

机器学习

第 8 章 概率图模型-Probabilistic Graphical Models

欧阳毅

浙江工商大学
管理工程与电子商务学院

2023

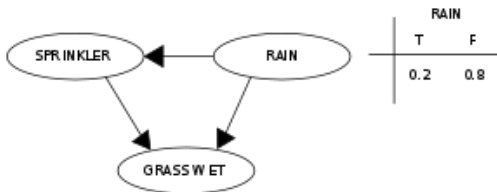
目录

- ① 概率图模型
 - 简单的贝叶斯网络模型推理
 - 隐马尔可夫模型
 - 马尔可夫随机场
 - 条件随机场
 - 蒙特卡罗 (Monte Carlo) 方法

简单的贝叶斯网络模型推理 I

假设有两个事件，可能导致草地是湿的：要么洒水或下雨。此外，假设在雨中使用的自动喷水灭火（即，下雨的时候，自动喷水灭火通常不开启）有直接的影响的情况，可以用贝叶斯网络模型预测草地是湿的原因。设这三个变量有两个可能的值，T/F

RAIN	SPRINKLER	
	T	F
F	0.4	0.6
T	0.01	0.99



	RAIN	
	T	F
	0.2	0.8

SPRINKLER	RAIN	GRASS WET	
		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

简单的贝叶斯网络模型推理 II

- 联合概率函数为：

$$P(G, S, R) = P(G|S, R)P(S|R)P(R) \quad (1)$$

- 草地是湿的概率，联合概率

令 $P(G_T) := P(G = T)$

$$P(G_T, S, R) = P(G_T|S_T, R_T)P(S_T|R_T)P(R_T) \quad (2)$$

$$+ P(G_T|S_T, R_F)P(S_T|R_F)P(R_F) \quad (3)$$

$$+ P(G_T|S_F, R_T)P(S_F|R_T)P(R_T) \quad (4)$$

$$+ P(G_T|S_F, R_F)P(S_F|R_F)P(R_F) \quad (5)$$

简单的贝叶斯网络模型推理 III

- 草地是湿的情况下，昨晚下雨的概率？ 令

$$P(X_T) := P(X = T), P(X_F) := P(X = F)$$

$$P(R_T|G_T) = \frac{P(G_T, R_T)}{P(G_T)} \quad (6)$$

$$= \frac{\sum_{S \in \{T, F\}} P(G_T, S, R_T)}{\sum_{S, R \in \{T, F\}} P(G_T, S, R)} \quad (7)$$

$$= \frac{0.00198_{TTT} + 0.1584_{TF T}}{0.00198_{TTT} + 0.288_{TTF} + 0.1584_{TF T} + 0.0_{TF F}} \quad (8)$$

$$= 0.3577 \quad (9)$$

$$0.00198_{TTT} = P(G_T|S_T, R_T)P(S_T|R_T)P(R_T) = 0.99 * 0.01 * 0.2$$

简单的贝叶斯网络模型推理 IV

- 草地是湿的情况下，昨晚自动喷水的概率？

$$P(S_T|G_T) = \frac{P(G_T, S_T)}{P(G_T)} \quad (10)$$

$$= \frac{\sum_{R \in \{T, F\}} P(G_T, S_T, R)}{\sum_{S, R \in \{T, F\}} P(G_T, S, R)} \quad (11)$$

$$= \frac{0.99*0.01*0.2_{TTT}+0.9*0.4*0.8_{TTF}}{0.99*0.01*0.2_{TTT}+0.9*0.4*0.8_{TTF}+0.8*0.99*0.2_{FTT}+0.0_{TFF}} \quad (12)$$

$$= 0.6467 \quad (13)$$

贝叶斯网络模型推理 I

福尔摩斯先生在办公室接到了他邻居华生的电话 $P(W=T)$ 。华生告诉他：他的家里可能进了窃贼 $P(B=T)$ ，因为他家的警铃响了 $P(A=T)$ 被告知有窃贼闯入，福尔摩斯迅速开车回家。

B — (盗贼)

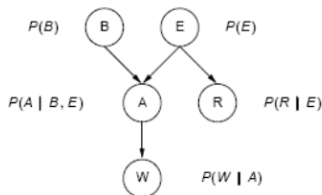
E — (地震)

R — (广播)

A — (警铃)

W — (华生致电福尔摩斯)

贝叶斯网络模型推理 II



W	A	
	n	y
n	0.99	0.35
y	0.01	0.65

R	E	
	n	y
n	0.999	0.01
y	0.001	0.99

$$P(E) = (0.99, 0.01), P(B) = (0.9, 0.1).$$

A	E = n		E = y	
	B = n	B = y	B = n	B = y
n	0.99	0.1	0.1	0.01
y	0.01	0.9	0.9	0.99

警铃响了 $P(A=T)$ 被告知有窃贼闯入的概率为：

$$P(B = T | A = T)$$

$$\text{令 } P(B_T | A_T) := P(B = T | A = T)$$

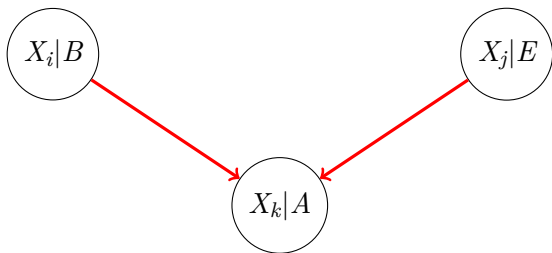
贝叶斯网络模型推理 III

根据收敛连接 (converging connection)

$$P(X_i, X_k, X_j) = P(X_i)P(X_j)P(X_k|X_i, X_j) \quad (14)$$

根据“条件局部独立性”可以得出这样一个结论：节点 i 和节点 j 是先验独立的

$$P(X_i, X_j) = P(X_i)P(X_j) \quad (15)$$



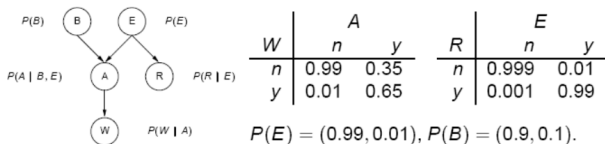
先来看看有警报情况下，发生盗贼入侵的概率？只有 3 个节点相关

贝叶斯网络模型推理 IV

$$\begin{aligned}P(B_T|A_T) &= \frac{P(A = T, B = T)}{P(A = T)} \\&= \frac{P(A_T|B_T, E_F)P(B_T)P(E_F) + P(A_T|B_T, E_T)P(B_T)P(E_T)}{\sum_{B, E \in \{T, F\}} P(A_T, B, E)} \\&= \frac{0.9 * 0.1 * 0.99 + 0.99 * 0.1 * 0.01}{0.99 * 0.1 * 0.01 + 0.9 * 0.1 * 0.99 + 0.9 * 0.9 * 0.01 + 0.01 * 0.99} \\&= 0.8412\end{aligned}$$

贝叶斯网络模型推理

华生来电话后 $P(W=T)$ 告知, 有窃贼闯入的概率为:



A	E = n		E = y	
	B = n	B = y	B = n	B = y
n	0.99	0.1	0.1	0.01
y	0.01	0.9	0.9	0.99

$$P(B = T | W = T) = \frac{P(B = T, W = T)}{P(W = T)} \quad (20)$$

$$= \frac{\sum_{A, E \in \{T, F\}} P(A, B = T, E, W = T)}{\sum_{A, B, E \in \{T, F\}} P(A, B, E, W = T)} \quad (21)$$

贝叶斯网络模型推理

$$P(B_T, W_T, A, E) = P(A_TB_TE_TW_T) + P(A_FB_TE_TW_T) \\ + P(A_TB_TE_FW_T) + P(A_FB_TE_FW_T)$$



$$P(A_TB_TE_TW_T)$$

如何确定?



$$P(A, B, E, W) = P(B) * P(E) * P(A|B, E) * P(W|A)$$

$$P(A_TB_TE_TW_T) = 0.1 * 0.01 * 0.99 * 0.65 = 0.0006435$$

$$P(A_FB_TE_TW_T) = 0.1 * 0.01 * 0.01 * 0.01 = 1.01e - 07$$

$$P(A_TB_TE_FW_T) = 0.1 * 0.99 * 0.9 * 0.65 = 0.05791500000000001$$

$$P(A_FB_TE_FW_T) = 0.1 * 0.99 * 0.1 * 0.01 = 9.9001e - 05$$

贝叶斯网络模型推理

$$P(B_T, W_T, A, E) = P(A_T B_T E_T W_T) + P(A_F B_T E_T W_T) \\ + P(A_T B_T E_F W_T) + P(A_F B_T E_F W_T)$$



$$P(A_T B_T E_T W_T)$$

如何确定?



$$P(A, B, E, W) = P(B) * P(E) * P(A|B, E) * P(W|A)$$

$$P(A_T B_T E_T W_T) = 0.1 * 0.01 * 0.99 * 0.65 = 0.0006435$$

$$P(A_F B_T E_T W_T) = 0.1 * 0.01 * 0.01 * 0.01 = 1.01e - 07$$

$$P(A_T B_T E_F W_T) = 0.1 * 0.99 * 0.9 * 0.65 = 0.057915000000000001$$

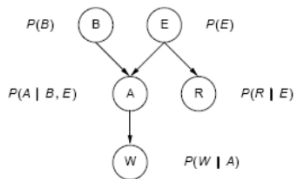
$$P(A_F B_T E_F W_T) = 0.1 * 0.99 * 0.1 * 0.01 = 9.9001e - 05$$

贝叶斯网络模型推理 I

$$\begin{aligned}P(B = T, W = T) &= P(A_T B_T E_T W_T) + P(A_F B_T E_T W_T) + P(A_T B_T E_F W_T) \\&\quad + P(A_F B_T E_F W_T) + P(A_T B_T E_T W_T) + P(A_F B_T E_T W_T) \\&\quad + P(A_T B_T E_F W_T) + P(A_F B_T E_F W_T) \\P(B = T | W = T) &= \frac{0.0587}{0.0785} = 0.7468\end{aligned}$$

贝叶斯网络模型推理 II

华生来电话后 $P(W=T)$ 告知，发生地震的概率为：



W	A	
	n	y
n	0.99	0.35
y	0.01	0.65

R	E	
	n	y
n	0.999	0.01
y	0.001	0.99

$$P(E) = (0.99, 0.01), P(B) = (0.9, 0.1).$$

A	$E = n$		$E = y$	
	$B = n$	$B = y$	$B = n$	$B = y$
n	0.99	0.1	0.1	0.01
y	0.01	0.9	0.9	0.99

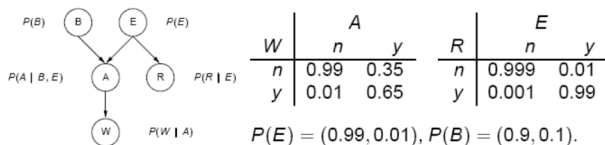
$$P(E = T | W = T) = \frac{P(E = T, W = T)}{P(W = T)} \quad (22)$$

$$= \frac{\sum_{A, B \in \{T, F\}} P(A, B, E = T, W = T)}{\sum_{A, B, E \in \{T, F\}} P(A, B, E, W = T)} \quad (23)$$

$$= 0.0753 \quad (24)$$

简单的贝叶斯网络模型推理 I

在路上，他听广播家那里发生了地震。请问他应该回家抓贼还是迅速撤离该地区以躲避地震？



A	E = n		E = y	
	B = n	B = y	B = n	B = y
n	0.99	0.1	0.1	0.01
y	0.01	0.9	0.9	0.99

$$P(E_T | W_T, R_T) = \frac{P(E = T, W = T, R = T)}{P(W = T, R = T)} \quad (25)$$

$$= \frac{\sum_{A, B \in \{T, F\}} P(A, B, E = T, W = T, R = T)}{\sum_{A, B, E \in \{T, F\}} P(A, B, E, W = T, R = T)} \quad (26)$$

$$= \frac{\sum_{A, B \in \{T, F\}} P(A, B, E = T, W = T, R = T)}{\sum_{A, B, E \in \{T, F\}} P(B) * P(E) * P(A|B, E) * P(R|E) * P(W|A)} \quad (27)$$

$$= 0.9878 \quad (28)$$

贝叶斯网络实现 I

```
from pgmpy.models import BayesianModel
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination
from pgmpy.models import BayesianModel
# pip install pgmpy
# 通过边来定义贝叶斯模型
model = BayesianModel([('B', 'A'), ('E', 'A'), ('A', 'W'),
                       ('E', 'R')])

# 定义条件概率分布
cpd_d = TabularCPD(variable='B', variable_card=2, values
                    =[[0.9, 0.1]])
cpd_i = TabularCPD(variable='E', variable_card=2, values
                    =[[0.99, 0.01]])

# variable: 变量
# variable_card: 基数
# values: 变量值
```

贝叶斯网络实现 II

```
# evidence:
cpd_g = TabularCPD(variable='A', variable_card=2,
                    values=[[0.99,0.1,0.1,0.01],
                             [0.01,0.9,0.9,0.99]],
                    evidence=['E', 'B'],
                    evidence_card=[2, 2])

cpd_l = TabularCPD(variable='W', variable_card=2,
                    values=[[0.99,0.35],
                             [0.01,0.65]],
                    evidence=['A'],
                    evidence_card=[2])

cpd_s = TabularCPD(variable='R', variable_card=2,
                    values=[[0.999, 0.01],
                             [0.001, 0.99]],
                    evidence=['E'],
                    evidence_card=[2])
```

贝叶斯网络实现 III

```
# 将有向无环图与条件概率分布表关联
model.add_cpds(cpd_d, cpd_i, cpd_g, cpd_l, cpd_s)

# 警铃响了 $P(A=T)$ 被告知有窃贼闯入的概率为：
#  $P(B=T|A=T)$ 
infer = VariableElimination(model)
print('P(B=T|A=T)')
print(infer.query(variables=['B'], evidence={'A': 1, 'W': 0}))
print(infer.query(variables=['B'], evidence={'A': 1}))

# 警铃响了 $P(A=T)$ 被告知有窃贼闯入的概率为：
#  $P(B=T|A=T) = \frac{P(A=T, B=T)}{P(A=T)} = 0.8412$ 

# 问题1： 我们想验证一下，计算过程，需要分别计算出分子和分母数值
```

贝叶斯网络实现 IV

请给出，分子和分母的具体数值（注意：利用`infer.query`，不需要手工计算）

```
PA=infer.query(variables=['A'])
PAB=infer.query(variables=['A','B'])
PA_V=PA.values
PAB_V=PAB.values
```

```
# ## 问题2：华生来电话后 $P(W=T)$ 告知，有窃贼闯入的概率为：
print(infer.query(variables=['E'], evidence={'W': 1}))
# ## 问题3：在路上，他听广播家那里发生了地震。请问他应该
      回家抓贼还是迅速撤离该地区以
      躲避地震

# $$P(E=T|W=T,R=T)$$
print(infer.query(variables=['E'], evidence={'W': 1, 'R': 1}))
```

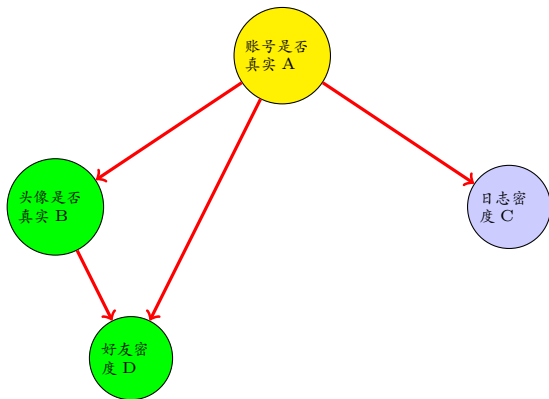
贝叶斯网络实现 Web 社区中不真实账号的检测 I

为了获取更准确的分类，可以将假设修改如下：

- i、真实账号比非真实账号平均具有更大的日志密度、各大的好友密度以及更多的使用真实头像。
- ii、日志密度与好友密度、日志密度与是否使用真实头像在账号真实性给定的条件下是独立的。
- iii、使用真实头像的用户比使用非真实头像的用户平均有更大的好友密度。

上述假设更接近实际情况，但问题随之也来了，由于特征属性间存在依赖关系，使得朴素贝叶斯分类不适用了。

贝叶斯网络实现 Web 社区中不真实账号的检测 II



$$P(A) \text{ --- } P(D|A, B)$$

		A=T	A=T	A=F	A=F
A=T	A=F	B=T	B=F	B=T	B=F
0.7	0.3	D=H	0.9	0.8	0.01
		D=L	0.1	0.2	0.99

贝叶斯网络实现 Web 社区中不真实账号的检测 III

第 1 种情况下:

$P(B|A) \text{ --- } P(C|A)$

	B=T	B=F		C=H	C=L
A=T	0.9	0.1	A=T	0.8	0.2
A=F	0.2	0.8	A=F	0.2	0.8

第 2 种情况下:

$P(B|A) \text{ --- } P(C|A)$

	B=T	B=F		C=H	C=L
A=T	0.9	0.1	A=T	0.9	0.1
A=F	0.2	0.8	A=F	0.3	0.7

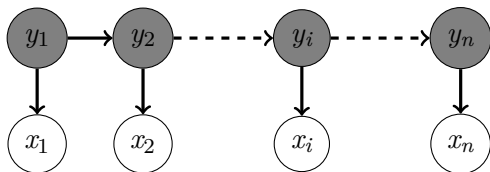
- 习题 2: 1. 贝叶斯网络实现 Web 社区中不真实账号检测的概率计算过程例如, 现随机抽取一个账户, 已知其头像为假, 但好友密度较高, 请判断其账号是真是假?
- 2. 进一步观察日志文件, 发现日志密度低, 请判断其账号是真是假?

目录

- ① 概率图模型
 - 简单的贝叶斯网络模型推理
 - 隐马尔可夫模型
 - 马尔可夫随机场
 - 条件随机场
 - 蒙特卡罗 (Monte Carlo) 方法

隐马尔可夫模型

- 使用无向图表示变量间的相互关系，称为无向图模型或马尔可夫网 (Markov Network)，属于生成式模型。
 - 隐马尔可夫模型 (Hidden Markov Model HMM): 结构最简单的动态贝叶斯网络。
 - 用于：时序数据建模，语音识别，自然语言处理。



- $y_i \in Y$ 表示第 i 时刻的系统状态（隐变量）
- $x_i \in X$ 表示第 i 时刻的观测值
- 箭头表示变量间的依赖关系。
- 马尔科夫链：系统下一时刻的状态仅由当前状态决定，不依赖于以前的任何状态

隐马尔可夫模型

马尔科夫链：所有变量的联合概率分布为

$$P(x_1, y_1, \dots, x_n, y_n) = P(y_1)P(x_1|y_1) \prod_{i=2}^n P(y_i|y_{i-1})P(x_i|y_i) \quad (29)$$

- 状态转移概率： $A = [a_{ij}]_{N \times N}$

$$a_{ij} = P(y_{t+1} = s_j | y_t = s_i), 1 \leq i, j \leq N$$

- 输出观察概率： $B = [b_{ij}]_{N \times M}$

$$b_{ij} = P(x_t = o_j | y_t = s_i), 1 \leq i \leq N, 1 \leq j \leq M$$

- 初始状态概率：模型在初始时刻各状态出现的概率，通常记为： $\pi = (\pi_1, \dots, \pi_N)$ ，其中

$$\pi_i = P(y_1 = s_i), 1 \leq i \leq N$$

- 隐马尔可夫模型参数： $\lambda = [A, B, \pi]$

隐马尔可夫模型-例子 I

我们考虑每年的气候状态，分为热年 (H)，冷年 (C)。

	H	C
H	0.7	0.3
C	0.4	0.6

假设当前研究表明，树的生长尺寸与天气温度相关。我们把树的年轮定义为小 (S)，中 (M)，大 (L) 三种不同大小。年度温度与树的年轮的概率关系为

	S	M	L
H	0.1	0.4	0.5
C	0.7	0.2	0.1

- 状态转移概率: $A = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$
- 输出观察概率: $B = \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{bmatrix}$
- 初始状态概率: $\pi = [0.6 \quad 0.4]$

隐马尔可夫模型-例子 II

- 隐马尔可夫模型参数: $\lambda = [A, B, \pi]$
- 现在假设我们观察到树的年轮变化为 S, M, S, L。令 0 表示 S, 1 表示 M, 2 表示 L

$$O = (O_0, O_1, O_2, O_3) = (0, 1, 0, 2)$$

- 我们希望根据树的年轮来确定最有可能的 Markov 过程的状态序列。也就是想知道这 4 年的天气冷热程度。

$$P(X, O) = \pi_{x_0} b_{x_0}(O_0) a_{x_0 x_1} b_{x_1}(O_1) a_{x_1 x_2} b_{x_2}(O_2) a_{x_2 x_3} b_{x_3}(O_3)$$

π_{x_0} 表示在起始状态 $x_0 \in [H, C]$ 时的概率; $b_{x_0}(O_0)$ 是观察到 O_0 的观察概率; $a_{x_0 x_1}$ 是从状态 x_0 转换到 x_1 的概率。

$$P(HHCC) = 0.6(0.1)(0.7)(0.4)(0.3)(0.7)(0.6)(0.1) = 0.00212$$

隐马尔可夫模型-例子 III

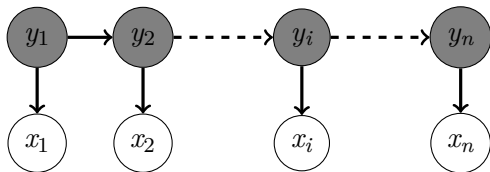
最高的概率序列为 *CCCH*

state	probability
HHHH	.000412
HHHC	.000035
HHCH	.000706
HHCC	.000212
HCHH	.000050
HCHC	.000004
HCCH	.000302
HCCC	.000091
CHHH	.001098
CHHC	.000094
CHCH	.001882
CHCC	.000564
CCHH	.000470
CCHC	.000040
CCCH	.002822
CCCC	.000847

隐马尔可夫模型

隐马尔可夫模型产生观测序列过程：

- ① 设置 $t = 1$, 并根据初始状态概率 π 选择初始状态 y_1 ;
- ② 根据状态 y_t 和输出观测概率 B 选择观测变量取值 x_t ;
- ③ 根据状态 y_t 和状态转移矩阵 A 转移模型状态, 即确定 y_{t+1} ;
- ④ 若 $t < n$, 设置 $t = t + 1$, 并转到第 (2) 步, 否则停止。



隐马尔可夫模型

隐马尔可夫模型的三个基本问题：

- ① Likelihood: 给定模型 $\lambda = [A, B, \pi]$, 如何有效计算产生观测序列?
- ② Decoding: 给定模型 $\lambda = [A, B, \pi]$, 和观测序列 $x = \{x_1, x_2, \dots, x_n\}$, 如何找到最匹配的状态序列 $y = \{y_1, y_2, \dots, y_n\}$?
- ③ Learning: 给的观测序列 $x = \{x_1, x_2, \dots, x_n\}$, 如何调整模型 $\lambda = [A, B, \pi]$?

基于公式

$$P(x_1, y_1, \dots, x_n, y_n) = P(y_1)P(x_1|y_1) \prod_{i=2}^n P(y_i|y_{i-1})P(x_i|y_i) \quad (30)$$

这三类问题都可求解

隐马尔可夫模型

Problem1:Computing Likelihood

马尔可夫假设:

$$P(y_i|y_1...y_{i-1}) = P(y_i|y_{i-1})$$

输出独立性:

$$P(O_i|y_1...y_T, O_1,..., O_T) = P(O_i|y_i)$$

Likelihood

给定 HMM 模型 $\lambda = [A, B, \pi]$ 和一个观察序列 O , 确定或然率 $P(O|\lambda)$

隐马尔可夫模型

Computing Likelihood

例子

若给定隐状态序列 (H,H,C)，我们可以计算观测 (L,M,L) 的 Likelihood,

$$P(O|Y) = \prod_{i=1}^T P(O_i|y_i)$$

若不知道隐状态序列

$$P(O) = \sum_Y P(O, Y) = \sum_Y P(O|Y)P(Y)$$

$$P(2, 1, 2) = P(2, 1, 2, CCC) + P(2, 1, 2, CCH) + P(2, 1, 2, CHH) + \dots$$

N 个状态 T 个观测序列计算复杂度为: $O(N^T)$

隐马尔可夫模型

Computing Likelihood

The forward procedure:

计算复杂度为: $O(N^2 T)$

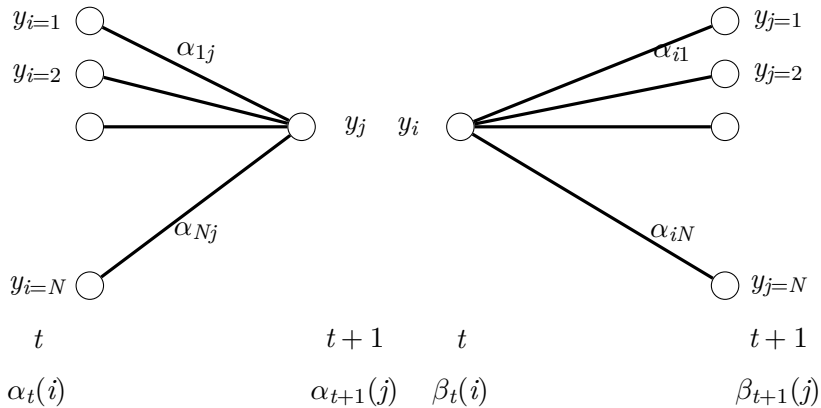
- ❶ $a_1(i) = \pi_i b_i(O_1), 1 \leq i \leq N$
- ❷ for $t = 1, 2, \dots, T - 1, 1 \leq i \leq N$

$$a_t(i) = \left[\sum_{j=1}^N a_{t-1}(j) \alpha_{ji} \right] b_i(O_t)$$

- ❸ $a_T = \sum_{i=1}^N a_{T-1}(i)$
- ❹ then , $P(O|\lambda) = \frac{\sum_{i=1}^N a_{T-1}(i)}{a_T}$

隐马尔可夫模型

forward



隐马尔可夫模型

Computing Likelihood

The backward procedure:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T | i_t = y_i, \lambda)$$

- ❶ $\beta_T(i) = 1, 1 \leq i \leq N$
- ❷ for $t=T-1, T-2, \dots, 1, 1 \leq i \leq N$

$$\beta_t(i) = \sum_{j=1}^N \alpha_{ij} b_j(O_{t+1}(j))$$

The forward-backward I

```
def fwd_bkw(Os, Y, Pi, A, emm_prob, end_st):  
    # Forward part of the algorithm  
    fwd = []  
    for i, O_i in enumerate(Os):  
        f_curr = {}  
        for st in Y:  
            if i == 0:  
                # base case for the forward part  
                prev_f_sum = Pi[st]  
            else:  
                prev_f_sum = sum(f_prev[k] * A[k][st]  
                                for k in Y)  
            f_curr[st] = emm_prob[st][O_i] * prev_f_sum  
  
        fwd.append(f_curr)  
        f_prev = f_curr  
    p_fwd = sum(f_curr[k] * A[k][end_st] for k in Y)
```

The forward-backward II

```
# Backward part of the algorithm
bkw = []
for i, 0_i in enumerate(reversed(0s[1:] + (None,))):
    b_curr = {}
    for st in states:
        if i == 0:
            b_curr[st] = A[st][end_st]
        else:
            b_curr[st] = sum(A[st][l] * emm_prob[l][
                                0_i] *
                                b_prev[l]
                            for l in Y)

    bkw.insert(0, b_curr)
    b_prev = b_curr
p_bkw = sum(Pi[l] * emm_prob[l][0s[0]] * b_curr[l]
            for l in Y)
```

The forward-backward III

```
# Merging the two parts
posterior = []
for i in range(len(Os)):
    posterior.append({st: fwd[i][st] * bkw[i][st] /
                      p_fwd for st in Y})

assert p_fwd == p_bkw
return fwd, bkw, posterior
```

隐马尔可夫模型 I

Problem2: Decoding : Viterbi

Decoding:

给定 $HMM\lambda = (A, B)$ 和观测序列 $O = \{O_1, \dots, O_T\}$, 找到最合适的隐状态序列 $Y = \{y_1, \dots, y_T\}$

① Step1: Initialization:

$$\delta_1(i) = \pi_i b_i(O_1)$$

$$\Psi_1(i) = 0$$

② Step2: Recursion

For $2 \leq t \leq T, 1 \leq j \leq N$

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) \alpha_{ij}] b_j(O_t)$$

$$\Psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) \alpha_{ij}]$$

隐马尔可夫模型 II

Problem2: Decoding : Viterbi

③ Termination

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$i_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

④ Path backtracking

$$\text{For } t = T - 1, T - 2, \dots, 1$$

$$i_t^* = \Psi_{t+1}(i_{t+1}^*)$$

隐马尔可夫模型 I

Problem3: Learning

Learning

给的观测序列 $x = \{x_1, x_2, \dots, x_n\}$, 如何调整模型 $\lambda = [A, B, \pi]$?

定义:

$$\begin{aligned}\gamma_t(i, j) &= P(y_t^i, y_{t+1}^j | O, \lambda) \\ &= \frac{a_t(i) \alpha_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)}\end{aligned}$$

$$\gamma_t(i) = \sum_{j=0}^{N-1} \gamma_t(i, j)$$

① For $i=0, 1, \dots, N-1$,

$$\pi_i = \gamma_0(i)$$

隐马尔可夫模型 II

Problem3: Learning

- ② For $i=0,1,\dots,N-1$ and $j=0,1,\dots,N-1$

$$\alpha_{ij} = \frac{\sum_{t=0}^{T-2} \gamma_t(i, j)}{\sum_{t=0}^{T-2} \gamma_t(i)}$$

- ③ For $j=0,1,\dots,N-1$ and $k=0,1,\dots,M-1$

$$b_j(k) = \frac{\sum_{t \in (0, \dots, T-1), O_t=k} \gamma_t(j)}{\sum_{t=0}^{T-1} \gamma_t(j)}$$

目录

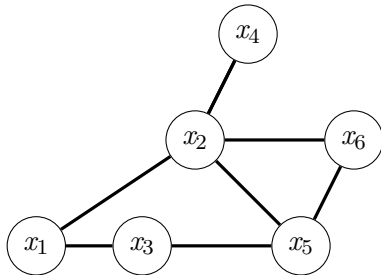
- ① 概率图模型
 - 简单的贝叶斯网络模型推理
 - 隐马尔可夫模型
 - 马尔可夫随机场
 - 条件随机场
 - 蒙特卡罗 (Monte Carlo) 方法

马尔可夫随机场

马尔可夫随机场 (Markov Random Field MRF): 是无向图模型, 结点表示随机变量, 连边表示变量之间的依赖关系。MRF 有一组势函数 (potential functions), 亦称因子 (factor), 定义在变量子集上的非负实函数, 主要用于定义概率分布函数。

团 : 团 (clique) 中任意两个结点都有连边。

极大团 : 加入另外任何一个结点都不再形成团。



马尔可夫随机场

马尔可夫随机场中，多个变量之间的联合概率分布能基于团分解为多个因子的乘积，每个因子仅与一个团相关。

$$P(x) = \frac{1}{Z} \prod_{Q \in C} \psi_Q(x_Q) \quad (31)$$

- ψ 为团 Q 对应的势能函数。
- $Z = \sum_x \prod_{Q \in C} \psi_Q(x_Q)$ 为规范化因子。

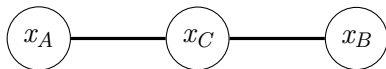
$$P(X) = \frac{1}{Z} \psi_{12}(x_1, x_2) \psi_{13}(x_1, x_3) \psi_{24}(x_2, x_4) \psi_{35}(x_3, x_5) \psi_{256}(x_2, x_5, x_6)$$

马尔可夫随机场

定义

全局马尔可夫性：给定两个变量子集的分离集，则这两个变量子集条件独立。

分离集：若从结点集 A 中的结点到 B 中的结点都必须经过结点集 C 中的结点，则称结点集 C 为“分离集”。



马尔可夫随机场

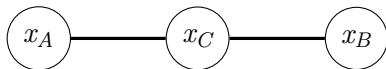
马尔可夫随机场中，多个变量之间的联合概率分布能基于团分解为多个因子的乘积，每个因子仅与一个团相关。

$$P(x) = \frac{1}{Z} \prod_{Q \in C} \psi_Q(x_Q) \quad (32)$$

- ψ 为团 Q 对应的势能函数。
- $Z = \sum_x \prod_{Q \in C} \psi_Q(x_Q)$ 为规范化因子。

$$P(X) = \frac{1}{Z} \psi_{12}(x_1, x_2) \psi_{13}(x_1, x_3) \psi_{24}(x_2, x_4) \psi_{35}(x_3, x_5) \psi_{256}(x_2, x_5, x_6)$$

马尔可夫随机场



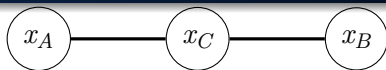
$$P(x_A, x_B, x_C) = \frac{1}{Z} \psi_{AC}(x_A, x_C) \psi_{BC}(x_B, x_C)$$

$$P(x_A, x_B | x_C) = \frac{P(x_A, x_B, x_C)}{P(x_C)} = \frac{P(x_A, x_B, x_C)}{\sum_{x'_A} \sum_{x'_B} P(x'_A, x'_B, x_C)} \quad (33)$$

$$= \frac{\frac{1}{Z} \psi_{AC}(x_A, x_C) \psi_{BC}(x_B, x_C)}{\sum_{x'_A} \sum_{x'_B} \frac{1}{Z} \psi_{AC}(x'_A, x_C) \psi_{BC}(x'_B, x_C)} \quad (34)$$

$$= \frac{\psi_{AC}(x_A, x_C)}{\sum_{x'_A} \psi_{AC}(x'_A, x_C)} * \frac{\psi_{BC}(x_B, x_C)}{\sum_{x'_B} \psi_{BC}(x'_B, x_C)} \quad (35)$$

马尔可夫随机场



$$P(x_A|x_C) = \frac{P(x_A, x_C)}{P(x_C)} = \frac{\sum_{x'_B} P(x_A, x'_B, x_C)}{\sum_{x'_A} \sum_{x'_B} P(x'_A, x'_B, x_C)} \quad (36)$$

$$= \frac{\sum_{x'_B} \frac{1}{Z} \psi_{AC}(x_A, x_C) \psi_{BC}(x'_B, x_C)}{\sum_{x'_A} \sum_{x'_B} \frac{1}{Z} \psi_{AC}(x'_A, x_C) \psi_{BC}(x'_B, x_C)} \quad (37)$$

$$= \frac{\psi_{AC}(x_A, x_C)}{\sum_{x'_A} \psi_{AC}(x'_A, x_C)} \quad (38)$$

同理可得：

$$P(x_B|x_C) \frac{\psi_{BC}(x_B, x_C)}{\sum_{x'_B} \psi_{BC}(x'_B, x_C)}$$

因此得到：

$$P(x_A, x_B|x_C) = P(x_A|x_C)P(x_B|x_C) \quad (39)$$

即 x_A 和 x_B 在给定 x_C 时条件独立。

马尔可夫随机场

由全局马尔可夫性可以得到两个推理：

- 局部马尔可夫性 (Local Markov property)：给定某变量的邻接变量，则该变量条件独立于其它变量。
- 成对马尔可夫性 (Pairwise Markov property)：给定所有其它变量，则两个非邻接变量条件独立。

目录

- ① 概率图模型
 - 简单的贝叶斯网络模型推理
 - 隐马尔可夫模型
 - 马尔可夫随机场
 - 条件随机场
 - 蒙特卡罗 (Monte Carlo) 方法

条件随机场

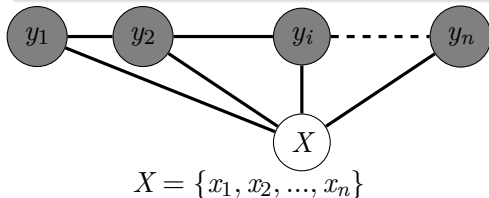
Conditional Random Field (CRF) 是一种判别式无向图模型。

条件随机场

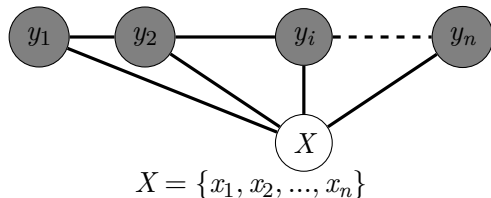
给定图 $G = (V, E)$ 表示结点与标记变量 y 中元素一一对应的无向图, y_v 表示与结点 v 对应的标记变量, $n(v)$ 表示结点 v 的邻接结点, 若图 G 的每个变量 y_v 都满足马尔可夫性, 即:

$$P(y_v | x, y_{V \setminus \{v\}}) = P(y_v | x, y_{n(v)}) \quad (40)$$

则 (y, x) 构成一个条件随机场



条件随机场



$$p(x|y) = \frac{1}{Z} \exp\left(\sum_j \sum_{i=1}^{n-1} \lambda_j t_j(y_{i+1}, y_i, x, i) + \sum_k \sum_{i=1}^n \mu_k s_k(y_i, x, i)\right)$$

- 通过选用指数势函数并引入特征函数
- t_i 是观测序列的两个相邻标记位置上的转移特征函数。
- s_k 是定义在观测序列的标记位置 i 上的状态特征函数。

目录

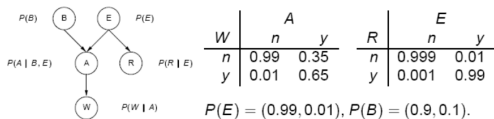
- ① 概率图模型
 - 简单的贝叶斯网络模型推理
 - 隐马尔可夫模型
 - 马尔可夫随机场
 - 条件随机场
 - 蒙特卡罗 (Monte Carlo) 方法

蒙特卡罗

- 从给定的分布中抽取样本
- 对于复杂分布产生一个随机表示。
- 如：使用基于样本的平均值去近似期望

$$E[f(x)] = \frac{1}{N} \sum_{t=1}^N f(x_t)$$

蒙特卡罗



A	E = n		E = y	
	B = n	B = y	B = n	B = y
n	0.99	0.1	0.1	0.01
y	0.01	0.9	0.9	0.99

E0	B0	A0	W0
E0	B0	A0	W0
E0	B0	A0	W1
E0	B0	A0	W0
E0	B0	A0	W0
E0	B0	A0	W0
E1	B0	A1	W1
E0	B0	A0	W0
E0	B0	A0	W0
E0	B0	A0	W0

• 计算频数

$$\begin{aligned}
 P(W|A0) &= \frac{P(W, A0)}{P(A0)} \\
 &= \langle 1/9, 8/9 \rangle
 \end{aligned}$$

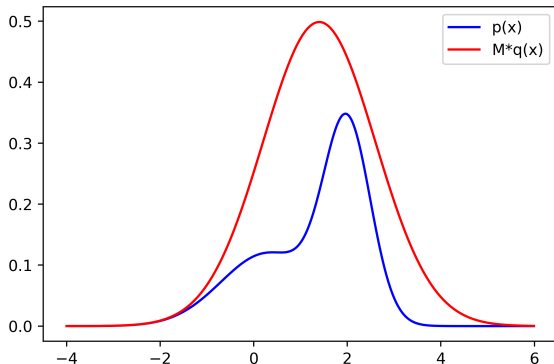
• 若我们想计算 $P(W|A1)$?

• 若我们想计算 $P(W|B1)$?

蒙特卡罗方法

假设我们希望从分布 $\pi(x)$ 中采样, $\pi(x)$ 是较难采样的, 但 $\hat{\pi}$ 是容易计算的

- 拒绝采样
- 重要性采样: Importance Sampling
- Metropolis-Hastings MH



拒绝采样: Rejection Sampling

假设已知一个容易采样的提议分布 $q(x)$ 和 $p(x)$ 之间满足:
 $Mq(x) \geq p(x)$ 。M 为某一个常数, 上式的意思即选择一个提议分布 $q(x)$ 和某一常数 M, 使得 $Mq(x)$ 包含 $p(x)$ 。

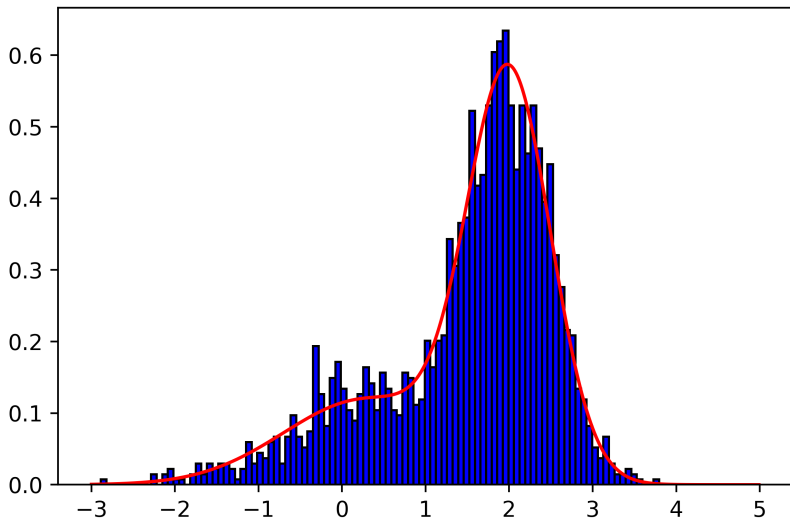
拒绝采样的步骤:

- 1. 确定需采样的样本个数 N;
- 2. 从提议分布 $q(x)$ 中采样样本 x_i ;
- 3. 从均匀分布 $[0, 1]$ 中采样样本 U ;
- 4. 计算 $\frac{p(x_i)}{Mq(x_i)}$, 若 $U < \frac{p(x_i)}{Mq(x_i)}$, 则接受当前采样样本 x_i , 否则, 拒绝该次采样;
- 5. 重复步骤 2-4, 直到采样样本数量等于 N。

拒绝采样: Rejection Sampling

```
def p(x):  
    gauss1=norm(loc=0.3,scale=1)  
    gauss2=norm(loc=2,scale=0.5)  
    y=0.3*gauss1.pdf(x)+0.7*gauss2.pdf(x)  
    return y  
qx=norm(loc=1.4,scale=1.2)  
M=2.5  
sample=[]  
for i in range(5000):  
    U=np.random.uniform(0, 1,1)  
    X=qx.rvs(1)[0]  
    if p(X)>=U*M*qx.pdf(X):  
        sample.append(X)
```

拒绝采样: Rejection Sampling



重要性采样: Importance Sampling

- 假设分布 $P(x)$ 中是较难采样的
- 假设我们可以从一个相对简单的提议分布 $Q(\cdot)$ 中采样.

$$\int h(x)p(x)dx = \int h(x)\frac{p(x)}{q(x)}q(x)dx = \int h(x)w(x)q(x)dx$$

- $p(x)$ 和 $q(x)$ 具有相同的样本空间
- $w(x)$ 被称为重要性函数
- 一个理想的重要性函数: 当被积分函数值变大它将变大,

重要性采样: Importance Sampling

例子: 使用 IS 计算积分的近似

考虑对一个函数的积分:

$$h(x) = 10 \exp(-2|x - 5|)$$

假设我们想要计算 $E[h(X)] = \int h(x)p(x)dx$, 其中 $X \sim \text{Uniform}(0, 10)$.

$$\int_0^{10} \exp(-2|x - 5|) dx$$

重要性采样: Importance Sampling

例子: 使用 IS 计算积分的近似

一种简单的方式是:

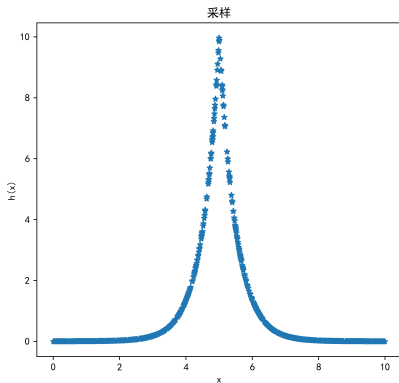
- 从 $\text{Uniform}(0,10)$ 分布中产生 x_i
- 计算 $h(x_i)$ 的样本均值, 均匀分布的 $p(x) = 1/10$

```
import numpy as np
X=np.random.uniform(0, 10,10000)
Y=10*np.exp(-2*np.abs(X-5))
print(np.mean(Y),np.var(Y))
```

$$\mu_x = \int_0^{10} xp(x)dx = \int_0^{10} \frac{x}{10} dx = 5$$

重要性采样: Importance Sampling

例子: 使用 IS 计算积分的近似



- $h(x)$ 的峰值在 $x=5$, 并下降较快, 在均匀分布中, 许多点对于期望的贡献是较小的
- 一些类高斯函数 (如: $ce^{-(x-5)^2}$), 在 $x=5$ 处也具有峰值

重要性采样: Importance Sampling

例子: 使用 IS 计算积分的近似

我们可以定义:

- $Q(x) = \frac{1}{\sqrt{2\pi}} e^{-(x-5)^2/2}$
- $p(x) = 1/10$

$$\int_0^{10} 10 * \exp(-2|x-5|) \frac{1/10}{\frac{1}{\sqrt{2\pi}} e^{-(x-5)^2/2}} \frac{1}{\sqrt{2\pi}} e^{-(x-5)^2/2} dx$$

- $w(x) = \frac{1}{10} * \sqrt{2\pi} e^{(x-5)^2/2}$

原式可变换为:

$$\int_0^{10} \exp(-2|x-5|) * \sqrt{2\pi} e^{(x-5)^2/2} * \frac{1}{\sqrt{2\pi}} e^{-(x-5)^2/2} dx$$

重要性采样: Importance Sampling

例子: 使用 IS 计算积分的近似

$$\int_0^{10} 10 * \exp(-2|x-5|) \frac{1/10}{\frac{1}{\sqrt{2\pi}} e^{-(x-5)^2/2}} \frac{1}{\sqrt{2\pi}} e^{-(x-5)^2/2} dx$$

计算 $E[h(X)]$, 其中 $X \sim N(5, 1)$.

$$w(x) = \frac{1/10}{\frac{1}{\sqrt{2\pi}} e^{-(x-5)^2/2}}$$

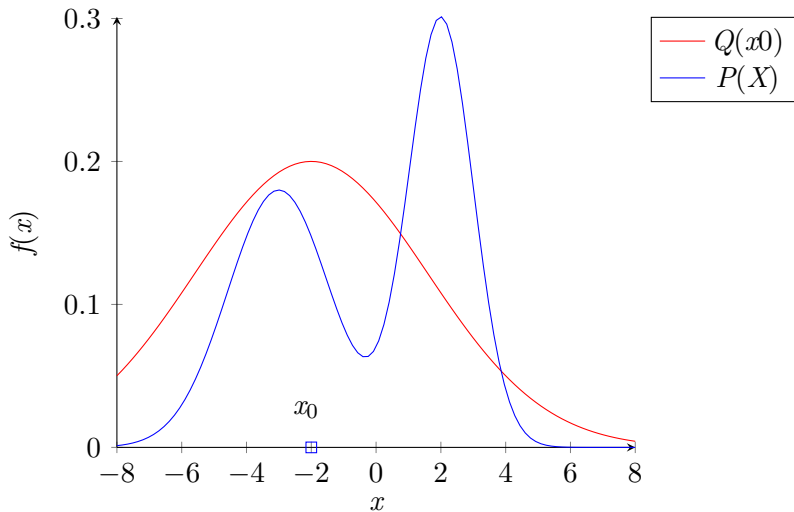
```
from scipy import stats
def weight(x):
    gauss=stats.norm(loc=5,scale=1)
    y1=gauss.pdf(x)
    uniform=stats.uniform(0,10)
    y2=uniform.pdf(x)
    return y2/y1
def hfun(x):
    return 10*np.exp(-2*np.abs(X-5))
X=np.random.normal(5,1,10000)
Y=weight(X)*hfun(X)
print(np.mean(Y), np.var(Y))
```

Metropolis-Hastings

- 初始化状态 $x^{(0)}$, 设置 $t = 0$
- Burn-in: 当采样还未收敛
 - $x = x^{(t)}$
 - $t = t + 1$
 - $x^* \sim Q(x^* | x)$
 - $u \sim \text{Uniform}(0, 1)$
 - if $u < A(x^* | x) = \min(1, \frac{P(x^*)Q(x|x^*)}{P(x)Q(x^*|x)})$
 - $x^{(t)} = x^*$ - 状态转移
 - else
 - $x^{(t)} = x$ - 保持
- 从 $P(x)$ 中采样: 重置 $t = 0$

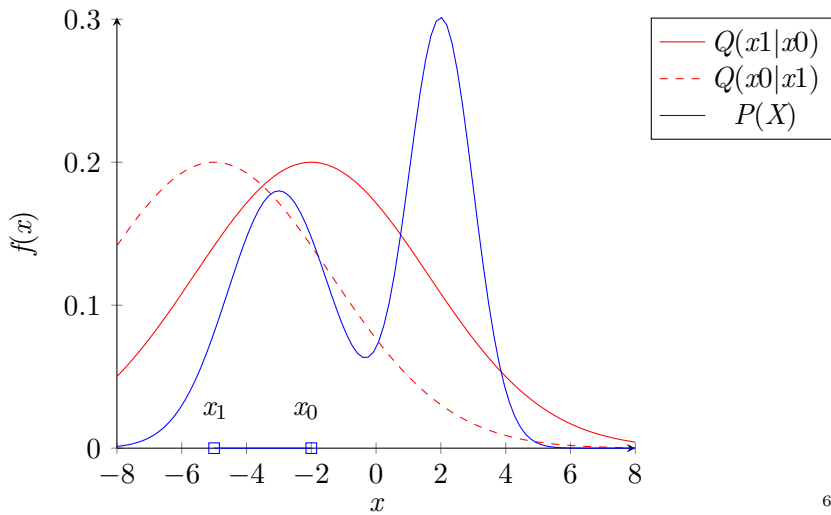
The MH Algorithm

- Let $Q(x'|x)$ be a Gaussian centered on x
- 试图从 $P(x)$ 分布中采样
- Draw ,accept x_0



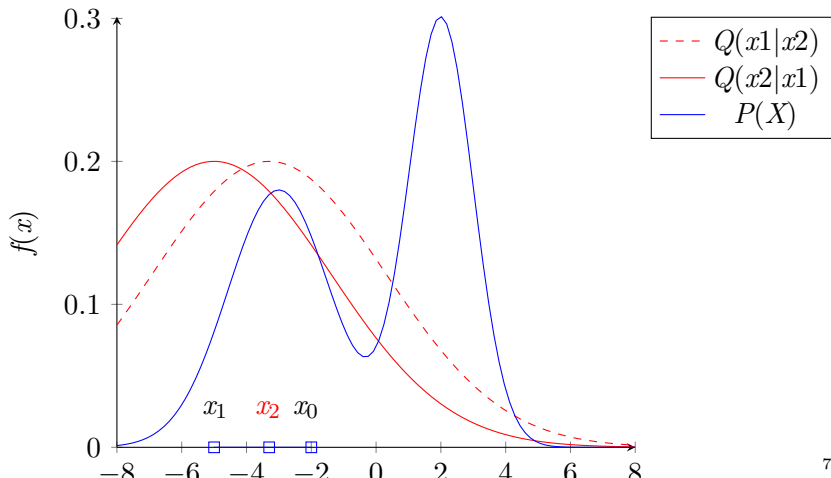
The MH Algorithm

- Let $Q(x'|x)$ be a Gaussian centered on x
- $A(x^*|x) = \min(1, \frac{P(x^*)Q(x|x^*)}{P(x)Q(x^*|x)})$
- Accept $x_1, P(x_1)/Q(x_1|x_0) < 1$ and $P(x_0)/Q(x_0|x_1) \approx 1$



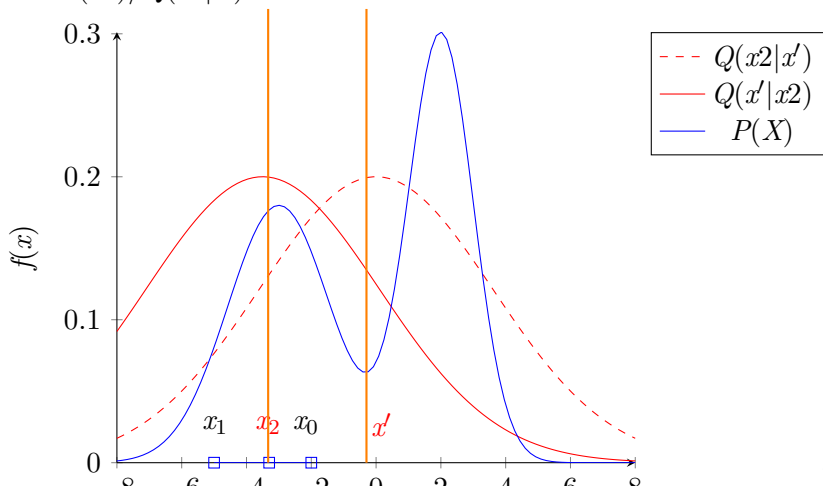
The MH Algorithm

- Let $Q(x'|x)$ be a Gaussian centered on x
- $A(x^*|x) = \min(1, \frac{P(x^*)Q(x|x^*)}{P(x)Q(x^*|x)})$
- Draw x_2 , accept x_2 if $P(x_2)/Q(x_2|x_1) \approx 1$ and $P(x_1)/Q(x_1|x_2) < 1$



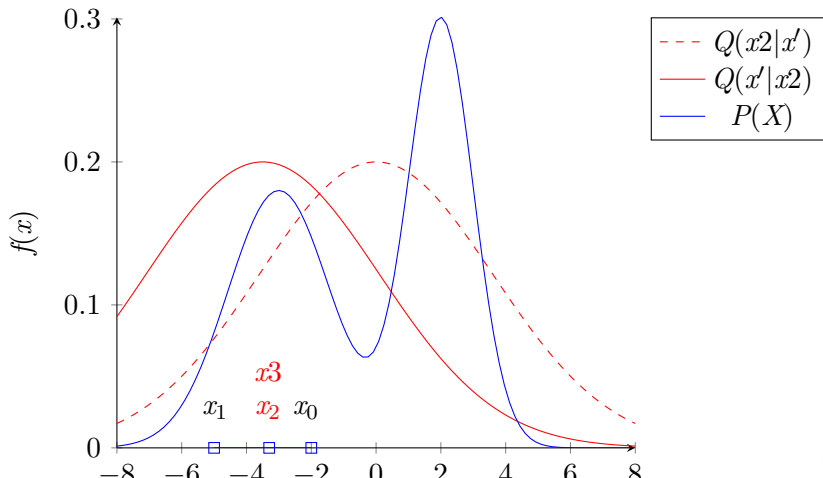
The MH Algorithm

- Let $Q(x'|x)$ be a Gaussian centered on x
- $A(x^*|x) = \min(1, \frac{P(x^*)Q(x|x^*)}{P(x)Q(x^*|x)})$
- Draw ,reject x' ; set $x_3=x_2$, 因为 $P(x')/Q(x'|x_2) < 1$ and $P(x_2)/Q(x_2|x') > 1$



The MH Algorithm

- Let $Q(x'|x)$ be a Gaussian centered on x
- $A(x^*|x) = \min(1, \frac{P(x^*)Q(x|x^*)}{P(x)Q(x^*|x)})$
- Draw x' , reject x' ; set $x_3=x_2$, 因为 $P(x')/Q(x'|x_2) < 1$ and $P(x_2)/Q(x_2|x') > 1$



Gibbs Sampling

已知：随机变量 X 服从的分布 $p(X|\Theta)$ ，样本 $\{x_1, x_2, \dots, x_N\}$ ，分布参数 Θ 的先验分布 $p(\theta_1, \theta_2)$ 。

执行下列步骤：

- 1. 初始化，随机得到第一个样本 $\Theta^1 = (\theta_1^1, \theta_2^1)$ ；
- 2. 利用 θ_2^i 从 $p(\theta_1|\theta_2 = \theta_2^i, x_1, \dots, x_n)$ 中抽取 θ_1^{i+1} ；
- 3. 利用 θ_1^{i+1} ，从 $p(\theta_2|\theta_1 = \theta_1^{i+1}, x_1, \dots, x_n)$ 中抽取 θ_2^{i+1} ；构成 Θ^{i+1} 。
- 4. 重复步骤 2 与 3，直到抽取的 Θ 样本个数满足需求。

在抽样过程中，每一个样本 Θ^i 总会受到前一个样本 Θ^{i-1} 的影响。因此，通过吉布斯抽样得到的样本序列 $\Theta^1, \dots, \Theta^n$ 是一条马尔可夫链。该马尔可夫链的平稳分布是目标联合概率分布 $p(\theta_1, \theta_2)$ 。

$$\pi(i)P(i, j) = \pi(j)P(j, i)$$

Gibbs Sampling I

```
from pgmpy.sampling import BayesianModelSampling,
                        GibbsSampling
from pgmpy.models import BayesianModel, MarkovModel
markov_model = MarkovModel([("A", "B"), ("C", "B"), ("B",
                                                    "D")])
factor_cb = DiscreteFactor(
    ["C", "B"], [4, 3], [3, 1, 4, 5, 7, 8, 1, 3, 10, 4,
                        5, 6]
)
factor_bd = DiscreteFactor(["B", "D"], [3, 2], [5, 7, 2,
                                                    1, 9, 3])
markov_model.add_factors(factor_ab, factor_cb, factor_bd
                        )

gibbs = GibbsSampling(markov_model)
sample = gibbs.sample(size=20)
print(sample)
```