

概率图模型

Probabilistic Graphical Models

- ① 背景
- ② 贝叶斯网络
- ③ 马尔可夫随机场
- ④ 推断
- ⑤ 概念补充

背景 (Background)

概率图模型关注的对象是高维随机变量。

高维随机变量 $P(x_1, x_2, \dots, x_p)$

边缘概率	$P(x_i)$
条件概率	$P(x_i x_j)$

下面是一些法则：(前三个法则假设高维随机变量的维度为2)

sum rule: $P(x_1) = \int p(x_1, x_2) dx_2$. (高散求数和，连续求数积分)

product rule: $p(x_1, x_2) = p(x_1)p(x_2|x_1) = p(x_2)p(x_1|x_2)$

Bayesian rule: $p(x_2|x_1) = \frac{p(x_1, x_2)}{p(x_1)} = \frac{p(x_2)p(x_1|x_2)}{\int p(x_1, x_2) dx_2} = \frac{p(x_2)p(x_1|x_2)}{\int p(x_2)p(x_1|x_2) dx_2}$

chain rule: $p(x_1, x_2, \dots, x_p) = p(x_1) \prod_{i=2}^p p(x_i | x_1, x_2, \dots, x_{i-1})$

困境：数据维度高、chain rule中的链过长， $p(x_1, x_2, \dots, x_p)$ 计算复杂。

↓ 简化

相互独立: $p(x_1, \dots, x_p) = \prod_{i=1}^p p(x_i)$ Naive Bayes: $p(x|y) = \prod_{i=1}^p p(x_i|y)$

↓ 放宽条件

Markov Property $x_j \perp x_{j+1} | x_i, j < i$

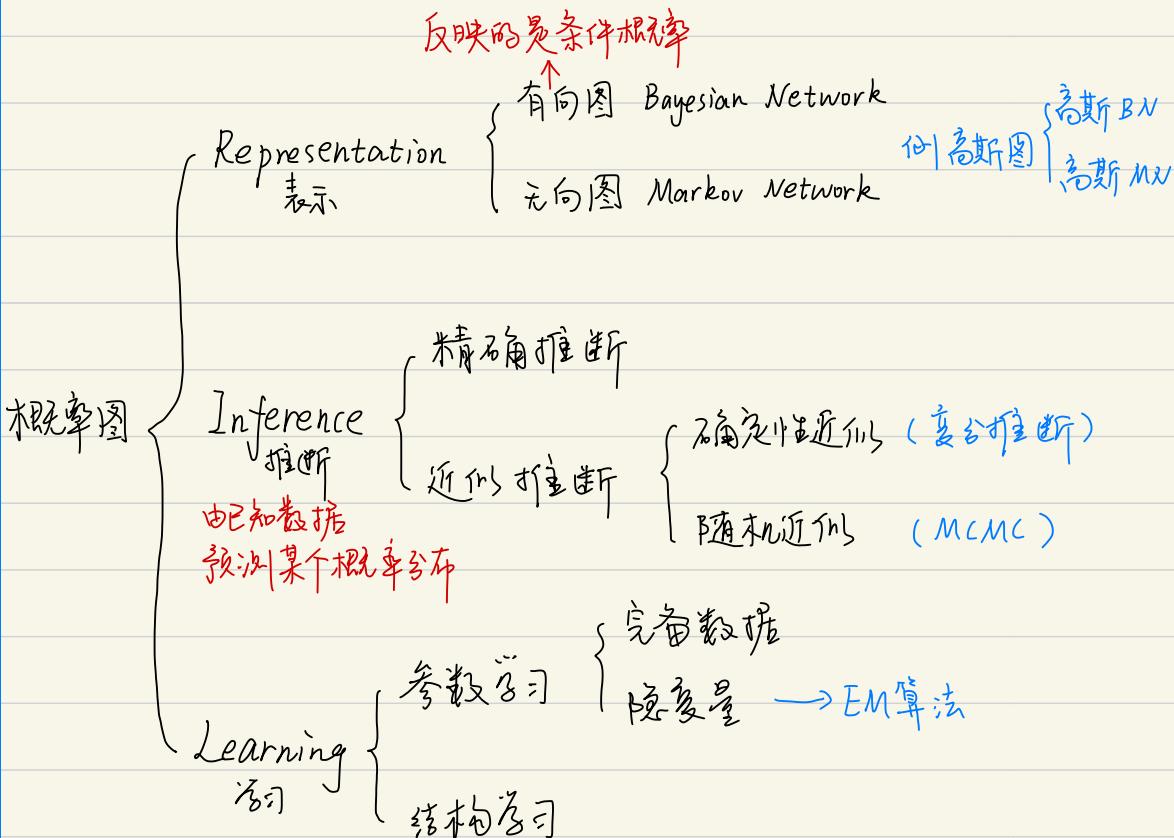
未来时刻独立于所有过去时刻

HMM (齐次Markov假设
观测独立假设)

↓ 继续放宽 (因为未来某时刻不一定独立于所有过去时刻) 对 Markov Property 适用

条件独立性假设: $X_A \perp X_B \mid X_C$.

概率图模型核心概念 其中, X_A, X_B, X_C 分别为三个集合, 且 $X_A \cap X_B = X_A \cap X_C = X_B \cap X_C = \emptyset$



其实还有一个 Decision (决策), 这里只关注上面三个部分,

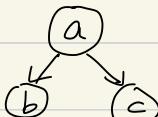
贝叶斯网络 (Bayesian Network)

条件独立性 (Conditional Independence)

构建有向图方法 — 拓扑排序.

示例: 假设存在 $p(x_i | x_j)$. 如图表示为 $x_j \rightarrow x_i$

可以直接根据有向图写出联合概率因子分解.

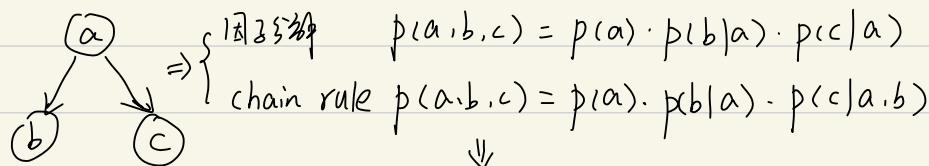
示例:  $\Rightarrow p(a, b, c) = p(a) \cdot p(b|a) \cdot p(c|a)$.

如通式: $p(x_1, x_2, \dots, x_p) = \prod_{i=1}^p p(x_i | \text{par}_i)$ par_i 是 x_i 父节点的集合.

"head \rightarrow tail" \Rightarrow "O \rightarrow O"

某些有向图无法以表示条件独立性. 见下面三种情况

①. tail to tail 结构.



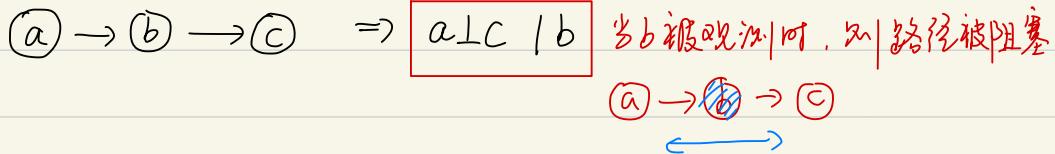
$$p(c|a) = p(c|a, b)$$

当 a 被观测时, 之路经被阻塞

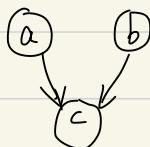
$$c \perp b \mid a$$



(2) head to tail 结构



(3) head to head 结构.



这种情况和上述两种不同，不能用路径阻塞
来解释，和上述两种情况相反
该种情况下，在默认情况下 a 和 b 相互独立
但是当 c 被观测时，a 和 b 就不独立

$$\begin{cases} p(a, b, c) = p(a)p(b)p(c|a, b) \\ p(a, b, c) = p(a)p(b|a)p(c|a, b) \end{cases}$$
$$\Rightarrow p(b) = p(b|a) \Rightarrow \text{无视条件 } c$$

如果图中没有环，那么就不满足条件独立性 (这里的 a, b, c, 指的是集合，
并不是说图中不能出现环结构)

```
graph TD; a((a)) --> c((c)); b((b)) --> c((c))
```

$$\begin{aligned} p(a, b, c) &= p(a|c)p(b|a)p(c|b) \\ p(a, b, c) &= p(a)p(b|a)p(c|a, b) \\ \Rightarrow p(a|c)p(c|b) &= p(a)p(c|a, b) \\ \Rightarrow p(a|b) &= p(c|b) \\ \Rightarrow a \text{ 和 } c \text{ 在条件 } b \text{ 下不独立.} \end{aligned}$$

D 分离 (D-Separation)

将上述3种情况加以综合概括，总结出了 D-separation.

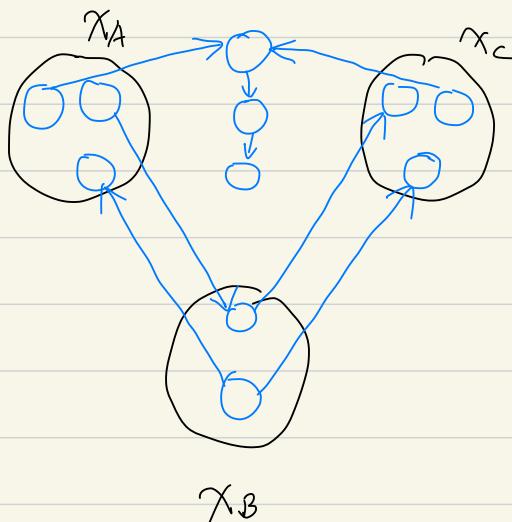
即，判断一个图是否满足条件独立性可以使用 D-Separation 规则如下。

假设 X_A , X_B 和 X_C 是总体的三个子集 ($X_A \cup X_B \cup X_C$ 不一定覆盖总体)

将 X_B 看作观察的变量。即

若 X_A 和 X_C 中至少有一个节点，两者之间存在路径

- ① 若路径为“tail to tail”，则该结构中的 head 必在 X_B 中
- ② 若路径为“head to tail”，则该结构中的中间节点必在 X_B 中
- ③ 若路径为“head to head”，则该结构中的 tail 必不在 X_B 中，并且该 tail 的后续节点也必不在 X_B 中



这种 D-Separation 规则一般被称作“全局 Markov Property”

Markov Blanket

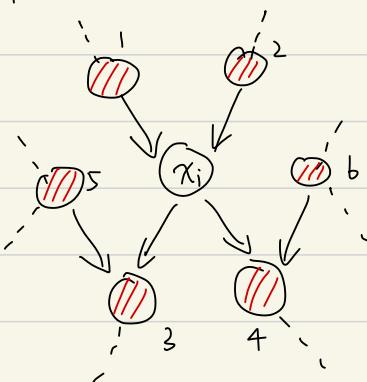
$$P(x_i | \underline{x_{-i}}) = \frac{P(x)}{P(x_{-i})} = \frac{\prod_{j=1}^p P(x_j | x_{pa_j})}{\int \prod_{j=1}^p P(x_j | x_{par_j}) dx_i}$$

\downarrow

$x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_p$

观察上式，设 $P(x_j | x_{par(j)})$ 中与 x_i 相关的为 Δ ，与 x_i 无关的为 $\bar{\Delta}$ 。
 那么 Δ 可以从父辈中被公因子中提出，并与分子约去。
 即最后 $P(x_i | x_{-i})$ 表达式中只有与 x_i 相关的 Δ 。

从概念图中表示：



观察上图，显然， $P(x_i | x_{par(i)})$ 会保留。又 1, 2 与 x_i 相关。

$P(x_{child(i)} | \underline{x_i}, \underline{x_{par(child(i))}})$ 也会保留。2-| 3, 4, 5, 6 与 x_i 相关。
 x_i 的子节点，除了 x_i 外的 x_i 的子节点的父节点集合

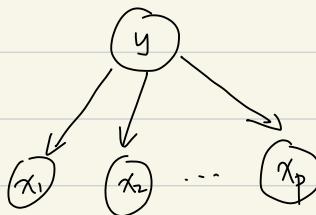
2-| 1, 2, 3, 4, 5, 6 被称为 Markov Blanket，即与 x_i 相关的节点。

贝叶斯网络的一些例子。

① 单一: Naive Bayes

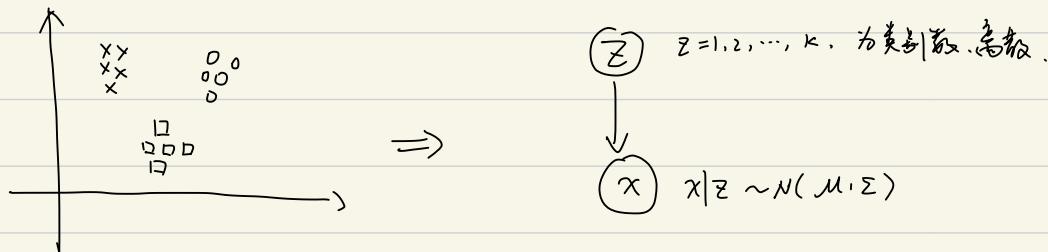
$$p(x|y) = \prod_{i=1}^p p(x_i|y)$$

概率图:



② 混合: GMM

(这里将GMM与贝叶斯网络联系在一起,一时没想明白)



③ 引入时间概念

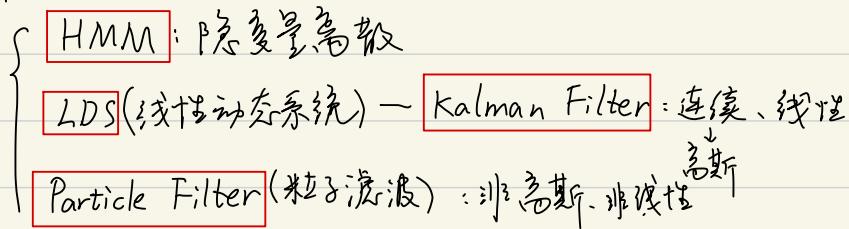
{ Markov chain

Gaussian Process (无限维高斯分布)

④ 连续: Gaussian Bayesian Network

⑤ 将混合与高斯结合.

⇒ 动态模型



总的来说，是从单一到混合

从有限到无限 → {
 时间
 空间 — 随机变量(高斯 → 连续)

马尔可夫随机场

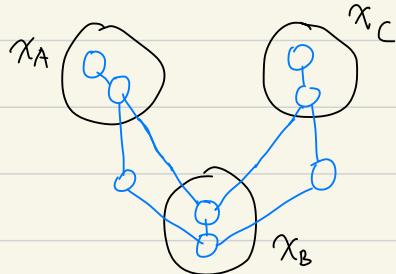
(Markov Random Field)

马尔可夫随机场也称作马尔可夫网络 (Markov Network)

条件独立性 体现在三个方面

① Global Markov

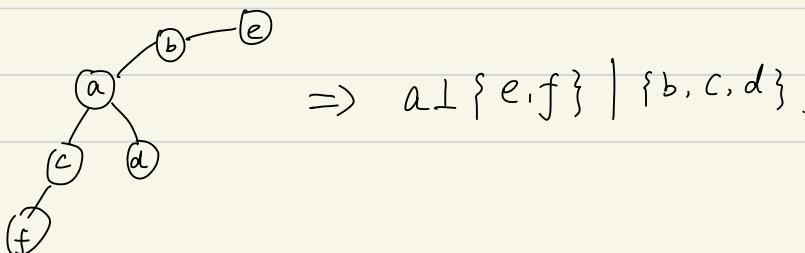
$$x_A \perp x_C \mid x_B$$



表述为： x_A 中任意节点到 x_C 中任意节点，
两条路径都至少有一个节点
在 x_B 中

② Local Markov

$$x_i \perp x - x_i - x_{\text{nei}(i)} \mid x_{\text{nei}(i)}, \text{ 其中 } x \text{ 表示所有节点, } x_{\text{nei}(i)} \text{ 表示 } x_i \text{ 的所有邻居}$$



③ 成对 Markov

$x_i \perp x_j \mid X_{-i-j}$, 其中 $i \neq j$, 且 x_i 和 x_j 不相邻.

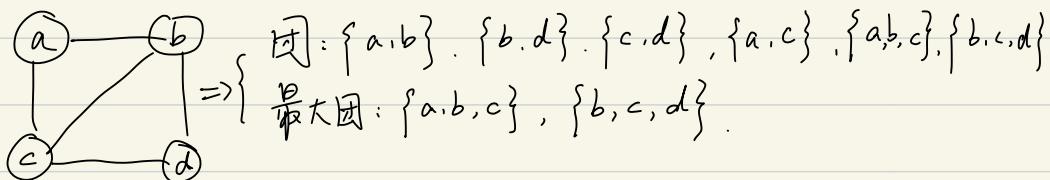
① \Leftrightarrow ② \Leftrightarrow ⑤, 三个方面是等价的.

无向图的 Markov Blanket 只由目标节点的所有相邻节点组成.

团子分解

团：团内的结点都是全连接的

最大团：向团内添加任一节点都会破坏团的性质.



将团子分解定义在团上面:

$$P(x) = \frac{1}{Z} \prod_{i=1}^k \psi(x_{c_i}), \quad x_{c_i} \text{ 表示第 } i \text{ 个最大团的随机变量的集合, } k \text{ 为团数}$$

$\psi(\cdot)$ 为势函数, 总为正

$$Z \text{ 为归一化因子, } Z = \frac{1}{x} \sum_{i=1}^k \psi(x_{c_i}) = \boxed{\frac{\sum \dots \sum}{x_1 x_2 \dots x_p}} \sum_{i=1}^k \psi(x_{c_i})$$

Hammersley-Clifford 定理能证明上述的因子分解服从条件独立性

这个定理太复杂，这里略去，无向图的难点就在于这个定理。

一般地，我们令

$$\psi(x_{ci}) = \exp\{-E(x_{ci})\}$$

能量函数

统计物理学概念

通过该种形式计算出来的 $P(x)$ 亦为 Gibbs Distribution

(Boltzmann Distribution)

$$P(x) = \frac{1}{Z} \prod_{i=1}^k \psi(x_{ci})$$

$$= \frac{1}{Z} \prod_{i=1}^k \exp\{-E(x_{ci})\}$$

$$= \frac{1}{Z} \exp\left\{-\sum_{i=1}^k E(x_{ci})\right\} \Rightarrow \text{指教族分布}$$

由前面指教族分布可以知道，在满足已知事实的情况下，指教族分布熵最大，
所以 Gibbs 分布，虽然也具备最大熵概念。

Markov Random Field \Leftrightarrow Gibbs Distribution

推断 (Inference)

总体介绍

总的来说，推断就是求概率

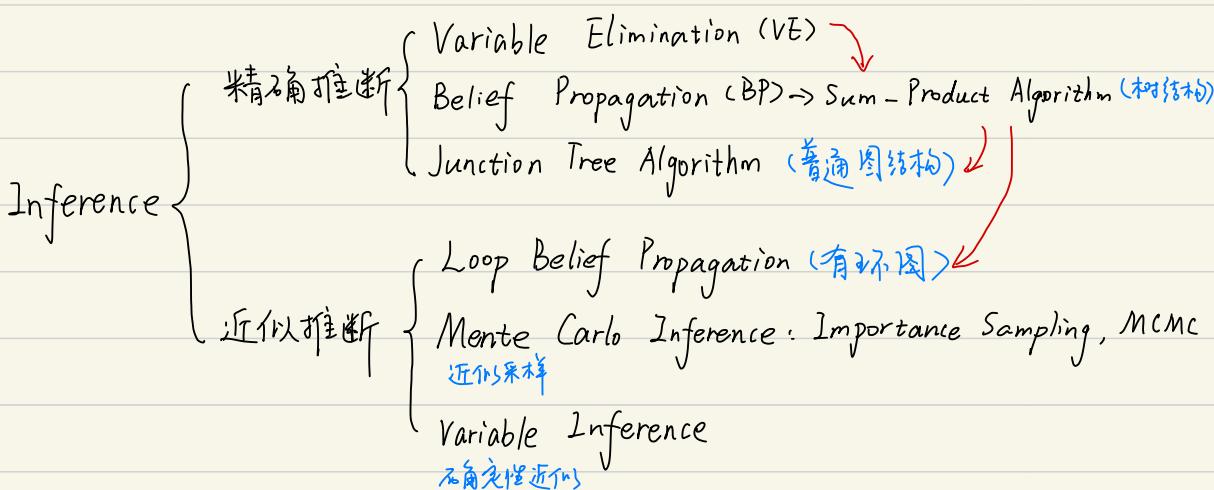
① 这样的概率： $P(x_i) = \sum_{x_1} \sum_{x_2} \dots \sum_{x_{i-1}} \sum_{x_{i+1}} \dots \sum_{x_p} P(x_1, x_2, \dots, x_p)$

扩展到集合： $P(x_A) = \sum_{x-x_A} P(x_1, x_2, \dots, x_p)$

② 条件概率： $P(x_A | x_B) \quad x = x_A \cup x_B$

③ MAP Inference： $P(z|x) = \frac{p(x,z)}{p(x)}$

$$z = \operatorname{argmax}_z p(z|x) \propto \operatorname{argmax}_z p(x,z) = \operatorname{argmax}_z p(x|z)p(z)$$



这里 Inference 部分只介绍 VE, BP 和 max-product algorithm

变量消除 (Variable Elimination)

$a \rightarrow b \rightarrow c \rightarrow d$, 假设 a, b, c, d 是离散的，且二值
 $a, b, c, d \in \{0, 1\}$

考虑加上马氏链：

$$p(d) = \sum_{a,b,c} p(a, b, c, d)$$

$$\begin{aligned} &= \sum_{a,b,c} p(a) p(b|a) p(c|b) p(d|c) \\ &= p(a=0) p(b=0|a=0) p(c=0|b=0) p(d|c=0) \\ &\quad + p(a=1) p(b=0|a=1) p(c=0|b=0) p(d|c=0) \\ &\quad + \dots \\ &\quad + p(a=1) p(b=1|a=1) p(c=1|b=1) p(d|c=1) \end{aligned} \quad \left. \right\} 8 \text{项.}$$

考虑针对 a 的求和， $p(a)p(b|a)$ 是唯一与 a 相关的因子。

$$\therefore p(d) = \sum_{b,c} \left[p(c|b) p(d|c) \underbrace{\sum_a p(a) p(b|a)}_{\phi_a(b)} \right] \quad \text{先对 } a \text{ 进行求和，消去变量 } a$$

$$p(a=0)p(b|a=0) + p(a=1)p(b|a=1)$$

$$= \sum_{b,c} p(c|b) p(d|c) \phi_a(b) \quad \text{乘法对加法的分配律.}$$

$$\text{同理: } = \sum_c \left[p(d|c) \underbrace{\sum_b p(c|b) \phi_a(b)}_{\phi_b(c)} \right] \quad \text{消去变量 } b$$

$$= \sum_c p(d|c) \phi_b(c) \quad \text{消去变量 } c.$$

$$= \phi_c(d).$$

上式的推导就是变量消除的方法，现在扩展到具有N个节点的无向图。



从得知：相邻两个节点构成一个最大团。

如 $p(x) = \frac{1}{Z} \prod_{i=1}^{N-1} \psi(x_i, x_{i+1})$ N-1 K^{N-1}

如 $p(x_n) = \frac{1}{Z} \sum_{x_1} \sum_{x_2} \dots \sum_{x_{n-1}} \sum_{x_{n+1}} \dots \sum_{x_N} \prod_{i=1}^{N-1} \psi(x_i, x_{i+1})$

考虑到对 x_n 求和时， $\psi(x_{n-1}, x_n)$ 是唯一与其相关的形式。

借助变量消除，我们可以将边缘概率写为

$$p(x_n) = \frac{1}{Z} \left[\sum_{x_{n-1}} \psi(x_{n-1}, x_n) \dots \left[\sum_{x_2} \psi(x_2, x_3) \left[\sum_{x_1} \psi(x_1, x_2) \right] \dots \right] \right]$$

$$\cdot \left[\sum_{x_{n+1}} \psi(x_n, x_{n+1}) \dots \left[\sum_{x_{N-1}} \psi(x_{N-2}, x_{N-1}) \left[\sum_{x_N} \psi(x_{N-1}, x_N) \right] \dots \right] \right]$$

这样做为什么可以简化计算？见下例。

$$ab + ac = a(b+c)$$

见，左式需要三次运算，而右式只需两次运算。

假设 x_i 有 K 个状态，则对于 $p(x_n)$ 来说

① 原式中， $\sum_{x_1} \sum_{x_2} \dots \sum_{x_{n-1}} \sum_{x_N}$ 的时间复杂度为 $O(K^{N-1})$ ， $\prod_{i=1}^{N-1} \psi(x_i, x_{i+1})$ 代价为 $O(K^{N-1})$

则总体为 $O((N-1)K^{N-1}) \rightarrow O(NK^N)$

→ 这里的分析是我自己分析的，可能时间复杂度不是这个。我是参照第一个例子分析的，后面第②点是 PRML 上的

其实是 $O(k(k-1))$, $k-1$ 代表对第
 i 行的 k 个数求和的代价, k 代表一共有

② VE 后, $\sum_{x_1} \psi(x_1, x_2)$ 只关于 x_1 , 对一个 $K \times K$ 表格求和, 复杂度为 $O(K^2)$ 行

$\sum_{x_2} \psi(x_2, x_3) [\sum_{x_1} \psi(x_1, x_2)]$, 其中 $\psi(x_2, x_3)$ 又是 $K \times K$ 表格,

括号中结果为 k 个数, 相应运算后, 复杂度为 $O(k^2)$
一共要进行 $n-1$ 次, 总代价 $O((n-1)k^2) \rightarrow O(Nk^2)$

其实为 $O(k(k-1+1))$,

$k-1$ 为求合, $+1$ 为将求合后的
结果与 $[\sum_{x_1} \psi(x_1, x_2)]$ 对应项相乘

可见, ② 的时间复杂度远小于①, 故 VE 能减小计算量.

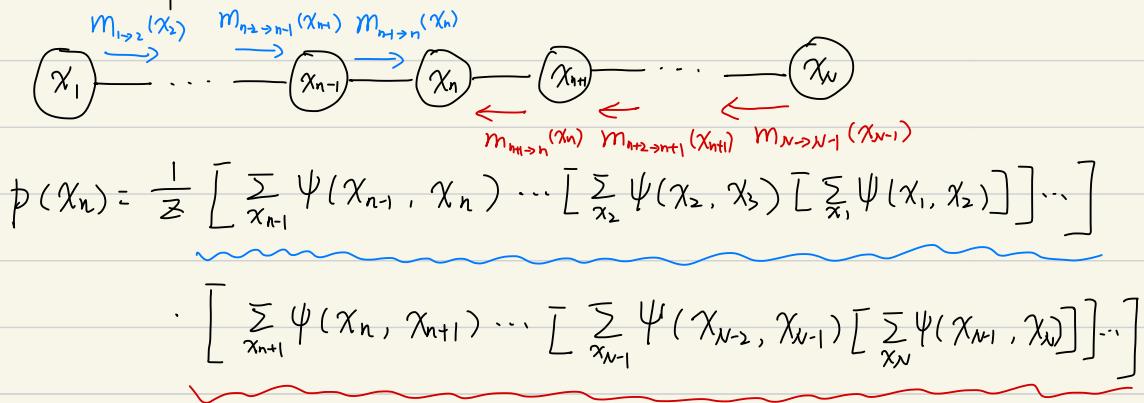
需要注意的是, VE 是利用条件独立性来简化运算的, 如果图是
全连接的, 则不满足条件独立性, 就必需计算完整的联合
概率分布.

① 注意 VE 一个缺点是没有传播数据的功能, 即计算 $P(x_m)$
需要 $O(NK^2)$ 代价, 计算 $P(x_m)$ 也需要 $O(NK^2)$ 代价, 计算
某一个概率后, 不会对其他产生帮助.

② 对于单链, 显然地我们有如上的办法, 但是对于不是链状
结构的图来说, 不同随机变量的求和顺序 (变量消除顺序) 对
计算结果影响很大, 选择最优顺序是一个 NP-Hard 问题.

Variable Elimination to Belief Propagation

考虑上述 n 个结点，马氏链。



可以看用 VE 法会有两部分，对应概率图中两种方向的箭头。

可以形象地将其看作 x_n 接收了其两侧的消息，箭头代表消息传递方向。

所以 $p(x_n)$ 可以改写成。

$$P(x_n) = \frac{1}{Z} m_{n \rightarrow n}(x_n) \cdot m_{n+1 \rightarrow n}(x_n)$$

其中 对于函数 $m_{i \rightarrow j}(x_j)$ ，代表消息从 x_i 传递到 x_j 。

$$\text{对应上式 } m_{1 \rightarrow 2}(x_2) = \sum_{x_1} \psi(x_1, x_2)$$

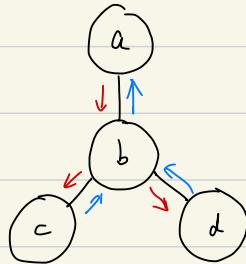
$$m_{2 \rightarrow 3}(x_3) = \sum_{x_2} \psi(x_2, x_3) \left[\sum_{x_1} \psi(x_1, x_2) \right] = \sum_{x_2} \psi(x_2, x_3) m_{1 \rightarrow 2}(x_2)$$

以此类推，最终得到 $p(x_n)$ 的表达式。

这也给我们一种启发，若将概率图中所有连通节点的消息传递表达式 $m_{i \rightarrow j}(x_j)$ 全求出来，最后在计算某一个具体概率时，直接就可以求得。

$$- \text{假设地 } p(x_{n-1}, x_n) = \frac{1}{Z} m_{n-2 \rightarrow n-1}(x_{n-1}) \psi(x_{n-1}, x_n) m_{n-1 \rightarrow n}(x_n)$$

现在将压缩结构推广到更一般的树结构.



$$p(a, b, c, d) = \frac{1}{Z} f_a(a) \cdot f_b(b) \cdot f_c(c) \cdot f_d(d) \\ \cdot f_{a,b}(a, b) \cdot f_{b,c}(b, c) \cdot f_{b,d}(b, d)$$

上式其实借助了因子图所写出的分步
PRML中有.

根据后上述的传递方式

$$m_{c \rightarrow b} = \sum_c f_c(c) f_{b,c}(b, c)$$

$$m_{d \rightarrow b} = \sum_d f_d(d) f_{b,d}(b, d)$$

$$m_{b-a} = \sum_b f_b(b) f_{a,b}(a, b) m_{c \rightarrow b} m_{d \rightarrow b}$$

$$m_{a \rightarrow b} = \sum_a f_a(a) f_{a,b}(a, b)$$

$$m_{b \rightarrow c} = \sum_b f_b(b) f_{b,c}(b, c) m_{a \rightarrow b} m_{d \rightarrow b}$$

$$m_{b \rightarrow d} = \sum_b f_b(b) f_{b,d}(b, d) m_{a \rightarrow b} m_{c \rightarrow b}$$

$$\therefore p(a) = \frac{1}{Z} f(a) m_{b \rightarrow a}$$

$$p(c) = \frac{1}{Z} f(c) m_{b \rightarrow c}$$

$$p(d) = \frac{1}{Z} f(d) m_{b \rightarrow d}$$

$$p(b) = \frac{1}{Z} f(b) m_{a \rightarrow b} m_{c \rightarrow b} m_{d \rightarrow b}$$

写成更一般的形式

此时 i, j 表示节点的下标	$\left\{ \begin{array}{l} m_{i \rightarrow j} = \sum_{x_i} f_{x_i}(x_i) f_{x_i, x_j}(x_i, x_j) \prod_{k \in \text{ne}(i)-j} m_{k \rightarrow i} \\ p(x_j) = \frac{1}{Z} f_{x_j}(x_j) \prod_{k \in \text{ne}(j)} m_{k \rightarrow j} \end{array} \right.$	$\prod_{k \in \text{ne}(i)-j} m_{k \rightarrow i}$ 表示不包括 x_j 和所有 x_i 的邻居.
-------------------	--	--

表示 x_j 的邻居

现在我们来分析一下代价(时间复杂度). 针对N个节点的链结构.

假设对N个节点中的所有节点都求边缘概率

① VE: 前面已经分析过, 求一个边缘概率的代价为 $O(NK^2)$. 则求N个边缘概率的代价为 $O(N^2K^2)$

② 消息传递: 只需对两个端点求边缘概率, 就可得到所有 $m_{i \rightarrow j}$. 此时代价为 $O(2NK^2)$, 此后, 求每个节点的边缘概率的代价为 $O(1)$. 则总代价为 $O(2NK^2 + N) = O((N+1)K^2) \rightarrow O(NK^2)$, 远小于 $O(N^2K^2)$.

Belief Propagation

$BP = VE + caching$, 其实就是利用上面得到的一般形式计算边缘概率.

稍微需要注意的是如何求 $m_{i \rightarrow j}$. 其实也很简单, 利用图的遍历就能做.

具体流程:

① Get Root. (选定一个结点作为根结点, 不是树型结构定义的根节点, 这里根结点只是遍历的起始节点.)

② Collect Message. (递归地, 向邻居节点索要信息, 索要的信息就是上述一般化的递推公式中的 $m_{i \rightarrow j}$)

③ Send Message (递归地, 向邻居节点发送信息)

算法运行过程中, 所有 $m_{i \rightarrow j}$ 都将被保留.

上一页将A作为Root, 蓝色箭头和蓝色框中的文字就是 Collect Message.

红色箭头和红色框中的文字就是 Send Message

上述的流程称为 Sequential Implementation, 还有 Parallel Implementation 版本 BP.

这里视频中也没有详细介绍, 后续如果接触到再补笔记

sum-product algorithm 其实是上述的 BP 算法的更一般化的形式。
 主要思想就是利用因子图，将任意概率图模型转化为树结构。
 然后应用消息传递的方式进行计算。(具体见 PRML)

上述的分析都是将所有变量当作是隐含变量，如果引入观测变量也是很容易处理的。借用上述一般树结构的推导结果，假设 d 为观测变量，在 d 的 K 个状态中观测到的值为 \hat{d} ，则只需在联合概率函数上函数 $I(d, \hat{d})$ ，当 $d = \hat{d}$ 时 $I(d, \hat{d}) = 1$ ，否则为 0。

$$\text{则 } p(a, b, c | d=\hat{d}) = I(d, \hat{d}) p(a, b, c, d).$$

$I(d, \hat{d})$ 的作用就是舍弃那些 $d \neq \hat{d}$ 的值，也就是在边缘概率时只考虑 $d = \hat{d}$ 的情况。

Max - Product Algorithm

对集合来说，是集合内
变量 = 联合概率

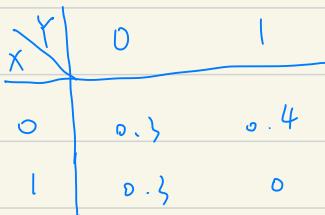
上述的 sum-product 求的是某个变量，或者是某个集合的边缘概率。

还有一种常见的任务，就是求最大概率情况下隐变量的值，即。

$$(x_1^*, x_2^*, \dots, x_N^*) = \arg \max_{(x_1, x_2, \dots, x_N)} p(x_1, x_2, \dots, x_N)$$

其实心理就是使仍然最大。

如果用 sum-product 来解，无非是求 (x_1, x_2, \dots, x_N) 使得各自的边缘概率最大，但这通常是错误的，如下例：



显然，当 $Y=0$ 时， $P(Y)$ 最大为 0.6

当 $X=0$ 时， $P(X)$ 最大为 0.7

此时得到的结果为 $(X=0, Y=0)$ 。

但是由联合分布可知 $(X=0, Y=1)$ 时， $P(X, Y)$ 最大。

所以，就需要用到 max-product。如下 (n 个节点，链结构为 12)

$$\text{目标: } \max_x p(x) = \max_{x_1} \max_{x_2} \dots \max_{x_N} p(x). \quad x \in \{x_1, x_2, \dots, x_N\}.$$

应用类似于 sum-product 中的乘法分配律，如下：

$$\max(ab, ac) = a \max(b, c). \quad (\text{由于条件独立性, 才可以使用该分配律})$$

得证。

$$\begin{aligned} \max_x p(x) &= \frac{1}{2} \max_x [\psi(x_1, x_2) \dots \psi(x_{n-1}, x_n)] \\ &= \frac{1}{2} \max_{x_1} \left[\max_{x_2} \psi(x_1, x_2) \left[\max_{x_3} \psi(x_2, x_3) \dots \left[\max_{x_n} \psi(x_{n-1}, x_n) \right] \dots \right] \right] \end{aligned}$$

$$\text{即: } m_{n \rightarrow n-1} = \max_{x_n} \psi(x_{n-1}, x_n) \cdot m_{n-1 \rightarrow n}$$

推广到一般 n 的树结构 (如 sum-product 中那样)

$$m_{i \rightarrow j} = \max_{x_i} f_i(x_i) f_{i \rightarrow j}(x_i, x_j) \prod_{k \in \text{nei}(i) \setminus j} m_{k \rightarrow i}$$

$$\max p(x) = \max_{x_i} f_i(x_i) \prod_{k \in \text{nei}(i)} m_{k \rightarrow i}$$

实际情况下，许多小概率的乘积会产生数值下溢问题，这时我们使用对数进行处理，从而 max-product 变为 max-sum

$$\ln(\max_x p(x)) = \max_x \ln p(x).$$

此时分配律仍然正确 $\rightarrow \max(a+b, a+c) = a + \max(b, c)$

$$\begin{cases} m_{i \rightarrow j} = \max_{x_i} (\ln f_i(x_i) + \ln f_{i \rightarrow j}(x_i, x_j) + \sum_{k \in \text{nei}(i) \setminus j} m_{k \rightarrow i}) \\ \ln(\max_x p(x)) = \max_{x_i} (\ln f_i(x_i) + \sum_{k \in \text{nei}(i)} m_{k \rightarrow i}) \end{cases}$$

回到 [b] 题中，我们要求的是 $x^* = \arg \max p(x)$ ，现在我们已经得到

$\max p(x)$ ($\max p(x)$ 与根节点的选择无关)，但是，可能存在多种 x_1, x_2, \dots, x_N 的组合能使 $p(x)$ 达到最大值，如果我们仅仅使用发送消息的方式去获取 x_i 的值，那么得到的 x_1, x_2, \dots, x_N 不能不是匹配的组合，所以为了获取正确的组合，我们需要跟踪能使得 $p(x)$ 最大的每一种组合，在反向跟踪时，我们只需要知道其中一个值，就能准确得知一种组合。(这里我想了半天也没想出能够解释发消息方法不行的例子，反向跟踪方法更详细的介绍参考 PRML)。

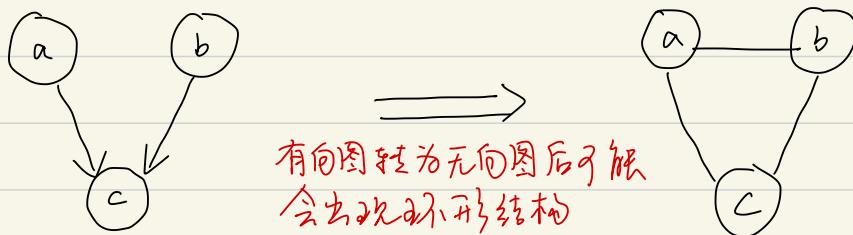
概念补充

道德图 (moral graph)

道德图二：将有向图转换成无向图，转换规则如下所示：

- ① 将同一个节点的所有父节点，两两相连
- ③ 将所有箭头去掉

上述的转换方式对“tail to tail”和“head to tail”的结构都不会发生改变，但会改变“head to head”结构，如下：



有向图转为无向图后才能
会出现环形结构

原因：道德图构建肯定不能破坏原有三致性性质。

针对有向图： $p(a, b, c) = p(a)p(b)p(c|a, b)$

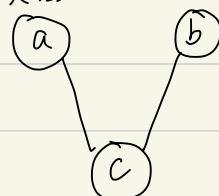
考虑只去除箭头的情形：此时， $p(a, b, c) = \frac{1}{2} \psi(a, c)\psi(b, c)$

显然， $p(c|a, b)$ 并不能分解到两个因子中，所以该转换是错误的。

再看正确的转换 $p(a, b, c) = \frac{1}{2} \psi(a, b, c)$

此时， $\psi(a, b, c) = p(a)p(b)p(c|a, b)$ 。

说明该转换合理。

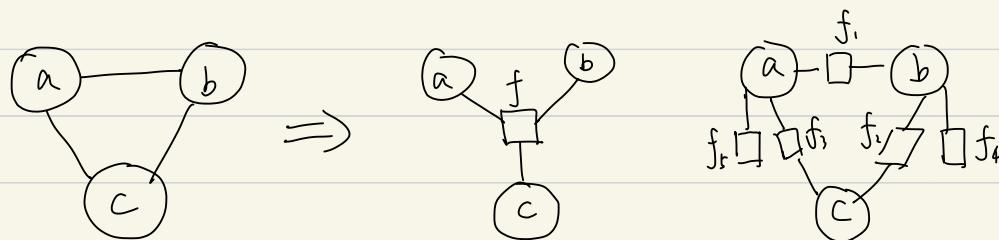


因子图 (factor graph)

因子图就是在原来的图上引入因子节点，联合概率就是因子的乘积：

$$p(x) = \prod_s f_s(x_s), x_s 表示变量的一个子集, f_s 就是该子集的函数(因子).$$

将图转化为因子图的方式就是引入因子节点，因子节点表示单个或多个变量之间的关系，所以因子图是不唯一的。如下例：



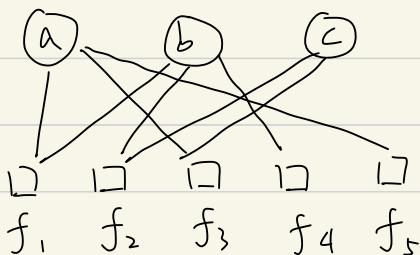
$$(1): p(x) = f(a, b, c)$$

(1)

$$(2): p(x) = f_1(a, b) f_2(b, c) f_3(a, c) f_4(b) f_5(a)$$

但无论怎么分解，最终的联合概率总是相同的。
（需要注意的是，因子图并不一定对应条件独立性质，如(2)的分解所示）

因子图所有因子连接的节点都是不同的变量，所以因子图是二分的，可以表示为



因子图的好处在于，只要选择合适的因子函数，无论有向树还是无向树转化成因子图后仍然为树结构，不存在环(即无圈)