

《神经网络与深度学习》



概率图模型

<https://nndl.github.io/>

如何表示高维随机向量的概率密度？

▶ $P(X_1, X_2, \dots, X_K)$

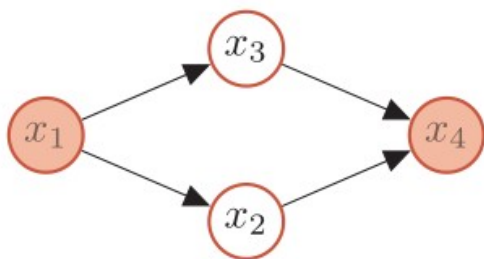
▶ 全概率公式

▶ 例子： $P(X_1, X_2, X_3)$

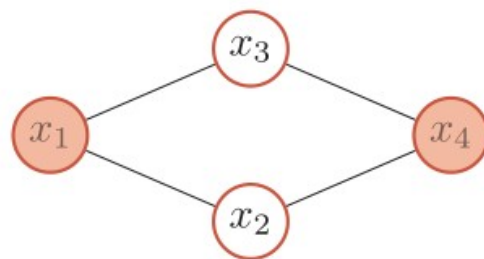
▶ 如果 $(X_1 || X_3) | X_2$ 是条件独立的，如何化简？

概率图模型

- ▶ 概率图模型是指一种用图结构来描述多元随机变量之间条件独立关系的概率模型。



(a) 有向图：贝叶斯网络



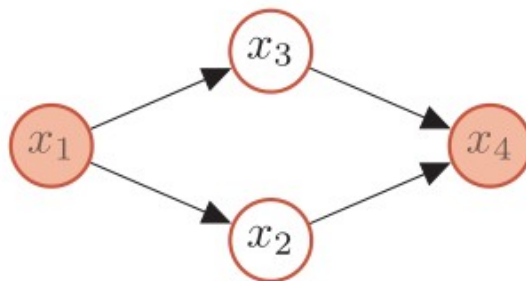
(b) 无向图：马尔可夫随机场

- ▶ 每个节点都对应一个随机变量，可以是观察变量，隐变量或是未知参数等；
- ▶ 每个连接表示两个随机变量之间具有依赖关系。

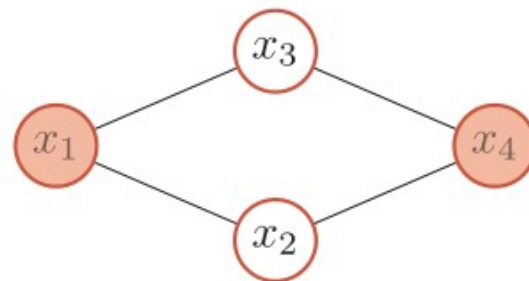
概率图模型

► 模型表示（图结构）

- 有向图
- 无向图



(a) 有向图：贝叶斯网络



(b) 无向图：马尔可夫随机场

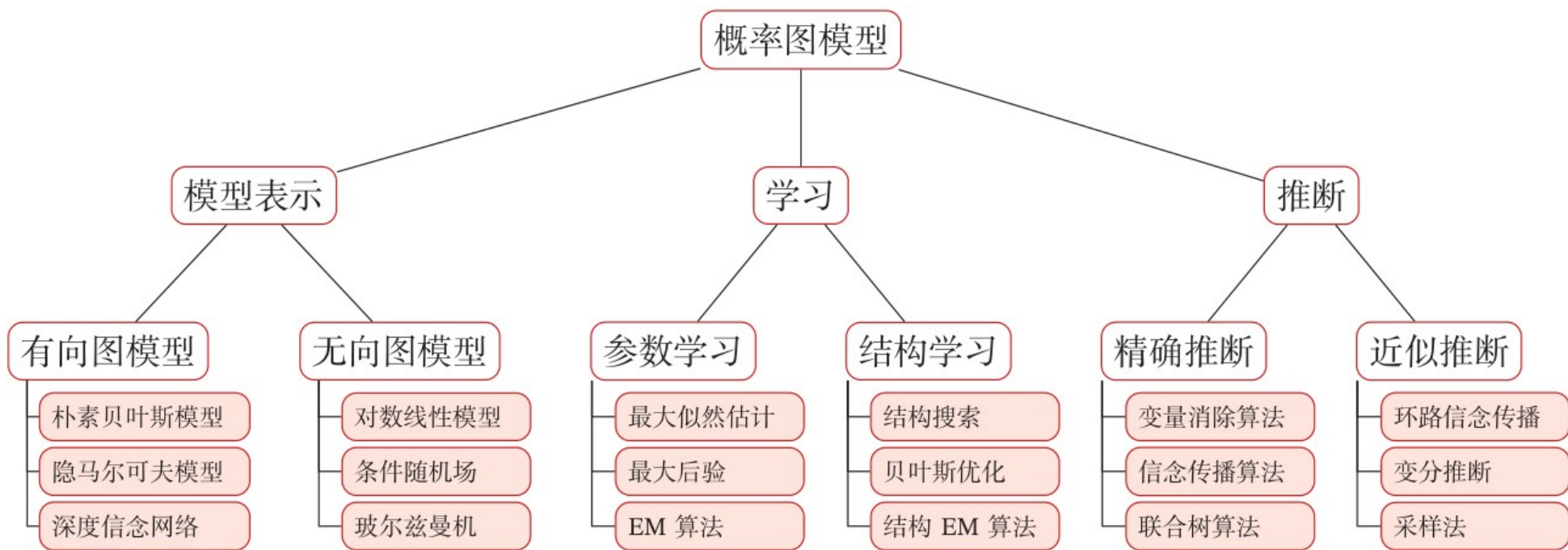
► 推断（ Inference ）

- 给定部分变量，推断另一部分变量的后验概率。

► 学习（ Learning ）

- 参数学习：给定一组训练样本，求解模型参数

概率图模型





模型表示

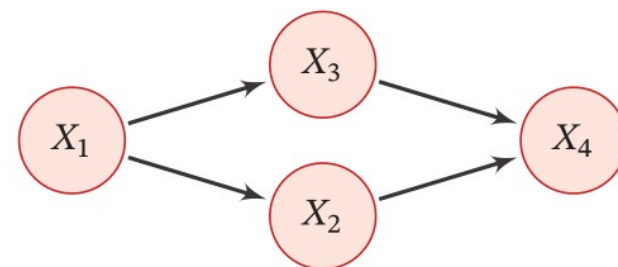
贝叶斯网络

- ▶ 有向图模型（Directed Graphical model），也称为贝叶斯网络（Bayesian Network），或信念网络（Belief Network，BN）。

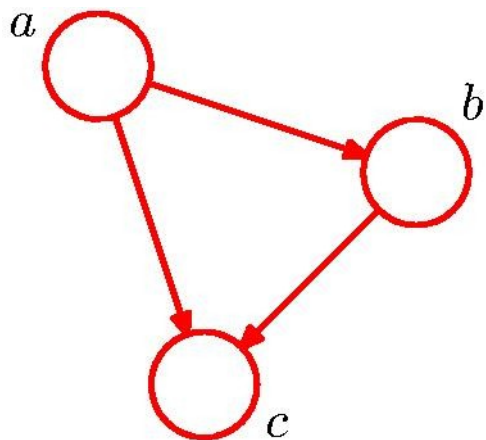
定义 11.1 – 贝叶斯网络： 对于一个 K 维随机向量 $\mathbf{X}$ 和一个有 K 个节点的有向非循环图 G , G 中的每个节点都对应一个随机变量, 每个连接 e_{ij} 表示两个随机变量 X_i 和 X_j 之间具有非独立的因果关系. 令 $\mathbf{X}_{\pi_k}$ 表示变量 X_k 的所有父节点变量集合, $P(X_k|\mathbf{X}_{\pi_k})$ 表示每个随机变量的局部条件概率分布 (Local Conditional Probability Distribution). 如果 $\mathbf{X}$ 的联合概率分布可以分解为每个随机变量 X_k 的局部条件概率的连乘形式, 即

$$p(\mathbf{x}) = \prod_{k=1}^K p(x_k|\mathbf{x}_{\pi_k}), \quad (11.8)$$

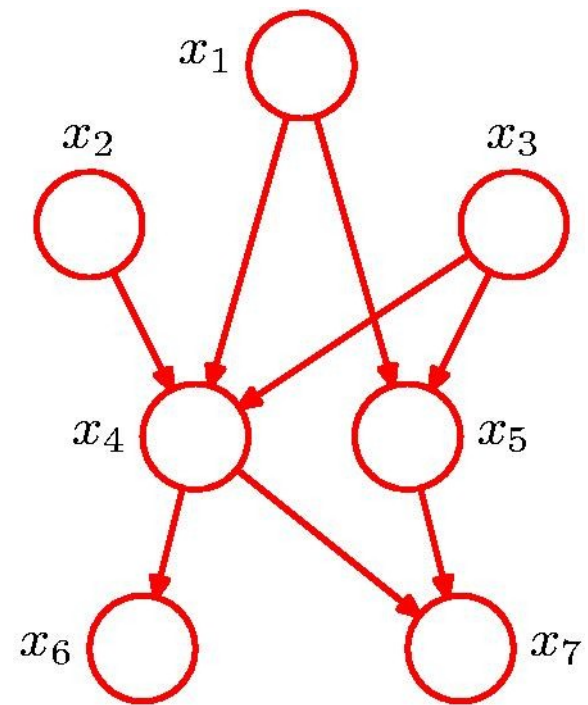
那么 $(G, \mathbf{X})$ 构成了一个贝叶斯网络.



练习



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



$$\begin{aligned} p(x_1, \dots, x_7) &= p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \\ &\quad p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5) \end{aligned}$$

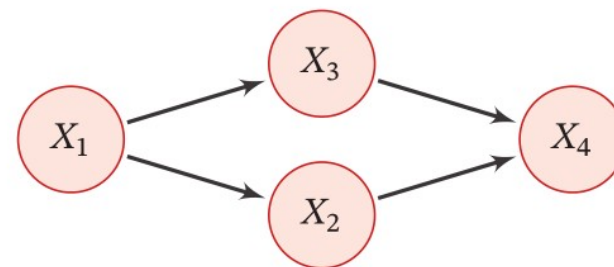
局部马尔可夫性质

- ▶ 贝叶斯网络的局部马尔可夫性质：每个随机变量在给定父节点的情况下，条件独立于它的非后代节点。

$$X_k \perp\!\!\!\perp Z | X_{\pi_k}$$

- ▶ 利用局部马尔可夫性，可以对多元变量的联合概率进行简化，从而降低建模的复杂度。
- ▶ 例子：

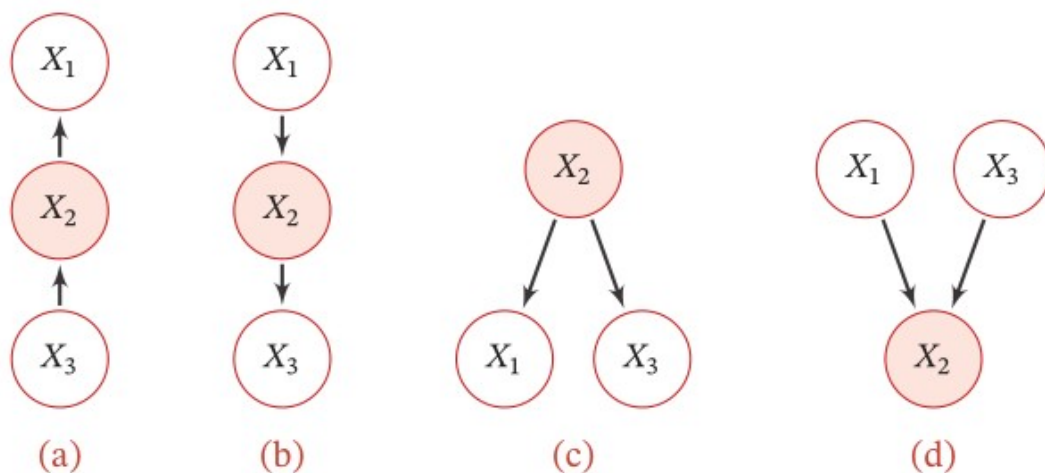
$$\begin{aligned} p(\mathbf{x}) &= p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3), \\ &= p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3), \end{aligned}$$



- ▶ 是 4 个局部条件概率的乘积，这样只需要 $1 + 2 + 2 + 4 = 9$ 个独立参数。

条件独立性

- ▶ 在贝叶斯网络中，如果两个节点是直接连接的，它们肯定是非条件独立的，是直接因果关系。
- ▶ 父节点是“因”，子节点是“果”。
- ▶ 如果两个节点不是直接连接的，但是它们之间有一条经过其他节点的路径连接互连接，它们之间的条件独立性就比较复杂。

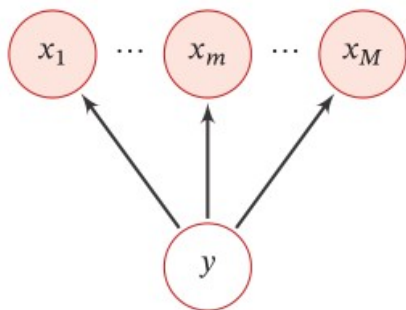


常见的有向图模型

朴素贝叶斯分类器

- 给定一个有 M 维特征的样本 $\mathbf{x}$ 和类别 y ，类别的后验概率为

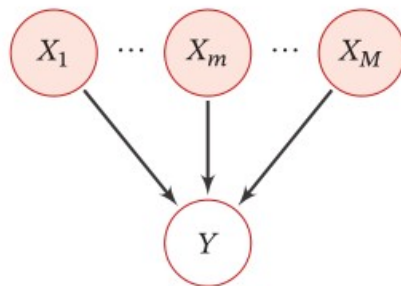
$$p(y|\mathbf{x}; \theta) \propto p(y|\theta_c) \prod_{m=1}^M p(x_m|y; \theta_m)$$



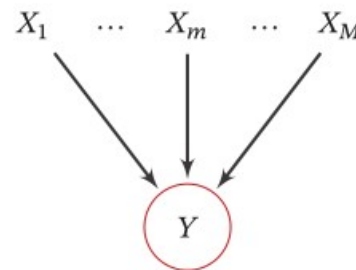
Sigmoid 信念网络

- Sigmoid 信念网络网络中的变量为二值变量，取值为 $\{0,1\}$ 。

$$p(x_k = 1|\mathbf{x}_{\pi_k}; \theta) = \sigma(\theta_0 + \sum_{x_i \in \mathbf{x}_{\pi_k}} \theta_i x_i)$$



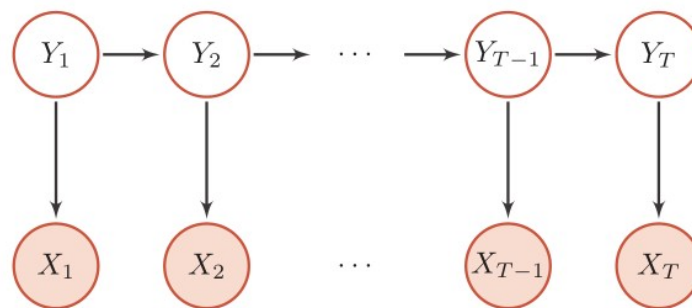
(a) 只有一层的简单 Sigmoid 信念网络



(b) Logistic 回归模型

隐马尔可夫模型（Hidden Markov Model，HMM）

- 表示一种含有隐变量的马尔可夫过程



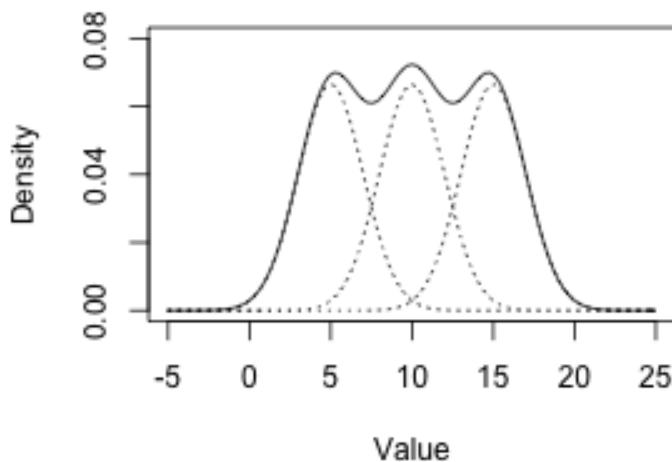
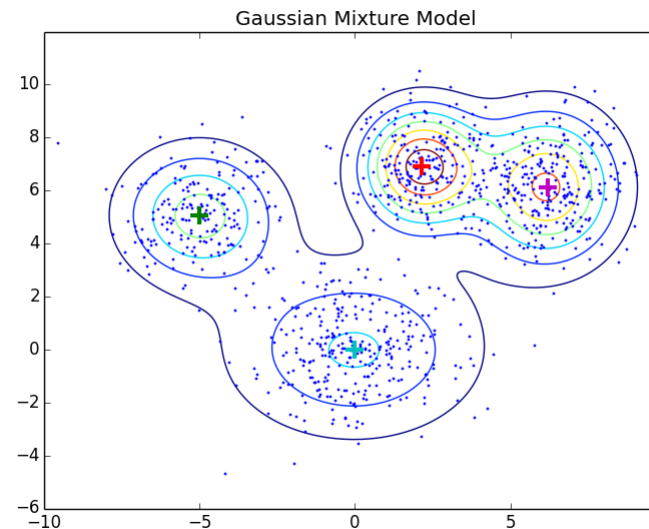
- 隐马尔可夫模型的联合概率可以分解为

$$p(\mathbf{x}, \mathbf{y}; \theta) = \prod_{t=1}^T p(y_t | y_{t-1}, \theta_s) p(x_t | y_t, \theta_t)$$

转移概率 输出概率

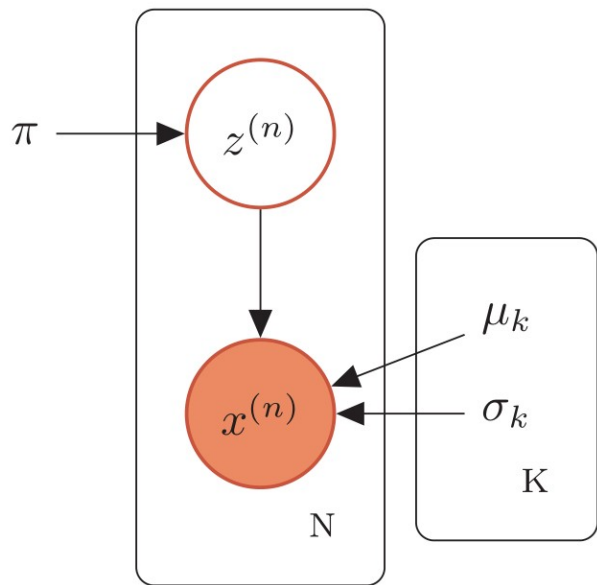
高斯混合模型

- ▶ 高斯混合模型（Gaussian Mixture Model，GMM）是由多个高斯分布组成的模型，其密度函数为多个高斯密度函数的加权组合。



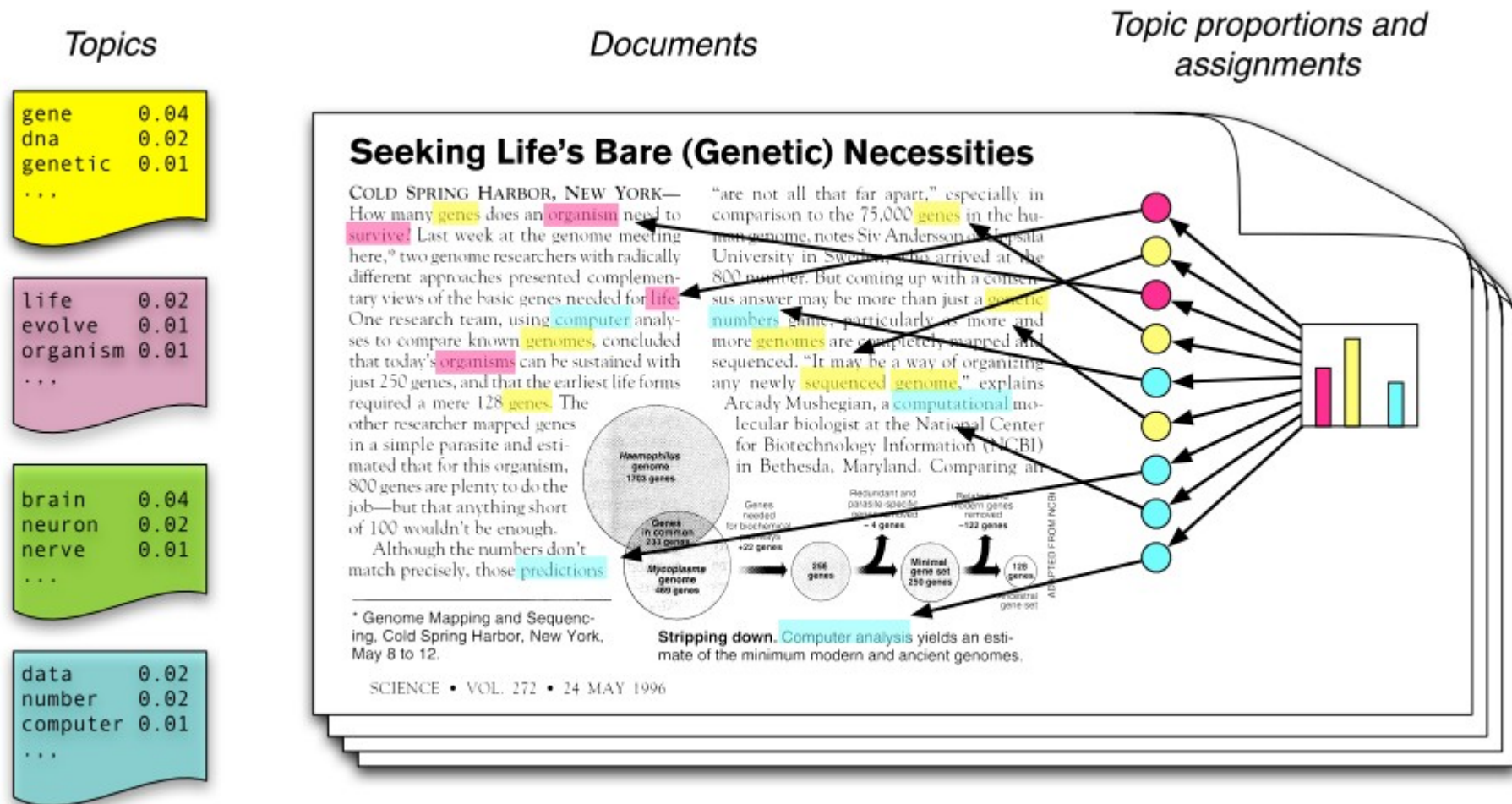
高斯混合模型

图模型表示

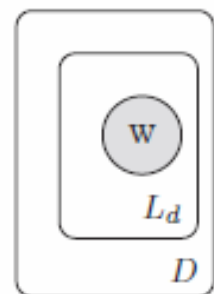


$$\mathcal{N}(x|\mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(x - \mu_k)^2}{2\sigma_k^2}\right)$$

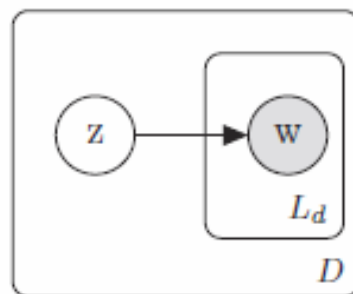
概率主题模型



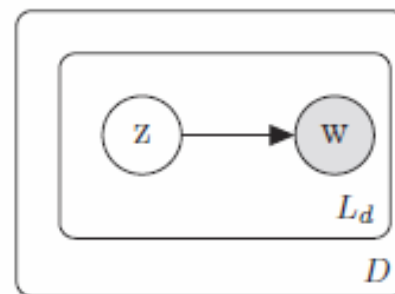
概率主题模型



(a) 一元语言模型



(b) 一元混合语言模型



(c) 概率主题模型

有向图模型

- ▶ 深度信念网络
- ▶ 变分自编码器

马尔可夫随机场

- ▶ 马尔可夫随机场，也称无向图模型，是一类用无向图来表示一组具有马尔可夫性质的随机变量 $\mathbf{X}$ 的联合概率分布模型。

定义 11.2 – 马尔可夫随机场： 对于一个随机向量 $\mathbf{X} = [X_1, \dots, X_K]^T$ 和一个有 K 个节点的无向图 $G(\mathcal{V}, \mathcal{E})$ (可以存在循环), 图 G 中的节点 k 表示随机变量 $X_k, 1 \leq k \leq K$. 如果 $(G, \mathbf{X})$ 满足**局部马尔可夫性质**, 即一个变量 X_k 在给定它的邻居的情况下独立于所有其他变量,

$$p(x_k | \mathbf{x}_{\setminus k}) = p(x_k | \mathbf{x}_{\mathcal{N}(k)}), \quad (11.15)$$

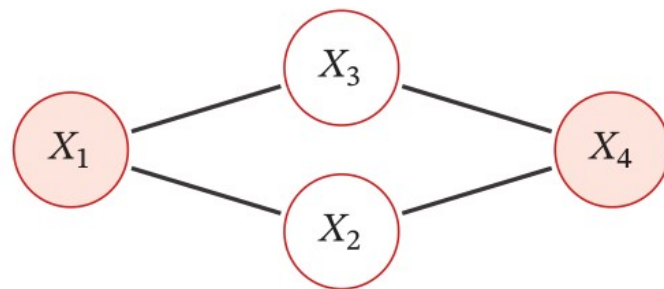
其中 $\mathcal{N}(k)$ 为变量 X_k 的邻居集合, $\setminus k$ 为除 X_k 外其他变量的集合, 那么 $(G, \mathbf{X})$ 就构成了一个**马尔可夫随机场**.

无向图的马尔可夫性

无向图的马尔可夫性 无向图中的马尔可夫性可以表示为

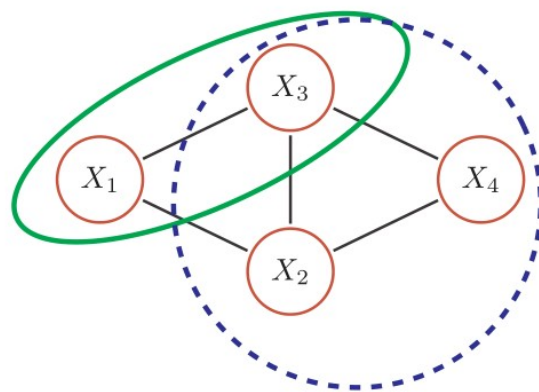
$$X_k \perp\!\!\!\perp \mathbf{X}_{\setminus N(k), \setminus k} \mid \mathbf{X}_{N(k)},$$

其中 $\mathbf{X}_{\setminus N(k), \setminus k}$ 表示除 $\mathbf{X}_{N(k)}$ 和 X_k 外的其它变量。



团 (Clique)

- ▶ 团：一个全连通子图，即团内的所有节点之间都连边。



共有 7 个
团

Hammersley-Clifford 定理

- ▶ 无向图的联合概率可以分解为一系列定义在最大团上的非负函数的乘积形式。

定理 11.1 – Hammersley-Clifford 定理： 如果一个分布 $p(\mathbf{x}) > 0$ 满足无向图 G 中的局部马尔可夫性质, 当且仅当 $p(\mathbf{x})$ 可以表示为一系列定义在最大团上的非负函数的乘积形式, 即

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{x}_c), \quad (11.16)$$

其中 $\mathcal{C}$ 为 G 中的最大团集合, $\phi_c(\mathbf{x}_c) \geq 0$ 是定义在团 c 上的**势能函数** (Potential Function), Z 是**配分函数** (Partition Function), 用来将乘积归一化为概率形式:

$$Z = \sum_{\mathbf{x} \in \mathcal{X}} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{x}_c), \quad (11.17)$$

其中 $\mathcal{X}$ 为随机向量 X 的取值空间.

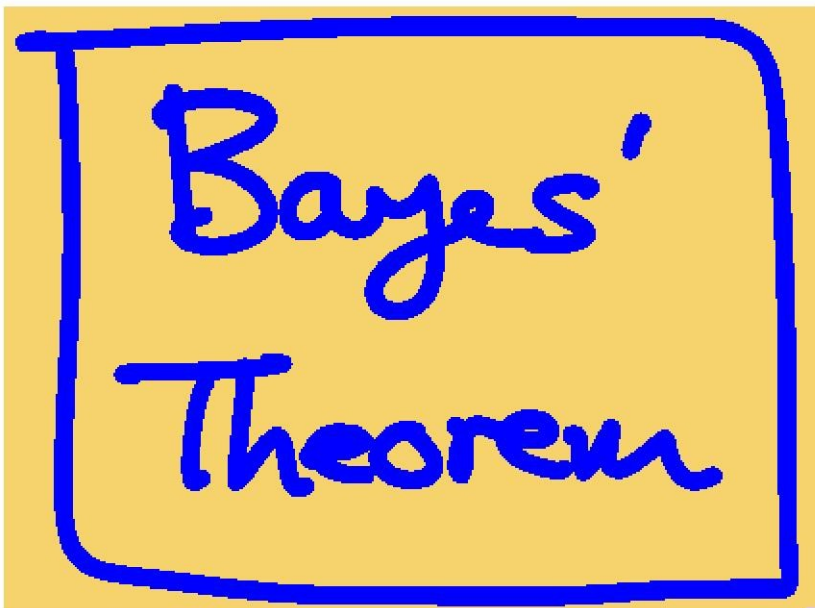
无向图模型

- ▶ 无向图模型的联合分布可以表示为

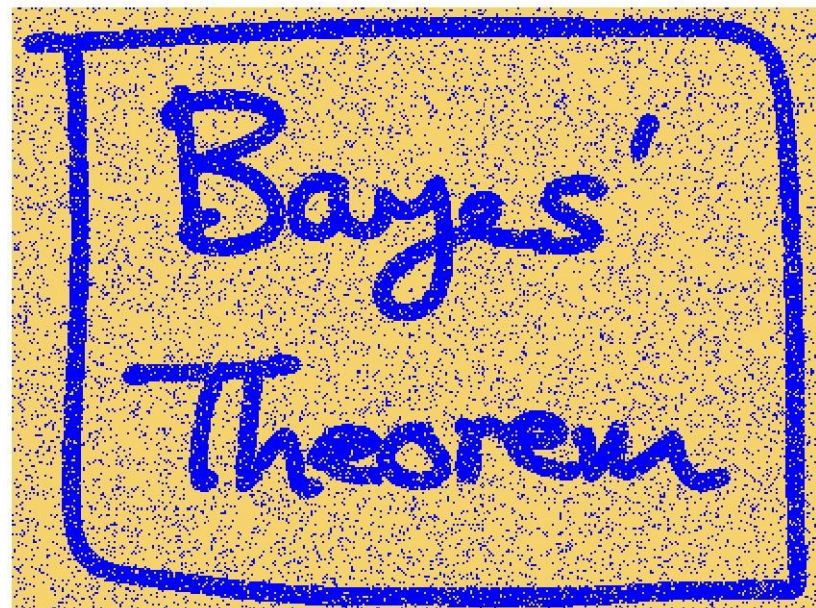
$$\begin{aligned} P(\mathbf{X} = \mathbf{x}) &= \frac{1}{Z} \prod_{c \in \mathcal{C}} \exp(-E(X_c)) \\ &= \frac{1}{Z} \exp\left(\sum_{c \in \mathcal{C}} -E(X_c)\right) \end{aligned}$$

- ▶ 其中 $E(X_c)$ 为能量函数， Z 是配分函数。

Illustration: Image De-Noising (1)

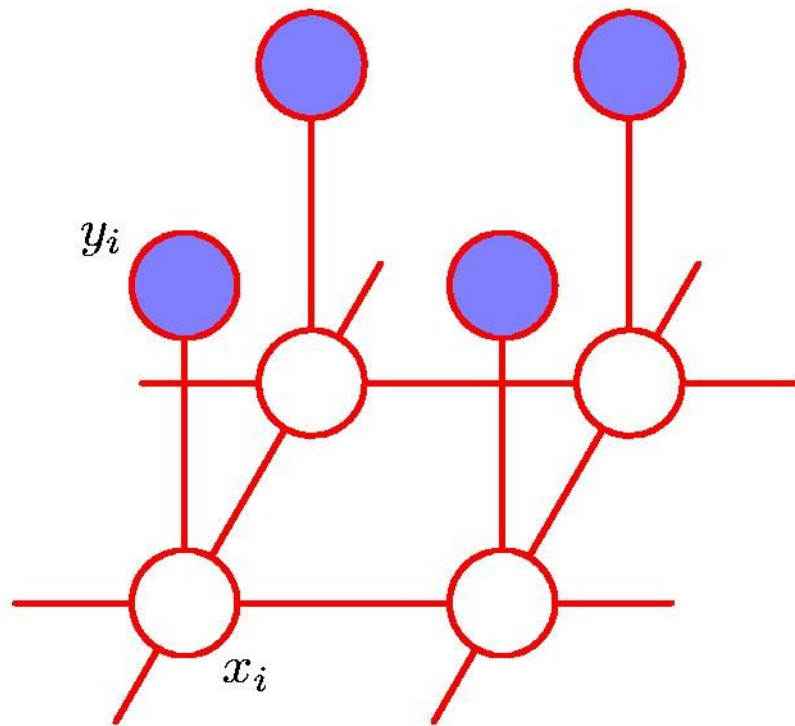


Original Image



Noisy Image

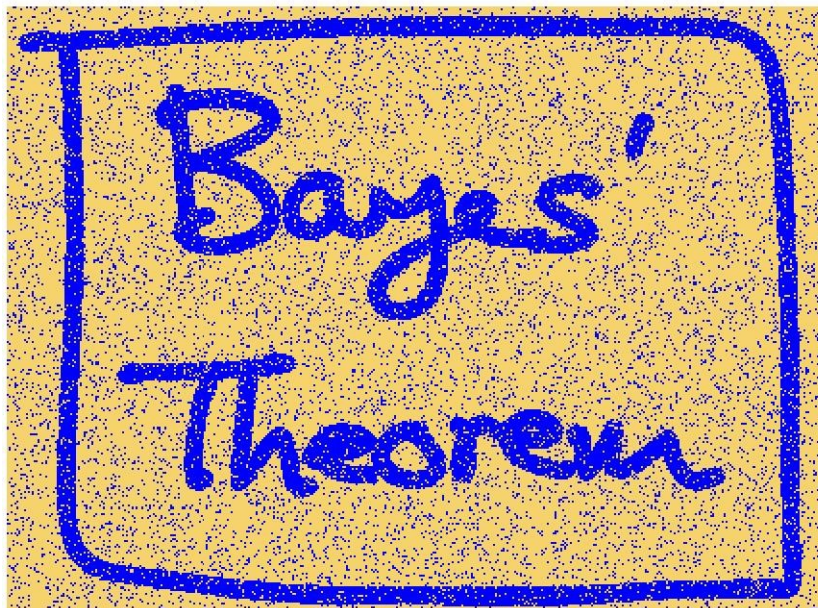
Illustration: Image De-Noising (2)



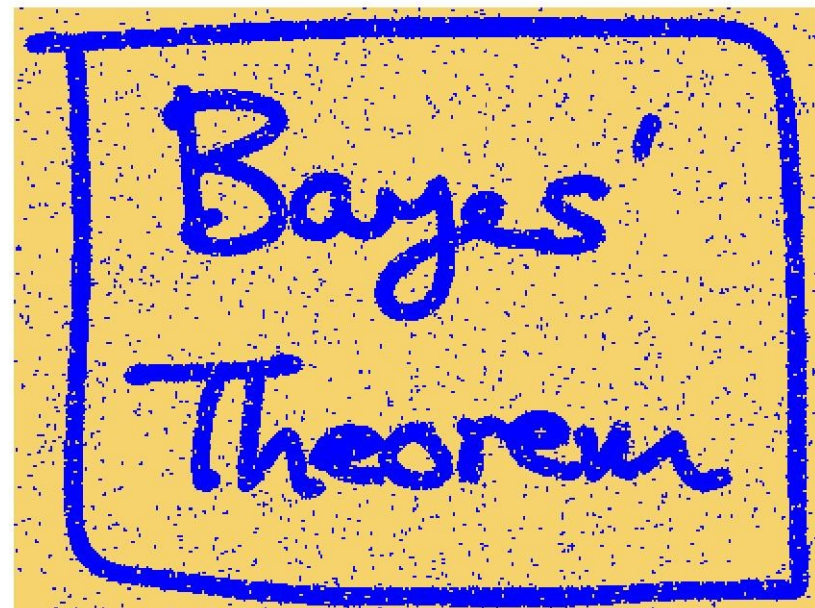
$$E(\mathbf{x}, \mathbf{y}) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i$$

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\{-E(\mathbf{x}, \mathbf{y})\}$$

Illustration: Image De-Noising (3)

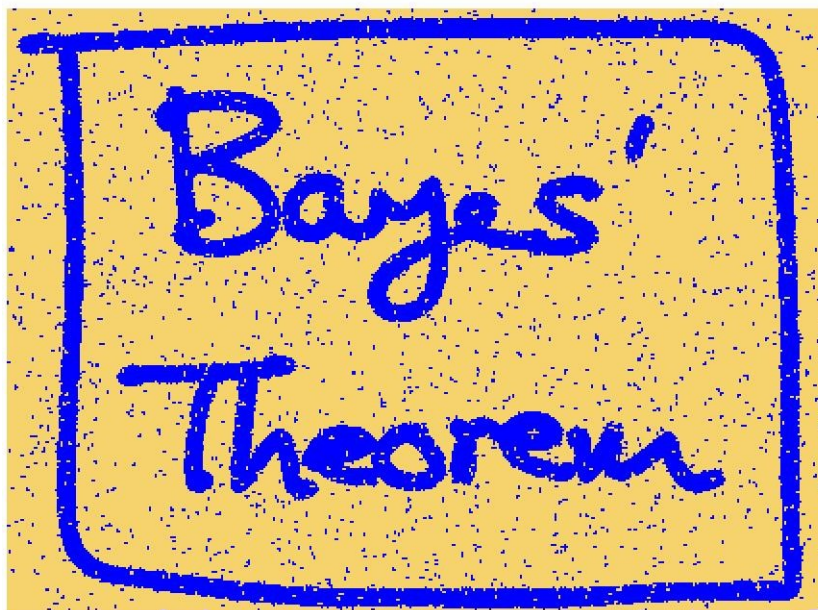


Noisy Image

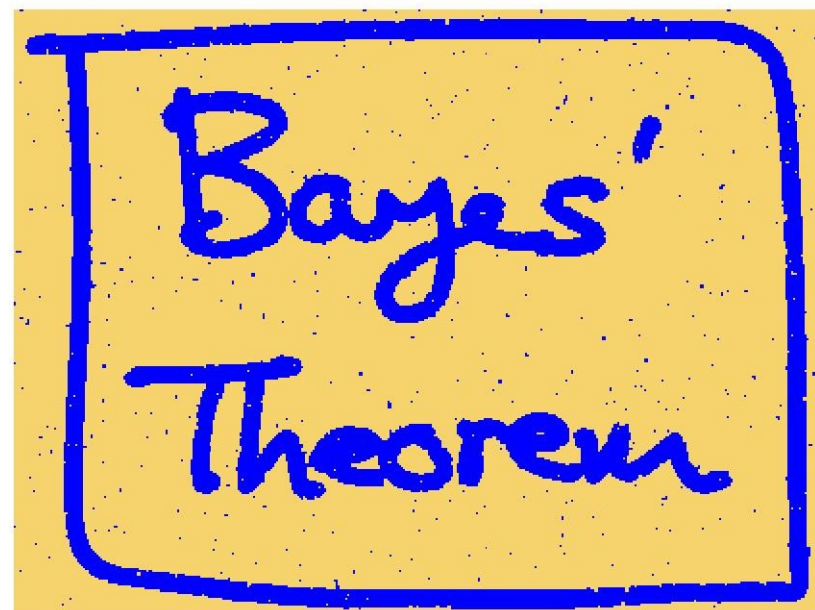


Restored Image (ICM)

Illustration: Image De-Noising (4)



Restored Image (ICM)



Restored Image (Graph cuts)

常见的无向图模型

▶ 对数线性模型

▶ 势能函数的一般定义为

$$\phi_c(\mathbf{x}_c|\theta_c) = \exp\left(\theta_c^T f_c(\mathbf{x}_c)\right)$$

▶ 联合概率 $p(\mathbf{x})$ 的对数形式为

$$\log p(\mathbf{x}|\theta) = \sum_{c \in C} \theta_c^T f_c(\mathbf{x}_c) - \log Z(\theta)$$

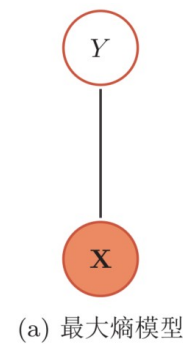
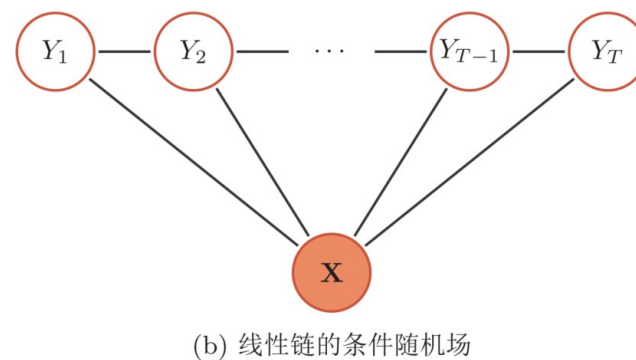
▶ 也称为最大熵模型

▶ 条件随机场

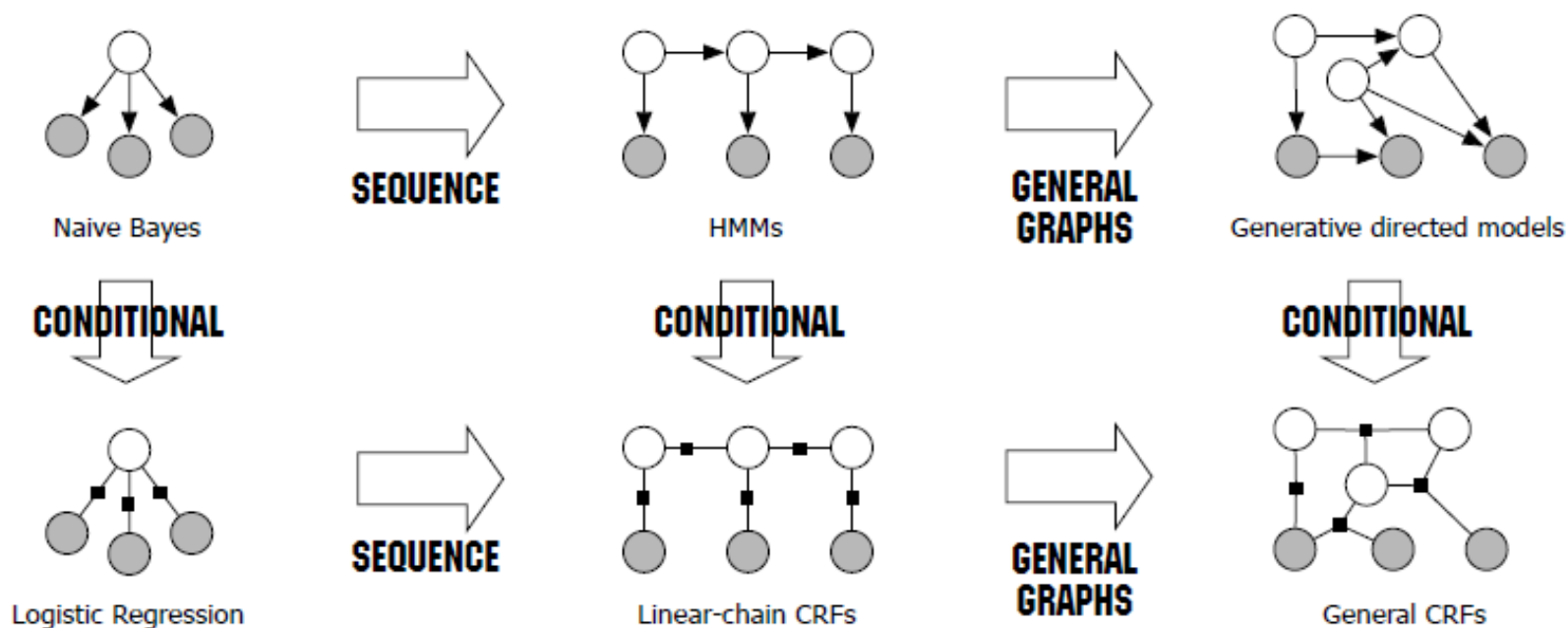
▶ y 一般为随机向量

▶ 条件概率 $p(y|\mathbf{x})$

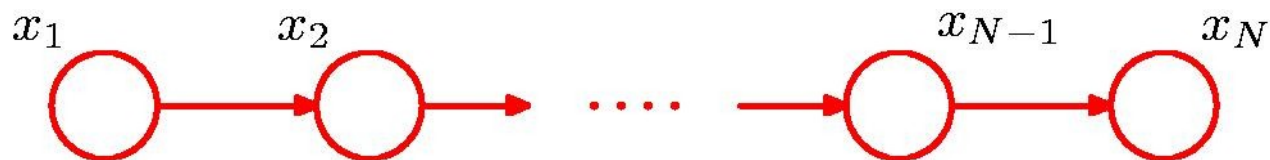
$$p(\mathbf{y}|\mathbf{x}, \theta) = \frac{1}{Z(\mathbf{x}, \theta)} \exp\left(\sum_{c \in C} \theta_c^T f_c(\mathbf{x}, \mathbf{y}_c)\right)$$



模型对比

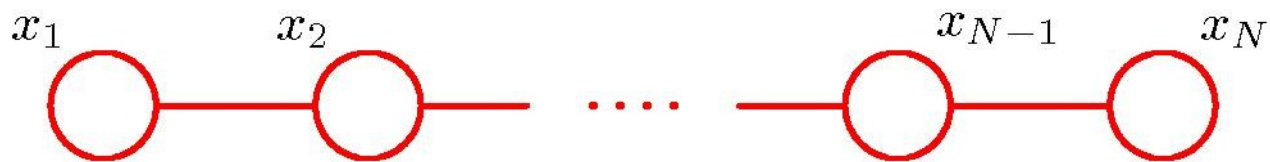


有向图和无向图的转换

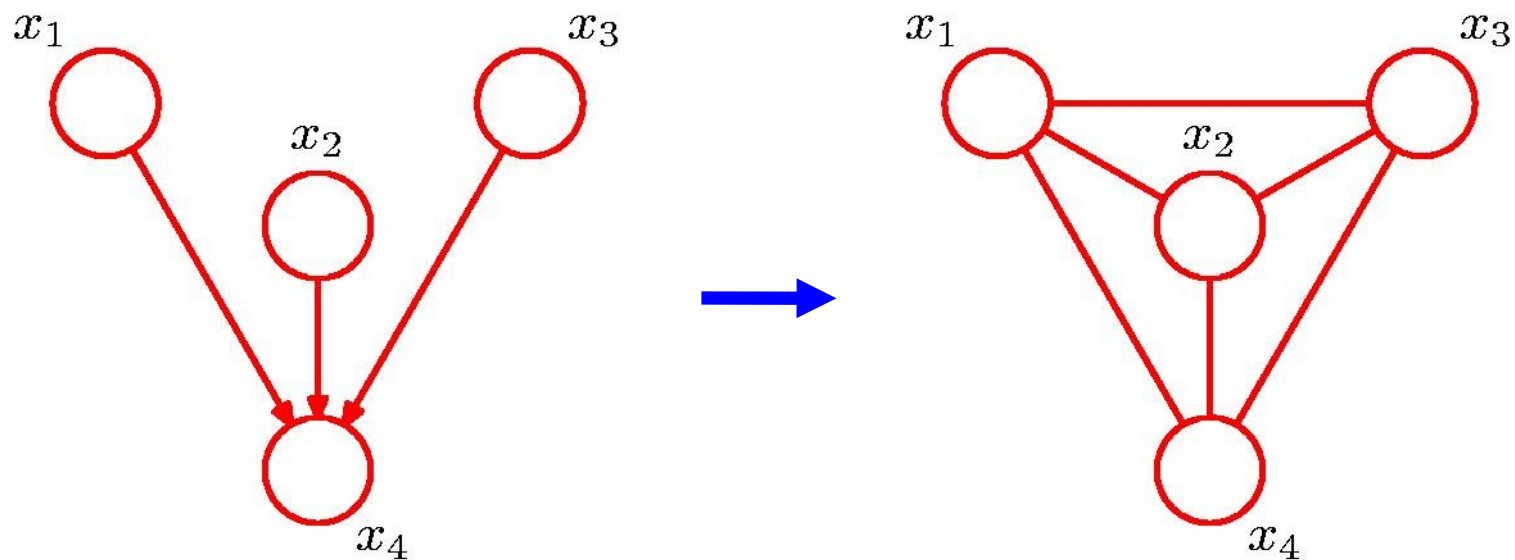


$$p(\mathbf{x}) = p(x_1) \underbrace{p(x_2|x_1)} \underbrace{p(x_3|x_2)} \cdots \underbrace{p(x_N|x_{N-1})}$$

$$p(\mathbf{x}) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N)$$



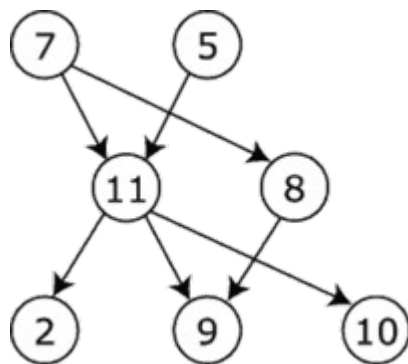
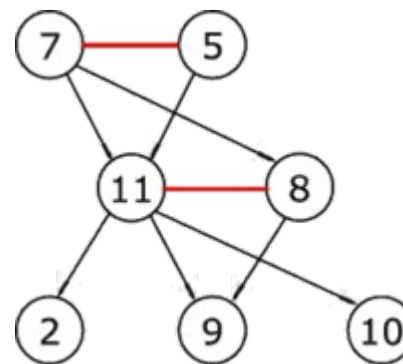
有向图和无向图的转换



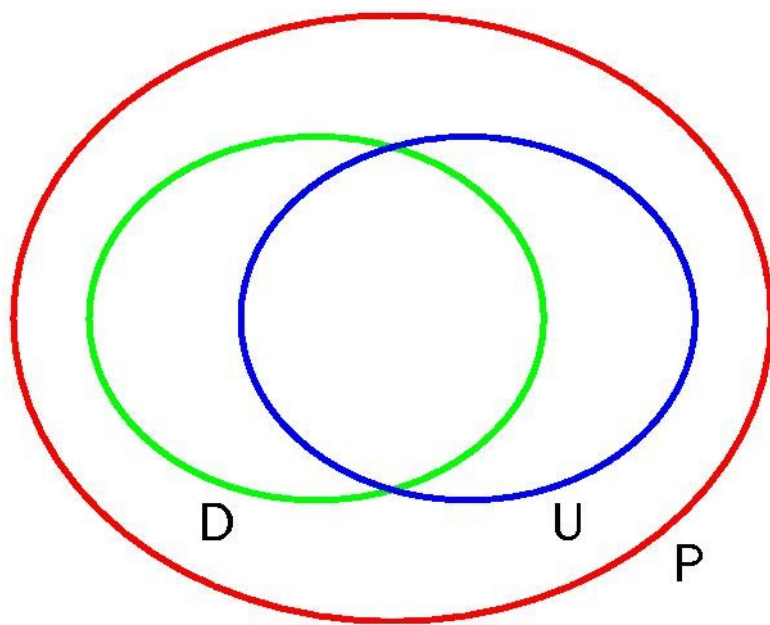
$$\begin{aligned} p(\mathbf{x}) &= p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \\ &= \frac{1}{Z} \psi_A(x_1, x_2, x_3) \psi_B(x_2, x_3, x_4) \psi_C(x_1, x_2, x_4) \end{aligned}$$

道德图 (Moral Graph)

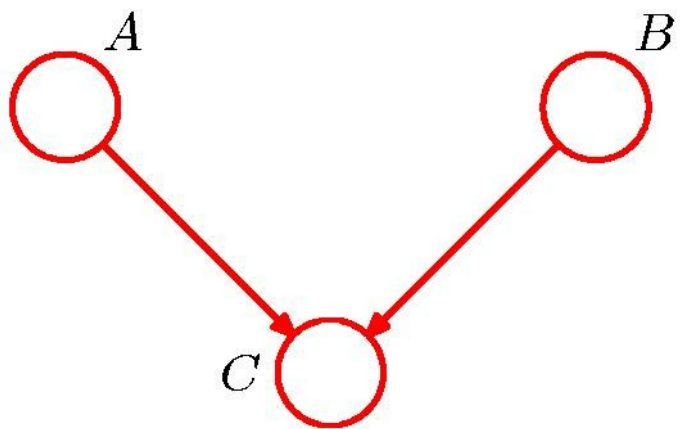
- A moral graph of a directed acyclic graph G is an undirected graph in which each node of the original G is now connected to its Markov blanket.



有向图和无向图

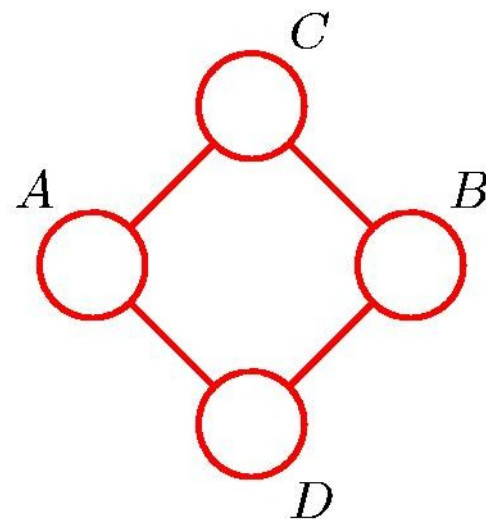


有向图和无向图



$$A \perp\!\!\!\perp B \mid \emptyset$$

$$A \not\perp\!\!\!\perp B \mid C$$



$$A \not\perp\!\!\!\perp B \mid \emptyset$$

$$A \perp\!\!\!\perp B \mid C \cup D$$

$$C \perp\!\!\!\perp D \mid A \cup B$$



学习

有向图模型

- ▶ 在贝叶斯网络中，所有变量 x 的联合概率分布可以分解为每个随机变量 x_k 的局部条件概率的连乘形式。
- ▶ 假设每个局部条件概率 $p(x_k | x_{\pi_k})$ 的参数为 θ_k ，则对数似然函数为

$$\begin{aligned}\mathcal{L}(\mathcal{D}; \theta) &= \frac{1}{N} \sum_{n=1}^N \log p(\mathbf{x}^{(n)}; \theta) \\ &= \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K \log p(x_k^{(n)} | x_{\pi_k}^{(n)}; \theta_k),\end{aligned}$$

- ▶ 分别最大化每个变量的条件似然来估计其参数

$$\theta_k = \arg \max \sum_{n=1}^N \log p(x_k^{(n)} | x_{\pi_k}^{(n)}; \theta_k)$$

无向图模型

► 以对数线性模型为例，

$$p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{c \in \mathcal{C}} \theta_c^\top f_c(\mathbf{x}_c)\right),$$

给定 N 个训练样本 $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^N$ ，其对数似然函数为

$$\begin{aligned} \mathcal{L}(\mathcal{D}; \theta) &= \frac{1}{N} \sum_{n=1}^N \log p(\mathbf{x}^{(n)}; \theta) \\ &= \frac{1}{N} \sum_{n=1}^N \left(\sum_{c \in \mathcal{C}} \theta_c^\top f_c(\mathbf{x}_c^{(n)}) \right) - \log Z(\theta), \end{aligned}$$

► 偏导数

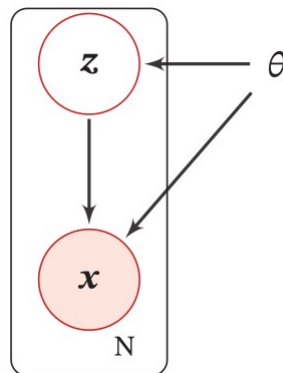
$$\begin{aligned} \frac{\partial \mathcal{L}(\mathcal{D}; \theta)}{\partial \theta_c} &= \frac{1}{N} \sum_{n=1}^N f_c(\mathbf{x}_c^{(n)}) - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}; \theta)} \left[f_c(\mathbf{x}_c) \right] \\ &= \mathbb{E}_{\mathbf{x} \sim \tilde{p}(\mathbf{x})} \left[f_c(\mathbf{x}_c) \right] - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}; \theta)} \left[f_c(\mathbf{x}_c) \right], \end{aligned}$$



含隐变量的参数估计

含隐变量的参数学习

- ▶ 隐变量即变量是不可观测的



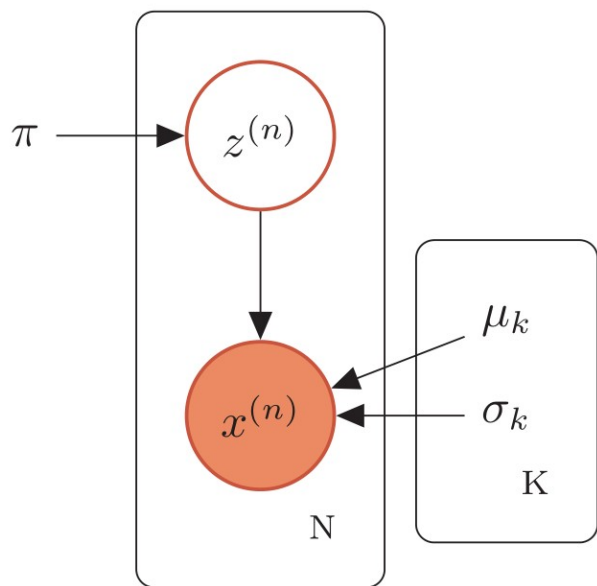
- ▶ 边际似然函数 (Marginal Likelihood)

$$p(\mathbf{x}; \theta) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta)$$

- ▶ 需要用 EM 算法进行参数估计

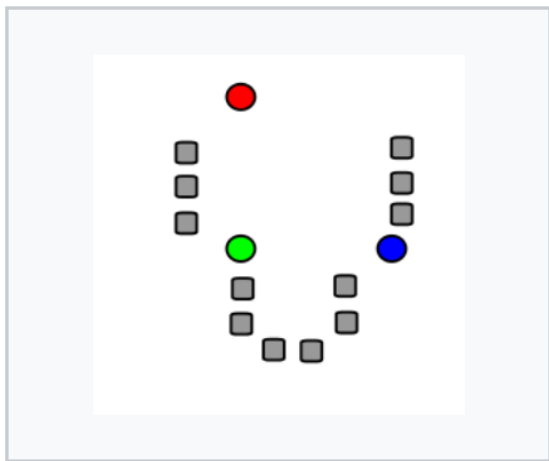
高斯混合模型

图模型表示

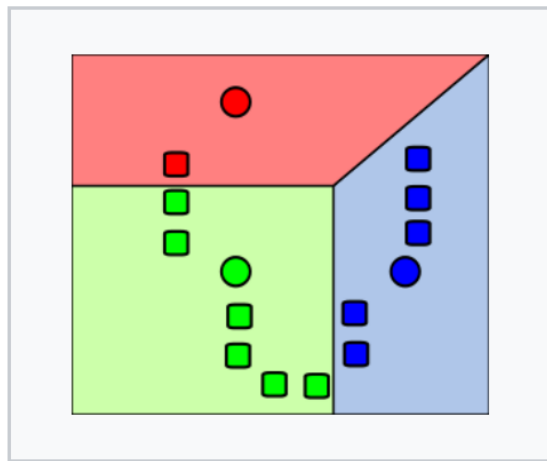


$$\mathcal{N}(x|\mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(x - \mu_k)^2}{2\sigma_k^2}\right)$$

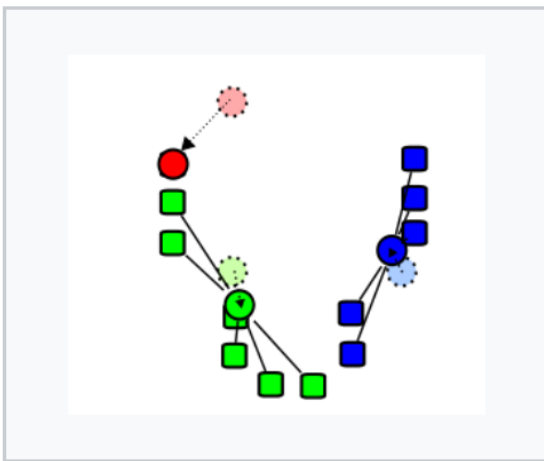
一个简单的解法： K-means



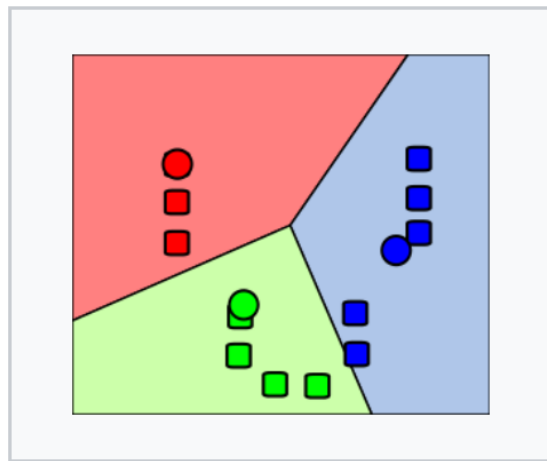
1. k initial "means" (in this case $k=3$) are randomly generated within the data domain (shown in color).



2. k clusters are created by associating every observation with the nearest mean. The partitions here represent the **Voronoi diagram** generated by the means.



3. The **centroid** of each of the k clusters becomes the new mean.



4. Steps 2 and 3 are repeated until convergence has been reached.

K-means 算法

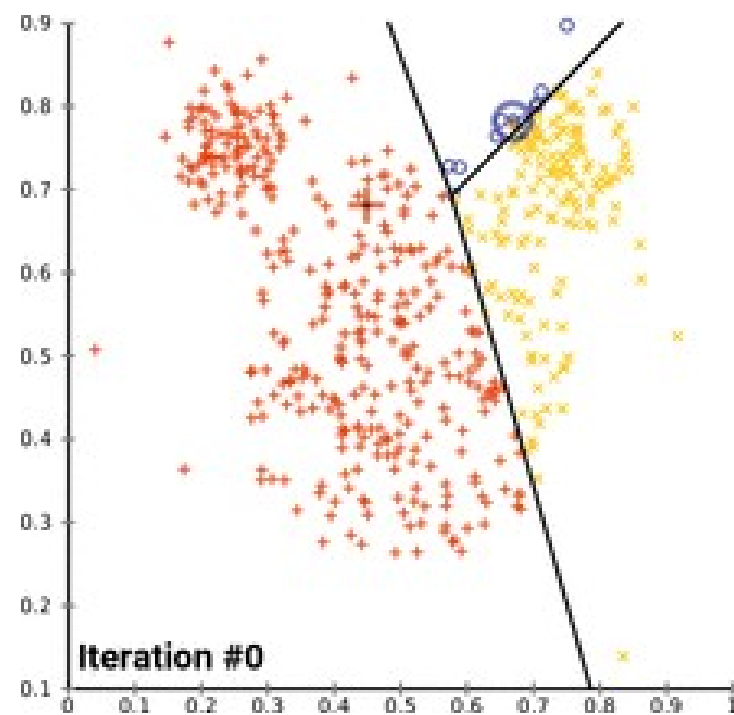
- ▶ 初始化中心点 $m_1^{(1)}, \dots, m_k^{(1)}$
- ▶ 迭代执行下面两步
 - ▶ 分配步 (Assignment step) :

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \forall j, 1 \leq j \leq k\},$$

- ▶ 更新步 (Update step)

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

K-means 算法



期望最大化（Expectation-Maximum，EM）算法

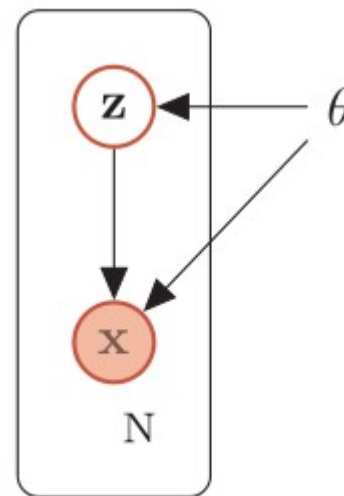
- 假设有一组变量，有部分变量是不可观测的，如何进行参数估计呢？

对数边际似然函数

$$\begin{aligned}\log p(\mathbf{x}; \theta) &= \log \sum_z q(z) \frac{p(\mathbf{x}, z; \theta)}{q(z)} \\ &\geq \sum_z q(z) \log \frac{p(\mathbf{x}, z; \theta)}{q(z)} \\ &\triangleq ELBO(q, \mathbf{x}; \theta),\end{aligned}$$

利用 Jensen 不等式

证据下界



另外一种推导

$$\begin{aligned}\log p(\mathbf{x}; \theta) &= \sum_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{x}; \theta) \\&= \sum_{\mathbf{z}} q(\mathbf{z}) \left(\log p(\mathbf{x}, \mathbf{z}; \theta) - \log p(\mathbf{z}|\mathbf{x}; \theta) \right) \\&= \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}; \theta)}{q(\mathbf{z})} - \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{z}|\mathbf{x}; \theta)}{q(\mathbf{z})} \\&= ELBO(q, \mathbf{x}; \theta) + \text{KL}(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x}; \theta)),\end{aligned}$$

EM 算法

$$ELBO(q, \mathbf{x}; \theta) + \text{KL}(q(\mathbf{z}) || p(\mathbf{z} | \mathbf{x}; \theta))$$

► E 步

$$q_{t+1}(\mathbf{z}) = \arg \max_q ELBO(q, \mathbf{x} | \theta_t)$$

► M 步

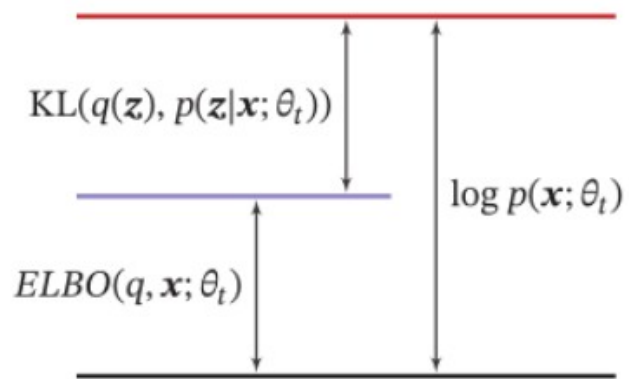
$$\theta_{t+1} = \arg \max_{\theta} ELBO(q_{t+1}, \mathbf{x} | \theta)$$

收敛性

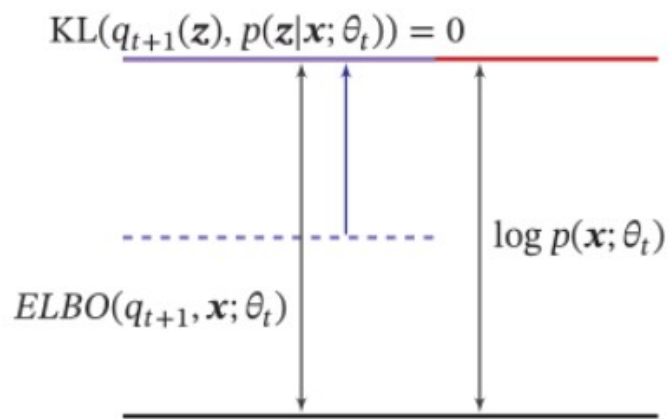
$$\log p(\mathbf{x}; \theta_{t+1}) \geq \boxed{ELBO(q_{t+1}, \mathbf{x}; \theta_{t+1})} \geq \boxed{ELBO(q_{t+1}, \mathbf{x}; \theta_t) = \log p(\mathbf{x}; \theta_t)}$$

M
步

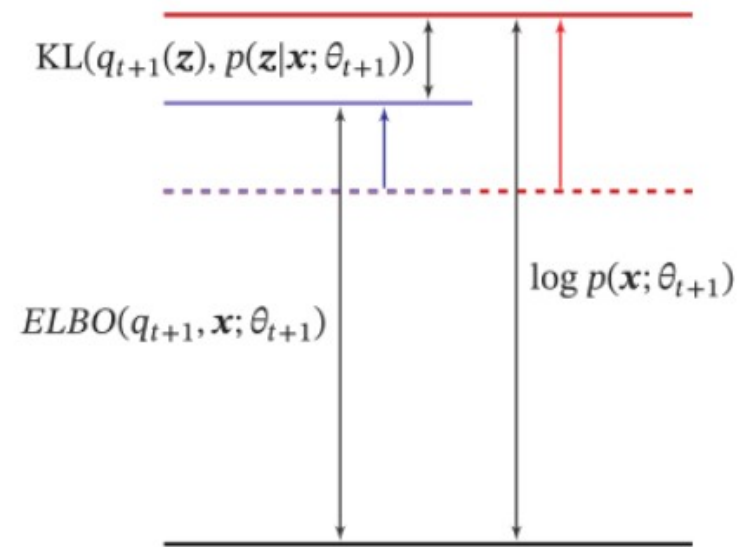
E
步



(a) 初始状态



(b) E 步更新



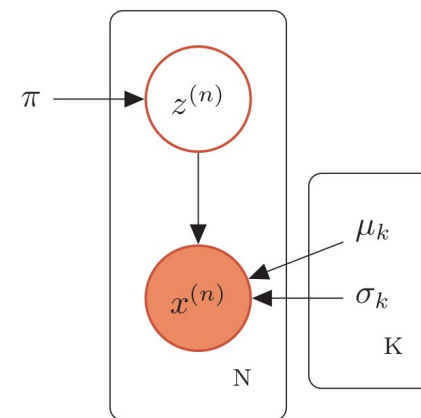
(c) M 步更新

GMM Revisit

E 步 先固定参数 μ, σ , 计算后验分布 $p(z^{(n)}|x^{(n)})$

$$\begin{aligned}\gamma_{nk} &\triangleq p(z^{(n)} = k|x^{(n)}) \\ &= \frac{p(z^{(n)})p(x^{(n)}|z^{(n)})}{p(x^{(n)})} \\ &= \frac{\pi_k \mathcal{N}(x^{(n)}|\mu_k, \sigma_k)}{\sum_{k=1}^K \pi_k \mathcal{N}(x^{(n)}|\mu_k, \sigma_k)},\end{aligned}$$

其中 γ_{nk} 定义了样本 $x^{(n)}$ 属于第 k 个高斯分布的后验概率。



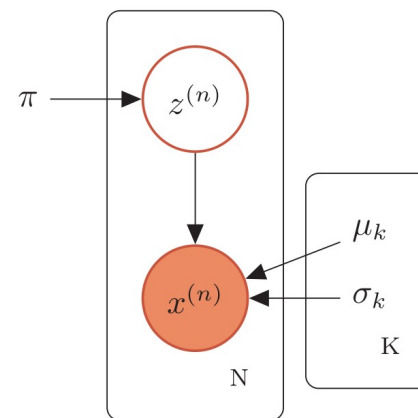
$$\mathcal{N}(x|\mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(x - \mu_k)^2}{2\sigma_k^2}\right)$$

GMM Revisit

M 步 令 $q(z = k) = \gamma_{nk}$, 训练集 $\mathcal{D}$ 的证据下界为

$$\begin{aligned} ELBO(\gamma, \mathcal{D} | \pi, \mu, \sigma) &= \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk} \log \frac{p(x^{(n)}, z^{(n)} = k)}{\gamma_{nk}} \\ &= \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk} \left(\log \mathcal{N}(x^{(n)} | \mu_k, \sigma_k) + \log \frac{\pi_k}{\gamma_{nk}} \right) \\ &= \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk} \left(-\frac{(x - \mu_k)^2}{2\sigma_k^2} - \log \sigma_k + \log \pi_k \right) + C, \end{aligned}$$

其中 C 为和参数无关的常数。



$$\mathcal{N}(x | \mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(x - \mu_k)^2}{2\sigma_k^2}\right)$$

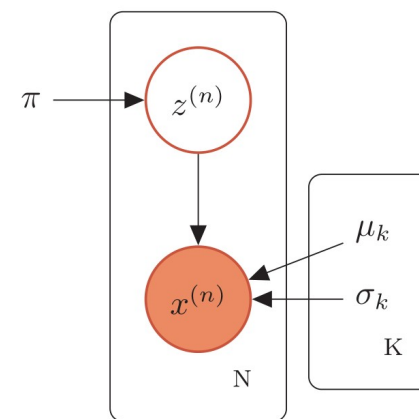
GMM Revisit

$$N_k = \sum_{n=1}^N \gamma_{nk}.$$

$$\pi_k = \frac{N_k}{N},$$

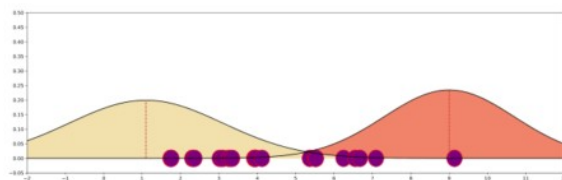
$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma_{nk} x^{(n)},$$

$$\sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma_{nk} (x^{(n)} - \mu_k)^2,$$

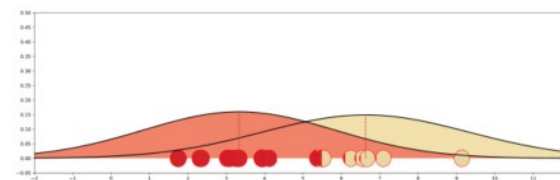


$$\mathcal{N}(x|\mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(x - \mu_k)^2}{2\sigma_k^2}\right)$$

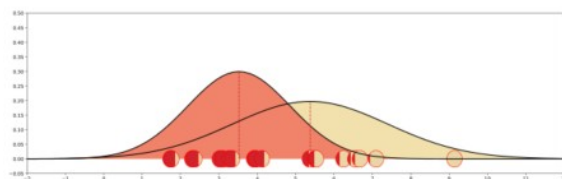
GMM 的参数学习



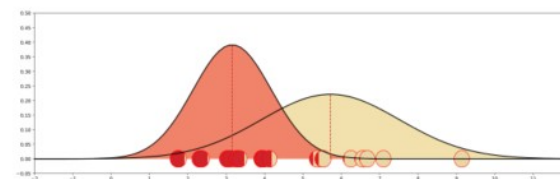
(a) 初始化



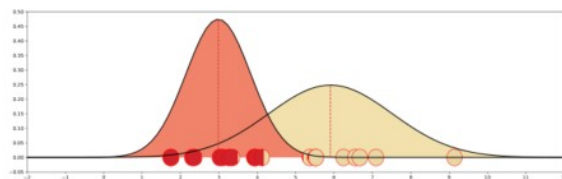
(b) 第 1 次迭代



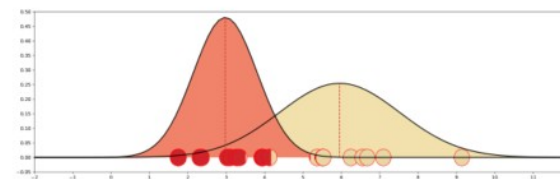
(c) 第 4 次迭代



(d) 第 8 次迭代

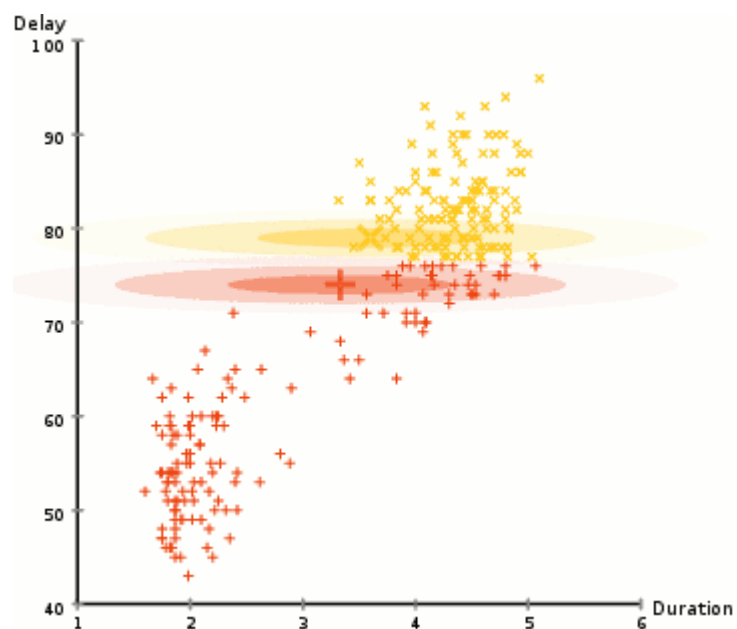
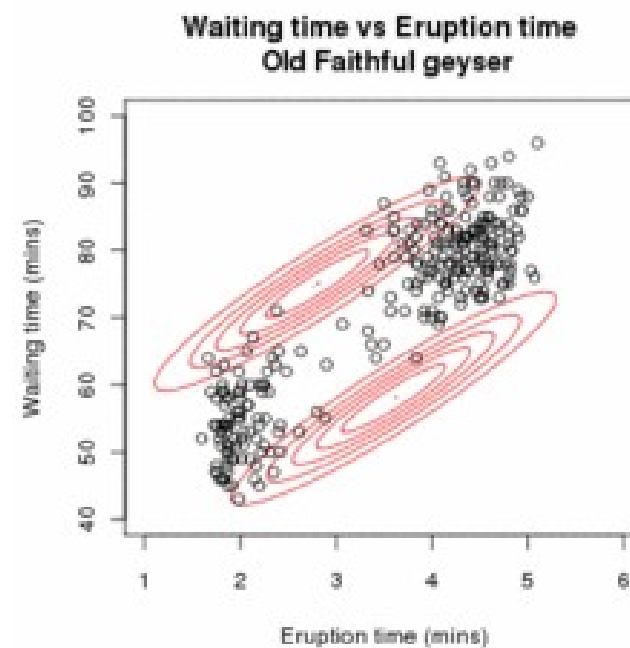


(e) 第 12 次迭代



(f) 第 16 次迭代

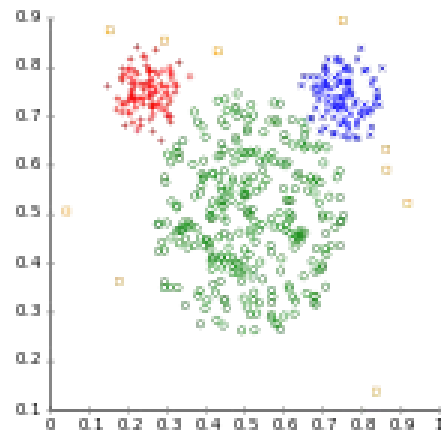
GMM 的参数学习



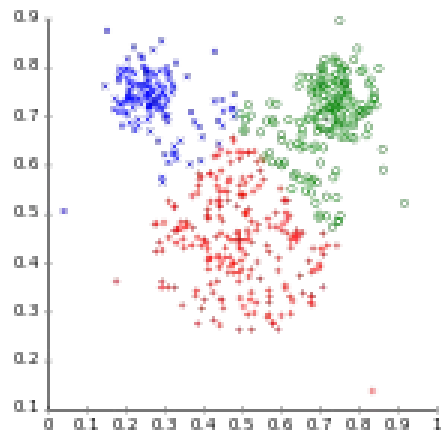
k-means clustering vs. EM clustering

Different cluster analysis results on "mouse" data set:

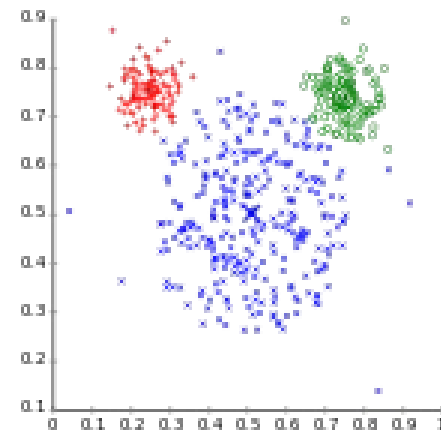
Original Data



k-Means Clustering



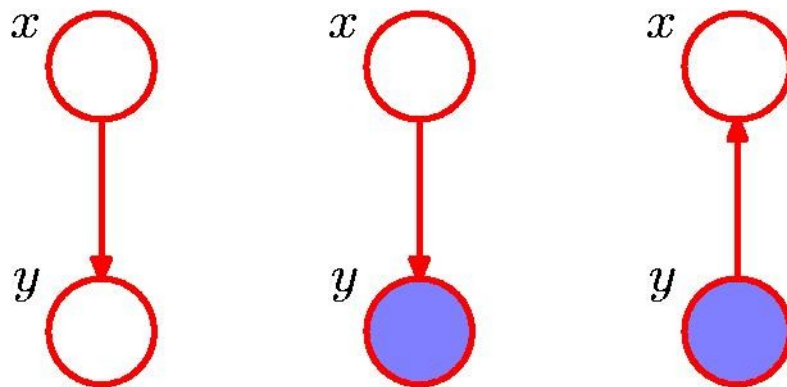
EM Clustering





推断 (Inference)

Inference in Graphical Models



$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

$$p(y) = \sum_{x'} p(y|x')p(x')$$

推断

- ▶ 精确推断 Exact Inference
 - ▶ 变量消去法 Variable Elimination Algorithm
 - ▶ 信念传播
 - ▶ 联合树算法
- ▶ 近似推断 Approximate Inference
 - ▶ 变分推断
 - ▶ 采样法 (蒙特卡罗方法)
- ▶ 主要在于如何计算边际概率 $p(y)$.

如何有效地计算边际概率？

► 边际概率

$$p(x_1, x_4) = \sum_{x_2, x_3} p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3)$$

► 如何减少计算量？

► 乘法的分配律

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

变量消去法

▶ 边际概率

$$p(x_1, x_4) = \sum_{x_2, x_3} p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3)$$

$$p(x_1, x_4) = p(x_1) \sum_{x_3} p(x_3|x_1) \sum_{x_2} p(x_2|x_1)p(x_4|x_2, x_3)$$

▶ 变量消除法可以按照不同的顺序来消除变量

▶ 边际概率 $p(x_4)$

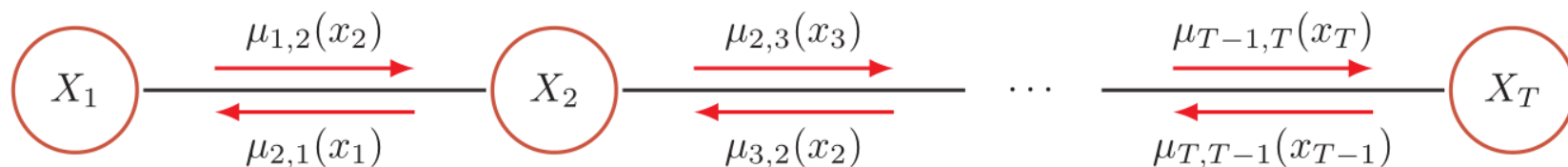
$$p(x_4) = \sum_{x_3} \sum_{x_2} p(x_4|x_2, x_3) \sum_{x_1} p(x_3|x_1)p(x_2|x_1)p(x_1)$$

如何减少重复计算？

信念传播算法

- ▶ 信念传播（Belief Propagation，BP）算法
 - ▶ 也称为和积（Sum-Product）算法或消息传递（Message Passing）算法，
- ▶ 将变量消除法中的和积（Sum-Product）操作看作是消息，并保存起来，这样可以节省大量的计算资源。

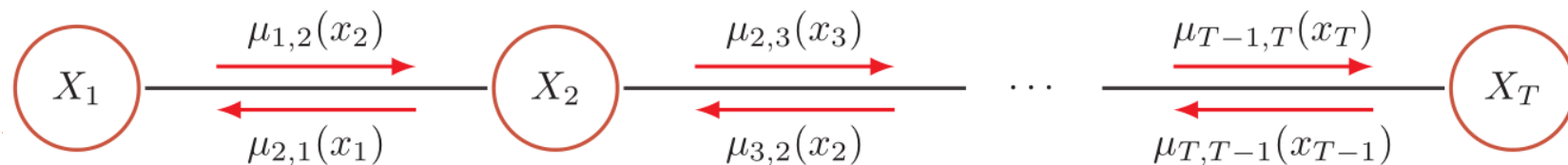
链上的消息传递



第 t 个变量的边际概率 $p(x_t)$ 为

$$\begin{aligned} p(x_t) &= \sum_{x_1} \cdots \sum_{x_{t-1}} \sum_{x_{t+1}} \cdots \sum_{x_T} p(\mathbf{x}) \\ &= \frac{1}{Z} \sum_{x_1} \cdots \sum_{x_{t-1}} \sum_{x_{t+1}} \cdots \sum_{x_T} \prod_{t=1}^{T-1} \phi(x_t, x_{t+1}). \end{aligned}$$

链上的消息传递



$$\begin{aligned}
 p(x_t) &= \frac{1}{Z} \left(\sum_{x_1} \cdots \sum_{x_{t-1}} \prod_{j=1}^{t-1} \phi(x_j, x_{j+1}) \right) \cdot \left(\sum_{x_{t+1}} \cdots \sum_{x_T} \prod_{j=t}^{T-1} \phi(x_j, x_{j+1}) \right) \\
 &= \frac{1}{Z} \left(\sum_{x_{t-1}} \phi(x_{t-1}, x_t) \cdots \left(\sum_{x_2} \phi(x_2, x_3) \left(\sum_{x_1} \phi(x_1, x_2) \right) \right) \right) \cdot \\
 &\quad \left(\sum_{x_{t+1}} \phi(x_t, x_{t+1}) \cdots \left(\sum_{x_{T-1}} \phi(x_{T-2}, x_{T-1}) \left(\sum_{x_T} \phi(x_{T-1}, x_T) \right) \right) \right) \\
 &= \frac{1}{Z} \mu_{t-1,t}(x_t) \mu_{t+1,t}(x_t), \tag{11.77}
 \end{aligned}$$

$\mu_{t-1,t}(x_t)$ 定义为变量 X_{t-1} 向变量 X_t 传递的消息, 定义为

$$\mu_{t-1,t}(x_t) \triangleq \sum_{x_{t-1}} \phi(x_{t-1}, x_t) \mu_{t-2,t-1}(x_{t-1}).$$

$\mu_{t+1,t}(x_t)$ 是变量 X_{t+1} 向变量 X_t 传递的消息, 定义为

$$\mu_{t+1,t}(x_t) \triangleq \sum_{x_{t+1}} \phi(x_t, x_{t+1}) \mu_{t+2,t+1}(x_{t+1}).$$

链式结构图模型的信念传播过程

1. 依次计算前向传递的消息 $\mu_{t-1,t}(x_t)$, $t = 1, \dots, T - 1$;
2. 依次计算反向传递的消息 $\mu_{t+1,t}(x_t)$, $t = T - 1, \dots, 1$;
3. 在任意节点 t 上计算配分函数 Z ,

$$Z = \sum_{x_t} \mu_{t-1,t}(x_t) \mu_{t+1,t}(x_t).$$

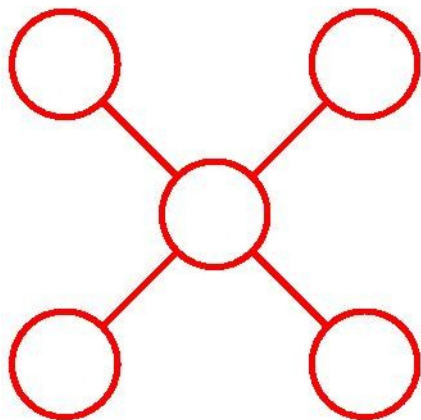
更一般的推断

► 树结构

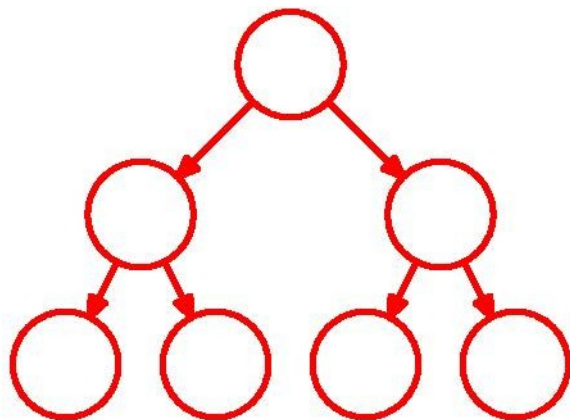
► 和积算法

► 因子图 (Factor Graph)

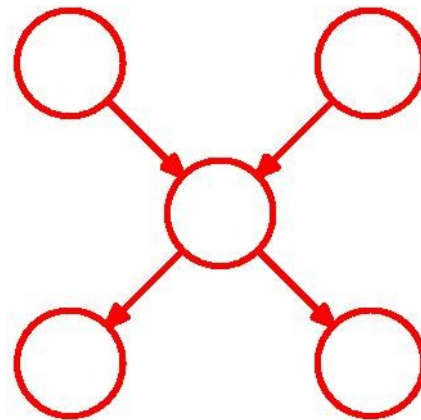
Undirected Tree



Directed Tree



Polytree



近似推断

▶ 环路信念传播 (Loopy Belief Propagation)

- ▶ 在具有环路的图上依然使用和积算法，即使得到不精确解，在某些任务上也可以近似精确解。

▶ 变分法 (Variational Method)

- ▶ 引入一个变分分布（通常是比较简单的分布）来近似这些条件概率，然后通过迭代的方法进行计算。

▶ 采样法 (Sampling Method)

- ▶ 通过模拟的方式来采集符合某个分布 $p(x)$ 的一些样本，并通过这些样本来估计和这个分布有关的运算



基于采样法的近似推断

采样法 (Sampling Method)

- ▶ 假设要推断的概率分布为 $p(x)$ ，并用来计算期望

$$\mathbb{E}_p[f(x)] = \int_x f(x)p(x)dx.$$

- ▶ 采样法

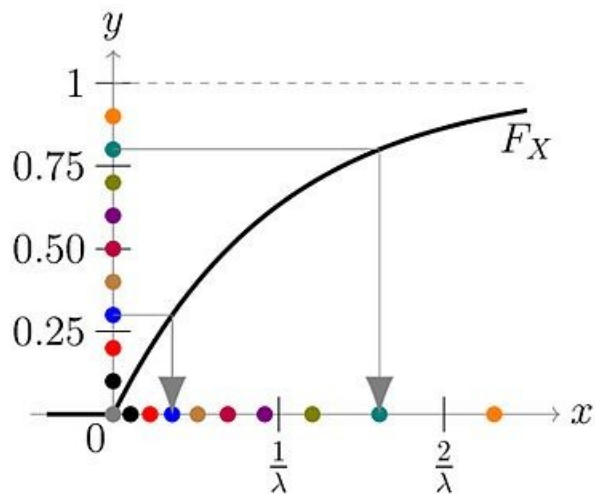
- ▶ 也称为蒙特卡罗方法 (Monte Carlo Method)
- ▶ 数值解：用数值方法求出解，给出一系列对应的自变量和解。
- ▶ 独立抽取 N 个样本，并计算

$$\bar{f}_N = \frac{1}{N} (f(x^{(1)}) + \dots + f(x^{(N)}))$$

- ▶ 根据大数定律，当 N 趋向于无穷大时，样本均值收敛于期望值。

采样法

- ▶ 直接采样
 - ▶ 均匀分布
- ▶ 逆变换采样 (Inverse Transform Sampling)



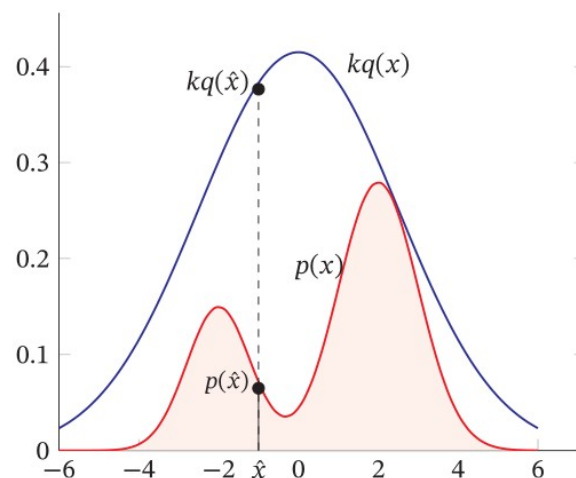
- ▶ Box-Muller 方法
 - ▶ 正态分布

- ▶ 拒绝采样
- ▶ 重要性采样
- ▶ 马尔可夫链蒙特卡罗 (Markov Chain Monte Carlo, MCMC) 方法
 - ▶ Metropolis-Hastings 算法
 - ▶ Metropolis 算法
 - ▶ Gibbs 采样法

拒绝采样

- ▶ 假设原始分布 $p(x)$ 难以直接采样
- ▶ 引入一个容易采样的分布 $q(x)$
 - ▶ 一般称为提议分布 (Proposal Distribution)
- ▶ 流程：
 - ▶ 先从分布 $q(x)$ 中采样
 - ▶ 再以某个标准来拒绝一部分的样本
- ▶ 使得最终采集的样本服从分布 $p(x)$

- ▶ 已知未归一化的分布 $\hat{p}(x)$,
- ▶ 构建一个提议分布 $q(x)$ 和一个常数 k , 使得 $kq(x)$ 可以覆盖函数 $\hat{p}(x)$, 即 $kq(x) \geq \hat{p}(x), \forall x$.



拒绝采样

算法 11.2 拒绝采样的采样过程

输入: 提议分布 $q(x)$, 常数 k , 样本集合 $\mathcal{V} = \emptyset$;

1 **repeat**

2 根据 $q(x)$ 随机生成一个样本 $\hat{x}$;

3 计算接受概率 $\alpha(\hat{x})$;

4 从 $(0, 1)$ 的均匀分布中随机生成一个值 z ;

5 **if** $z \leq \alpha(\hat{x})$ **then**

6 $\mathcal{V} = \mathcal{V} \cup \{\hat{x}\}$;

7 **end**

8 **until** 获得 N 个样本 ($|\mathcal{V}| = N$);

输出: 样本集合 $\mathcal{V}$

// 以 $\alpha(\hat{x})$ 的概率接受 $\hat{x}$

$$\alpha(\hat{x}) = \frac{\hat{p}(\hat{x})}{kq(\hat{x})}$$

接受概率 (Acceptance
Probability)

重要性采样

- 如果采样的目的是计算分布 $p(x)$ 下函数 $f(x)$ 的期望，那么实际上抽取的样本不需要严格服从分布 $p(x)$ 。也可以通过另一个分布，即提议分布 $q(x)$ ，直接采样并估计 $E_p[f(x)]$ 。

$$\begin{aligned} \mathbb{E}_p[f(x)] &= \int_x f(x)p(x)dx \\ &= \int_x f(x)\frac{p(x)}{q(x)}q(x)dx \\ &= \int_x f(x)w(x)q(x)dx \\ &= \mathbb{E}_q[f(x)w(x)]. \end{aligned}$$



$$\hat{f}_N = \frac{1}{N} (f(x^{(1)})w(x^{(1)}) + \dots + f(x^{(N)})w(x^{(N)}))$$

$w(x)$ 为重要性权重

马尔可夫过程

- ▶ 随机过程 (stochastic process) 是一组随机变量 X_t 的集合，其中 t 属于一个索引 (index) 集合 T 。
 - ▶ 引集合 T 可以定义在时间域或者空间域。
- ▶ 在随机过程中，马尔可夫性质 (Markov property) 是指一个随机过程在给定现在状态及所有过去状态情况下，其未来状态的条件概率分布仅依赖于当前状态。

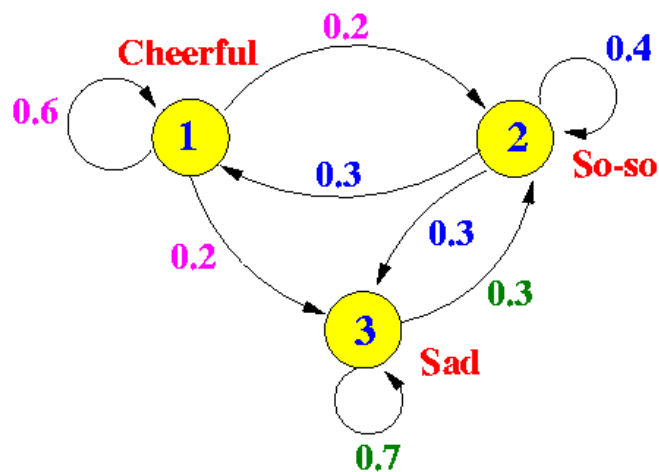
$$P(X_{t+1} = x_{t+1} | X_{0:t} = x_{0:t}) = P(X_{t+1} = x_{t+1} | X_t = x_t)$$

马尔可夫链

- ▶ 离散时间的马尔可夫过程也称为马尔可夫链（Markov chain）
 - ▶ 状态转移矩阵

$$P(X_{t+1} = s | X_t = s') = m_{ss'}$$

- ▶ 例子



平稳分布 (Stationary Distribution)

平稳分布 假设状态空间大小为 K , 向量 $\boldsymbol{\pi} = [\pi_1, \dots, \pi_K]^\top$ 为状态空间中的一个分布, 满足 $0 \leq \pi_k \leq 1$ 和 $\sum_{k=1}^K \pi_k = 1$.

对于状态转移矩阵为 $\mathbf{M}$ 的时间同质的马尔可夫链, 若存在一个分布 $\boldsymbol{\pi}$ 满足

$$\boldsymbol{\pi} = \mathbf{M}\boldsymbol{\pi}, \quad (\text{D.47})$$

则称分布 $\boldsymbol{\pi}$ 为该马尔可夫链的**平稳分布** (Stationary Distribution). 根据特征向量的定义可知, $\boldsymbol{\pi}$ 为矩阵 $\mathbf{M}$ 的 (归一化) 的对应特征值为 1 的特征向量.

细致平稳条件

定理 D.1 – 细致平稳条件 (Detailed Balance Condition): 给定一个状态空间中的分布 $\pi \in [0, 1]^K$, 如果一个状态转移矩阵为 $M \in \mathbb{R}^{K \times K}$ 的马尔可夫链满足

$$\pi_i m_{ij} = \pi_j m_{ji}, \quad \forall 1 \leq i, j \leq K \quad (\text{D.49})$$

则该马尔可夫链经过一定时间的状态转移后一定会收敛到分布 π .

细致平稳条件只是马尔可夫链收敛的充分条件, 不是必要条件. 细致平稳条件保证了从状态 i 转移到状态 j 的数量和从状态 j 转移到状态 i 的数量相一致, 互相抵消, 所以数量不发生改变.

马尔可夫链蒙特卡罗方法

► Markov Chain Monte Carlo, MCMC

- 核心思想是将采样过程看作是一个马尔可夫链。

$$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \dots$$

第 $t + 1$ 次采样依赖于第 t 次抽取的样本 $\mathbf{x}_t$ 以及状态转移分布（即提议分布） $q(\mathbf{x}|\mathbf{x}_t)$. 如果这个马尔可夫链的平稳分布为 $p(\mathbf{x})$, 那么在状态平稳时抽取的样本就服从 $p(\mathbf{x})$ 的分布.

► MCMC方法的关键是如何构造出平稳分布为 $p(x)$ 的马尔可夫链

- 并且其状态转移分布 $q(x|x_t)$ 一般为比较容易采样的分布。

Metropolis-Hastings 算法

在 MH 算法中,假设第 t 次采样的样本为 $\mathbf{x}_t$,首先根据提议分布 $q(\mathbf{x}|\mathbf{x}_t)$ 抽取一个样本 $\hat{\mathbf{x}}$,并以概率 $A(\hat{\mathbf{x}}, \mathbf{x}_t)$ 来接受 $\hat{\mathbf{x}}$ 作为第 $t + 1$ 次的采样样本 $\mathbf{x}_{t+1}$,

$$A(\hat{\mathbf{x}}, \mathbf{x}_t) = \min \left(1, \frac{p(\hat{\mathbf{x}})q(\mathbf{x}_t|\hat{\mathbf{x}})}{p(\mathbf{x}_t)q(\hat{\mathbf{x}}|\mathbf{x}_t)} \right). \quad (11.107)$$

在 MH 算法中,因为每次 $q(\mathbf{x}|\mathbf{x}_t)$ 随机生成一个样本 $\hat{\mathbf{x}}$,并以概率 $A(\hat{\mathbf{x}}, \mathbf{x}_t)$ 的方式接受,所以修正的马尔可夫链状态转移概率为

$$q'(\hat{\mathbf{x}}|\mathbf{x}_t) = q(\hat{\mathbf{x}}|\mathbf{x}_t)A(\hat{\mathbf{x}}, \mathbf{x}_t), \quad (11.108)$$

►该修正的马尔可夫链可以达到平稳状态,且平稳分布为 $q(\mathbf{x})$.

$$\begin{aligned} p(\mathbf{x}_t)q'(\hat{\mathbf{x}}|\mathbf{x}_t) &= p(\mathbf{x}_t)q(\hat{\mathbf{x}}|\mathbf{x}_t)A(\hat{\mathbf{x}}, \mathbf{x}_t) \\ &= p(\mathbf{x}_t)q(\hat{\mathbf{x}}|\mathbf{x}_t) \min \left(1, \frac{p(\hat{\mathbf{x}})q(\mathbf{x}_t|\hat{\mathbf{x}})}{p(\mathbf{x}_t)q(\hat{\mathbf{x}}|\mathbf{x}_t)} \right) \\ &= \min \left(p(\mathbf{x}_t)q(\hat{\mathbf{x}}|\mathbf{x}_t), p(\hat{\mathbf{x}})q(\mathbf{x}_t|\hat{\mathbf{x}}) \right) \\ &= p(\hat{\mathbf{x}})q(\mathbf{x}_t|\hat{\mathbf{x}}) \min \left(\frac{p(\mathbf{x}_t)q(\hat{\mathbf{x}}|\mathbf{x}_t)}{p(\hat{\mathbf{x}})q(\mathbf{x}_t|\hat{\mathbf{x}})}, 1 \right) \\ &= p(\hat{\mathbf{x}})q(\mathbf{x}_t|\hat{\mathbf{x}})A(\mathbf{x}_t, \hat{\mathbf{x}}) \\ &= p(\hat{\mathbf{x}})q'(\mathbf{x}_t|\hat{\mathbf{x}}). \end{aligned}$$

算法 11.3 Metropolis-Hastings 算法的采样过程

输入: 提议分布 $q(\mathbf{x}|\mathbf{x}')$, 采样间隔 M , 样本集合 $\mathcal{V} = \emptyset$;

1 随机初始化 $\mathbf{x}_0, t = 0$;

2 **repeat**

 // 预热过程

3 根据 $q(\mathbf{x}|\mathbf{x}_t)$ 随机生成一个样本 $\hat{\mathbf{x}}$;

4 计算接受概率 $A(\hat{\mathbf{x}}, \mathbf{x}_t)$;

5 从 $(0, 1)$ 的均匀分布中随机生成一个值 z ;

6 **if** $z \leq A(\hat{\mathbf{x}}, \mathbf{x}_t)$ **then** // 以 $A(\hat{\mathbf{x}}, \mathbf{x}_t)$ 的概率接受 $\hat{\mathbf{x}}$

7 $\mathbf{x}_{t+1} = \hat{\mathbf{x}}$;

8 **else** // 拒绝接受 $\hat{\mathbf{x}}$

9 $\mathbf{x}_{t+1} = \mathbf{x}_t$;

10 **end**

11 $t++$;

12 **if** 未到平稳状态 **then** continue;

13 **if** $t \bmod M = 0$ **then** // 采样过程, 每隔 M 次采一个样本

14 $\mathcal{V} = \mathcal{V} \cup \{\mathbf{x}_t\}$;

15 **end**

16 **until** 获得 N 个样本 ($|\mathcal{V}| = N$);

输出: 样本集合 $\mathcal{V}$

Metropolis 算法

如果 MH 算法中的提议分布是对称的,即 $q(\hat{\mathbf{x}}|\mathbf{x}_t) = q(\mathbf{x}_t|\hat{\mathbf{x}})$, 第 $t + 1$ 次采样的接受率可以简化为

$$A(\hat{\mathbf{x}}, \mathbf{x}_t) = \min\left(1, \frac{p(\hat{\mathbf{x}})}{p(\mathbf{x}_t)}\right). \quad (11.115)$$

这种 MCMC 方法称为 **Metropolis 算法**.

吉布斯采样

- ▶ 吉布斯采样（Gibbs Sampling）是一种有效地对高维空间中的分布进行采样的 MCMC 方法，可以看作是 Metropolis-Hastings 算法的特例。

- ▶ 全条件概率（Full Conditional

$$\begin{aligned} \text{Prob } p(x_m | \mathbf{x}_{\setminus m}) &\triangleq P(X_m = x_m | X_{\setminus m} = \mathbf{x}_{\setminus m}) \\ &= p(x_m | x_1, x_2, \dots, x_{m-1}, x_{m+1}, \dots, x_M), \end{aligned}$$

- ▶ 吉布斯采样使用全条件概率作为提议分布来依次对每个维度进行采样，并设置接受率为 $A = 1$ 。

- ▶ 吉布斯采样可以按照任意的顺序根据全条件分布依次对每个变量进行采样。假设从一个随机的初始化状态 $\mathbf{x}^{(0)}$ 开始，按照下标顺序依次对 M 个变量进行采样。

$$\begin{aligned} x_1^{(1)} &\sim p(x_1 | x_2^{(0)}, x_3^{(0)}, \dots, x_M^{(0)}), \\ x_2^{(1)} &\sim p(x_2 | x_1^{(1)}, x_3^{(0)}, \dots, x_M^{(0)}), \\ &\vdots \\ x_M^{(1)} &\sim p(x_M | x_1^{(1)}, x_2^{(1)}, \dots, x_{M-1}^{(1)}), \\ &\vdots \\ x_1^{(t)} &\sim p(x_1 | x_2^{(t-1)}, x_3^{(t-1)}, \dots, x_M^{(t-1)}), \\ x_2^{(t)} &\sim p(x_2 | x_1^{(t)}, x_3^{(t-1)}, \dots, x_M^{(t-1)}), \\ &\vdots \\ x_M^{(t)} &\sim p(x_M | x_1^{(t)}, x_2^{(t)}, \dots, x_{M-1}^{(t)}), \end{aligned}$$

使用 MCMC 方法的注意事项

- ▶ 马尔可夫链需要经过一段时间的随机游走才能达到平稳状态，这段时间称为预烧期（Burn-in Period）。
 - ▶ 预烧期内的采样点并不服从分布 $p(x)$ ，需要丢弃。
- ▶ 基于马尔可夫链抽取的相邻样本是高度相关的。而在机器学习中，我们一般需要抽取的样本是独立同分布的。
 - ▶ 为了使得抽取的样本之间独立，我们可以每间隔 M 次随机游走，抽取一个样本。如果 M 足够大，可以认为抽取的样本是独立的。



谢 谢

<https://nndl.github.io/>