符号学习简介

概率逻辑系统

(Press ? for help, n and p for next and previous slide)

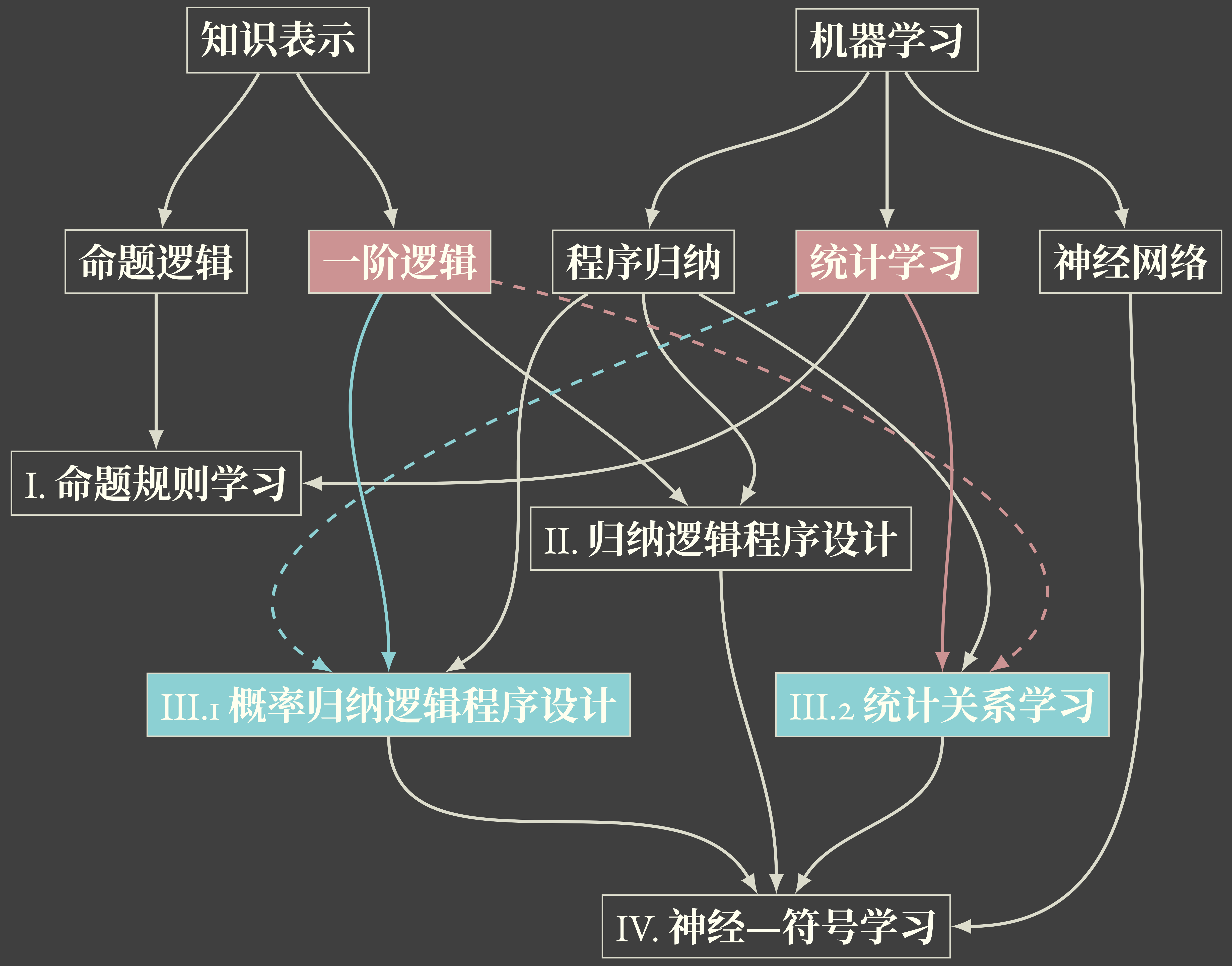
戴望州

南京大学智能科学与技术学院

2025年-秋-季

https://daiwz.net

路径图


知识表示
机器学习
命题逻辑
一阶逻辑
程序归纳
统计学习
神经网络
I. 命题规则学习
II. 归纳逻辑程序设计
III.1 概率归纳逻辑程序设计
III.2 统计关系学习
IV. 神经—符号学习

Final Project

Science World

https://github.com/allenai/ScienceWorld

Task

https://sciworld.apps.allenai.org/explore

1. 选取其中的 1 个task，通过自己玩 or 用（比较强的）LLM 收集完成该任务的 examples（trajectory）
2. 运用你学到的关于“程序归纳”的知识，实现一个简单的归纳算法，将上面的 examples 归纳为一个成功率较高的 program / workflow（命题 or 一阶均可）
3. 撰写课程报告（ddl, 2026 年 1 月 11 日）
4. 其他要求：
   - 若你的 program / workflow 需调用大模型，请使用 10b 以下模型
   - 为降低难度，可提供一定先置背景知识（如地图等），但关于任务执行本身的 Program / workflow 不能是手写的
   - 报告中必须严格描述你如何将该任务形式化为一个逻辑归纳问题，并描述你的归纳算法
5. 打分标准：技术报告的完成程度 + 选取任务的难度

符号学习简介・概率逻辑系统

1. **从非经典逻辑到概率逻辑**
2. KBMC
3. 分布语义

“真度”与“概率”

如果雨下的越大，身上打湿得越厉害。

“真度”与“概率”

$$rain \to wet$$

其中$rain$和wet均为**fuzzy set**

“真度”与“概率”

确定规则：

若身上某处淋到雨，则该处被打湿

“真度”与“概率”

确定规则：

若**身上每处**都淋到雨，则**全身**被打湿

“真度”与“概率”

概率事实：

0.8:: 身上每处都淋到雨

“真度”与“概率”

$$\frac{\begin{array}{r}\text{身上每处都淋到雨} \rightarrow \text{全身被打湿}\\ 0.8 :: \text{身上每处都淋到雨}\end{array}}{0.8 :: \text{全身被打湿}}$$

概率逻辑的语义

`0.8::身上每处都淋到雨`被称为**概率原子公式**，它表示：

$$\Pr\left(v(\text{身上每处都淋到雨}) = True\right) = 0.8$$

其中v为赋值函数。为了简便起见，一般省略该函数将$P(v(a) = True)$记为$P(a)$。

概率逻辑的语义

逻辑连接词的概率语义是什么：

- $\Pr(a \wedge b) = ?$
- $\Pr(a \vee b) = ?$
- $\Pr(\neg a) = ?$
- $\Pr(a \rightarrow b) = ?$

概率关系逻辑解释是什么：

- $\Pr(a \mid b) = ?$
- $\Pr(a, b) = ?$

符号学习简介・概率逻辑系统

1. 从非经典逻辑到概率逻辑
2. **KBMC**
3. 分布语义

基于知识的模型构建

Knowledge-Based Model Construction （KBMC）：构建一个具有逻辑结构的**概率模型**

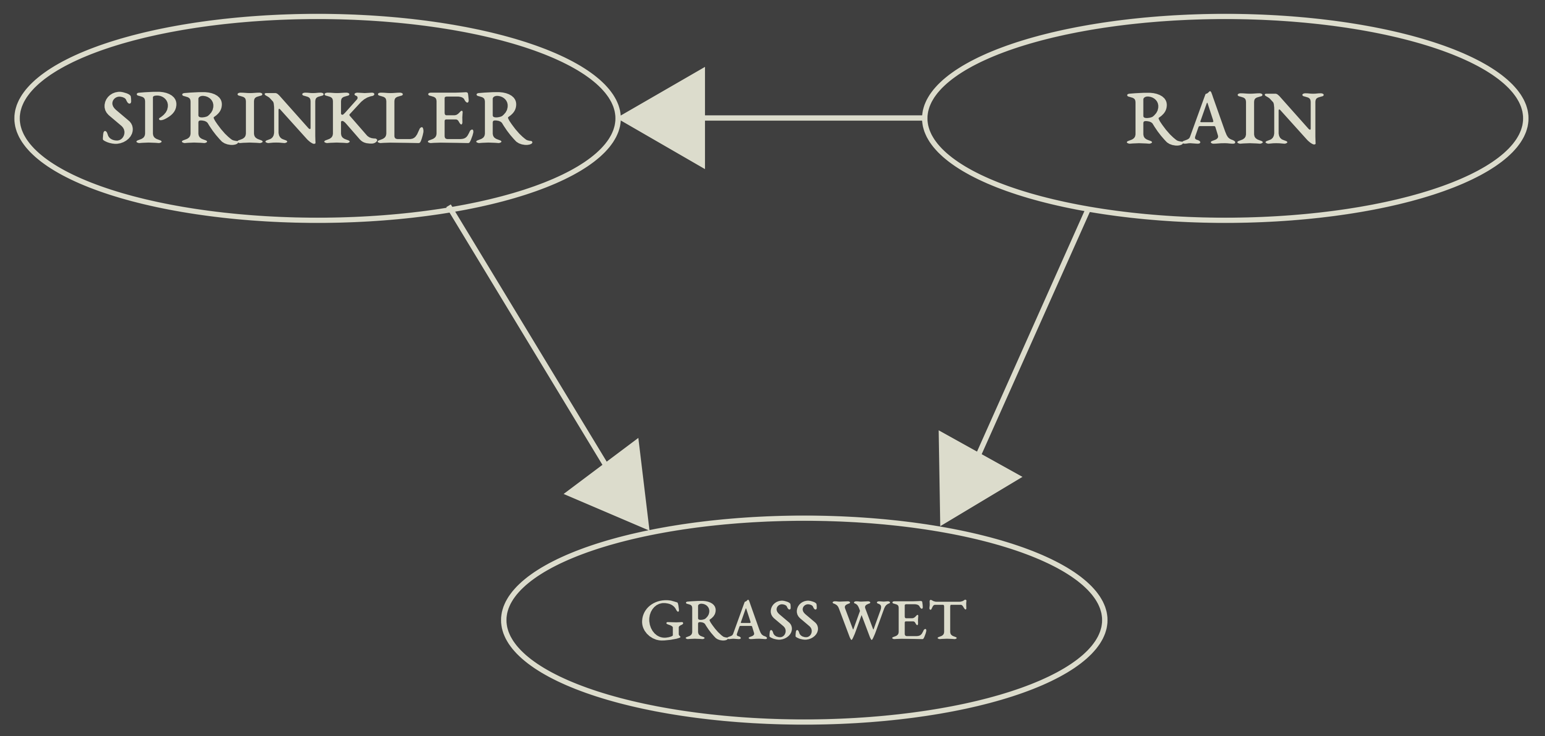
- 起源于知识工程应对不确定推理的不足
  - 决策支持系统（Decision Support System, DSS），1970s
- 先决条件是贝叶斯网的出现，1988

贝叶斯决策系统（Bayesian Decision Theory）：

1. 事件构成的集合：
   - 原因事件（causation, support）
   - 结果事件（outcome, decision）
2. 事件之间条件概率关系和概率分布，称为**模型**
3. 结果事件间的优先级

贝叶斯网

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



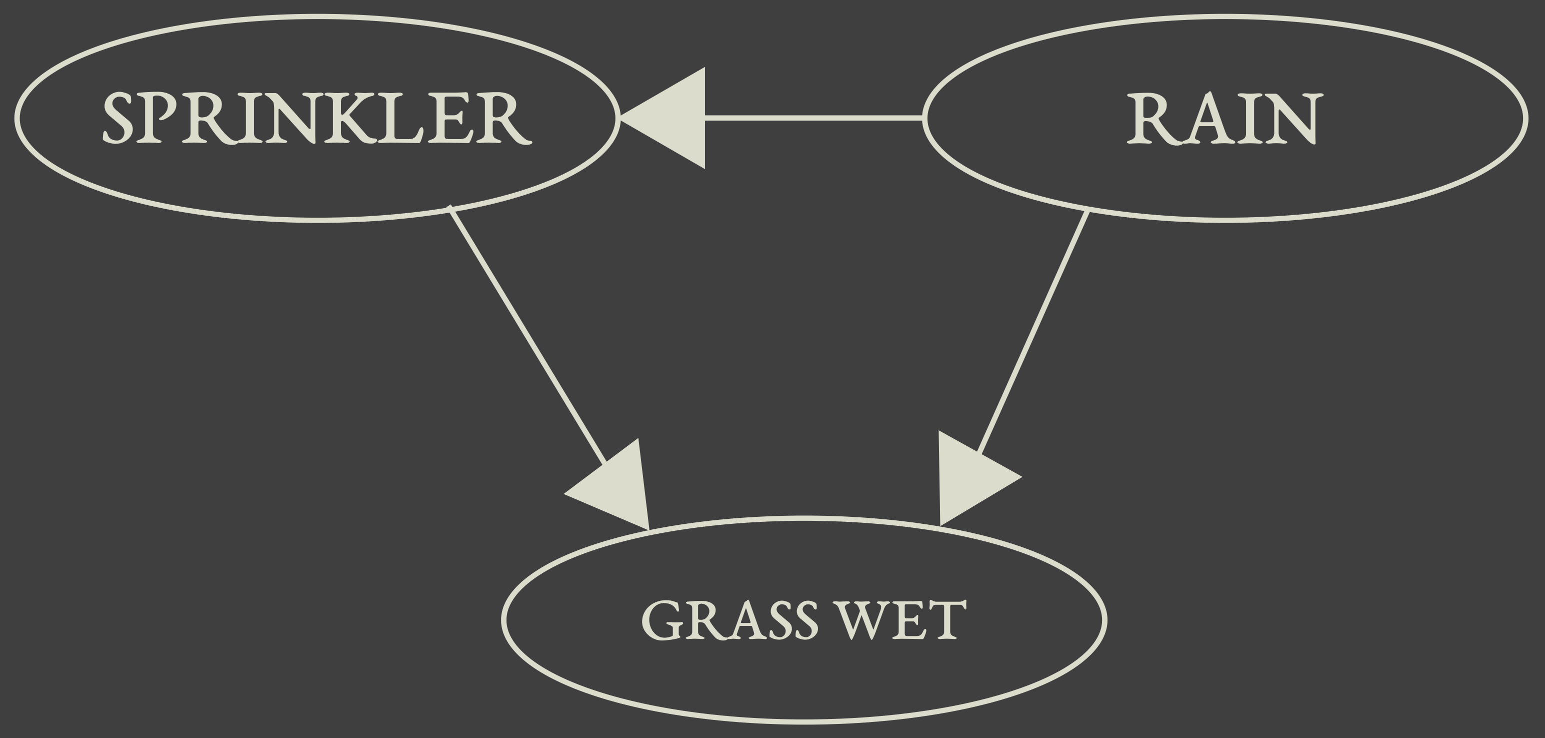
RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

图中的条件概率分布：$\Pr(R), \Pr(S \mid R), \Pr(G \mid S, R)$

贝叶斯网

RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99



RAIN T	RAIN F
0.2	0.8

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

图中的**联合概率分布**：$\Pr(G, S, R) = \Pr(R) \cdot \Pr(S \mid R) \cdot \Pr(G \mid S, R)$

贝叶斯网

1. 有向无环图（directed acyclic graph, DAG）
2. 每条有向边代表一个条件概率关系
3. 子节点与其所有父节点构成一个条件概率分布
4. 联合概率分布由所有条件概率分布组成

贝叶斯网的概率分布函数

给定一个图模型为$G=(V,E)$的贝叶斯网，其中V为节点（随机变量）集合、E为有向边集合，那么所有随机变量的联合概率分布为:

$$\Pr(X)=\prod_{v\in V}\Pr(X_v \mid \mathbf{X}_{\mathrm{pa}(v)})$$

其中X_v是v的随机变量取值，$\mathrm{pa}(v)$为v的全部父节点集合。

知识在概率模型中的作用

在概率模型中引入**知识**是为了简化推理的运算与存储复杂度。

例如：计算 n 个布尔随机变量构成的联合概率分布 $\Pr(X_1, X_2, X_3, \ldots, X_n)$

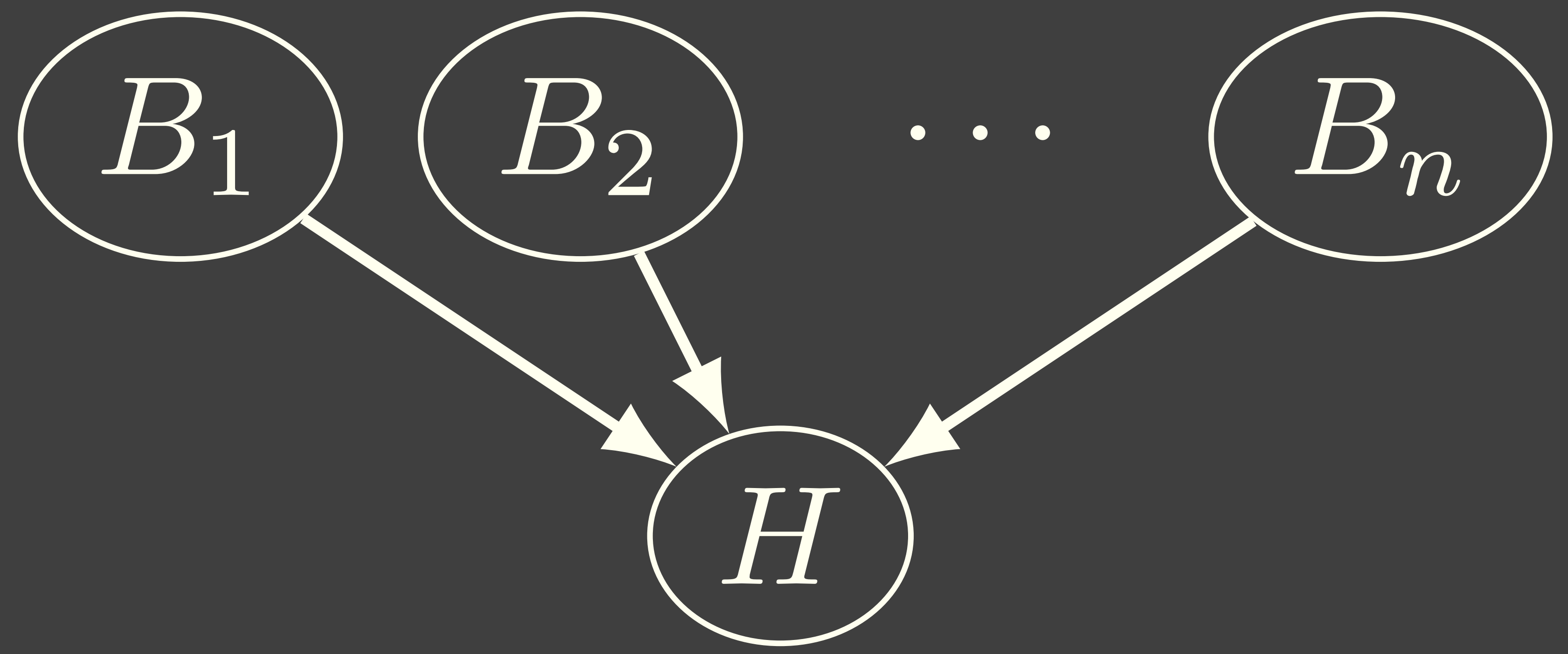
- 样本空间大小为 2^n
- 计算 X_i 为真的概率（边际概率）需要对 2^{n-1} 个事件的概率值求和：

$$\Pr(X_i) = \sum_{x_{j \neq i}} \Pr(X_1, X_2, \ldots, X_n)$$

知识可以描述随机变量间的独立关系，对联合概率进行分解

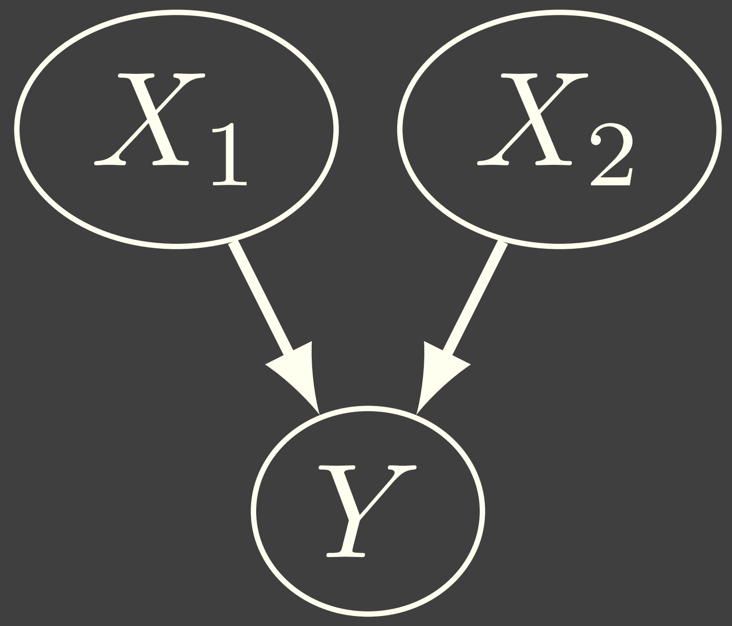
蕴涵的语义

$$\Pr(H \leftarrow B_1 \land B_2 \land \ldots \land B_n) \triangleq \Pr(H \mid B_1, B_2, \ldots, B_n)$$



Noisy Or

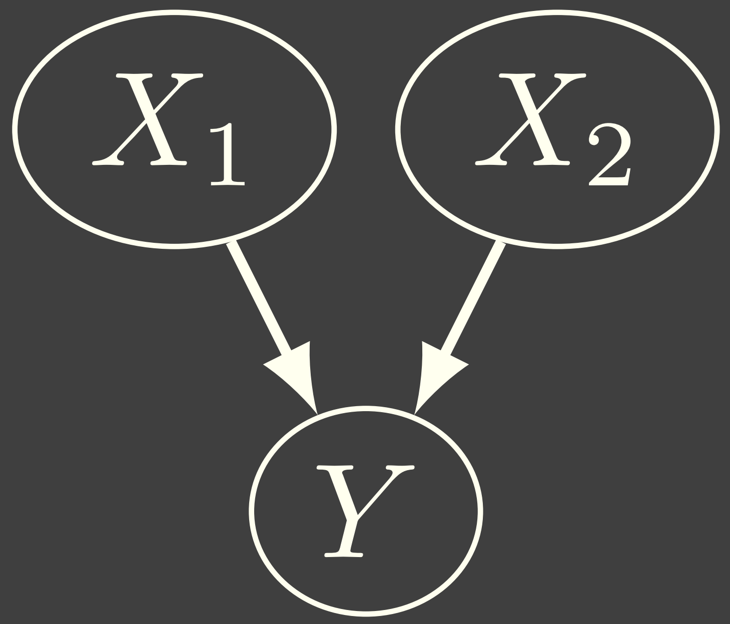
$$\Pr(X_1 \vee X_2 \vee \ldots \vee X_n) = 1 - \prod_{i=1}^{n} \Pr(\neg X_i)$$



- 原始的$\vee$若用贝叶斯网来表示需要$O(2^n)$个参数
- 引入独立性以后将参数缩小为$O(n)$个

Noisy And

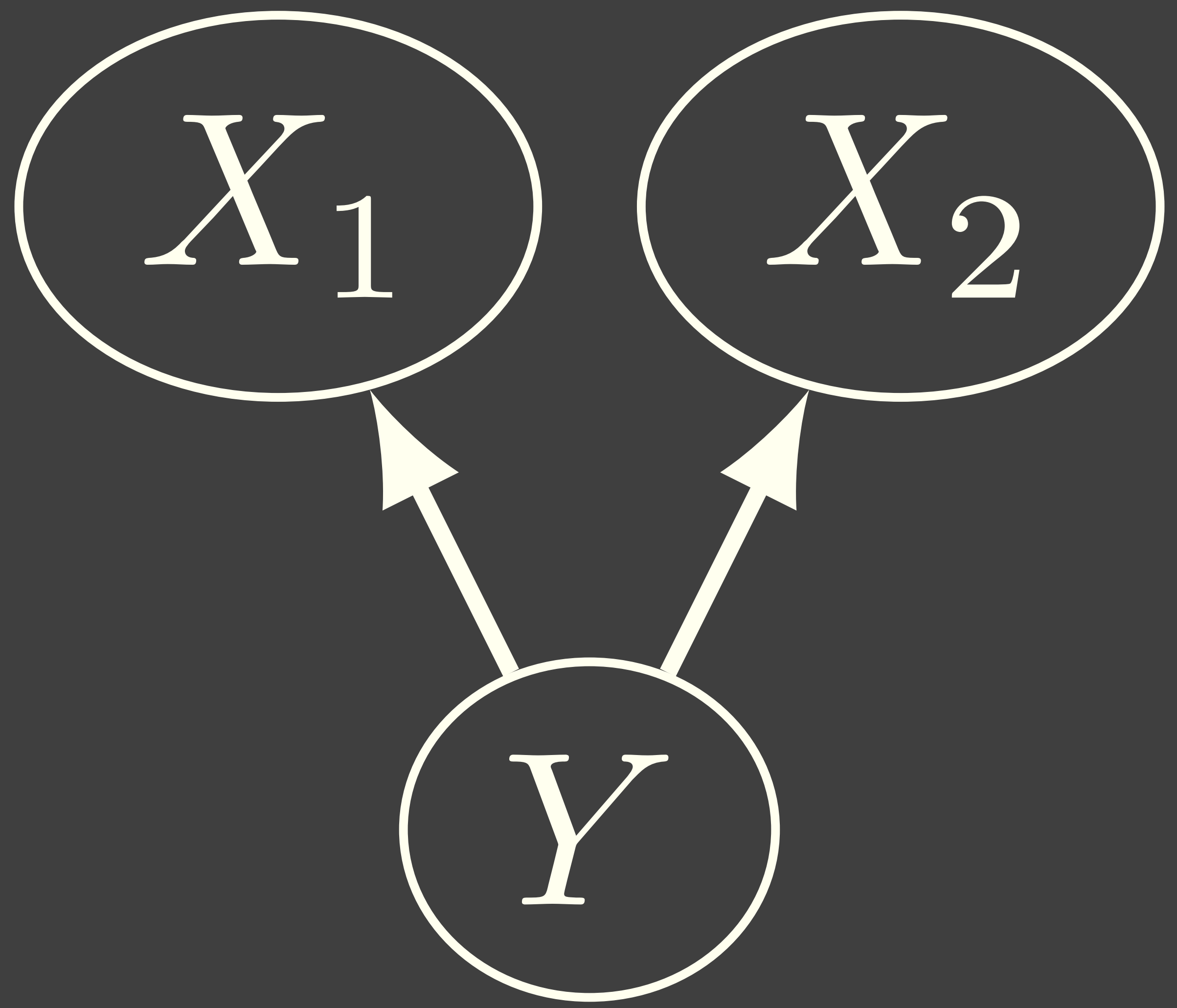
$$\Pr(X_1 \wedge X_2 \wedge \ldots \wedge X_n) = \prod_{i=1}^{n} \Pr(X_i)$$



- 原始的$\wedge$若用贝叶斯网来表示需要$O(2^n)$个参数
- 引入独立性以后将参数缩小为$O(n)$个

条件独立性

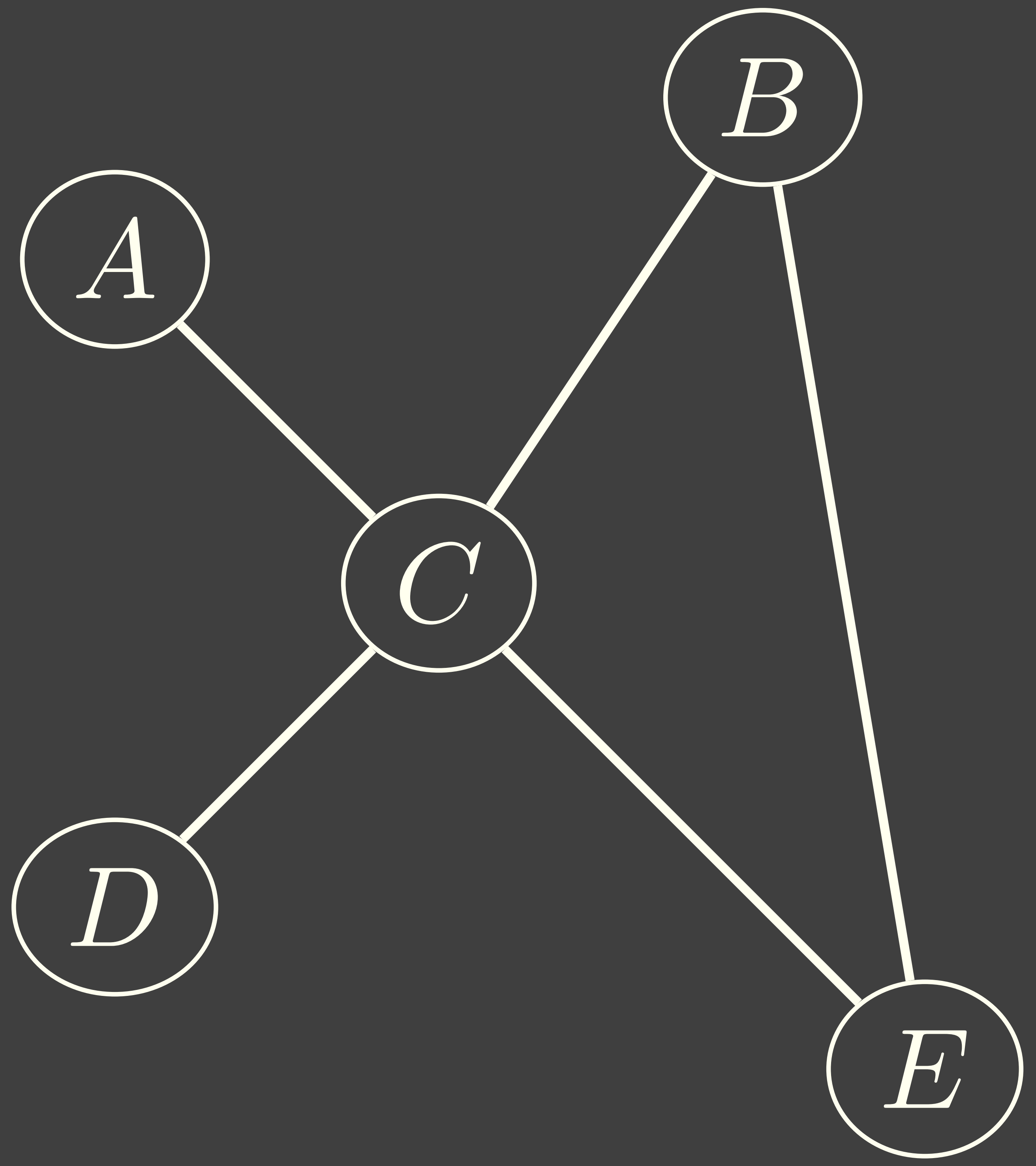
- $X_1 \top\!\!\!\top X_2$
- $X_1 \perp\!\!\!\perp X_2 \mid Y$



马尔可夫网

通过无向图构建随机变量间的依赖关系。

- 通过一种叫做**团**（clique）的子图来分解随机联合概率分布
- 每个团是一个**完全连通图**
- 团拥有自己的**势函数**，用来描述它对联合概率分布的贡献



马尔可夫网的概率分布函数

右图的联合概率分布定义如下：

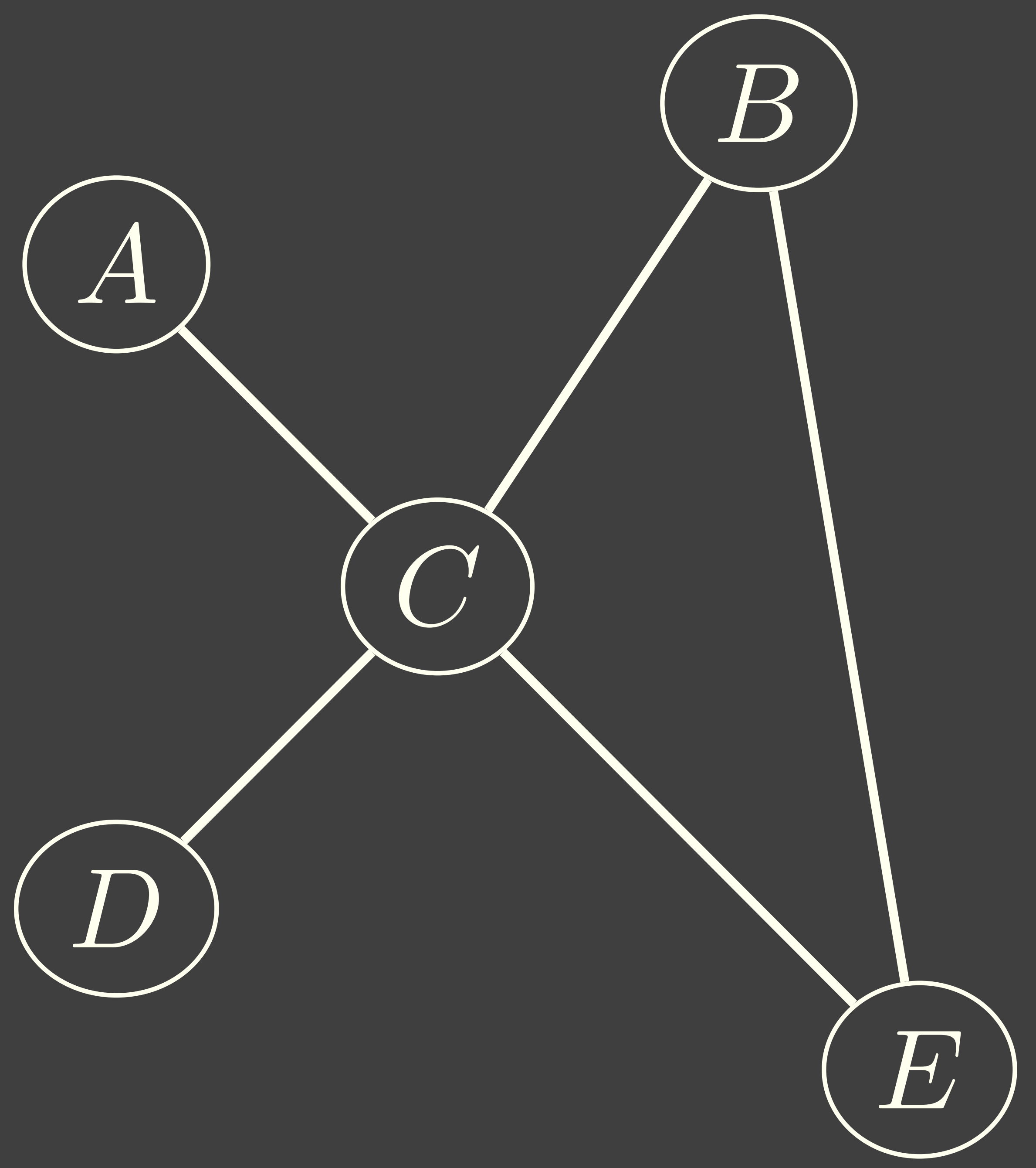
$$\Pr(A,B,C,D,E)=\frac{1}{Z}\phi(A,C)\phi(C,D)\phi(B,C,E)$$

其中 Z 为用来归一的**配分函数**（ partition function ）：

$$Z=\sum_{A,B,C,D,E}\phi_{AC}(A,C)\phi_{CD}(C,D)\phi_{BCE}(B,C,E)$$

而 ϕ 为每个团的势函数，例如：

A	C	$\phi_{AC}(A,C)$
0	0	1
0	1	10
1	0	10
1	1	100

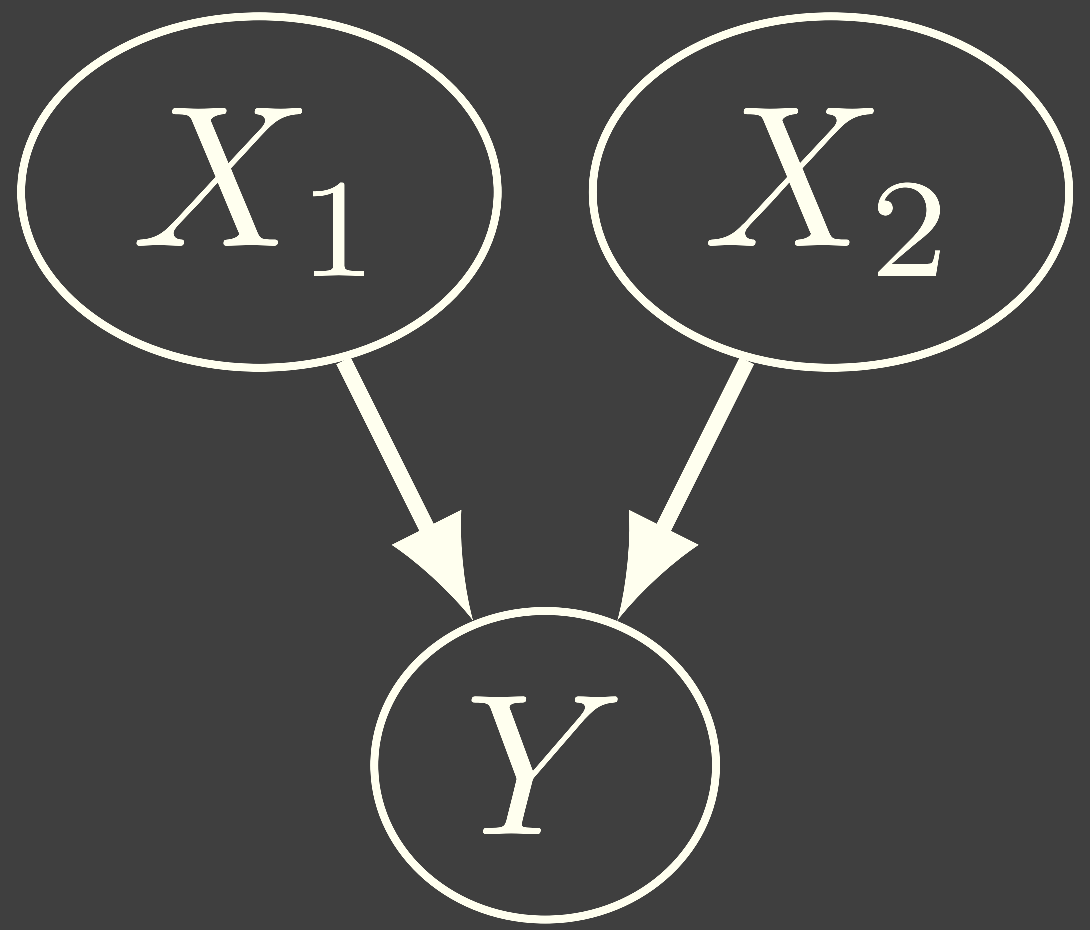


马尔可夫网中的条件独立

贝叶斯网中的条件概率

$$\Pr(Y \mid X_1, X_2) = \phi_{X_1 X_2 Y}(X_1, X_2, Y)$$

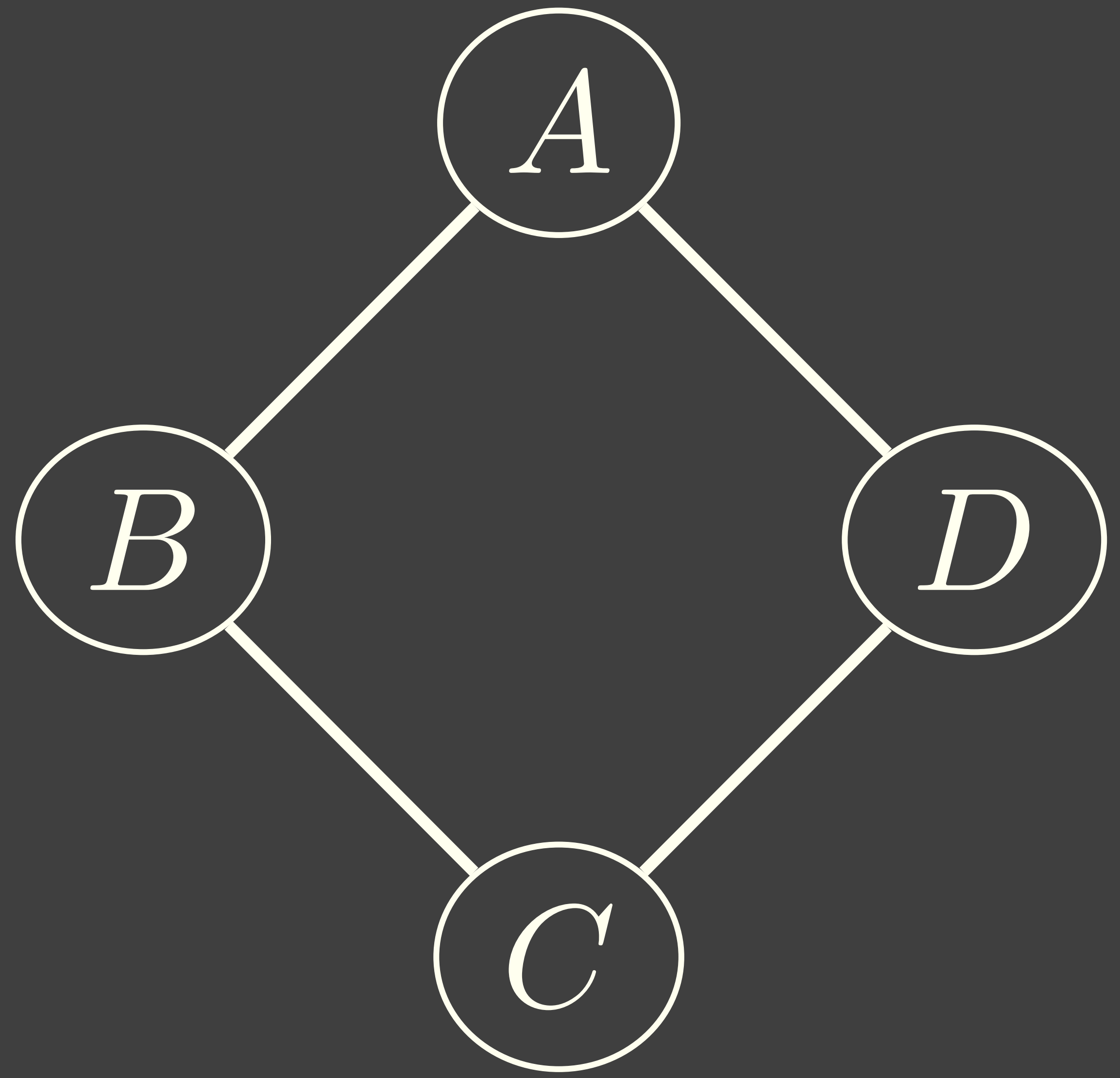
因此，在马尔可夫网中右图会变成一个大小为3的团。



马尔可夫网中的条件独立

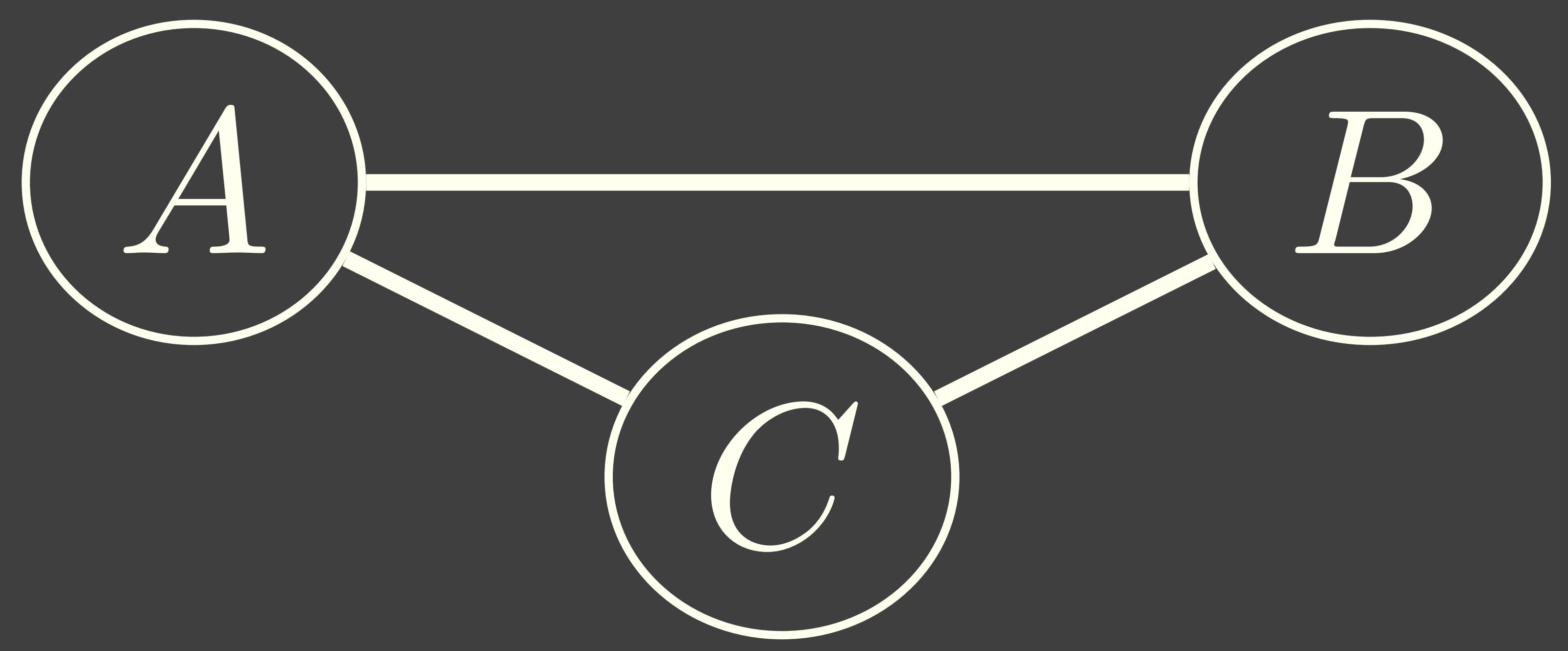
一个节点所有相邻的节点被称为它的**马尔可夫毯**（Markov Blanket）。给定马尔可夫毯时，该节点与图中其他节点条件独立。

$$\begin{aligned}
\Pr(B \mid A, C, D) &= \frac{\mathcal{P}r(A, B, C, D)}{\mathcal{P}r(A, C, D)} \\
&= \frac{\phi_{AB}(A, B)\phi_{BC}(B, C)\phi_{CD}(C, D)\phi_{AD}(A, D)}{\phi_{CD}(C, D)\phi_{AD}(A, D)\sum_B \phi_{AB}(A, B)\phi_{BC}(B, C)} \\
&= \frac{\phi_{AB}(A, B)\phi_{BC}(B, C)}{\sum_B \phi_{AB}(A, B)\phi_{BC}(B, C)} \\
&= \Pr(B \mid A, C)
\end{aligned}$$



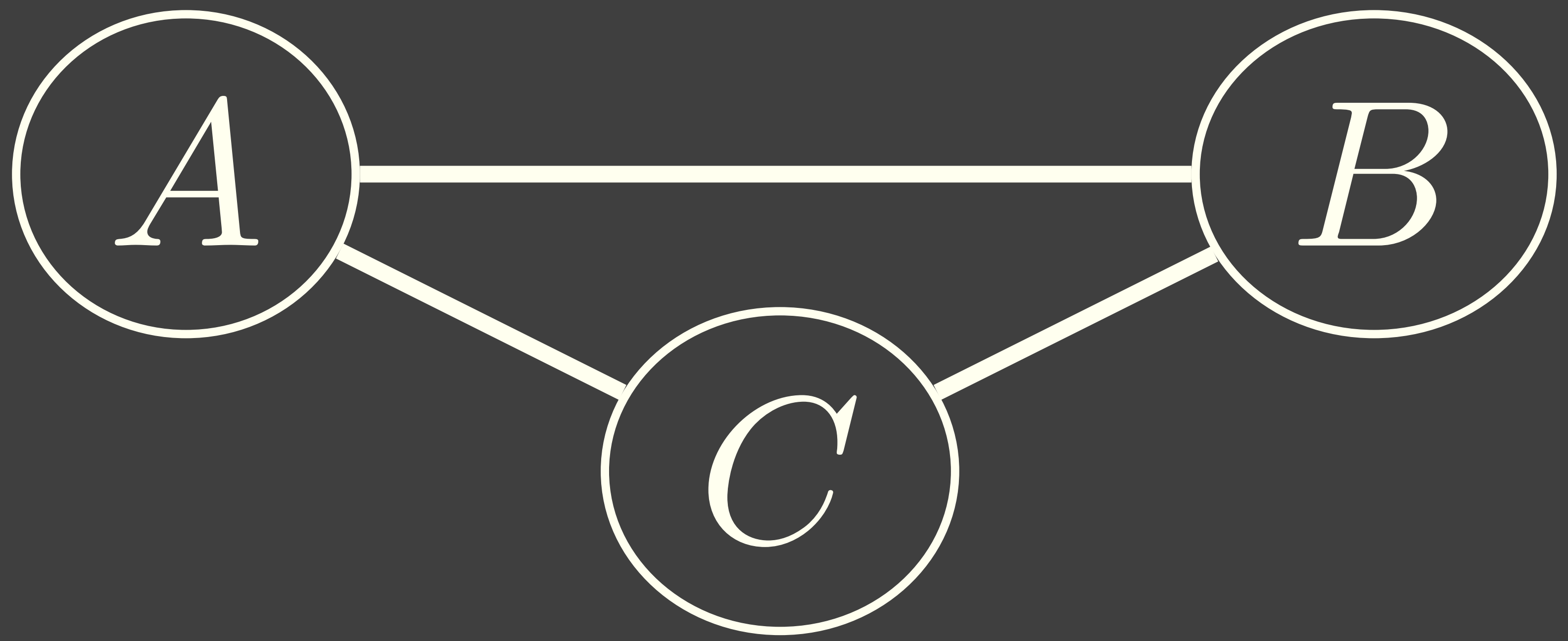
合取的语义

$$\Pr(A \wedge B \wedge C) \propto \phi_{ABC}(A, B, C)$$



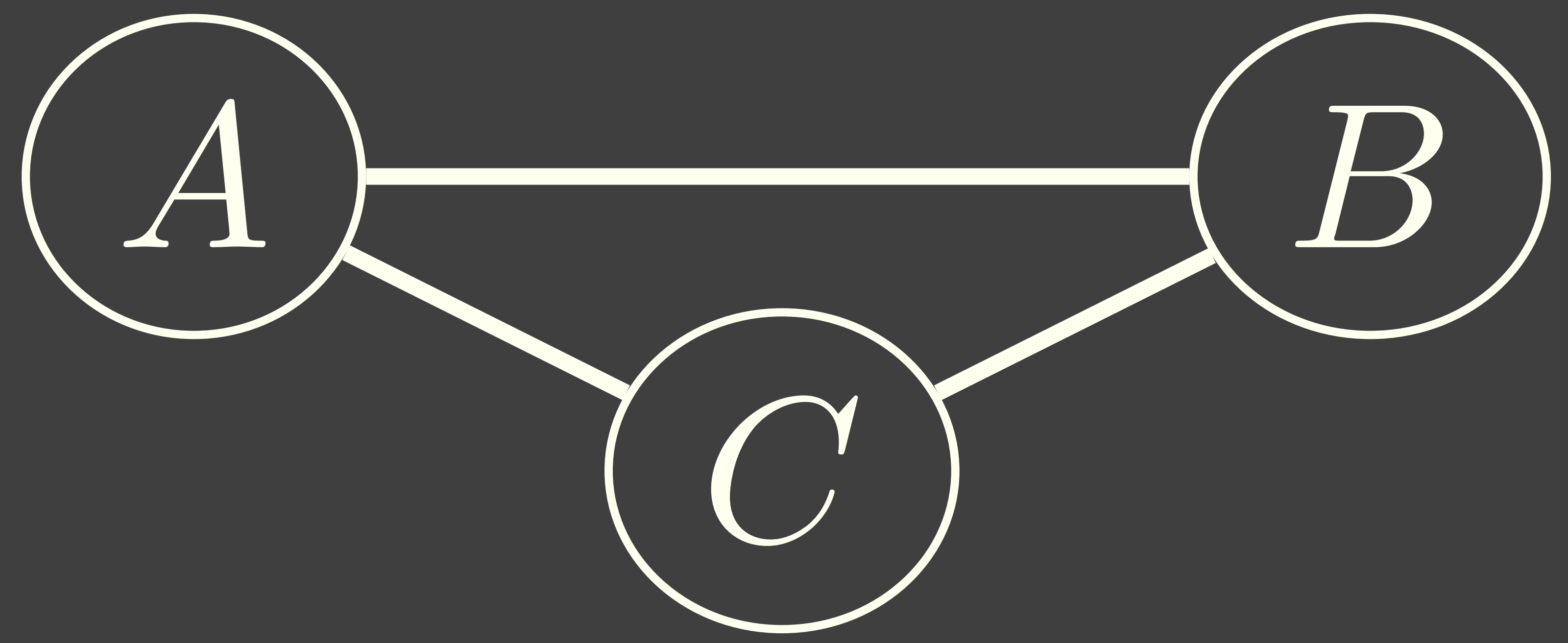
析取的语义

$$\Pr(A \vee B \vee C) \propto \phi_{ABC}(A, B, C)$$



蕴涵的语义

$$\Pr(A \leftarrow B \wedge C) \propto \phi_{ABC}(A, B, C)$$

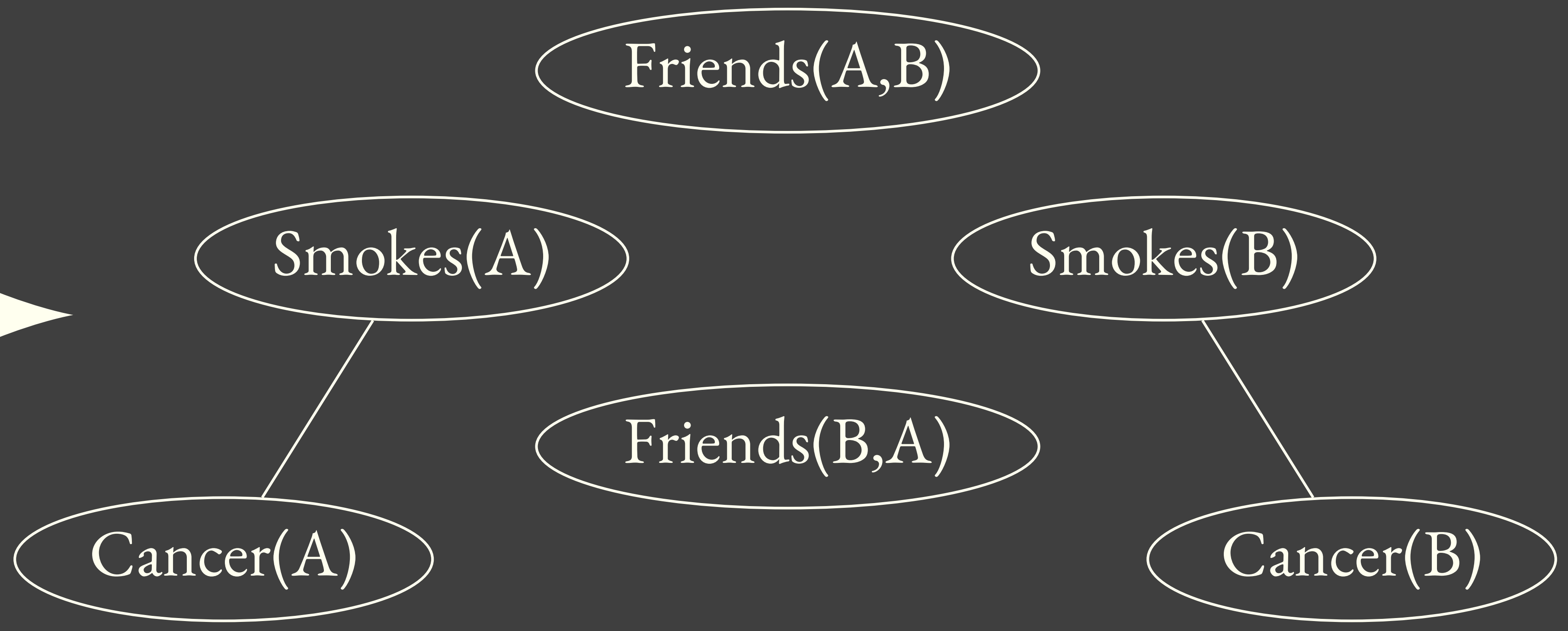


马尔可夫逻辑网

将逻辑公式的语义用马尔可夫网表达，即得到**马尔可夫逻辑网**（Markov Logic Network, MLN）。

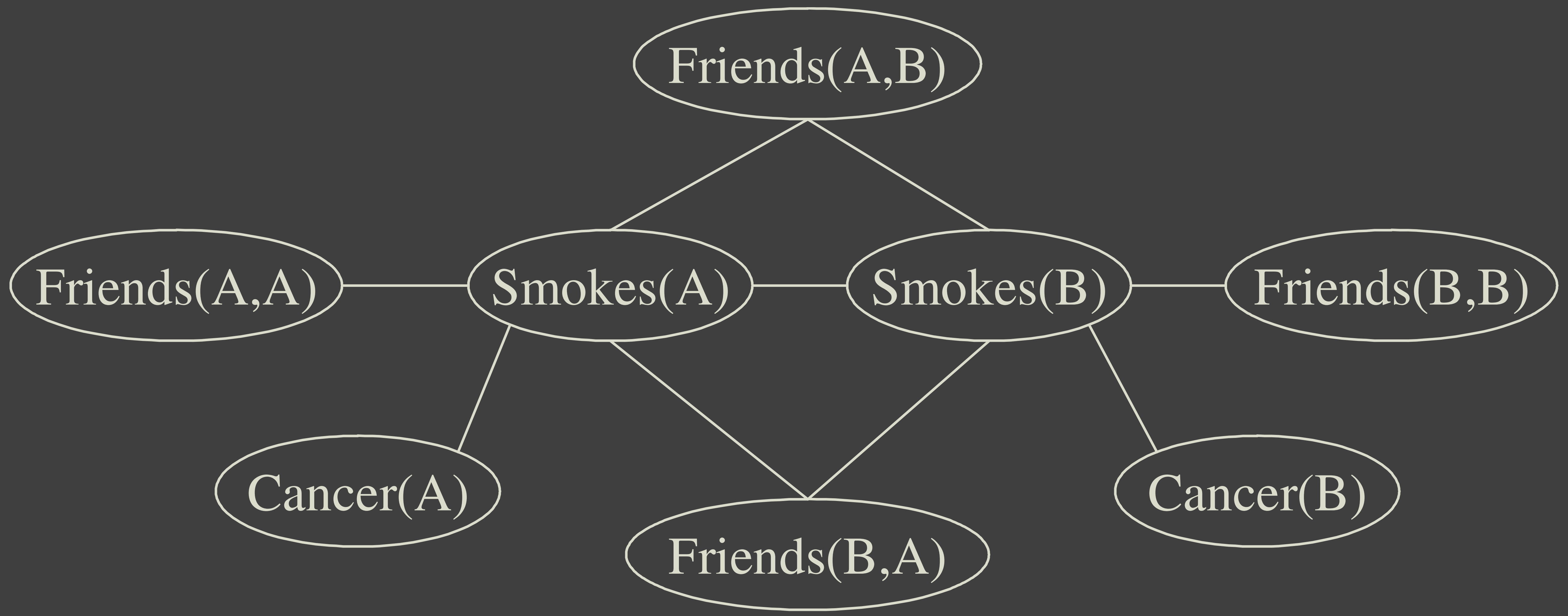
- MLN 可看作用一阶逻辑公式作为**模板**生成的具体马尔可夫网
- 将每个具体逻辑公式作为一个**团**
- 一阶逻辑公式的权重作为势函数的来源

$$\forall x\ Smokes(x) \Rightarrow Cancer(x)$$



马尔可夫逻辑网

English	First-Order Logic	Clausal Form	Weight
Friends of friends are friends.	$\forall x \forall y \forall z\, \mathtt{Fr}(x,y) \land \mathtt{Fr}(y,z) \Rightarrow \mathtt{Fr}(x,z)$	$\neg\mathtt{Fr}(x,y) \lor \neg\mathtt{Fr}(y,z) \lor \mathtt{Fr}(x,z)$	0.7
Friendless people smoke.	$\forall x\, (\neg(\exists y\, \mathtt{Fr}(x,y)) \Rightarrow \mathtt{Sm}(x))$	$\mathtt{Fr}(x, g(x)) \lor \mathtt{Sm}(x)$	2.3
Smoking causes cancer.	$\forall x\, \mathtt{Sm}(x) \Rightarrow \mathtt{Ca}(x)$	$\neg\mathtt{Sm}(x) \lor \mathtt{Ca}(x)$	1.5
If two people are friends, either	$\forall x \forall y\, \mathtt{Fr}(x,y) \Rightarrow (\mathtt{Sm}(x) \Leftrightarrow \mathtt{Sm}(y))$	$\neg\mathtt{Fr}(x,y) \lor \mathtt{Sm}(x) \lor \neg\mathtt{Sm}(y)$,	1.1
both smoke or neither does.		$\neg\mathtt{Fr}(x,y) \lor \neg\mathtt{Sm}(x) \lor \mathtt{Sm}(y)$	1.1



马尔可夫逻辑网

马尔可夫逻辑网的概率分布函数:

$$\Pr(\mathbf{X}) = \frac{1}{Z}\prod_i \phi_i(\mathbf{X}) = \frac{1}{Z}\exp\left(\sum_i w_i n_i(\mathbf{X})\right)$$

- $\mathbf{X}$ 为所有命题原子（随机变量）的集合
- ϕ_i 为第 i 个公式的势函数，是一个指数函数
  - w_i 为第 i 个命题公式的权重
  - $n_i(\mathbf{X})$ 是随机变量取值为\mathbf{X}时，命题公式 i（子句形式）中为真的原子命题个数

符号学习简介·概率逻辑系统

1. 从非经典逻辑到概率逻辑
2. KBMC
3. **分布语义**

用逻辑来表达概率

公式 φ 为真的概率至少为 $\frac{1}{2}$

$$w(\varphi) \geq \frac{1}{2}$$

用逻辑来表达概率

公式φ_1为真的概率至少为φ_2的两倍

$$w(\varphi_1) \geq 2w(\varphi_2)$$

用逻辑来表达概率

$$M \models a_1 w(\varphi_1) + \ldots + a_k w(\varphi_k) \geq c$$

用逻辑来表达概率

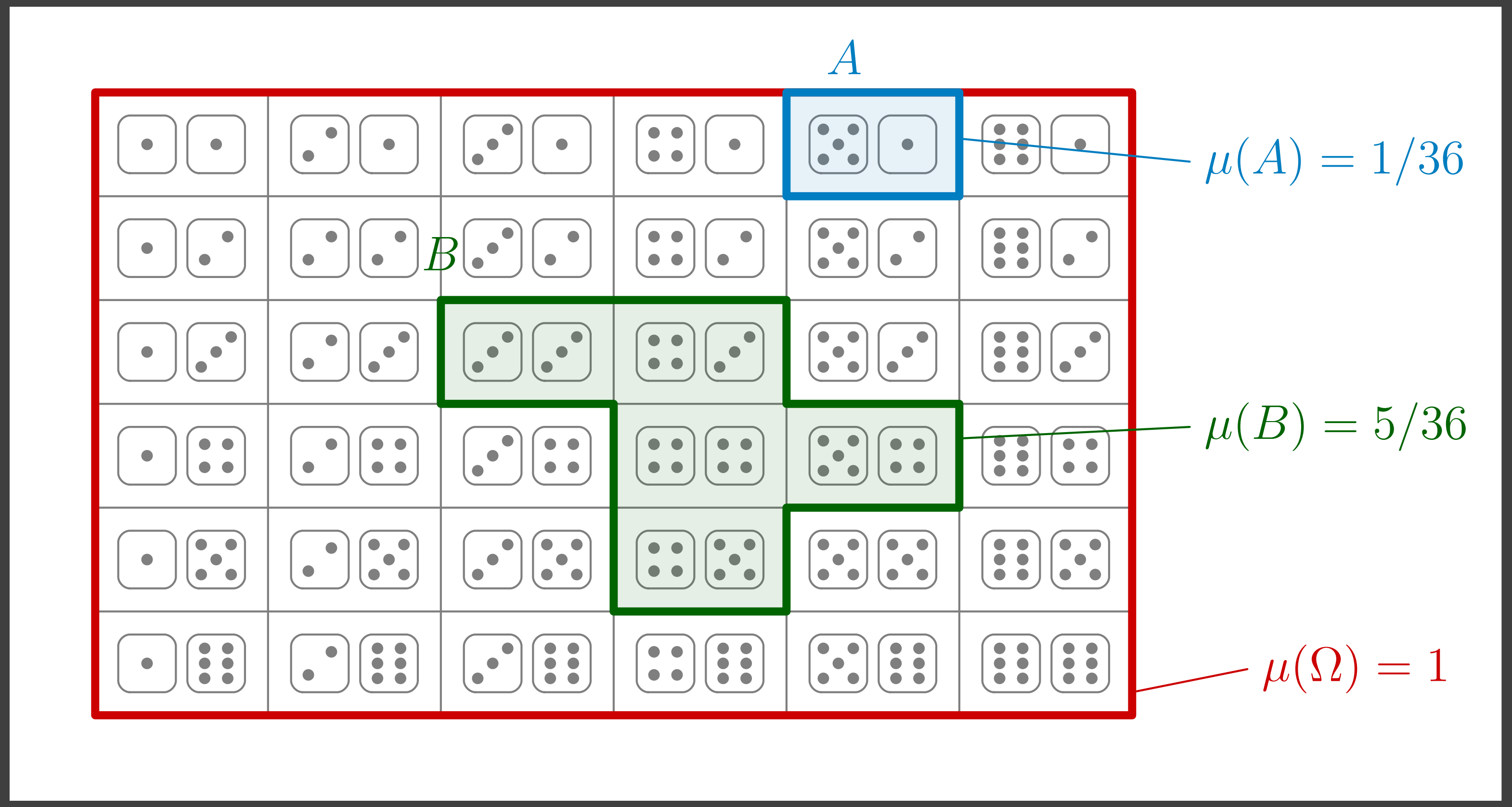
M 是什么？

概率空间

概率空间是一个三元组$(\Omega, \mathcal{F}, \Pr)$，其中：

- Ω是一个非空集，被称为**样本空间**
- $\mathcal{F} \subseteq 2^{\Omega}$是一个$\sigma$-代数，代表**可能事件**
  - 它关于可数并和补运算封闭
- $\Pr : \mathcal{F} \mapsto [0,1]$是**概率测度函数**
  - $\Pr(\Omega) = 1$
  - 它可数可加：若$\{A_i\}_{i=1}^{\infty} \subseteq \mathcal{F}$两两不相交，则

$$\Pr(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \Pr(A_i)$$



概率逻辑语言

我们仍然有一个自由代数，它包含一系列基本符号：

- $\Phi = \{p_1, p_2, ...\}$为所有的原子命题
- 实数符号
- $\neg$与$\vee$为逻辑连词
- $w(\varphi)$是一个权重函数，φ为以上符号构成的公式
- 算数符号
- 大小于符号

概率逻辑结构

四元组$M=(\Omega,\mathcal{F},\Pr,\pi)$是一个**概率逻辑结构**（probabilistic logic structure），其中:

- $(\Omega,\mathcal{F},\Pr)$是一个概率空间
  - Ω是关于Φ的所有**可能世界**
  - $\mathcal{F}\subseteq 2^{\Omega}$是一个$\sigma$-代数
  - $\Pr:\mathcal{F}\mapsto[0,1]$是概率测度函数
- π为**真值指派函数**，例如:
  - 原子命题$p\in\Phi$在可能世界$s\in\Omega$中的真值$\pi(s)(p)\in\{True,False\}$

概率逻辑结构的语义

定义：对于$p \in \Phi$和它们组成的公式φ：

- $p^M = \{s \in \Omega \mid \pi(s)(p) = True\}$
- $\varphi^M = \{s \in \Omega \mid \pi(s)(\varphi) = True\}$
- $w^M(p) = \mu(p^M)$
- $w^M(\varphi) = \mu(\varphi^M)$

概率逻辑结构的语义

$$M \models a_1 w(\varphi_1) + \ldots + a_k w(\varphi_k) \geq c$$

当且仅当

$$a_1 w^M(\varphi_1) + \ldots + a_k w^M(\varphi_k) \geq c$$

基于可能世界的推理

考虑一个逻辑理论的Herbrand universe：

1. 每个具体事实可看作一个随机变量；
2. 对它们的真值指派可看作一个**可能世界**（Possible world）；
3. 每个可能世界中均能够进行纯逻辑推理；
4. 对问句（query）求它在所有可能世界中为真的期望；
5. 概率分布只需定义在该Herbrand universe上即可

这种概率逻辑语义被称为“分布语义”（Distribution Semantics）

分布语义的公理系统

分布语义涵盖了命题逻辑与概率论公理，具体包含了：

1. 命题逻辑公理
   - 所有命题逻辑重言式
   - MP规则
2. 关于线性不等式的一切重言式
3. 关于概率论的公理：
   - $0 \le w(\varphi) \le 1$
   - $w(True) = 1$
   - $w(\varphi \wedge \psi) + w(\varphi \wedge \neg\psi) = w(\varphi)$
   - $w(\varphi) = w(\psi)$仅当$\varphi \equiv \psi$是一个命题逻辑中的重言式

该公理系统可靠（sound）且完备（complete）

概率逻辑程序的例子

```
%% Probabilistic facts:
0.5::heads1.
0.6::heads2.

%% Rules:
twoHeads :- heads1, heads2.
someHeads :- heads1.
someHeads :- heads2.

%% Queries:
query(twoHeads).
query(someHeads).
```

```
prob(twoHeads, 0.3).
prob(someHeads, 0.8).
```

概率逻辑程序的例子

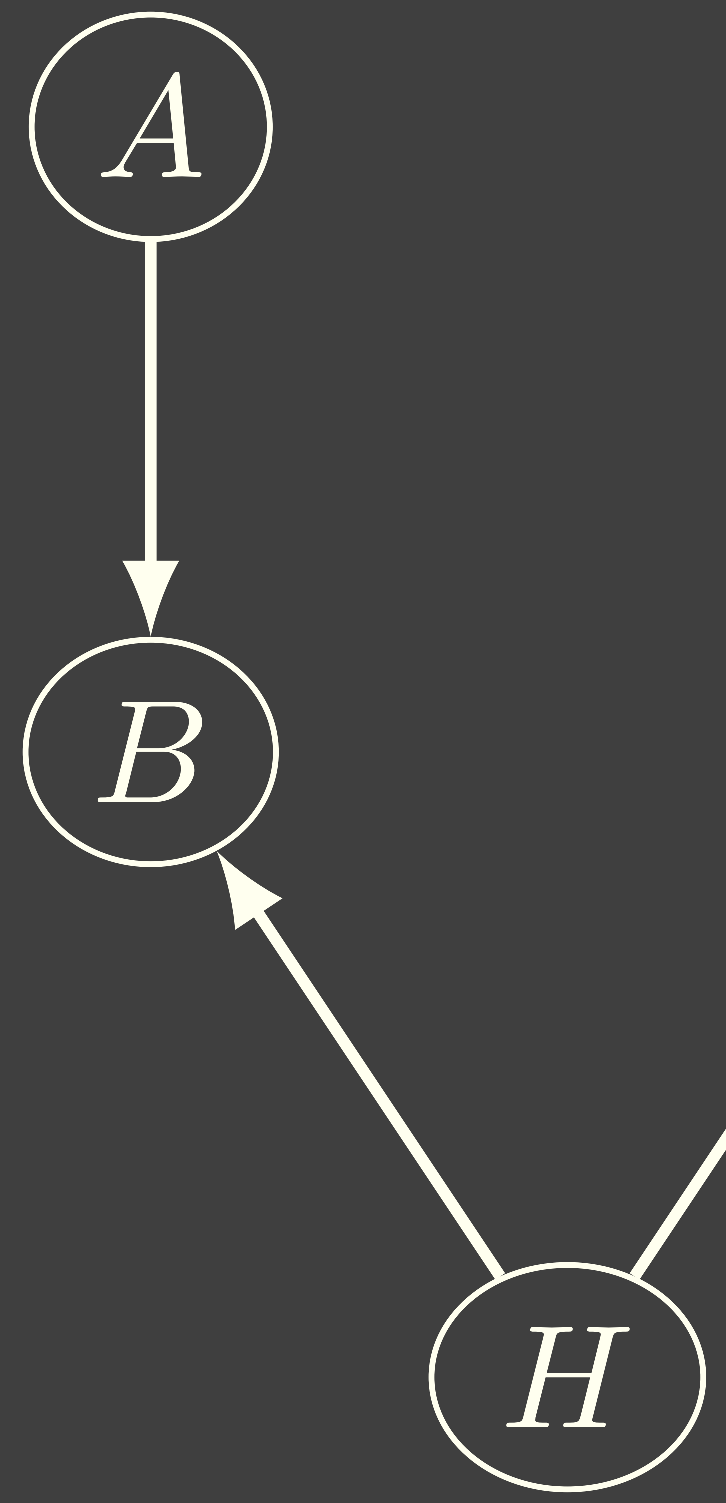
```
%% Probabilistic rules:
0.3::stress(X) :- person(X).
0.2::influences(X,Y) :- person(X), person(Y).
0.4::cancer(X) :- smokes(X).

%% Non-probabilistic rules:
smokes(X) :- stress(X).
smokes(X) :- friend(X,Y), influences(Y,X), smokes(Y).

%% Facts
person(angelika).
person(joris).
person(jonas).
person(dimitar).

friend(joris,jonas).
friend(joris,angelika).
friend(joris,dimitar).
friend(angelika,jonas).

%% Query
```

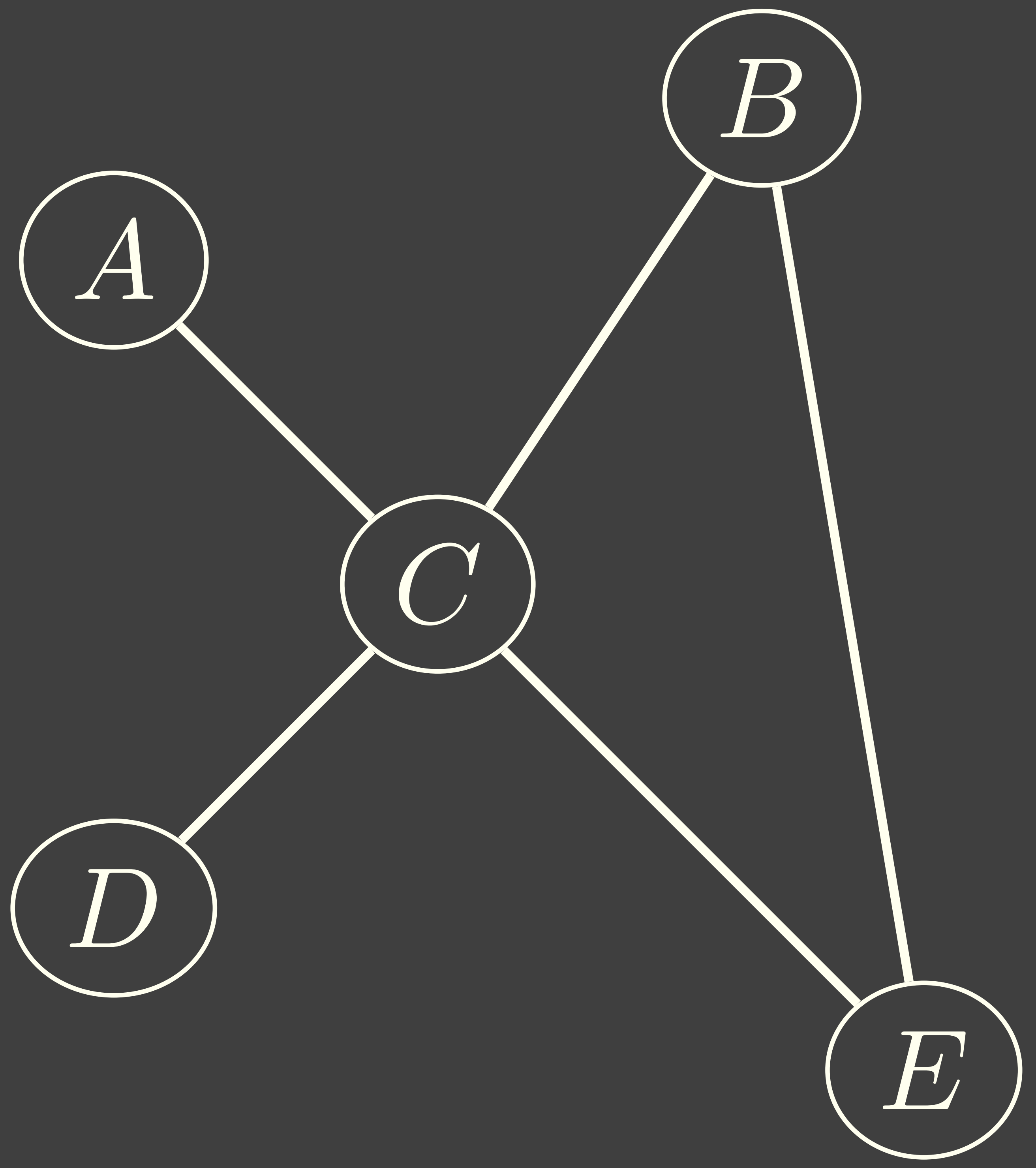
符号学习简介 · 概率逻辑推断

1. **图模型推断**
2. 从SAT到WMC
3. 知识编译

统计关系模型

Startistical Relational AI (StaRAI)的本质是将ground rules转换为布尔随机变量构成的概率图模型（ Probabilistic Graphical Model, PGM ）。





PGM 中的推理

统计推断（Statistical Inference）：

- 通过基于概率分布的计算来回答关于领域的问题。

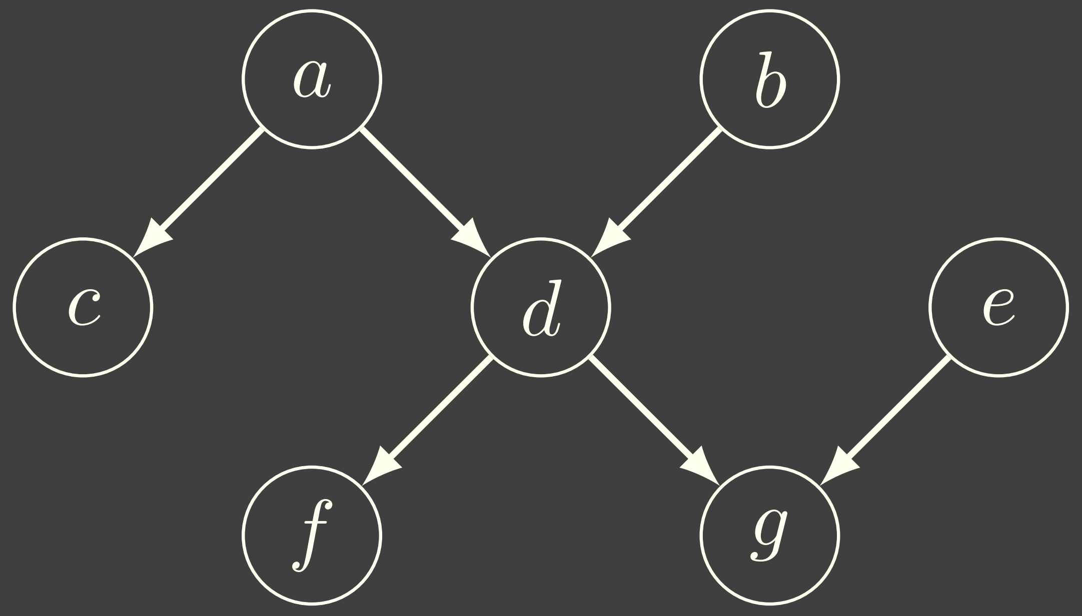
1. $\Pr(x=1 \mid y=4, z=3)$ 的概率是多少？
2. $\Pr(x, y)$ 联合概率分布中最可能出现的 (X, Y) 是什么？
3. 联合概率分布 $\Pr(x, y, z)$ 的熵是多少？
4. 明天大盘下跌的概率是多少？

消元法

计算如下边际概率：

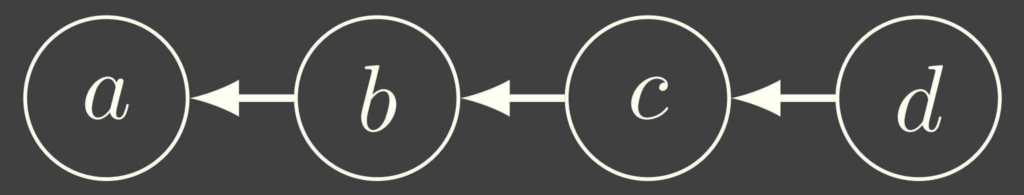
$$\Pr(f) = \sum_{a,b,c,d,e,g} \Pr(a, b, c, d, e, f, g)$$

$$= \sum_{a,b,c,d,e,g} \Pr(f \mid d)\Pr(g \mid d, e)\Pr(c \mid a)\Pr(d \mid a, b)\Pr(a)\Pr(b)\Pr(e)$$



消息传递

将边际概率看作“消息传递”（message passing）。考虑如下布尔马尔可夫链：



其联合概率分布的分解形式为：$\Pr(a, b, c, d) = \Pr(a \mid b)\Pr(b \mid c)\Pr(c \mid d)\Pr(d)$

计算$\Pr(a = 0) = ?$：

1. **第一种方式**：直接计算$\Pr(a = 0) = \sum_{b,c,d} \Pr(a = 0 \mid b)\Pr(b \mid c)\Pr(c \mid d)\Pr(d)$
   - 共需要计算7次求和。
2. **第二种方式**：将求和操作分解
   - 先将d推进去：

$$\Pr(a = 0) = \sum_{b,c} \Pr(a = 0 \mid b)\Pr(b \mid c)\underbrace{\sum_{d} \Pr(c \mid d)\Pr(d)}_{\gamma_d(c)}$$

   - 对于$\gamma_d(c)$需要计算2次求和。

消息传递

1. 先将$\sum_d$推进去：

$$\Pr(a=0)=\sum_{b,c}\Pr(a=0\mid b)\Pr(b\mid c)\underbrace{\sum_{d}\Pr(c\mid d)\Pr(d)}_{\gamma_d(c)}$$

   - 对于$\gamma_d(c)$需要计算2次求和。

2. 再将$\sum_c$推进去：

$$\Pr(a=0)=\sum_{b}\Pr(a=0\mid b)\underbrace{\sum_{c}\Pr(b\mid c)\gamma_d(c)}_{\gamma_c(b)}$$

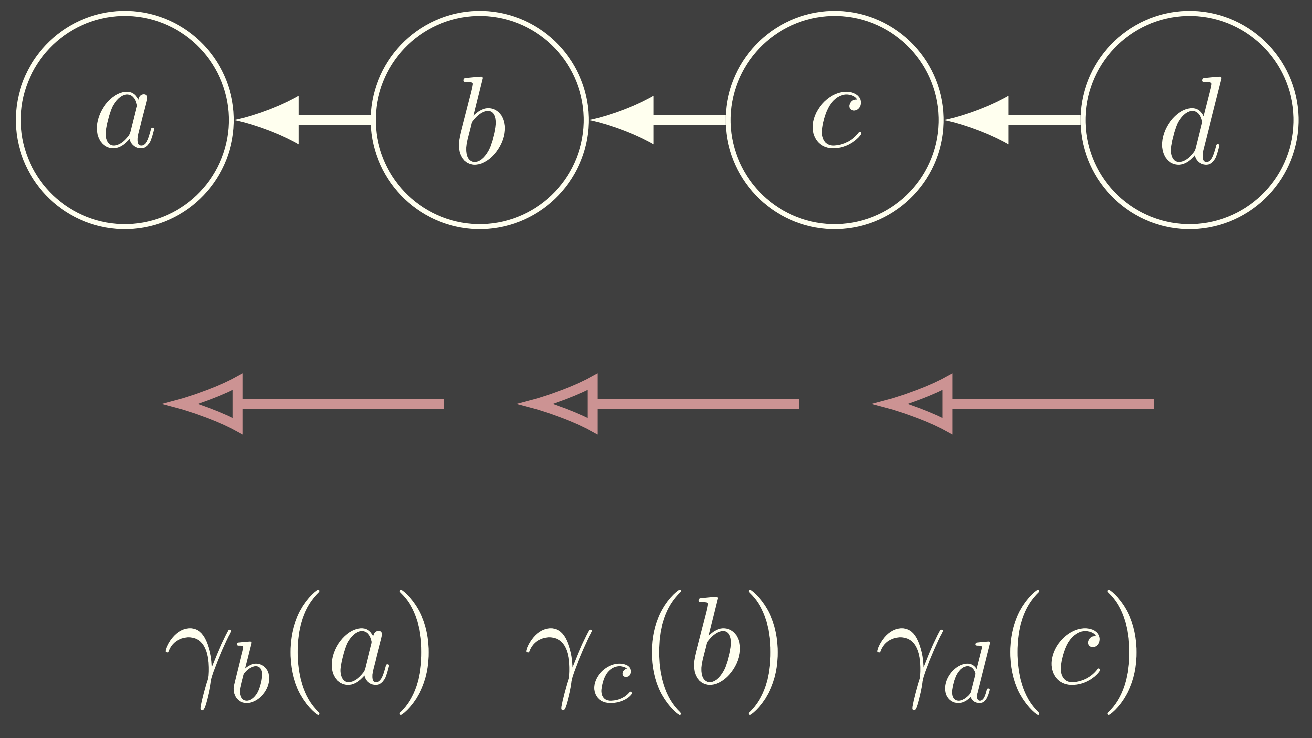
   - 对于$\gamma_c(b)$需要计算2次求和。

3. 最后将$\sum_b$推进去：

$$\Pr(a=0)=\underbrace{\sum_{b}\Pr(a=0\mid b)\gamma_c(b)}_{\gamma_b(a)}$$

   - 对于$\gamma_b(a)$需要计算1次求和。

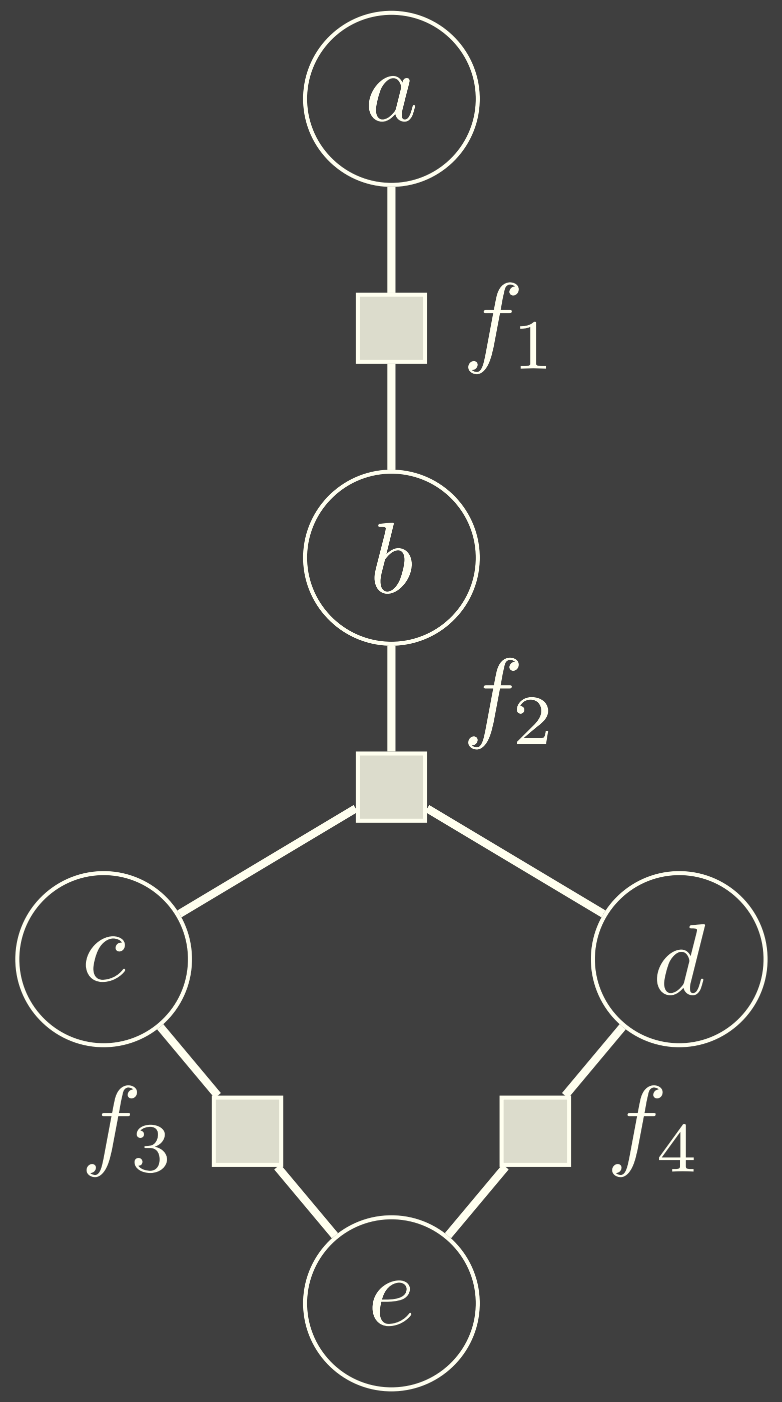
消息传递



$$\Pr(a=0) = \underbrace{\sum_b \Pr(a=0 \mid b) \underbrace{\sum_c \Pr(b \mid c) \underbrace{\sum_d \Pr(c \mid d)\Pr(d)}_{\gamma_d(c)}}_{\gamma_c(b)}}_{\gamma_b(a)}$$

因子图

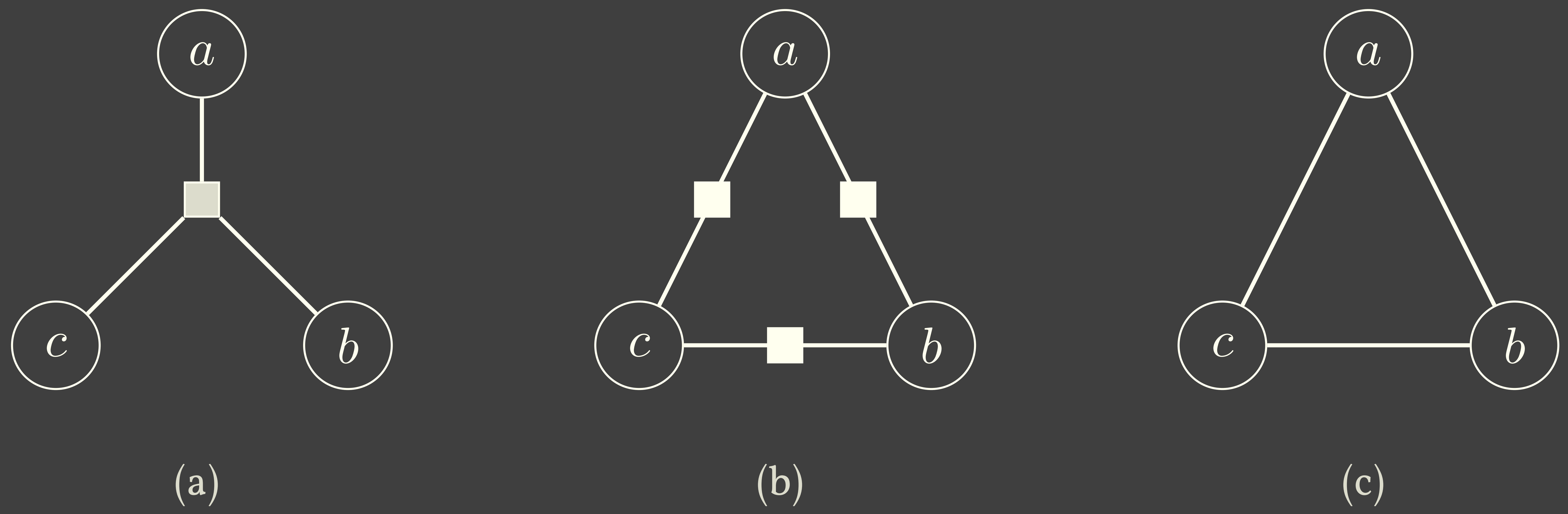
图模型中用方形节点表示其相邻节点随机变量构成的因子（factor），它是一个非负函数。



左边因子图表示是4个因子的乘积：

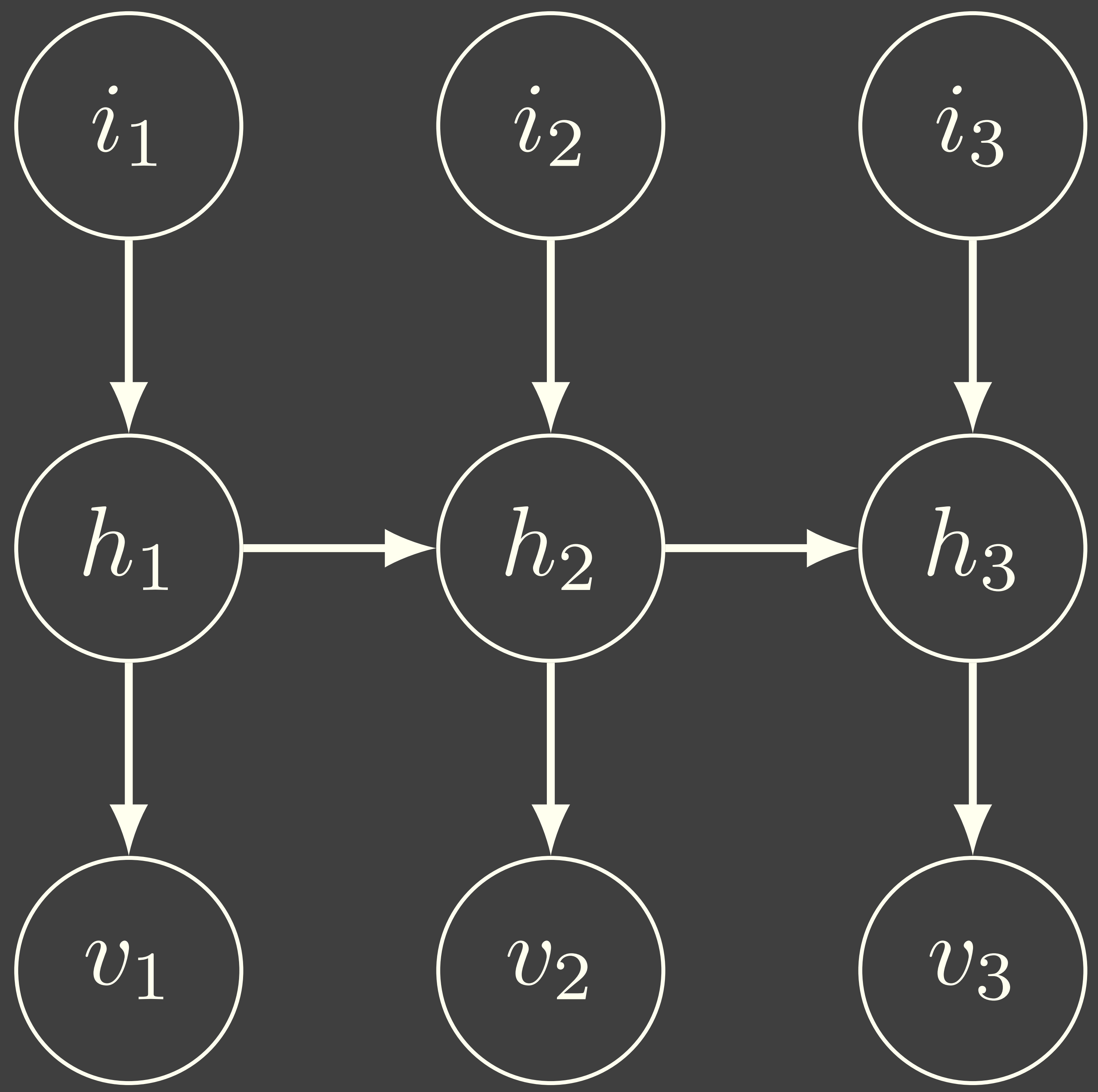
$$f(a,b,c,d,e)=f_1(a,b)f_2(b,c,d)f_3(c,e)f_4(d,e)$$

马尔可夫网的因子图



- (a)的势函数因子：$\phi(a, b, c)$
- (b)的势函数因子：$\phi_1(a, b)\phi_2(b, c)\phi_3(a, c)$
- (a)和(b)有着相同的马尔可夫网结构(c)
- 但(b)表达了更多关于势函数的约束（例如逻辑规则）

隐马尔可夫模型的因子图

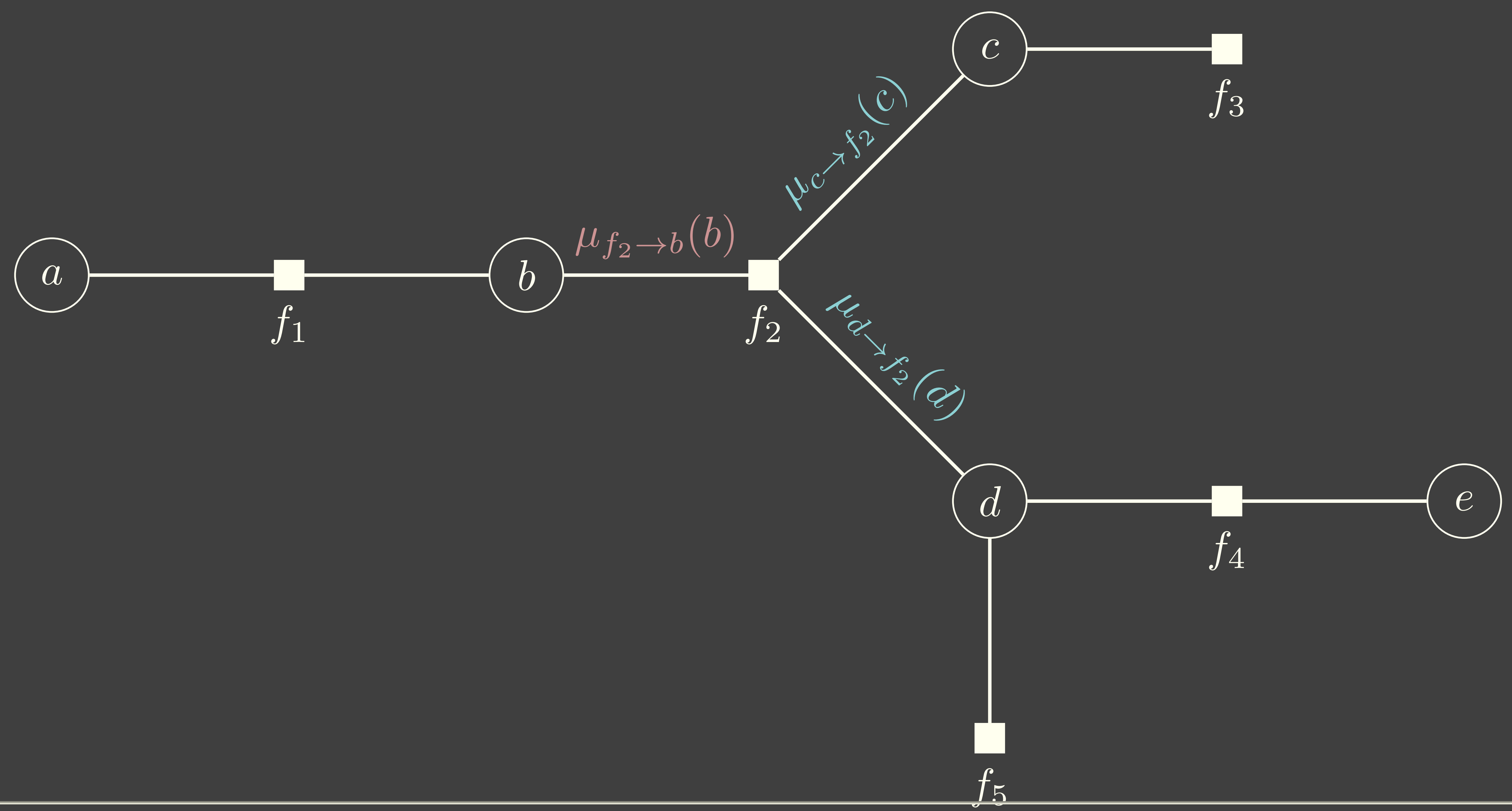


信念传播

Belief Propagation：

因子图中一个节点只有在**完全收集到邻居传来的“信息”时**才可以**将自己的“信息”传递出去**。

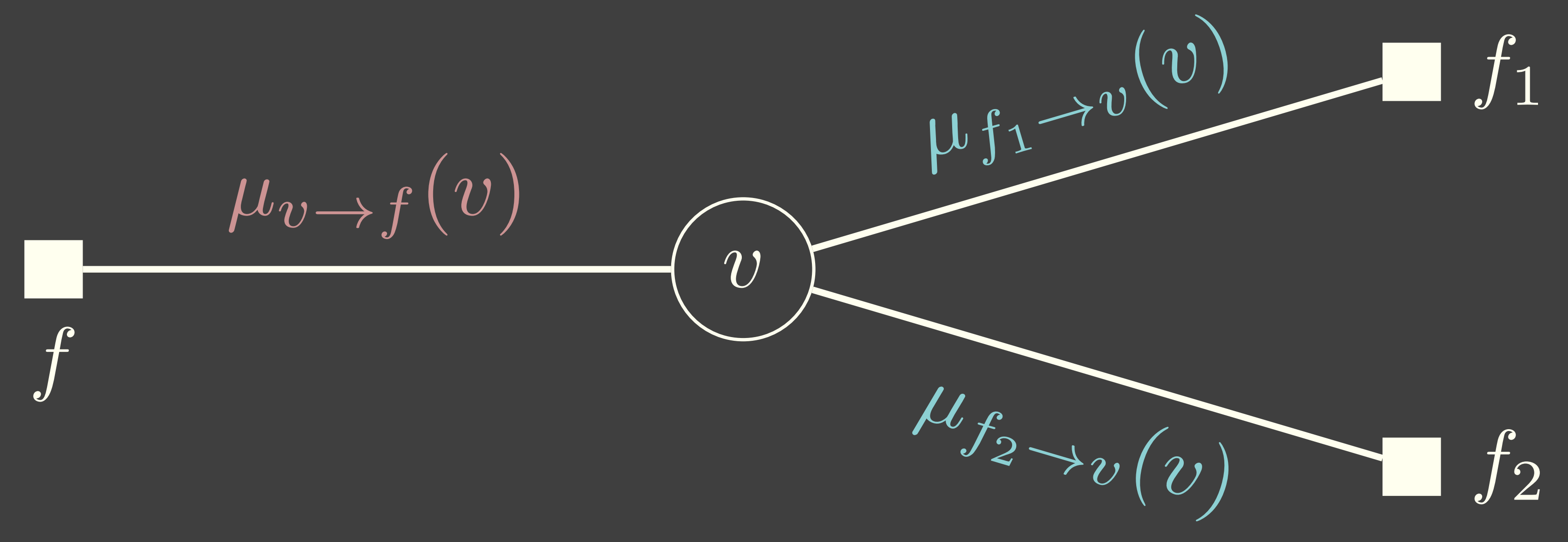
$$\Pr(a,b,c,d,e) \propto f_1(a,b)f_2(b,c,d)f_3(c)f_4(d,e)f_5(d)$$



信念传播 · Sum-Product

随机变量到因子节点

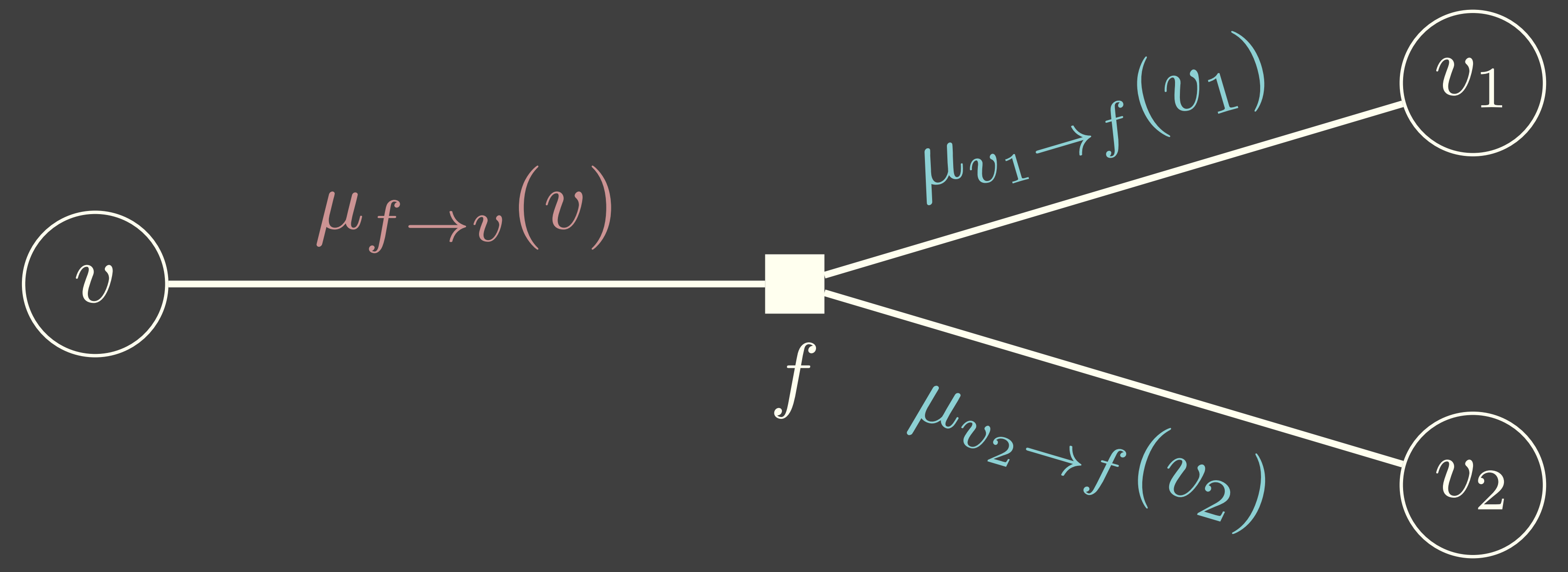
$$\mu_{v\to f}(v) = \prod_{f_i \sim v \backslash f} \mu_{f_i\to v}(v)$$



边界变量（无输入邻居因子）的信息为1。

因子节点到随机变量

$$\mu_{f\to v}(v) = \sum_{\{v_i\}} f(v, \{v_i\}) \prod_{v_i \sim f \backslash v} \mu_{v_i\to f}(v_i)$$



边界因子（无输入邻居变量）的信息为其本身。

符号学习简介・概率逻辑推断

1. 图模型推断
2. **从 SAT 到 WMC**
3. 知识编译

可满足性问题（SAT）

输入：

- 一组布尔变量集合（游戏中的$1, \ldots, 9$）
- 一组形如$l_1 \vee \ldots \vee l_k$的子句（在游戏中$k=3$，因此它是3-SAT），其中每个文字l_i是布尔变量或其否定形式

输出：

- 对所有布尔变量的一个真值指派，使得它能够满足（satisfies）所有子句

可满足性问题（SAT）

例如，下列两个子句:

$$A \vee B \vee C$$
$$\neg A \vee \neg B \vee \neg C$$

存在一组解（一个模型）为:

$$A = True, B = False, C = True$$

许多时候我们面对的是更一般形式的逻辑表达式，而不仅仅是k-CNF（conjunctive normal form，合析范式）

A	B	C	**SAT?**
0	0	0	No
0	0	1	Yes
0	1	0	Yes
0	1	1	Yes
1	0	0	Yes
1	0	1	Yes
1	1	0	Yes
1	1	1	No

模型计数（model counting）：一个逻辑理论（规则集）拥有多少个模型？[6]

可满足性问题（SAT）

- 是计算机科学中最基础的问题之一
- 是可计算理论的基石之一，号称“NP之壁”
- 许多问题能够归约为SAT，例如规划（planning）、约束求解（constraint solving）、验证（verification）……
- 大型SAT问题依然是计算机科学面临解决的核心问题之一，每年都有竞赛，且SotA榜一直在更新
- **SAT可以拓展至其他领域，例如模型计数、贝叶斯网中的概率推断等等。**

贝叶斯网与SAT

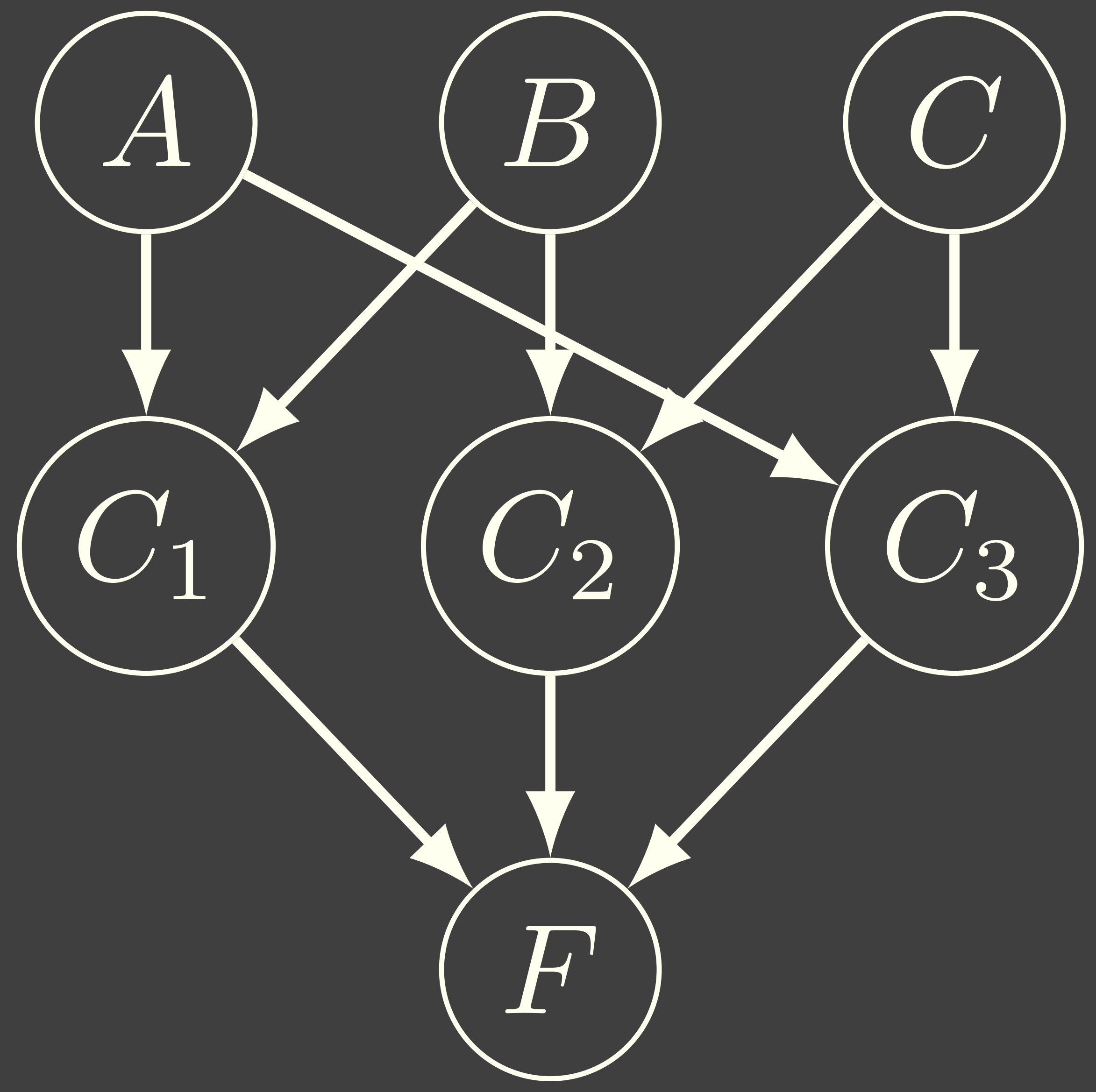
- 能不能将SAT问题转换为一个贝叶斯网推断问题?
- 考虑如下的例子:

$$(\neg A \vee B) \wedge (\neg B \vee C) \wedge (A \vee C)$$

贝叶斯网与SAT

$$(\neg A \vee B) \wedge (\neg B \vee C) \wedge (A \vee C)$$

- 给 A, B, C 各自为真的事件赋概率值0.5，并令它们到 C_1, C_2, C_3 的条件概率分布表为以上表达式（例如，C_1 为真当且仅当 $\neg A \vee B$）
- 若 $\Pr(F) > 0$ 当且仅当上面的CNF可满足
- 因此，**贝叶斯网推断的复杂度至少与SAT等价**（NP-难）
- **那么，$\Pr(F) = 3/8$ 代表什么意思？**



从SAT到#SAT到WMC

- **SAT**：对于一个逻辑理论，它的模型**是否存在**？
- **#SAT**：一共**有多少个**模型？（模型计数）
- **WMC**：若每个模型拥有不同的**权重**，计数所得的**权重**是多少？

从SAT到#SAT到WMC

从前面的例子可以看出，贝叶斯网的概率推断与带权模型计数有着极强的关联：

- 该例子中的CNF一共有3个模型
- 在该贝叶斯网中，每个**可能世界**有1/8的概率，每个变量为真的概率（权重）为0.5。我们可按如下方式定义**带权模型计数**（weighted model counting, WMC）：
  - 给定一个逻辑理论T（通常为CNF）
  - 它的任意文字l均被赋有一个（非负）权重$w(l)$
- 那么，带权模型计数问题的结果$wmc(T)$即为：
  - $wmc(T) = \sum_{M \models T} w(M)$，其中$M$是$T$的模型（例如least Herbrand model）
  - 假设所有文字之间相互独立，那么$w(M) = \prod_{l \in M} w(l)$

从SAT到#SAT到WMC

$$(\neg A \vee B) \wedge (\neg B \vee C) \wedge (A \vee C)$$

$$w(A) = w(\neg A) = w(B) = w(\neg B) = w(C) = w(\neg C) = 0.5$$

A	B	C	模型?	权重	$wmc(T)$
0	0	0	0	0.5^3	0
0	0	1	1	0.5^3	0.5^3
0	1	0	0	0.5^3	0
0	1	1	1	0.5^3	0.5^3
1	0	0	0	0.5^3	0
1	0	1	0	0.5^3	0
1	1	0	0	0.5^3	0
1	1	1	1	0.5^3	0.5^3

符号学习简介·概率逻辑推断

1. 图模型推断
2. 从SAT到WMC
3. **知识编译**

知识编译

Knowledge Compilation： 将逻辑表达式编译为一个电路（circuit）以进行高效运算

- 编译一次，重复利用
  - SAT 求解算法的 DPLL 每次计算都需要从头开始
  - 贝叶斯网推断方法中的 SoTA
  - 针对不同种类的问题，存在许多电路类型
  - 已经被应用在许多领域（软件验证、计算机体系结构、数据挖掘等等）

知识编译

贝叶斯网可以编码为如下布尔表达式：

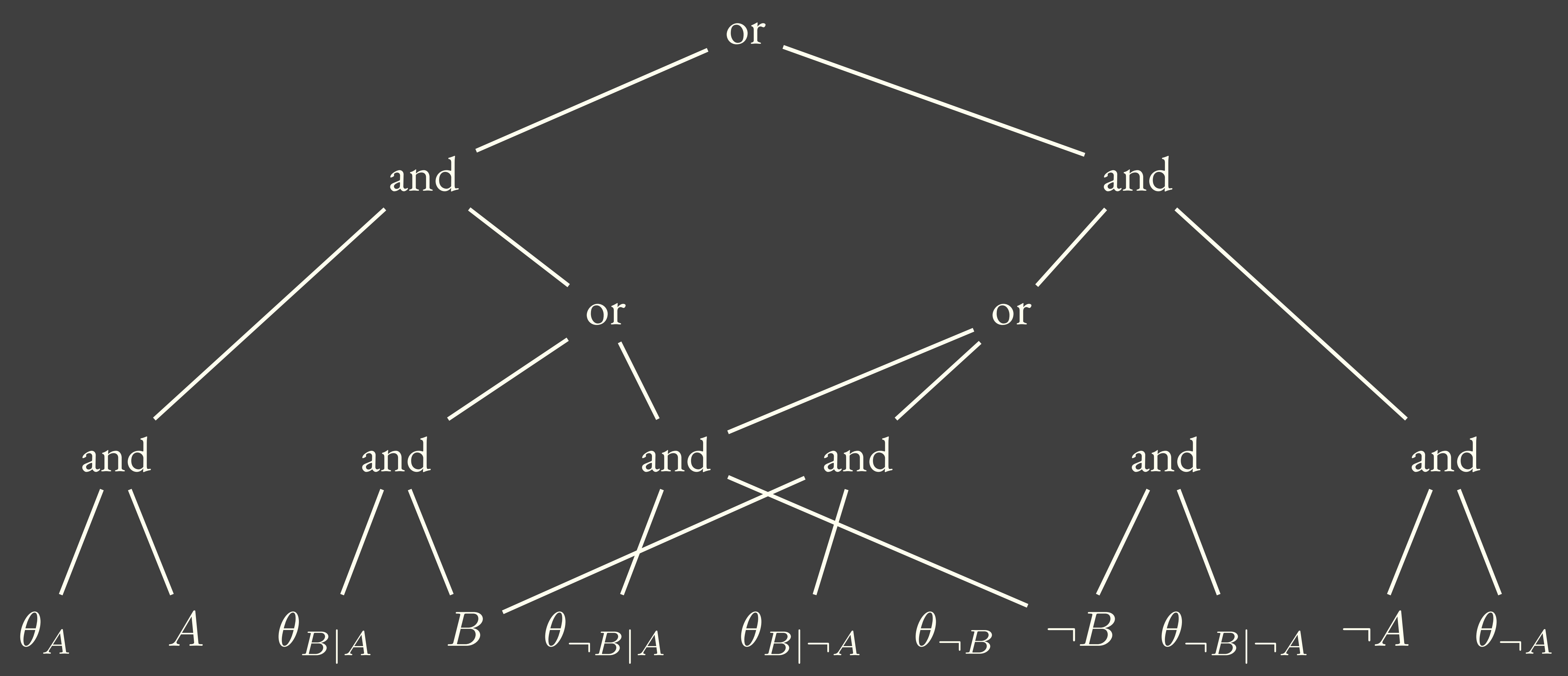
$$\{(\theta_A \wedge A) \wedge ((\theta_{B|A} \wedge B) \vee (\theta_{\neg B|A} \wedge \neg B))\} \vee \{(\theta_{\neg A} \wedge A) \wedge ((\theta_{B|\neg A} \wedge B) \vee (\theta_{\neg B|\neg A} \wedge \neg B))\}$$

- A:

$$A \Leftrightarrow \theta_A$$
$$\neg A \Leftrightarrow \theta_{\neg A}$$

- $A \rightarrow B$:

$$A \wedge B \Leftrightarrow \theta_{B|A}$$
$$A \wedge \neg B \Leftrightarrow \theta_{\neg B|A}$$
$$\neg A \wedge B \Leftrightarrow \theta_{B|\neg A}$$
$$\neg A \wedge \neg B \Leftrightarrow \theta_{\neg B|\neg A}$$



知识编译

$$(\theta_A \wedge A) \wedge ((\theta_{B|A} \wedge B) \vee (\theta_{\neg B|A} \wedge \neg B)) \vee$$
$$(\theta_{\neg A} \wedge A) \wedge ((\theta_{B|\neg A} \wedge B) \vee (\theta_{\neg B|\neg A} \wedge \neg B))$$

对应着:

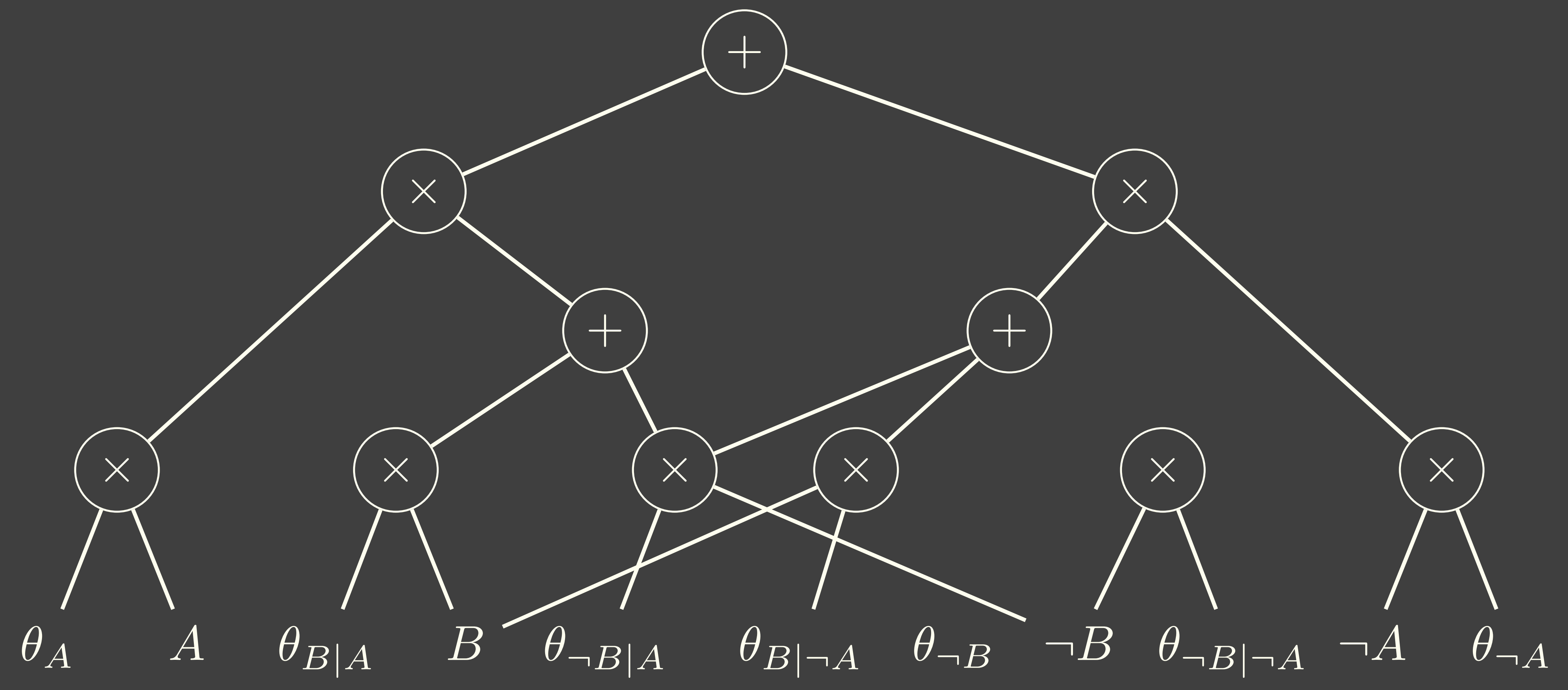
$$(\Pr(A = true) \cdot A) \cdot ((\Pr(B = true \mid A = true) \cdot B) + (\Pr(B = false \mid A = true) \cdot \neg B)) +$$
$$(\Pr(A = false) \cdot \neg A) \cdot ((\Pr(B = true \mid \neg A = true) \cdot B) + (\Pr(B = false \mid A = false) \cdot \neg B))$$

当我们将$A, B, \neg A, \neg B \in \{0, 1\}$代入时，即可获得欲求得的概率值。

例如，令$A = 1, B = 1$可得:

$$\begin{aligned}&(\Pr(A = true) \cdot A) \cdot (\Pr(B = true \mid A = true) \cdot B) \\ &= \Pr(A = true) \cdot \Pr(B = true \mid A = true)\end{aligned}$$

转化为算数电路

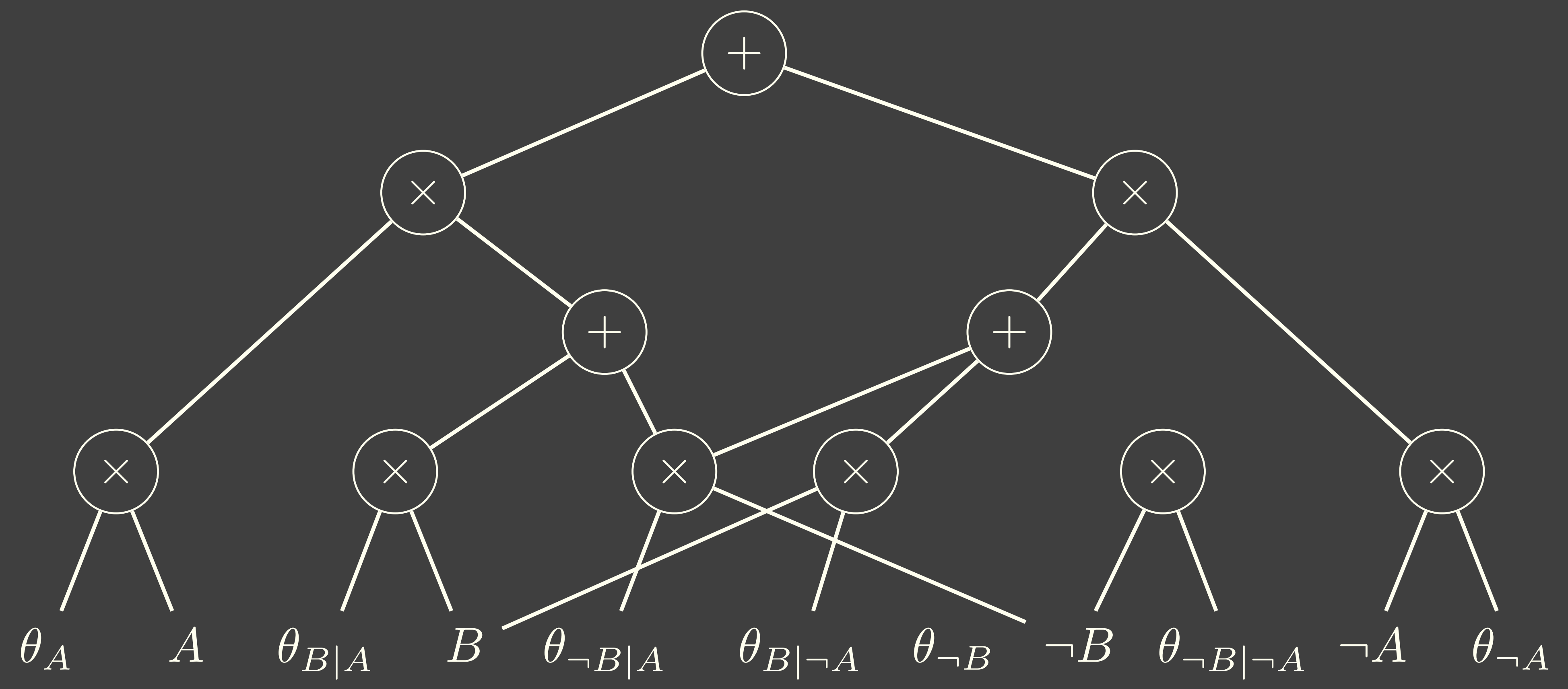


将and和or分别换成$\times$和$+$，可将贝叶斯网转化为一个多项式：

$$\begin{aligned}&\theta_A A(\theta_{B|A}B+\theta_{\neg B|A}\neg B)+\theta_{\neg A}\neg A(\theta_{B|\neg A}B+\theta_{\neg B|\neg A}\neg B)\\ =\ &\theta_A A\theta_{B|A}B+\theta_A A\theta_{\neg B|A}\neg B+\theta_{\neg A}\neg A\theta_{B|\neg A}B+\theta_{\neg A}\neg A\theta_{\neg B|\neg A}\neg B\end{aligned}$$

可以看到，这里的$A, B, \neg A, \neg B$均为布尔表达式θ_x的**权重**！

转化为算数电路



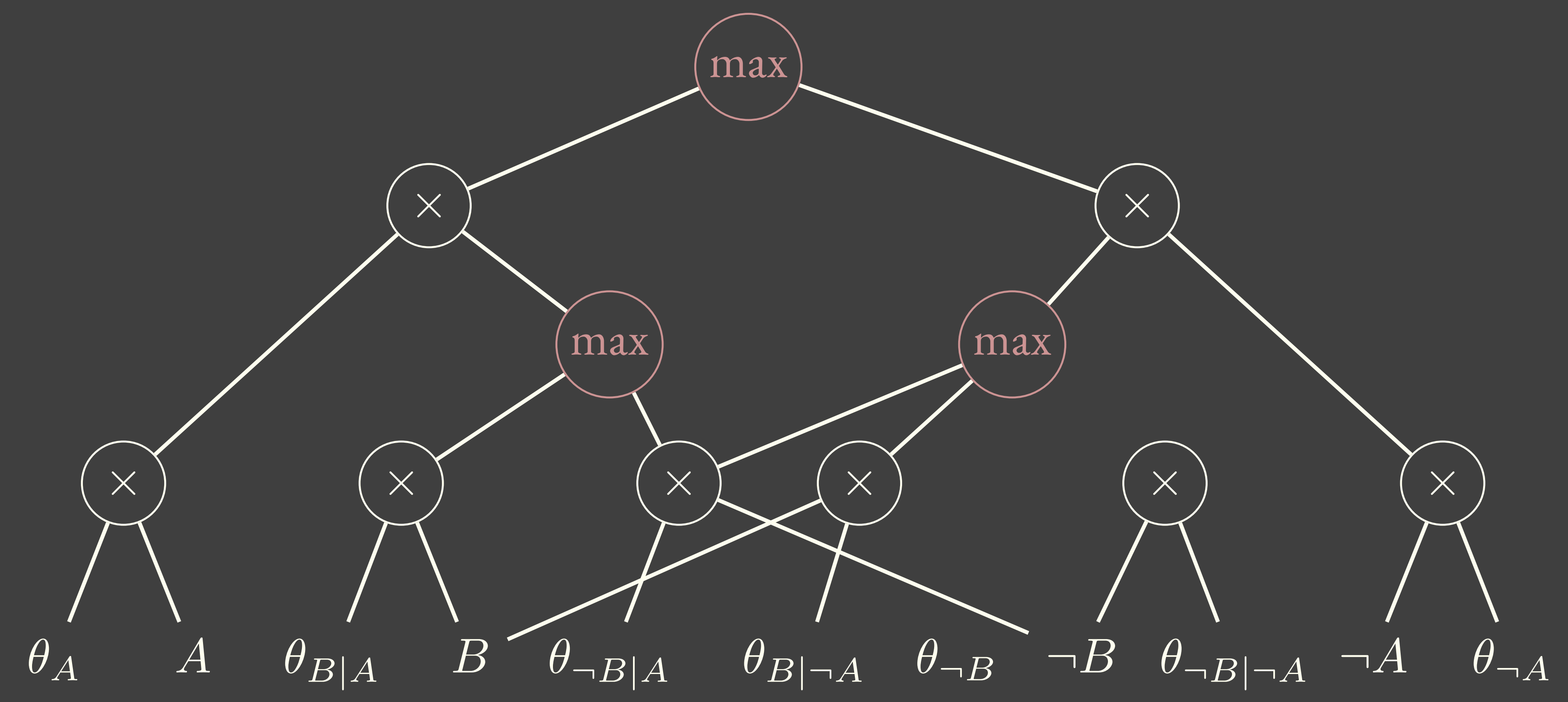
$$\theta_A A \theta_{B|A} B + \theta_A A \theta_{\neg B|A} \neg B + \theta_{\neg A} \neg A \theta_{B|\neg A} B + \theta_{\neg A} \neg A \theta_{\neg B|\neg A} \neg B$$

计算$\Pr(A = false)$时，只需令权重$A = 0, \neg A = 1, B = 1, \neg B = 1$

$$\begin{aligned} \Pr(A = false) &= 0 + 0 + \theta_{\neg A} \neg A \theta_{B|\neg A} B + \theta_{\neg A} \neg A \theta_{\neg B|\neg A} \neg B \\ &= 0 + 0 + \theta_{\neg A} \theta_{B|\neg A} + \theta_{\neg A} \theta_{\neg B|\neg A} = \theta_{\neg A} \end{aligned}$$

Max-Product

当需要计算MPE（most-probable explanation）时，可编译一个max-product电路：



将and和or分别换成×和max：

$$\max(\theta_A A \cdot \max(\theta_{B|A} B, \theta_{\neg B|A} \neg B), \theta_{\neg A} \neg A \cdot \max(\theta_{B|\neg A} B, \theta_{\neg B|\neg A} \neg B))$$

计算$\arg\max_b \Pr(A = false, B = b)$时，只需令权重$A = 0, \neg A = 1, B = 1, \neg B = 1$并计算上式。

算数电路中的概率推断

1. 计算观测变量（或称为证据，evidence）$\Pr(evidence)$的概率只需要重新设置权重：
   - 若$e = true$则令权重$e = 1$且$\neg e = 0$，反之亦然
   - 若v不是观测变量，则令$v = 1$且$\neg v = 1$
   - 对于所有条件概率表中的条目：令$\neg\theta_x = 1$且$\theta_x = CPT(x)$
2. 因为权重可以**重新设置**，算数电路可被用于多次重复运算
   - 仅需要编译一次（花费较高），但以后每次推断速度都很快
   - 可类比于信念传播，完成传播后可获得所有变量的边际分布
3. 如何保证算数电路能够计算出正确的结果？
   - sd-DNNF（smooth, deterministic, Decomposable Negation Normal Form）

D-DNNF 的编译算法

Huang and Darwiche. The language of search. JAIR, 2007.

概率规则学习

1. **参数学习**
2. 结构学习

参数学习

$$\underbrace{\Pr(\theta \mid \mathcal{D})}_{\text{posterior}} = \frac{\overbrace{\Pr(\mathcal{D} \mid \theta)}^{\text{likelihood}} \cdot \overbrace{\Pr(\theta)}^{\text{prior}}}{\underbrace{\Pr(\mathcal{D})}_{\text{evidence}}}$$

- θ：参数，即概率事实、概率规则的权重
- $\mathcal{D}$：训练数据，$\{X = x\}_{i=1}^{n}$

参数学习

考虑模型 M:

$$\Pr(\theta \mid \mathcal{D}, M) = \frac{\Pr(\mathcal{D} \mid \theta, M) \cdot \Pr(\theta \mid M)}{\underbrace{\Pr(\mathcal{D} \mid M)}_{\text{marginal likelihood}}}$$

- 最大似然（ML）

$$\theta^* = \arg\max_{\theta} \Pr(\mathcal{D} \mid \theta, M)$$

- 最大后验概率（MAP）

$$\theta^* = \arg\max_{\theta} \Pr(\theta \mid \mathcal{D}, M)$$

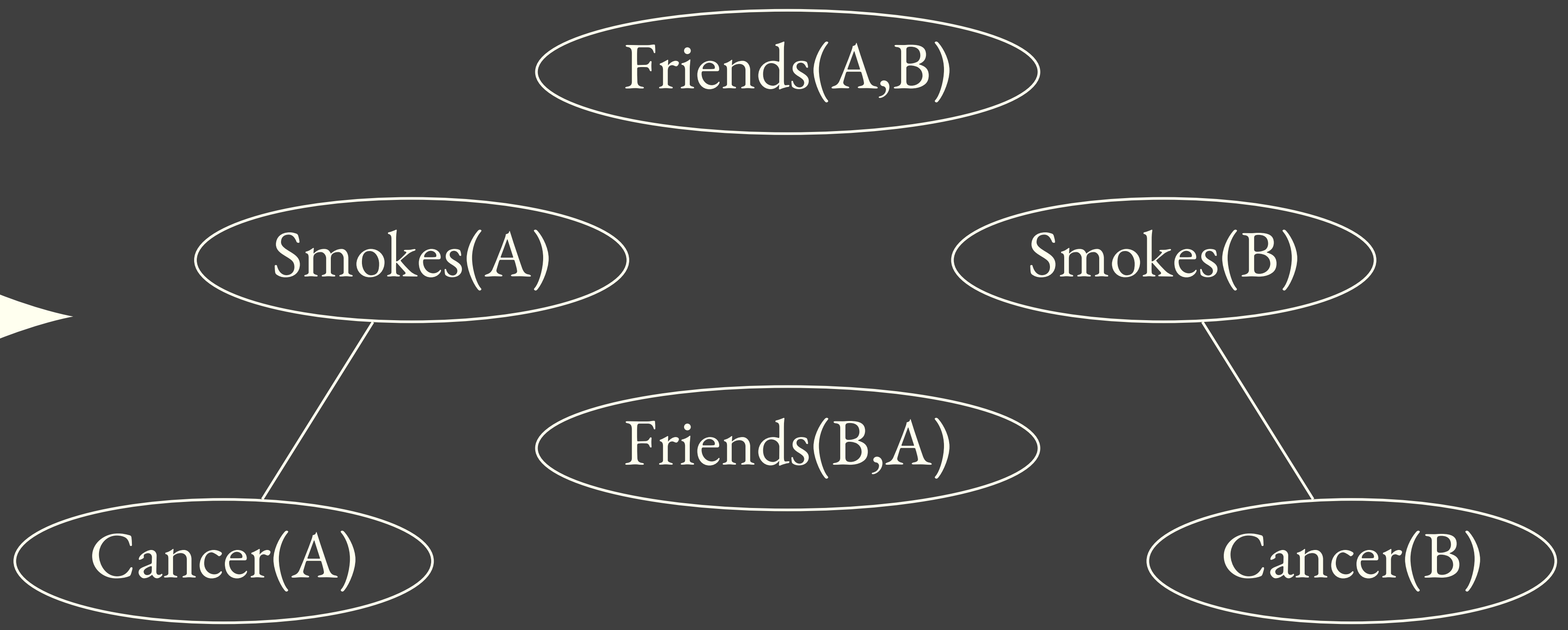
- 若 $\Pr(\theta \mid M) = \text{const.}$，那么MAP=ML

$$\theta^* = \arg\max_{\theta} \Pr(\theta \mid \mathcal{D}, M) = \arg\max_{\theta} \frac{\Pr(\mathcal{D} \mid \theta, M)\Pr(\theta \mid M)}{\Pr(\mathcal{D} \mid \theta)} = \arg\max_{\theta} \Pr(\mathcal{D} \mid \theta, M)$$

马尔可夫逻辑的参数估计

$$\Pr_w(\mathbf{X}) = \frac{1}{Z}\prod_i \phi_i(\mathbf{X}) = \frac{1}{Z}\exp\left(\sum_i w_i n_i(\mathbf{X})\right)$$

$$\forall x\ Smokes(x) \Rightarrow Cancer(x)$$



马尔可夫逻辑的参数估计

判别式学习：预测条件似然 $\Pr(y \mid x)$

$$\Pr(y \mid x) = \frac{1}{Z_x} \exp\left(\sum_{i \in F_Y} w_i n_i(x, y)\right)$$

- w_i：第 i 个子句的权重
- Z_x：取值 $X = x$ 的所有 possible world 权重之和（条件配分函数）
- F_Y：包含问句原子 Y 的所有子句（马尔可夫毯）
- $n_i(x, y)$：当变量取值为 $(X, Y) = (x, y)$ 时，第 i 个子句中为真的原子个数

参数 w_i 关于对数似然函数的梯度

$$\begin{aligned}\frac{\partial}{\partial w_i} \log \Pr_w(y \mid x) &= n_i(x, y) - \sum_{y'} \Pr_w(y' \mid x) n_i(x, y') \\ &= n_i(x, y) - E_w\left[n_i(x, y)\right]\end{aligned}$$

马尔可夫逻辑的参数估计

生成式学习：预测联合概率似然 $\Pr(x)$

$$\Pr_w(X = x) = \frac{1}{Z}\exp\left(\sum_{i\in F} w_i n_i(x)\right)$$

参数 w_i 关于对数似然函数的梯度

$$\frac{\partial}{\partial w_i}\log \Pr_w(X = x) = n_i(x) - \sum_{x'} \mathrm{P}_w(X = x')n_i(x')$$

配分函数和model的概率分布计算是intractable的：

- MCMC采样，如WalkSAT
- MAP估计

马尔可夫逻辑的参数估计

the process). Considering this must be done iteratively as L-BFGS searches for the minimum, we estimate it would take anywhere from 20 to 400 days to complete the training, even with a weak convergence threshold such as $R = 2.0$. Experiments confirmed the poor quality of the models that resulted if we ignored the convergence threshold and limited the training process to less than ten hours. With a better choice of initial state, approximate counting, and improved MCMC techniques such as the Swendsen-Wang algorithm (Edwards & Sokal, 1988), MC-MLE may become practical, but it is not a viable option for training in the current version. (Notice that during learning MCMC is performed over the full ground network, which is too large to apply MaxWalkSat to.)

对于含递归规则的一阶概率逻辑问题来说
基于MCMC（如Gibbs采样）的推理基本很难在短时间内收敛

伪似然函数

$$\Pr_w^*(X = x) = \prod_{l=1}^{n} \Pr(X_l = x_l \mid MB_x(X_l))$$

- $X_l, l \in \{1, \ldots, n\}$：Markov网中的所有随机变量（groundings）
- $MB_x(X_l)$：变量X_l的马尔可夫毯在$X = x$时的取值

伪似然函数

参数w_i关于对数伪似然函数的梯度

$$\frac{\partial}{\partial w_i}\log \Pr_w^*(X=x)=\sum_{l=1}^{n}\left[n_i(x)-\mathrm{P}_w(X_l=0\mid MB_x(X_l))n_i(x_{[X_l=0]})-\mathrm{P}_w(X_l=1\mid MB_x(X_l))n_i(x_{[X_l=1]})\right]$$

- $n_i(x_{[X_l=0]})$：在第i个子句中令$X=x$并强行设置$X_l=0$时真原子的个数

1. 不再需要用WMC对全局模型进行推理，**速度极快**
2. 只关心**每个grounding的局部信息**
3. 当推理链较长、依赖关系复杂时，**准确率非常差**

Max-margin Learning

对于判别式学习，除了之间最大化条件对数似然函数，还可以最大化如下“margin”：

$$\frac{\Pr_{\mathbf{w}}(y|x)}{\Pr_{\mathbf{w}}(\hat{y}|x)}$$

- 其中$\hat{y}$是参数$\mathbf{w}$下随机变量Y除了y以外**最可能的**赋值：

$$\hat{y} = \arg\max_{\bar{y}\in Y\setminus y} \Pr_{\mathbf{w}}(\bar{y} \mid x)$$

借助structural SVM，可以将上面的目标表达为如下优化问题：

$$\min_{\mathbf{w},\xi\geq 0} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\xi$$
$$s.t. \ \forall \bar{y} \in Y : \mathbf{w}^T[\mathbf{n}(x,y) - \mathbf{n}(x,\hat{y})] \geq \Delta(y,\bar{y}) - \xi$$

ProbLog 中的参数学习

一阶ProbLog的概率分布函数:

$$\Pr(L \mid \mathcal{T}) = \prod_{f_n\theta_{n,k}\in L} p_n \prod_{f_n\theta_{n,k}\in L_{\mathcal{T}}\setminus L} (1-p_n)$$

其中

- $\{\theta_{n,1},\ldots,\theta_{n,K}\}$是第$n$个一阶概率原子的所有可能变量替换
- $\mathcal{T}=\mathcal{F}\cup\mathcal{BK}=\{p_1::f_1,\ldots p_N::f_N\}\cup\mathcal{BK}$为ProbLog程序，其中$p_i::f_i$为概率原子公式，$BK$是一阶逻辑程序
- 概率原子公式及它们的变量替换定义了一个在其Herbrand Universe $L_{\mathcal{T}}=\{f_1\theta_{1,1},\ldots,f_1\theta_{1,K_1},\ldots,f_N\theta_{N,1},\ldots,f_N\theta_{N,K_N}\}$上的分布语义

ProbLog 中的参数学习

概率事实$\mathcal{F}$

```
0.1::burglary.
0.2::earthquake.
0.7::heard(X).
```

一阶知识库$\mathcal{BK}$

```
person(mary).
person(john).
alarm :- burglary; earthquake.
calls(X) :- person(X), alarm, heard(X).
```

$\mathcal{F}$构成的Herbrand Universe为:

```
heard(mary), heard(john), burglary, earthquake
```

ProbLog 中的“解释”

```
0.1::burglary.
0.2::earthquake.
0.7::heard(X).
person(mary).
person(john).
alarm :- burglary; earthquake.
calls(X) :- person(X), alarm, heard(X).
```

部分解释（partial interpretation）

$$
\begin{aligned}
I^+ &= \{person(mary), person(john), burglary, alarm, heard(john), calls(john)\} \\
I^- &= \{calls(mary), heard(mary)\}
\end{aligned}
$$

$$
\Pr_w((I^+, I^-)) = 0.1 \times 0.7 \times (1 - 0.7) \times \underbrace{(0.2 + (1 - 0.2))}_{\text{earthquake}}
$$

LFI-Problog

Learning from Interpretation：

- 从部分或完整观测数据（即证据包含部分/全部groundings）中学习。

本质上，LFI是一种最大似然估计：

$$\hat{\theta} = \arg\max_{\theta} \Pr(\mathcal{D} \mid \mathcal{T}_\theta) = \arg\max_{\theta} \prod_{m=1}^{M} \Pr(I_m \mid \mathcal{T}_\theta)$$

其中，

- 训练数据$\mathcal{D} = \{I_1, \ldots, I_M\}$所有Interpretation的集合
- $\theta = \{p_1, p_2, \ldots, p_n\}$为所有概率事实的权重

LFI-Problog

当$\mathcal{D}$中的训练数据均为“**完整解释**”时：

- 每个I_m中均包含了所有概率事实的真值
- 第n个概率事实$p_n :: f_n$的参数p_n的估计可通过数数来完成

$$\hat{p}_n = \frac{1}{Z_n} \sum_{m=1}^{M} \sum_{k=1}^{K_n^m} \delta_{n,k}^m$$

其中，

$$\delta_{n,k}^m = \begin{cases} 1, & \text{if } f_n \theta_{n,k}^m \in I_m \\ 0, & \text{else} \end{cases}$$

- m代表第m个解释：$f_n \theta_{n,k}^m$是f_n在I_m中利用第k个替换得到的grounding
- $Z_n = \sum_{m=1}^{M} K_n^m$是$f_n$在所有解释中的grounding个数

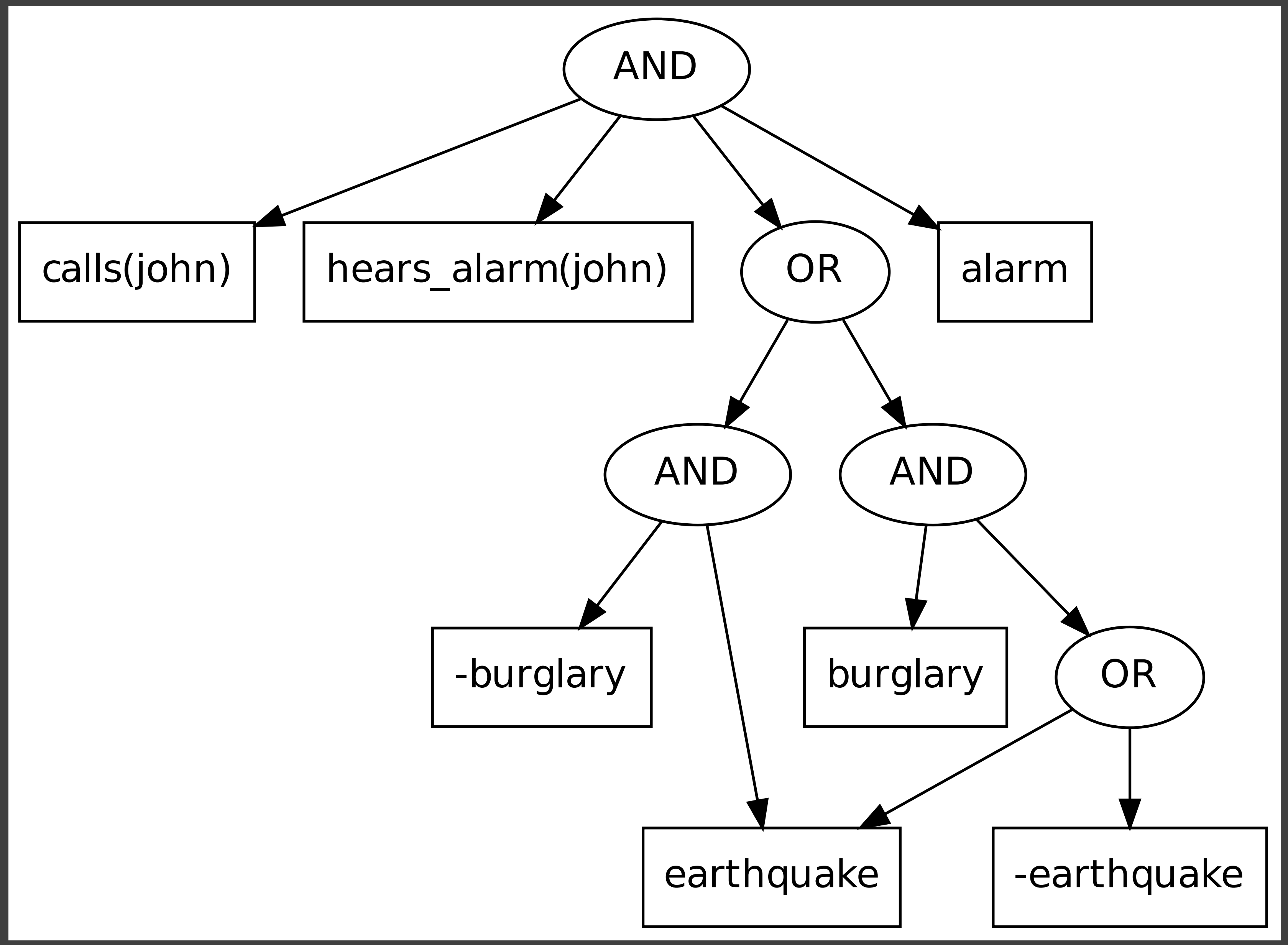
LFI-Problog

当$\mathcal{D}$中的训练数据为“**部分解释**”时：

- I_m中包含隐变量（无真值指派的 groundings）
- 通过EM算法求解

$$\hat{p}_n^{(i+1)} = \frac{1}{Z_n}\sum_{m=1}^{M}\sum_{k=1}^{K_n^m}\Pr_{\mathcal{T}_{\hat{\mathrm{p}}}^{(i)}}\left[f_{n,k} \mid I_m\right]$$

其中计算期望的步中需要将I_m编译为d-DNNF来推理隐变量的边际概率。



LFI-ProbLog

Algorithm 1 The main loop of LFI-ProbLog. The ProbLog program is compiled into a d-DNNF for each partial interpretation $\mathbf{E_m} = \mathbf{e_m}$. After the compilation step, the algorithm follows an EM update scheme, first using the current model to complete the data and then estimating the new model parameters from the resulting counts until convergence.

```
function LFI-ProbLog(T = {p_1 :: f_1, ..., p_N :: f_N} ∪ R, D = {E_1 = e_1, ..., E_M = e_M})
    for 1 ≤ n ≤ N do
        p_n^0 ← rand(0,1)          ▷ The fact probabilities are initialized with a random probability
    for 1 ≤ m ≤ M do                                    ▷ Loop over training examples
        d-DNNF_m ← COMPILE(P_{T_0}(E_m = e_m))
    i ← 0
    while not converged do                                ▷ EM algorithm
        i ← i + 1
        for 1 ≤ m ≤ M do
            for 1 ≤ n ≤ N do
                for 1 ≤ k ≤ K_n^m do
                    compute P_{T_{i-1}}(f_{n,k} | E_m) using d-DNNF_m    ▷ E Step
        for 1 ≤ n ≤ N do                                  ▷ Loop over probabilistic facts
            p_n^i ← (1/Z_n) Σ_{m=1}^{M} Σ_{k=1}^{K_n^m} P_{T_{i-1}}(f_{n,k} | E_m)    ▷ M Step (cf. Eq. 2)
    return {p_n^i :: f_n | f_n ∈ F} ∪ R
```

LFI-Problog

« Inhibition effects (../basic/10_inhibitioneffects.html) | Social networ... » (02_smokers.html)

Bayesian networks

The probabilities in a ProbLog program can be learned from partial interpretations. The LFI (Learning From Interpretations) parameter estimation algorithm was introduced in the following paper:

- B. Gutmann, I. Thon and L. De Raedt. **Learning the parameters of probabilistic logic programs from interpretations**. Machine Learning and Knowledge Discovery in Databases, volume 6911, pp. 581 - 596, Springer, 2011. PDF (https://lirias.kuleuven.be/bitstream/123456789/310534/3/gutmann_489.pdf) DOI (https://doi.org/10.1007/978-3-642-23780-5_47) TR (http://www.cs.kuleuven.be/publicaties/rapporten/cw/CW584.pdf) ERR (https://lirias.kuleuven.be/bitstream/123456789/310534/4/erratum.pdf)

Below we learn the parameters of the Alarm program. We use the 'plain' ProbLog encoding (learning in combination with annotated disjunctions is currently not yet supported).

When learning, the probability annotation in a probabilistic fact can be one of three possible forms.

- Of the form **t(_)**, as in for instance *t(_)::p_alarm1*. This indicates that the probability of this fact is to be learned from data. ProbLog2 uses an EM algorithm to perform parameter learning. In the first iteration of EM, each probability is initialized to a random value. Note that *t* is an abbreviation of 'tunable'.
- Of the form **t(p)**, with *p* a probability, as in for instance *t(0.5)::burglary*. This again indicates that the probability of this fact is to be learned from data, but instead of initializing this probability randomly, it will be set to the value *p* in the first iteration of EM.
- Of the form **p**, with *p* a probability, as in for instance *0.2::earthquake*. This indicates that the

概率规则学习

1. 参数学习
2. **结构学习**

结构学习的一般框架

1. （初始化）精化规则结构；
2. 对该规则结构进行参数优化；
3. 评估学习质量，例如用测试数据上的AUC或条件对数似然（CLL）对概率逻辑模型打分；
   - 如果达到终止条件则输出
   - 否则重复步骤1

Beam search

Kok and Domingos, ICML'05

```
function Markov_Structure_Learning
    while beam ≠ ∅
        add unit clauses to beam
        while beam has changed
            for each clause c ∈ beam
                c' ← add a literal to c
                newClauses ← newClauses ∪ c'
            end
            beam ← k best clauses in beam ∪ newClauses
        end
        add best clause in beam to MLN
    end
end
```

Relational Path Finding

Richards and Mooney, AAAI'92

- 将常元看作图中的点、谓词看作边
- 多个原子公式的合取看作一条路径

```
advises(pete, sam), teaches(pete, cs1), tas(sam, cs1).
```

构成了一条$(sam, pete, cs1)$路径

对路径进行泛化:

```
advises(X, Y), teaches(X, Z), tas(Y, Z).
```

BUSL

Mihalkova and Mooney, ICML'07

- 利用 relational pathfinding 找到数据中的短路径并泛化
- 将这些路径作为布尔变量 $\rightarrow$ 马尔可夫网中的节点
- 连接这些节点，形成团簇（cliques）
  - 将团簇重构为子句形式
- 对子句打分，并加入模型

BUSL

例如在上面的例子中，可先找到长度小于 2 的短路径

```
[advises(pete, sam), teaches(pete, cs1)], [tas(sam, cs1)].
```

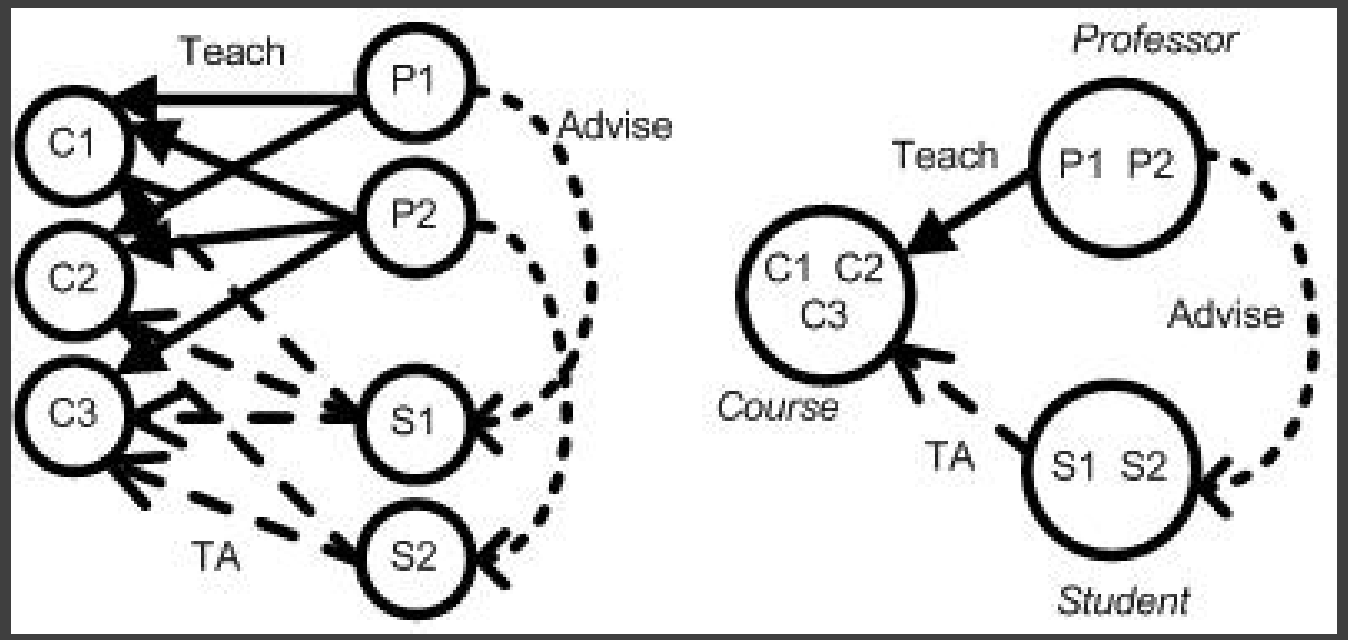
将相通的短路径进行连接，并枚举所有可能的子句

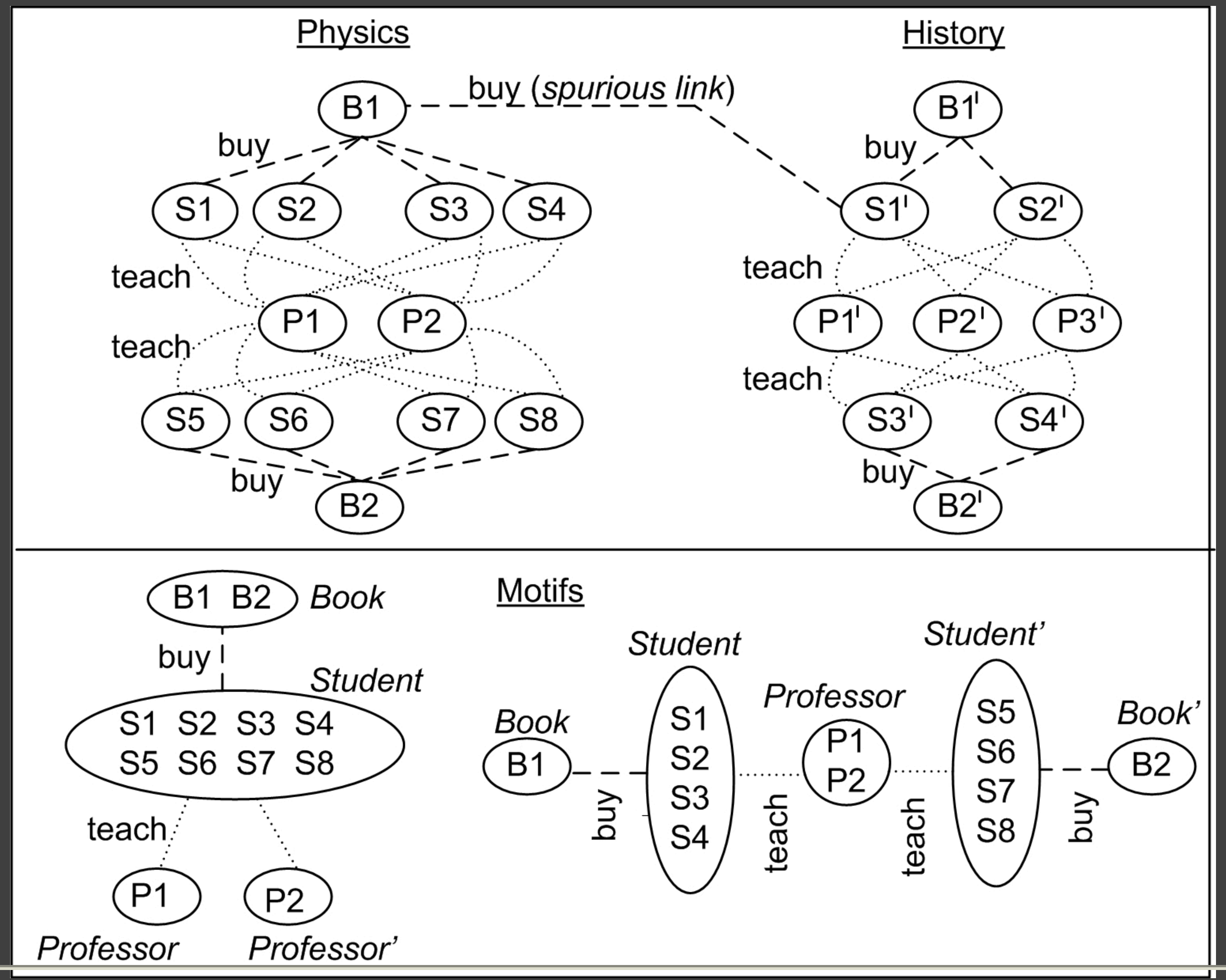
$$\begin{aligned}
advices(X,Y) \lor teaches(X,Z) \lor tas(Y,Z) \\
\neg advices(X,Y) \lor teaches(X,Z) \lor tas(Y,Z) \\
\neg advices(X,Y) \lor \neg teaches(X,Z) \lor tas(Y,Z) \\
\cdots
\end{aligned}$$

利用 Lifting 技术剪枝

Kok and Domingos, ICML'09; ICML'10

- 利用 relational pathfinding 找到 ground paths
- 根据 ground paths 的结构对领域中的常元进行聚类
- 用 random walk 找到图中具有代表性的子结构
  - 常见子图
  - 对称路径
  - …





其他方法

1. 利用 ILP 或其他规则学习技术初始化规则结构
   - ProbFOIL
   - ALEPH++
2. 利用决策树与集成学习同时归纳规则结构和参数
   - BoostSRL
   - Statistical Unfolded Learning

小结

概率逻辑系统

1. 真度与概率的区别
2. KBMC：从概率出发，引入逻辑
   - 只做概率推断
   - 借助逻辑知识结构来建模联合概率分布
3. 分布语义：从逻辑出发，引入概率
   - 同时进行概率和逻辑推理
   - 概率分布从哪来？