测量等概念和约束。
    - 贝叶斯网络、Gibbs分布、爱因斯坦标记（求和约定）等。
    - 实际上就是构件（gadget），常见于计数问题之间的归约中。

张量网络

- 第二定义更一般：无外部边时，是个0元函数，定义式也给了$2^0 = 1$个值，就是之前的封闭张量网络的值。
- 定义域D只是个起标记作用的符号的集合，
  例如，D取$[3] = \{0, 1, 2\}$或$\{a, b, c\}$或$\{R, B, G\}$都一样。
- 张量网络是多种定义的部分本质：
  - 因子图、布尔线路，是其特殊情况。
  - 量子线路要加入测量等概念和约束。
  - 贝叶斯网络、Gibbs分布、爱因斯坦标记（求和约定）等。
  - 实际上就是构件（gadget），常见于计数问题之间的归约中。
- 很多线性代数运算是其特殊情况。

张量网络

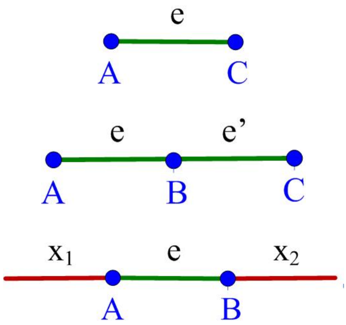
- 第二定义更一般：无外部边时，是个0元函数，定义式也给了$2^0=1$个值，就是之前的封闭张量网络的值。
- 定义域D只是个起标记作用的符号的集合，例如，D取$[3]=\{0,1,2\}$或$\{a,b,c\}$或$\{R,B,G\}$都一样。
- 张量网络是多种定义的部分本质：
  - 因子图、布尔线路，是其特殊情况。
  - 量子线路要加入测量等概念和约束。
  - 贝叶斯网络、Gibbs分布、爱因斯坦标记（求和约定）等。
  - 实际上就是构件（gadget），常见于计数问题之间的归约中。
- 很多线性代数运算是其特殊情况。
- 回顾一对概念的等同：二元函数$F(i,j)$即矩阵$(F_{i,j})$。类似的，一元函数$F(i)$即向量(F_i)。

张量网络例子:向量矩阵乘法

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

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

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



张量积定义

- 两个矩阵$\mathbf{M}_{dd}, \mathbf{N}_{dd}$，其中

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1d} \\ m_{21} & m_{22} & \dots & m_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ m_{d1} & m_{d2} & \dots & m_{dd} \end{pmatrix}$$

张量积定义

- 两个矩阵$\mathbf{M}_{dd},\mathbf{N}_{dd}$，其中

$$\mathbf{M}=\begin{pmatrix} m_{11} & m_{12} & \dots & m_{1d} \\ m_{21} & m_{22} & \dots & m_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ m_{d1} & m_{d2} & \dots & m_{dd} \end{pmatrix}$$

- 它们的张量积是一个$d^2\times d^2$矩阵，有如下分块矩阵形式：

$$\mathbf{M}\otimes\mathbf{N}=\begin{pmatrix} m_{11}N & m_{12}N & \dots & m_{1d}N \\ m_{21}N & m_{22}N & \dots & m_{1d}N \\ \vdots & \vdots & \ddots & \vdots \\ m_{d1}N & m_{d2}N & \dots & m_{dd}N \end{pmatrix}$$

张量积定义

- 两个矩阵$\mathbf{M}_{dd}, \mathbf{N}_{dd}$，其中

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1d} \\ m_{21} & m_{22} & \dots & m_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ m_{d1} & m_{d2} & \dots & m_{dd} \end{pmatrix}$$

- 它们的张量积是一个$d^2 \times d^2$矩阵，有如下分块矩阵形式：

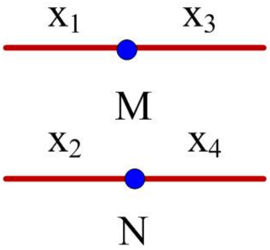
$$\mathbf{M} \otimes \mathbf{N} = \begin{pmatrix} m_{11}N & m_{12}N & \dots & m_{1d}N \\ m_{21}N & m_{22}N & \dots & m_{1d}N \\ \vdots & \vdots & \ddots & \vdots \\ m_{d1}N & m_{d2}N & \dots & m_{dd}N \end{pmatrix}$$

- 可用$\mathbf{M}$的行指标变量x_1和$\mathbf{N}$的行指标变量x_2联合起来的$x_1x_2 \in [d^2]$作为$\mathbf{M} \otimes \mathbf{N}$ 的行指标。

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

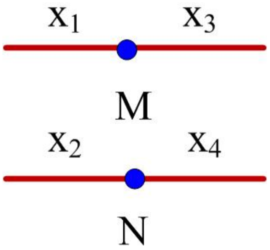
（x_1x_2是个符号二元组，不是数字乘法。行指标只是标记

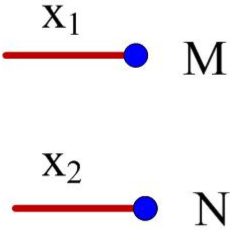
张量积



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

张量积





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

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

广义矩阵形式（行标朝左、列标朝右）

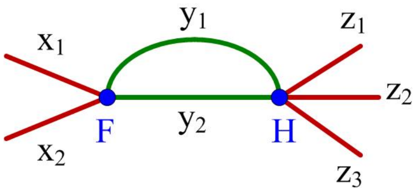
- 设F是一个$n+m$元布尔函数，$x_1,\dots,x_n,y_1,\dots,y_m$是它的输入。
  对应$2^n\times 2^m$的矩阵$M=(M_{x_1\cdots x_n,y_1\cdots y_m})$，其中
  $$M_{x_1\cdots x_n,y_1\cdots y_m}=F(x_1,\dots,x_n,y_1,\dots,y_m)。$$
  若取$m=0$（$n=0$），就成了列（行）向量。

广义矩阵形式（行标朝左、列标朝右）

- 设F是一个$n+m$元布尔函数，$x_1,\ldots,x_n,y_1,\ldots,y_m$是它的输入。
  对应$2^n \times 2^m$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：

广义矩阵形式（行标朝左、列标朝右）

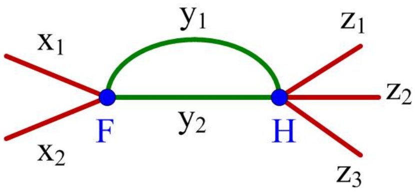
- 设F是一个$n+m$元布尔函数，$x_1,\ldots,x_n,y_1,\ldots,y_m$是它的输入。
  对应$2^n\times 2^m$的矩阵$M=(M_{x_1\cdots x_n,y_1\cdots y_m})$，其中
  $$M_{x_1\cdots x_n,y_1\cdots y_m}=F(x_1,\ldots,x_n,y_1,\ldots,y_m)。$$
  若取$m=0$（$n=0$），就成了列（行）向量。
- 如下张量网络的函数可用矩阵乘法表示：



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

广义矩阵形式（行标朝左、列标朝右）

- 设F是一个$n+m$元布尔函数，$x_1,\ldots,x_n,y_1,\ldots,y_m$是它的输入。
  对应$2^n\times 2^m$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：



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

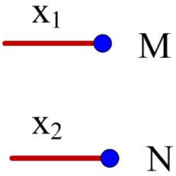
- 遵循行（列）标的变量边画在左（右）边。
  转置矩阵怎么画？

向量的张量积与矩阵乘法

$$(MN')_{x_1,x_2} = M_{x_1}N_{x_2}$$

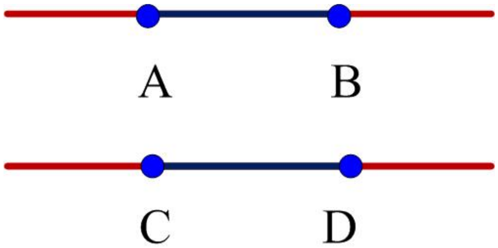
向量的张量积与矩阵乘法

$$(MN')_{x_1,x_2} = M_{x_1}N_{x_2}$$



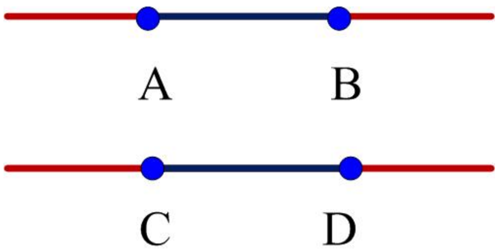
$$(M \otimes N)_{x_1,x_2} = M_{x_1}N_{x_2}$$

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？

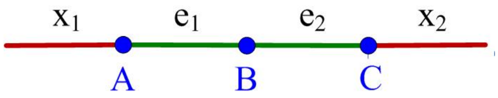
张量积与矩阵乘法



除了定义外，怎样用矩阵运算写出此张量网络的函数？

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

结合律的证明



此张量网络定义了矩阵$F = ABC$。如下三式等价。

- 按照张量网络的定义

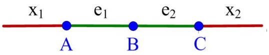
$$F(x_1, x_2) = \sum_{e_1, e_2} A(x_1, e_1)B(e_1, e_2)C(e_2, x_2)$$

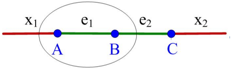
- 按照矩阵乘法依次乘，A, B先算,$\sum_{e_1} A(x_1, e_1)B(e_1, e_2)$。

$$F(x_1, x_2) = \sum_{e_2}(C(e_2, x_2)\sum_{e_1} A(x_1, e_1)B(e_1, e_2))$$

- 先乘B, C，$\sum_{e_1}(A(x_1, e_1)\sum_{e_2} B(e_1, e_2)C(e_2, x_2))$

结合律

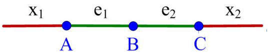


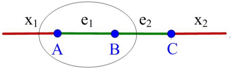


$$\sum_{e_1,e_3} A(x_1,e_1)B(e_1,e_2)C(e_2,x_2)$$

$$\sum_{e_2}(C(e_2,x_2)\sum_{e_1} A(x_1,e_1)B(e_1,e_2))$$

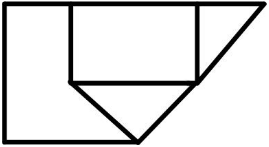
结合律

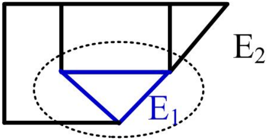




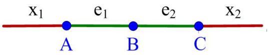
$$\sum_{e_1,e_3} A(x_1,e_1)B(e_1,e_2)C(e_2,x_2)$$

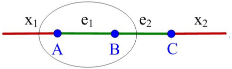
$$\sum_{e_2}(C(e_2,x_2)\sum_{e_1} A(x_1,e_1)B(e_1,e_2))$$





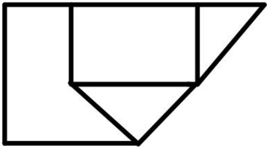
结合律

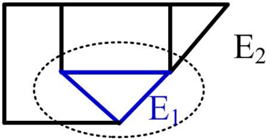




$$\sum_{e_1,e_3} A(x_1,e_1)B(e_1,e_2)C(e_2,x_2)$$

$$\sum_{e_2}(C(e_2,x_2)\sum_{e_1} A(x_1,e_1)B(e_1,e_2))$$

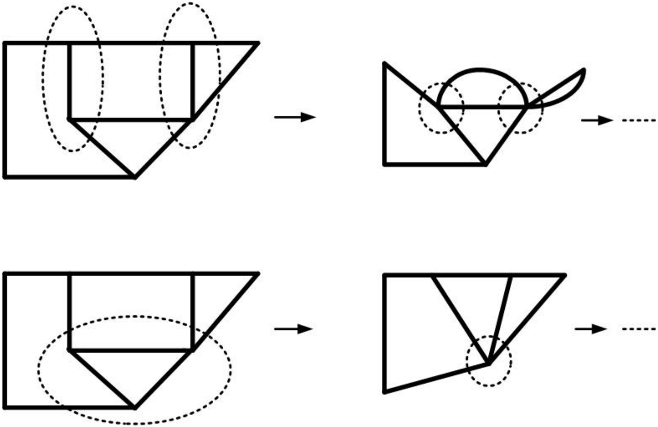




$$\sum_{e\in E_1\cup E_2}\prod_{v\in V_1\cup V_2} F_v$$

$$\sum_{e\in E_2}(\prod_{v\in V_2} F_v \sum_{e\in E_1}\prod_{v\in V_1} F_v)$$

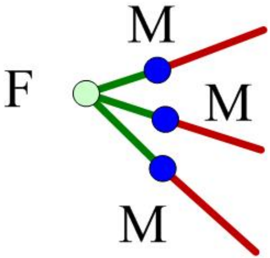
定义的不同嵌套次序，结果总相同



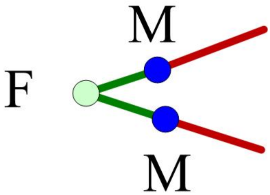
称之为张量网络的结合律。

矩阵乘法和张量积的联合使用例子

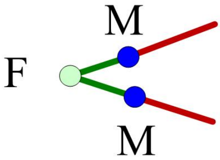
$F \cdot M^{\otimes 3}$



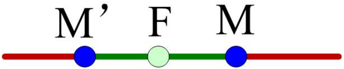
$F \cdot M^{\otimes 2}$



$F \cdot M^{\otimes 2}$



$M' F M$



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$x_1, \ldots, x_n$，

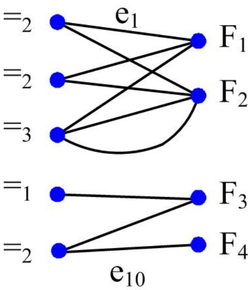
$$F(x_1, \ldots, x_n) = F(x_{\pi(1)}, \ldots, x_{\pi(n)})$$

- 布尔变量$x_j \in \{0, 1\}$。
- F值取决于输入中有多少个0和1。
- 用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。

布尔变量#CSP问题的第二易解类Product Type回顾

- $=_k$表示$[1, 0, \ldots, 0, 1]$，是$[a, 0, \ldots, 0, b]$函数类的子集。

布尔变量#CSP问题的第二易解类Product Type回顾

- $=_k$表示$[1,0,\ldots,0,1]$，是$[a,0,\ldots,0,b]$函数类的子集。
- $\neq_2$表示二元不等关系$[0,1,0]$。

布尔变量#CSP问题的第二易解类Product Type回顾

- $=_k$表示$[1,0,\ldots,0,1]$，是$[a,0,\ldots,0,b]$函数类的子集。
- $\neq_2$表示二元不等关系$[0,1,0]$。
- 从一个$[a,0,\ldots,0,b]$函数的外部边中选几个连$\neq_2$函数，得到一个张量网络，其函数就是$\mathcal{E}$中的函数。

布尔变量#CSP问题的第二易解类Product Type回顾

- $=_k$表示$[1,0,\ldots,0,1]$，是$[a,0,\ldots,0,b]$函数类的子集。
- $\neq_2$表示二元不等关系$[0,1,0]$。
- 从一个$[a,0,\ldots,0,b]$函数的外部边中选几个连$\neq_2$函数，得到一个张量网络，其函数就是$\mathcal{E}$中的函数。
- 回忆$F\in\mathcal{E}$当且仅当存在长n的$0,1$串$\alpha\in\{0,1\}^n$，$\forall X, X\notin\{\alpha,\bar{\alpha}\}\Rightarrow F(X)=0$。
  而Product Type中的函数是$\mathcal{E}$中的函数的乘积。

布尔变量#CSP问题的第二易解类Product Type回顾

- $=_k$表示$[1,0,\ldots,0,1]$，是$[a,0,\ldots,0,b]$函数类的子集。
- $\neq_2$表示二元不等关系$[0,1,0]$。
- 从一个$[a,0,\ldots,0,b]$函数的外部边中选几个连$\neq_2$函数，得到一个张量网络，其函数就是$\mathcal{E}$中的函数。
- 回忆$F\in\mathcal{E}$当且仅当存在长n的$0,1$串$\alpha\in\{0,1\}^n$，$\forall X, X\notin\{\alpha,\bar{\alpha}\}\Rightarrow F(X)=0$。
  而Product Type中的函数是$\mathcal{E}$中的函数的乘积。
- #CSP($\mathcal{P}$) 即计算$\{=_k\}$和$\mathcal{P}$构成的张量网络，等价于

布尔变量#CSP问题的第二易解类Product Type回顾

- $=_k$表示$[1,0,\dots,0,1]$，是$[a,0,\dots,0,b]$函数类的子集。
- $\neq_2$表示二元不等关系$[0,1,0]$。
- 从一个$[a,0,\dots,0,b]$函数的外部边中选几个连$\neq_2$函数，得到一个张量网络，其函数就是$\mathcal{E}$中的函数。
- 回忆$F\in\mathcal{E}$当且仅当存在长n的$0,1$串$\alpha\in\{0,1\}^n$，$\forall X, X\notin\{\alpha,\bar{\alpha}\}\Rightarrow F(X)=0$。
  而Product Type中的函数是$\mathcal{E}$中的函数的乘积。
- #CSP($\mathcal{P}$) 即计算$\{=_k\}$和$\mathcal{P}$构成的张量网络，等价于
- 计算$\{=_k\}$和$\mathcal{E}$构成的张量网络，等价于

布尔变量#CSP问题的第二易解类Product Type回顾

- $=_k$表示$[1,0,\ldots,0,1]$，是$[a,0,\ldots,0,b]$函数类的子集。
- $\neq_2$表示二元不等关系$[0,1,0]$。
- 从一个$[a,0,\ldots,0,b]$函数的外部边中选几个连$\neq_2$函数，得到一个张量网络，其函数就是$\mathcal{E}$中的函数。
- 回忆$F\in\mathcal{E}$当且仅当存在长n的$0,1$串$\alpha\in\{0,1\}^n$，$\forall X, X\notin\{\alpha,\bar{\alpha}\}\Rightarrow F(X)=0$。
  而Product Type中的函数是$\mathcal{E}$中的函数的乘积。
- #CSP($\mathcal{P}$) 即计算$\{=_k\}$和$\mathcal{P}$构成的张量网络，等价于
- 计算$\{=_k\}$和$\mathcal{E}$构成的张量网络，等价于
- 计算$\{[a,0,\ldots,0,b]\}$和$[0,1,0]$构成的张量网络。

布尔变量#CSP问题的第二易解类Product Type回顾

- $=_k$表示$[1,0,\ldots,0,1]$，是$[a,0,\ldots,0,b]$函数类的子集。
- $\neq_2$表示二元不等关系$[0,1,0]$。
- 从一个$[a,0,\ldots,0,b]$函数的外部边中选几个连$\neq_2$函数，得到一个张量网络，其函数就是$\mathcal{E}$中的函数。
- 回忆$F\in\mathcal{E}$当且仅当存在长n的0,1串$\alpha\in\{0,1\}^n$，$\forall X, X\notin\{\alpha,\bar{\alpha}\}\Rightarrow F(X)=0$。
  而Product Type中的函数是$\mathcal{E}$中的函数的乘积。
- #CSP($\mathcal{P}$) 即计算$\{=_k\}$和$\mathcal{P}$构成的张量网络，等价于
- 计算$\{=_k\}$和$\mathcal{E}$构成的张量网络，等价于
- 计算$\{[a,0,\ldots,0,b]\}$和$[0,1,0]$构成的张量网络。
- $\{[a,0,\ldots,0,b]\}$和$[0,1,0]$构成的连通的张量网络，至多只有两个变量赋值对应的乘积项（被求和项）非零。

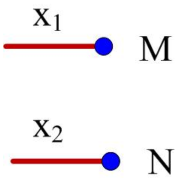
布尔变量#CSP问题的第二易解类Product Type回顾

- $=_k$表示$[1,0,\ldots,0,1]$，是$[a,0,\ldots,0,b]$函数类的子集。
- $\neq_2$表示二元不等关系$[0,1,0]$。
- 从一个$[a,0,\ldots,0,b]$函数的外部边中选几个连$\neq_2$函数，得到一个张量网络，其函数就是$\mathcal{E}$中的函数。
- 回忆$F\in\mathcal{E}$当且仅当存在长n的$0,1$串$\alpha\in\{0,1\}^n$，
  $\forall X, X\notin\{\alpha,\bar{\alpha}\}\Rightarrow F(X)=0$。
  而Product Type中的函数是$\mathcal{E}$中的函数的乘积。
- #CSP($\mathcal{P}$) 即计算$\{=_k\}$和$\mathcal{P}$构成的张量网络，等价于
- 计算$\{=_k\}$和$\mathcal{E}$构成的张量网络，等价于
- 计算$\{[a,0,\ldots,0,b]\}$和$[0,1,0]$构成的张量网络。
- $\{[a,0,\ldots,0,b]\}$和$[0,1,0]$构成的连通的张量网络，至多只有两个变量赋值对应的乘积项（被求和项）非零。
- 总结：找$\mathcal{P}$的生成元，然后只讨论生成元的张量网络算法。

图同态数目问题的一个易解类

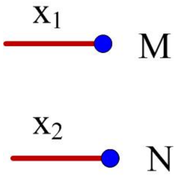
- 图H同态数目问题，问输入图G到H的同态映射数目。
- 就是一个二元函数H定义的#CSP问题。
- 如果二元函数H的矩阵形式的秩小于等于1，有多项式时间算法。
  （值域非负实数时，假设H联通，这是二分定理的一个易解类。）

H秩为1时的算法



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

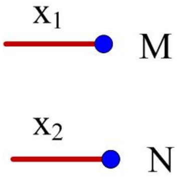
H秩为1时的算法



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

- 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$H = ab'$。

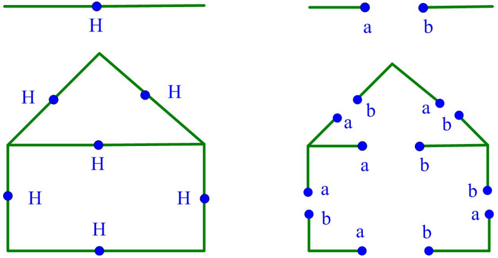


Figure: 作为输入的张量网络的两种等价形式

计算一个星

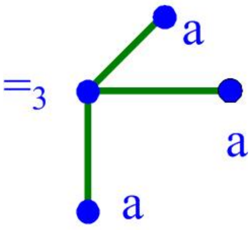


Figure: 每个连通分支是一个星

- 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

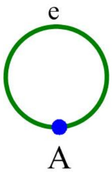


Figure: $\sum_e A_{e,e}$

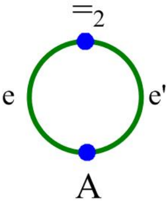


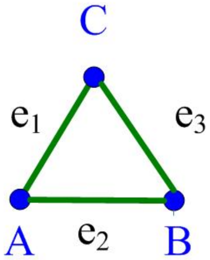
Figure: $\sum_{e,e'} A(e,e')\text{“}=_2\text{”}(e',e)$

迹

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

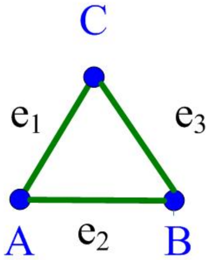
迹

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



迹

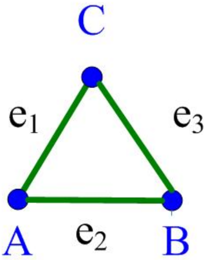
- $$\mathrm{trace}(ABC) = \sum_{e_1,e_2,e_3\in[d]} A_{e_1e_2}B_{e_2e_3}C_{e_3e_1}$$



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

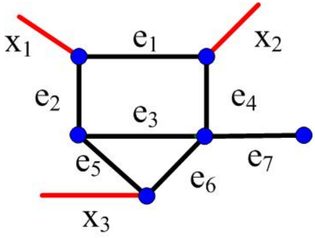
迹

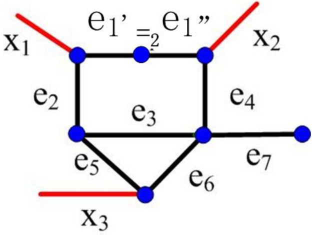
- $$\mathrm{trace}(ABC) = \sum_{e_1,e_2,e_3\in[d]} A_{e_1e_2}B_{e_2e_3}C_{e_3e_1}$$



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

定义两个问题集合：
Holant问题类$\{\#\mathcal{F}|\mathcal{F}$是个函数集合$\}$
#CSP问题类$\{\#CSP(\mathcal{F})|\mathcal{F}$是个函数集合$\}$

- 后者是前者的子集。
- #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。

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$\mathcal{F}$。
  $\#\{=_j | j \in \mathbb{N}\}|\mathcal{F}$

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$\mathcal{F}$。
  $\#\{=_2\}|\mathcal{F}$

构件归约

- 最朴素的计数问题$A \leq B$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。

构件归约

- 最朴素的计数问题$A \leq B$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。

构件归约

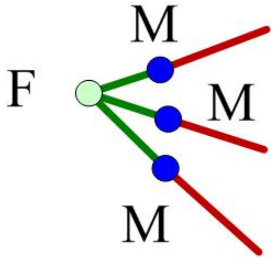
- 最朴素的计数问题$A \le B$归约方法，就是构造B中的构件（Gadget,即张量网络）模拟A中的函数。原因是结合律。
- 平行的，把计数问题换成判定问题，张量网络中的$\sum\prod$换成$\bigvee\bigwedge$，也有归约，也有结合律。

例

在证明$\#\mathit{CSP}(\mathcal{F}) = \#\mathcal{F} \cup \{=_1, =_2, \ldots, =_k, \ldots,\}$时，如果后者中有几个相等函数连接在一起，这个构件的函数还是相等，可以被$\#\mathit{CSP}$中一个变量模拟。

回顾一个张量网络

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

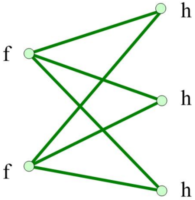


全息归约：Holant定理

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

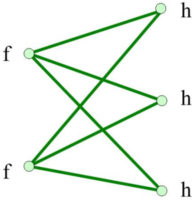
全息归约：Holant定理

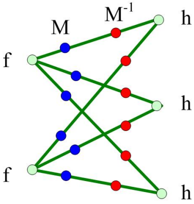
$AB = AEB = AMM^{-1}B = (AM)(M^{-1}B)$。($E$是单位阵)



全息归约：Holant定理

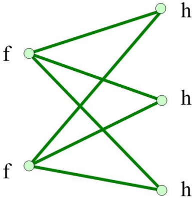
$AB = AEB = AMM^{-1}B = (AM)(M^{-1}B)$。($E$是单位阵)

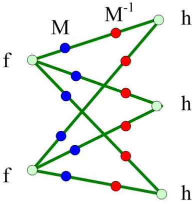


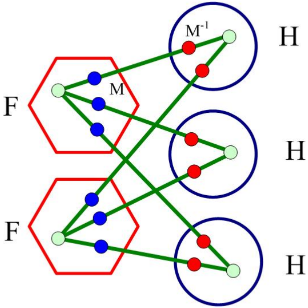


全息归约：Holant定理

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

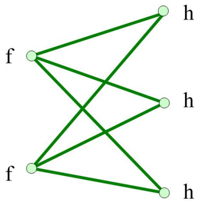


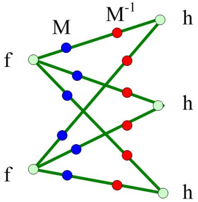


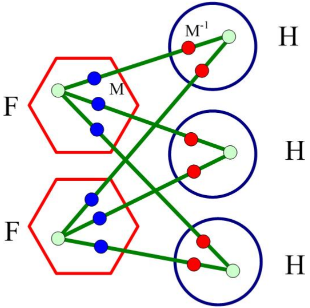


全息归约：Holant定理

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







定理 (Valiant 2004)

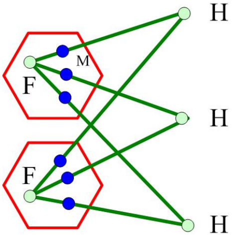
$\#\{F\}|\{H\}$和$\#\{f\}|\{h\}$在相同的图上的值相等。其中，

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

$$(M^{-1})^{\otimes 2}h = H。$$

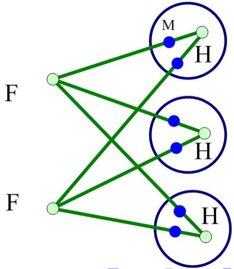
全息归约另一个一般形式

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）

- Holant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)