电子科技大学
University of Electronic Science and Technology of China

最大后验概率推理

牛婷婷&刘沁源

2016/5/25

Data Mining Lab, Big Data Research Center, UESTC
Email：junmshao@uestc.edu.cn
http://staff.uestc.edu.cn/shaojunming

数据挖掘实验室
Data Mining Lab

- 大纲:

1. 综述 – 解决什么样的问题
2. MAP的变量消除 – 如何解决这个问题
3. 团树中的最大积 – 如何快速解决问题
4. 总结

数据挖掘实验室
Data Mining Lab

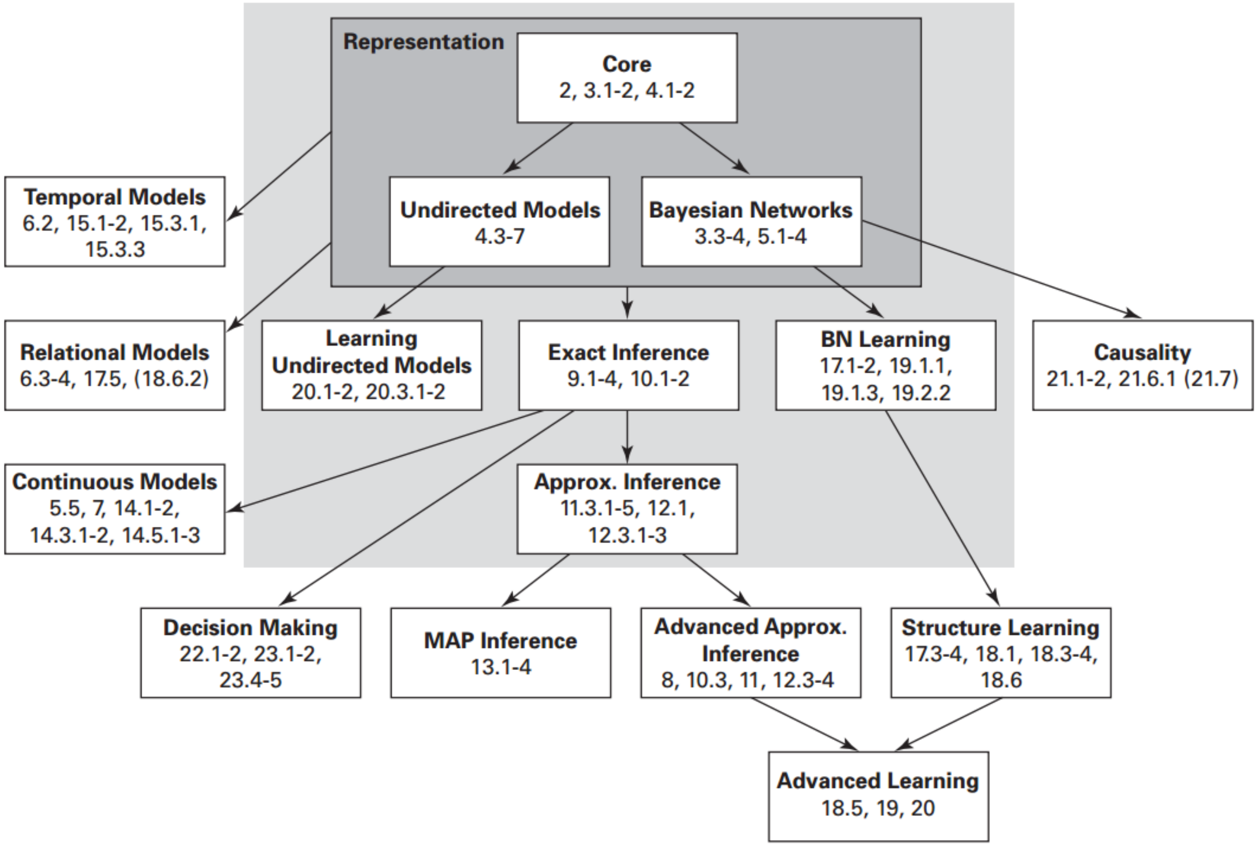


Figure 1.2 A reader's guide to the structure and dependencies in this book

数据挖掘实验室
Data Mining Lab

1. 综述：

- MAP： Maximum a posteriori
- Inference：推理是通过计算回答查询问题
- MAP查询的目的：对所有的变量（非证据）找到最可能的赋值
- 边缘MAP查询的目的：对变量的一个子集找到最可能的赋值，并将其它变量边缘化

数据挖掘实验室
Data Mining Lab

- MAP查询的目标函数：

$$\xi^{map} = \arg\max_{\xi} P_{\Phi}(\xi) = \arg\max_{\xi} \frac{1}{Z} \tilde{P}_{\Phi}(\xi) = \arg\max_{\xi} \tilde{P}_{\Phi}(\xi).$$

- 边缘MAP查询的目标函数：

$$\boldsymbol{y}^{m\text{-}map} = \arg\max_{\boldsymbol{y}} P_{\Phi}(\boldsymbol{y}) = \arg\max_{\boldsymbol{y}} \sum_{\boldsymbol{W}} \tilde{P}_{\Phi}(\boldsymbol{y}, \boldsymbol{W}).$$

- Max-product推理任务(Max-Sum推理任务)
- 任务目标：为得分最大（非归一化概率）的变量集找到一个赋值（最优解问题）

- ***def***：

  1. 最大—边缘：对任意的赋值*y*∈*Val(Y)*，函数*f*相对于变量集*Y*的最大边缘是：

$$MaxMarg_f(\boldsymbol{y}) = \max_{\xi\langle \mathbf{Y} \rangle = \boldsymbol{y}} f(\xi),$$

  2. 解码：将全局最优问题作为个体变量的一系列局部最优问题来解决（分治法的思想）

- 解决方案：

Step 1. 首先为χ中的所有变最计算一个精确的或近似的最大-边缘集

Step 2. 然后尝试从这些最大-边缘中提取一个精确的或近似的 **MAP** 赋值。

- 技巧

1. 第一阶段通常使用诸如在团树或聚类图中变量消除或者消息传递

2. 第二阶段就是回溯，用来确认变量的最优赋值

最基本的推理算法——变量消除

2. MAP的变量消除

- –2.1 最大积变量消除
- –2.2 找到最可能的赋值
- –2.3 边缘MAP的变量消除*

2.1 最大 - 积变量消除

- *def.* 令χ是变量的一个集合，$Y \notin X$是一个变量。令$\phi(X,Y)$是一个因子。Y在ϕ中的因子最大化定义为X上的因子ψ，使得

$$\psi(\boldsymbol{X}) = \max_{Y} \phi(\boldsymbol{X}, Y).$$

- Eg：$\Psi(A,C) = \max_B \phi(A,B,C)$

  注：如果Xi在这个消除过程中是最终的变量，我们已经最大化了除Xi外的所有变量，所以产生的因子是Xi上的**<u>最大 - 边缘</u>**

a^1	b^1	c^1	0.25
a^1	b^1	c^2	0.35
a^1	b^2	c^1	0.08
a^1	b^2	c^2	0.16
a^2	b^1	c^1	0.05
a^2	b^1	c^2	0.07
a^2	b^2	c^1	0
a^2	b^2	c^2	0
a^3	b^1	c^1	0.15
a^3	b^1	c^2	0.21
a^3	b^2	c^1	0.09
a^3	b^2	c^2	0.18

a^1	c^1	0.25
a^1	c^2	0.35
a^2	c^1	0.05
a^2	c^2	0.07
a^3	c^1	0.15
a^3	c^2	0.21

- # 最大-积变量消除算法：

与和积变量消除算法的区别

1.Max-Product

$$
\begin{aligned}
&\Phi' \leftarrow \{\phi \in \Phi : Z \in Scope[\phi]\} \\
&\Phi'' \leftarrow \Phi - \Phi' \\
&\psi \leftarrow \prod_{\phi \in \Phi'} \phi \\
&\tau \leftarrow \max_Z \psi \\
&\textbf{return } (\Phi'' \cup \{\tau\}, \psi)
\end{aligned}
$$

2.Max-Sum

$$
\begin{aligned}
&\Phi' \leftarrow \{\phi \in \Phi : Z \in Scope[\phi]\} \\
&\Phi'' \leftarrow \Phi - \Phi' \\
&\psi \leftarrow \prod_{\phi \in \Phi'} \phi \\
&\tau \leftarrow \sum_Z \psi \\
&\textbf{return } \Phi'' \cup \{\tau\}
\end{aligned}
$$

Algorithm 13.1 Variable elimination algorithm for MAP. The algorithm can be used both in its max-product form, as shown, or in its max-sum form, replacing factor product with factor addition.

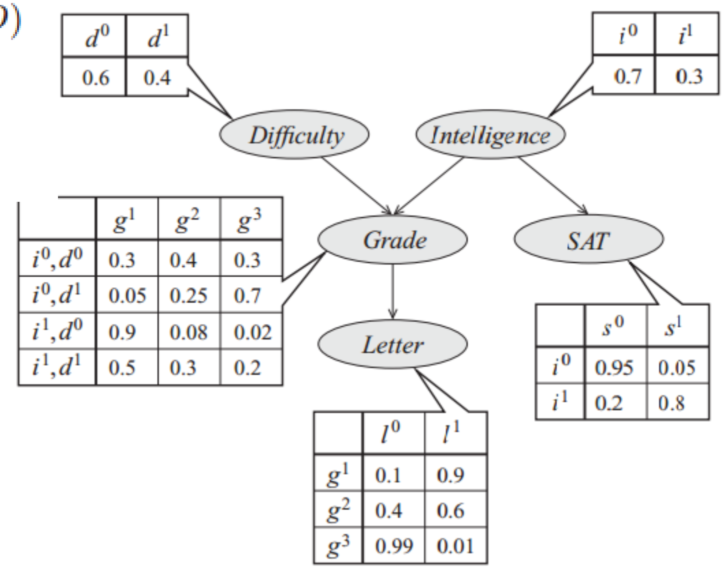
```
Procedure Max-Product-VE (
  Φ,  // Set of factors over X
  ≺   // Ordering on X
)
  Let X_1, ..., X_k be an ordering of X such that
    X_i ≺ X_j iff i < j
  for i = 1, ..., k
    (Φ, φ_{X_i}) ← Max-Product-Eliminate-Var(Φ, X_i)
  x* ← Traceback-MAP({φ_{X_i} : i = 1, ..., k})
  return x*, Φ   // Φ contains the probability of the MAP

Procedure Max-Product-Eliminate-Var (
  Φ,  // Set of factors
  Z   // Variable to be eliminated
)
  Φ' ← {φ ∈ Φ : Z ∈ Scope[φ]}
  Φ'' ← Φ − Φ'
  ψ ← ∏_{φ∈Φ'} φ
  τ ← max_Z ψ
  return (Φ'' ∪ {τ}, ψ)

Procedure Traceback-MAP (
  {φ_{X_i} : i = 1, ..., k}
)
  for i = k, ..., 1
    u_i ← (x*_{i+1}, ..., x*_k)⟨Scope[φ_{X_i}] − {X_i}⟩
      // The maximizing assignment to the variables eliminated after
         X_i
    x*_i ← arg max_{x_i} φ_{X_i}(x_i, u_i)
      // x*_i is chosen so as to maximize the corresponding entry in
         the factor, relative to the previous choices u_i
  return x*
```

- Eg：如图所示的学生网络，目标是在没有证据的情况下，计算整个网络的最可能实例。使用的消除顺序是S,I,D,L,G。

Step	Variable eliminated	Factors used	Intermediate factor	New factor
1	S	$\phi_S(I,S)$	$\psi_1(I,S)$	$\tau_1(I)$
2	I	$\phi_I(I)$, $\phi_G(G,I,D)$, $\tau_1(I)$	$\psi_2(G,I,D)$	$\tau_2(G,D)$
3	D	$\phi_D(D)$, $\tau_2(G,D)$	$\psi_3(G,D)$	$\tau_3(G)$
4	L	$\phi_L(L,G)$	$\psi_4(L,G)$	$\tau_4(G)$
5	G	$\tau_4(G)$, $\tau_3(G)$	$\psi_5(G)$	$\tau_5(\emptyset)$



最后的因子τ_5仅仅是一个数，其值为

$$max_{S,I,D,L,G} P(S,I,D,L,G) = 0.184338$$

数据挖掘实验室
Data Mining Lab

解：$max_{S,I,D,L,G}P(S,I,D,L,G) = \ max_{S,I,D,L,G}P(D)P(I)P(S|I)P(G|D,I)P(L|G)$

$= max_G\ max_L P(L|G) max_D P(D) max_I P(I)P(G|I,D) max_S P(S|I)$

①： $max_S P(S|I)$ → τ_1(I)

②： $max_I P(I)P(G|I,D)\tau_1$(I) → τ_2(G,D)

③： $max_D P(D)\tau_2$(G,D) → τ_3(G)

④： $max_L P(L|G)\tau_3$(G) → τ_4(G)

⑤： $max_G\tau_4$(G) → 0.184338

数据挖掘实验室
Data Mining Lab

- ①：$max_S P(S|I)$ → $\tau_1(I)$

i^0	i^1
0.95	0.8

- ②：$max_I P(I)P(G|I,D)\tau_1(I)$ → $\tau_2(G,D)$

d^0,g^1	d^0,g^2	d^0,g^3	d^1,g^1	d^1,g^2	d^1,g^3
0.9×0.3×0.8	0.4×0.7×0.95	0.3×0.7×0.95	0.5×0.3×0.8	0.3×0.3×0.8	0.7×0.7×0.95

- ③：$max_D P(D)\tau_2(G,D)$ → $\tau_3(G)$

g^1	g^2	g^3
0.6×0.9×0.3×0.8	0.6×0.4×0.7×0.95	0.4×0.7×0.7×0.95

- ④：$max_L P(L|G)\tau_3(G)$ → $\tau_4(G)$

g^1	g^2	g^3
0.9×0.6×0.9×0.3×0.8	0.6×0.6×0.4×0.7×0.95	0.99×0.4×0.7×0.7×0.95

- ⑤：$max_G \tau_4(G)$ → 0.184338

2.2 找到最可能的赋值：

- MAP推理的目标：找到其自身最可能的赋值，即解码任务
- 变量的取值不能在消除变量时确定，需要一个回溯的过程来确定变量最可能的赋值。（最后一个变量除外，因为此时已经是该变量的最大—边缘，可以直接确定）

- Eg：针对上例，现在考虑回溯阶段，确定所有的变量的值。
  - $g^* = argmax_g \tau_5(g)$
  - $l^* = argmax_l \tau_4(g^*, l)$
  - $d^* = argmax_d \tau_3(g^*, d)$
  - $i^* = argmax_i \tau_2(g^*, i, d^*)$
  - $s^* = argmax_s \tau_1(i^*, s)$

- ①： $max_S P(S|I)$ → τ_1(I)

i^0（s = 0）	i^1
0.95	0.8

- ②： $max_I P(I)P(G|I,D)\tau_1$(I) → τ_2(G,D)

d^0,g^1	d^0,g^2	d^0,g^3	d^1,g^1	d^1,g^2	d^1,g^3 （I = 0）
0.9×0.3×0.8	0.4×0.7×0.95	0.3×0.7×0.95	0.5×0.3×0.8	0.3×0.3×0.8	0.7×0.7×0.95

- ③： $max_D P(D)\tau_2$(G,D) → τ_3(G)

g^1	g^2	g^3（d = 1）
0.6×0.9×0.3×0.8	0.6×0.4×0.7×0.95	0.4×0.7×0.7×0.95

- ④： $max_L P(L|G)\tau_3$(G) → τ_4(G)

g^1	g^2	g^3（l = 0）
0.9×0.6×0.9×0.3×0.8	0.6×0.6×0.4×0.7×0.95	0.99×0.4×0.7×0.7×0.95

- ⑤： $max_G \tau_4$(G) → 0.184338

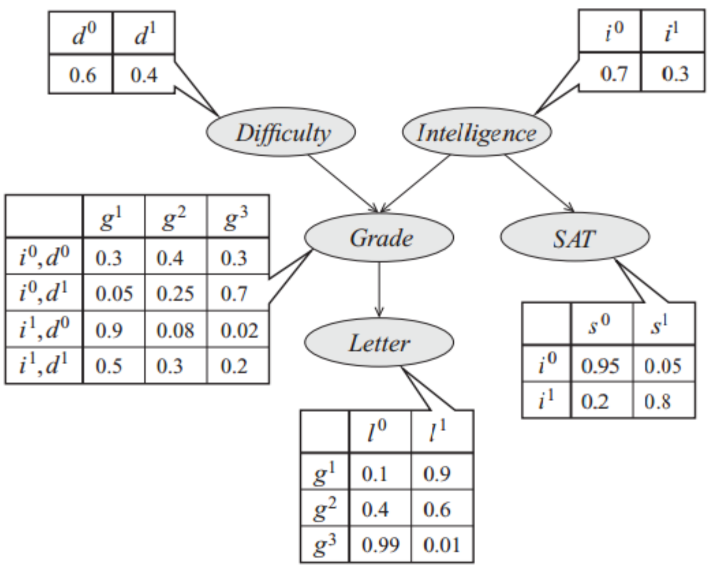
- $g^* = argmax_g \tau_5(g)$
  - $g = 3$
- $l^* = argmax_l \tau_4(g^*, l)$
  - $l = 0$
- $d^* = argmax_d \tau_3(g^*, d)$
  - $d = 1$
- $i^* = argmax_i \tau_2(g^*, i, d^*)$
  - $i = 0$
- $s^* = argmax_s \tau_1(i^*, s)$
  - $s = 0$

- 2.3 边缘MAP的变量消除*
- 由边缘MAP定义的公式可知，这是一个最大 -和-积的形式

$$\max_{\boldsymbol{Y}} \sum_{\boldsymbol{W}} \prod_{\phi \in \Phi} \phi.$$

- 算法仅仅是将用于概率查询和用于MAP查询的想法结合起来，因此，为了计算该表达式的值，必须通过对*W*中的变量求和消除*W*，以及最大化*Y*中的变量消除*Y*

- eg：考虑如下所示的网络，假定想要找到SAT结果和推荐信好坏的最可能实例的概率 $\max_{S,L}\sum_{G,I,D} P(I,D,G,S,L).$



$$
\begin{aligned}
\psi_1(I,G,D) &= \phi_D(D)\cdot\phi_G(G,I,D)\\
\tau_1(I,G) &= \sum_D \psi_1(I,G,D)\\
\\
\psi_2(S,G,I) &= \phi_I(I)\cdot\phi_S(S,I)\cdot\tau_1(I,G)\\
\tau_2(S,G) &= \sum_I \psi_2(S,G,I)\\
\\
\psi_3(S,G,L) &= \tau_2(S,G)\cdot\phi_L(L,G)\\
\tau_3(S,L) &= \sum_G \psi_3(S,G,L)\\
\\
\psi_4(S,L) &= \tau_3(S,L)\\
\tau_4(L) &= \max_S \psi_4(S,L)\\
\\
\psi_5(L) &= \tau_4(L)\\
\tau_5(\emptyset) &= \max_L \psi_5(L).
\end{aligned}
$$

3. 团树中的最大—积

– 3.1 计算最大—边缘
– 3.2 信息传递——再参数化
– 3.3 最大—边缘解码

数据挖掘实验室
Data Mining Lab

- 解决问题的流程是：
  - 1. 计算每个团的最大—边缘
  - 2. 计算每个团的置信和割集的置信
  - 3. 知道这些置信集之后我们其实就已经知道了每个团中的分布的再参数化(可以视为忽略某个参数影响后其它变量的分布情况)，使用这些数据可以表示团中的变量分布
  - 4. 已知团中分布的情况下通过解码来找到变量集合的赋值，基于局部最优性质

数据挖掘实验室
Data Mining Lab

- *defs*：
  - 聚类树：变量消除过程中的执行得到的聚类图就是聚类树
  - 执行相交性：设T是因子集Φ上的一颗聚类树，V_T是T的顶点，E_T是其边。如果每当有变量X使得$X \in C_i$且$X \in C_j$，则X也在C_i于C_j之间路径的每个聚类中。那么T称为有执行相交性质。
  - 团树：令Φ是X上的因子集， Φ上满足执行相交性质的聚类树称为团树，在团树情况下，聚类也称为团。
  - 消息传递：变量消除过程中两个簇之间传递的消息，用来实现变量消除
  - 置信：置信就是变量消除过程中，当本簇就绪后所蕴藏的分布信息的一种称呼
  - 校准：就是簇中的置信收敛，分布收敛
  - 再参数化：将校准后的团内的置信所表示的分布信息展示出来

数据挖掘实验室
Data Mining Lab

- 团树上的MAP推理算法：
  - 第一部分：计算最大－边缘
  - 第二部分：解码他们来抽取一个MAP赋值

3.1 计算最大 - 边缘

- 最大 – 积置信传播算法

Procedure CTree-Calibrate(***Φ***//因子的集合,***T***//***Φ***上的团树)

1.初始化团树

2.当存在i,j使得i向j的发送就绪时

$\delta_{i\to j}(S_{i,j}) \leftarrow Max - \text{Message}$（i，j）

3.对每个团I

$\beta_i = \Psi_i \prod_{k\in Nb_i} \delta_{k\to i}$

4.返回{$\boldsymbol{\beta_i}$}

Procedure Max-Message(i//传送团,j//接收团)

1.$\Psi(C_i) \leftarrow \Psi_i \prod_{k\in (Nb_i - \{j\})} \delta_{k\to i}$

2.$\tau(S_{i,j}) \leftarrow max_{C_i - S_{i,j}} \Psi(C_i)$

3.返回$\tau(S_{i,j})$

实质就是消息传递过程 + 上置信计算过程

- 考虑最大－积团树算法的一次运行，其中，我们用因子集ϕ进行初始化。令β_i是由该算法的一次向上和向下传递所产生的置信集。那么对于每一个团C_i以及C_i的每个赋值c_i，我们有

$$\beta_i(\boldsymbol{c}_i) = MaxMarg_{\tilde{P}_\Phi}(\boldsymbol{c}_i)$$

- 注：最大-积消息传递过程中不计算拆分函数，因此最大—边缘中代表的不是精确概率获得，又拆分函数是一个常数，所以仍然可以比较与每个赋值相关联的值，并计算最大化$\tilde{P}_\Phi(\xi)$的赋值ξ

- 团树上的最大—积消息传递在每个团中产生了最大—边缘，并且最大－边缘必须一致，所以任意两个相邻的团必须在其割集上保持一致，即

$$\max_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \beta_i = \max_{\boldsymbol{C}_j - \boldsymbol{S}_{i,j}} \beta_j = \mu_{i,j}(\boldsymbol{S}_{i,j})$$

  此时，聚类称为是最大－校准的，如果团树中所有相邻的团均是最大校准的，那么该团树称为最大－校准的

- **推论：**经由最大—积团树算法中一个向上和向下传递所产生的团树的置信是最大校准的。

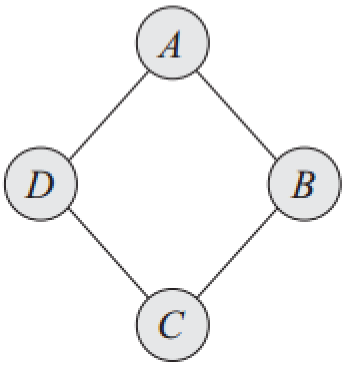
- 团的割集上的最大-边缘可以根据最大 – 积消息计算获得

$$\mu_{i,j}(\boldsymbol{S}_{i,j}) = \delta_{j\to i}(\boldsymbol{S}_{i,j}) \cdot \delta_{i\to j}(\boldsymbol{S}_{i,j}).$$

- Proof：

$$\begin{aligned}
\mu_{i,j}(S_{i,j}) &= max_{C_i - S_{i,j}} \beta_i(C_i) \\
&= max_{C_i - S_{i,j}} \Psi_i \prod_{k\in Nb_i} \delta_{k\to i} \\
&= max_{C_i - S_{i,j}} \Psi_i \delta_{j\to i} \prod_{k\in (Nb_i - j)} \delta_{k\to i} \\
&= \delta_{j\to i} max_{C_i - S_{i,j}} \Psi_i \prod_{k\in (Nb_i - j)} \delta_{k\to i} \\
&= \delta_{j\to i}\delta_{i\to j}
\end{aligned}$$

- Eg：考虑如图所示的马尔可夫网，其联合分布如右图所示。该网络的一个具有割集{B,D}的团树包含两个团{A,B,D}，{B,C,D}。



Assignment				*Unnormalized*	*Normalized*
a^0	b^0	c^0	d^0	300,000	0.04
a^0	b^0	c^0	d^1	300,000	0.04
a^0	b^0	c^1	d^0	300,000	0.04
a^0	b^0	c^1	d^1	30	$4.1 \cdot 10^{-6}$
a^0	b^1	c^0	d^0	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^0	d^1	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^1	d^0	5,000,000	0.69
a^0	b^1	c^1	d^1	500	$6.9 \cdot 10^{-5}$
a^1	b^0	c^0	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^0	d^1	1,000,000	0.14
a^1	b^0	c^1	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^1	d^1	100	$1.4 \cdot 10^{-5}$
a^1	b^1	c^0	d^0	10	$1.4 \cdot 10^{-6}$
a^1	b^1	c^0	d^1	100,000	0.014
a^1	b^1	c^1	d^0	100,000	0.014
a^1	b^1	c^1	d^1	100,000	0.014

这个例子中团和割集的最大－边缘置信如图所示，很容易验证该团树是校准的

Assignment			$\max_C$
a^0	b^0	d^0	300,000
a^0	b^0	d^1	300,000
a^0	b^1	d^0	5,000,000
a^0	b^1	d^1	500
a^1	b^0	d^0	100
a^1	b^0	d^1	1,000,000
a^1	b^1	d^0	100,000
a^1	b^1	d^1	100,000

$\beta_1(A, B, D)$

Assignment		$\max_{A,C}$
b^0	d^0	300,000
b^0	d^1	1,000,000
b^1	d^0	5,000,000
b^1	d^1	100,000

$\mu_{1,2}(B, D)$

Assignment			$\max_A$
b^0	c^0	d^0	300,000
b^0	c^0	d^1	1,000,000
b^0	c^1	d^0	300,000
b^0	c^1	d^1	100
b^1	c^0	d^0	500
b^1	c^0	d^1	100,000
b^1	c^1	d^0	5,000,000
b^1	c^1	d^1	100,000

$\beta_2(B, C, D)$

3.2 作为再参数化的信息传递

- 最大积消息传递步骤保持分布不变的方式视为对原分布的再参数化，更准确地说，可以将最大积团树中的置信集β_i和割集消息$\mu_{i,j}$视为一个度量

$$Q_{\mathcal{T}} = \frac{\prod_{i \in \mathcal{V}_{\mathcal{T}}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j) \in \mathcal{E}_{\mathcal{T}}} \mu_{i,j}(\boldsymbol{S}_{i,j})}.$$

- Eg：接上例，证实，原始度量 $\tilde{P}_\Phi(\xi)$ 可以直接从最大—边缘和割集消息中重构。例如考虑表值

$$\tilde{P}_\Phi(a^1, b^0, c^1, d^0) = 100.$$

根据公式

$$Q_\mathcal{T} = \frac{\prod_{i \in \mathcal{V}_\mathcal{T}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j) \in \mathcal{E}_\mathcal{T}} \mu_{i,j}(\boldsymbol{S}_{i,j})}.$$

知团树的度量为：

$$\frac{\beta_1(a^1, b^0, d^0)\beta_2(b^0, c^1, d^0)}{\mu_{1,2}(b^0, d^0)} = \frac{100 \cdot 300,000}{300,000} = 100,$$

3.3 最大边缘解码

- 在变量消除算法中我们采用的是回溯的方法来计算变量的赋值，现在我们有网络中所有变量的最大—边缘，怎么办？
- 解决方案：对每个X_i利用最大—边缘来计算其自身的最优赋值，从而构成所有变量的一个完整联合赋值。（有没有感觉像贪心算法呢？所以面临同样的问题，并不是什么时候都可以用这种方法）
- 联合赋值不能通过优化单个最大—边缘来实现

数据挖掘实验室
Data Mining Lab
LESS IS MORE

- 命题13.3：以下两个条件是等价的：
  - 节点置信集$\{MaxMarg_{\tilde{P}_\Phi}(X_i) \,:\, X_i \in \mathcal{X}\}$是确定的，并且
    $$x_i^* = \arg\max_{x_i} MaxMarg_{\tilde{P}_\Phi}(X_i)$$
    是Xi的唯一最优值。
  - $\tilde{P}_\Phi$有唯一的MAP赋值$(x_1^*, \ldots, x_n^*)$.

- 定义13.3：令$\beta_i(C_i)$是最大校准团树中的一个置信。如果对树中的每个团C_i, 我们有，

$$\xi^*\langle \boldsymbol{C}_i\rangle \in \arg\max_{\boldsymbol{c}_i} \beta_i(\boldsymbol{c}_i),$$

- 也就是说，C_i在ξ^*中的赋值优化了C_i置信，则称赋值ξ^*有**局部最优性质**，给定最大—校准置信的一个集合，找到局部最优赋值ξ^*的任务称为**解码**任务。

- 因此，如果在任意校准的节点置信之间没有连结，我们可以通过分别局部最优化对每个变量的赋值来找到唯一的MAP赋值。如果节点之间置信存在，那么就需要借助于前文所说的回溯(Traceback)机制来寻找最优值，此时的局部最优性为验证给定的赋值是否是一个MAP赋值提供了一个简单的测试。
- （这里有个问题，就是再动态规划过程中的每一个值并不是最大值就好，为什么这里强调要符合局部最优性质呢？思考一下（在计算最大边缘的时候这个性质就已经考虑在内了，因此这里不必再考虑，只是一种线性执行关系喽））

附：多圈聚类图中的置信传播

Def：什么叫做多圈聚类图

- 将最大—积消息传递算法推广到聚类图中就产生了标准最大—积累消息传递算法，和在团树上的消息传递算法的区别就在于是否验证图已经校准，在团树中我们可以知道经过消息传递后图是校准的，然而在最大—积消息传递过程中这一条件并不满足。

1 消息传递算法

Procedure Max-Message(i//传送团,j//接收团)

1.$\Psi(C_i) \leftarrow \Psi_i \prod_{k\in(Nb_i-\{j\})} \delta_{k\to i}$

2.$\tau(S_{i,j}) \leftarrow max_{C_i-S_{i,j}} \Psi(C_i)$

3.返回$\tau(S_{i,j})$

- 消息传递算法如算法13.2
- 在聚类图中执行消息传递
- 过程，多次迭代后，如果
- 到达收敛点，结果是校准的聚类集，那么就产生精确的最大—边缘，采用在团树上的算法即可，如果未达到收敛点，产生的置信不是精确的最—边缘，此时产生的是伪最大—边缘。
- 需要注意的是，在聚类图中的消息传递过程中，仍然满足分布的再参数化。因为分布不变性是基于消息传递的和聚类是树或者图无关。

2 解码伪最大—边缘

- 寻求最优赋值的过程：
- 在团树中，我们可以肯定存在一个最优联合赋值满足局部最优性质，但是在聚类图中这样的赋值并不一定存在，例如：

例 13.11 考虑有三个聚类$\{A,B\}$,$\{B,C\}$,$\{A,C\}$的一个聚类图和置信：

	a^1	a^0
b^1	1	2
b^0	2	1

	b^1	b^0
c^1	1	2
c^0	2	1

	a^1	a^0
c^1	1	2
c^0	2	1

- 在团树中，我们可以在团中简单的为置信选择任意的最大化赋值，并且确保其能够扩展为一个联合MAP赋值。
- 在聚类图中，并不是所有的局部最优赋值的一种选择都可以产生一个一致的联合赋值（上例是一种极端，实际中并不存在这样的情况）。

数据挖掘实验室
Data Mining Lab

- 由于在聚类图中的我们并不能简单的选取每个簇中的最优值进而扩展为MAP最优赋值，因此现在必须为χ搜索一个对每个聚类都导致一个合法值(合法值指的是该赋值在对应的簇上达到优化置信)的赋值。该问题敲好是一个约束满足问题，其中约束源自局部最优性质。

- **总结**
- 1 目标：找到MAP赋值
- 2 解决方法：VE
- 3 优化：团树上的最大—积消息置信传播算法，通过再参数化以产生置信的一个最大的校准集，将全局优化问题——即找到一个一致的联合赋值的问题——转换为一个局部最优化的问题——即找到一系列优化其(校准的)置信的局部赋值的问题。
- 4 扩展到了聚类图中
- 问题是：边缘MAP的问题还是很难解决

Thanks

Tingting Niu
www.827331138@qq.com
Qinyuan Liu
liuqinyuan1992@163.com