1. Probabilistic Reasoning

$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$；$P(A|B) = \frac{P(A,B)}{P(B)}$

P(A): 先验概率 (Prior Probability)；P(A|B): 后验概率 (Posterior Probability)；P(B|A): 似然 (Likelihood)；P(B): 证据的边缘概率 (Marginal Probability)。

$P(X) = \sum_Z P(X,Z)$ 边缘化 (Marginalization): Z 未观测的随机变量。

$P(Y|E) = \alpha \cdot \sum_Z P(Y,E,Z)$ E 已观测(证据); Z 隐藏变量; α 归一化

$P(X,Y|Z) = P(X|Z) \cdot P(Y|Z)$ 条件独立性，BN 中用 d-separation 判断

独立的符号： $A \perp B$ 表示 A 和 B 独立，即 P(A,B)=P(A)P(B)。$A \perp B|C$ 表示 A 和 B 在给定 C 的条件下是独立的，即：P(A,B｜C)=P(A｜C)P(B｜C)。或者说，给定 C 后，A 的状态不会影响 B 的概率分布，反之亦然。

前向推理（Forward Inference）：已知因，推导果（如 P(Y｜X)）；**后向推理（Backward Inference）**：已知果，推导因（如 P(X｜Y)）。

BN 三大经典结构：因果链、共同原因、共同效果

• **因果链**：X→Z→Y（X 是 Z 的原因，而 Z 是 Y 的原因）联合概率分布：$P(X,Y,Z) = P(X)P(Z|X)P(Y|Z)$ **未知 Z** 时，X 和 Y 是**依赖的**；知 Z 时，X 与 Y 条件独立，Z 屏蔽了 X 和 Y 的依赖。$P(Y|X,Z) = P(Y|Z)$

• **共同原因**：Z→X, Z→Y（Z 是 X 和 Y 的共同原因^）联合概率分布：$P(X,Y,Z) = P(Z)P(X|Z)P(Y|Z)$ **未知 Z** 时，X 和 Y 是**依赖的**，Z 引入了关联；已知 Z 时，X 和 Y 条件独立：$P(X,Y|Z) = P(X|Z)P(Y|Z)$

• **共同效果**：X→Z, Y→Z（X 和 Y 是 Z 的原因 V）联合概率分布：$P(X,Y,Z) = P(X)P(Y)P(Z|X,Y)$ 未知 Z 时，X 和 Y 独立；已知 Z 时，X 和 Y 依赖：$P(X,Y|Z) \neq P(X|Z)P(Y|Z)$

马尔可夫毯组成部分：**父节点**（Parents）直接指向该变量的节点；**子节点**（Children）由该变量直接指向的节点；**子节点的父节点**（Children's Parents）这些节点会影响子节点，但不直接影响该变量本身。如果已知马尔可夫毯内的所有节点，随机变量与网络中其他所有变量条件独立。markov 毯记为 M(X) $P(X|\text{所有其他变量}) = P(X|M(X))$ ——只需知道 X 的马尔可夫毯，便可以推断 X 的概率分布，忽略网络中其他变量。

2. Probabilistic Reasoning over Time

矩阵乘法 $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$

c11=a11 · b11+a12 · b21；c12=a11 · b12+a12 · b22；c21=a21 · b11+a22 · b21；c22=a21 · b12+a22 · b22

马尔科夫链：当前的状态只取决于前一个时间点，概率转移矩阵 *T* 不变，P(Xt)=T · P(Xt−1)平稳性。转移矩阵每行元素之和为 1（表示概率归一化）。

隐马尔可夫模型状态变量：Xt 描述不可观测的隐藏态；**观测变量** Et 描述隐藏状态观测值；**初始概率** P(X1)；**状态转移概率** P(Xt｜Xt−1)；**观测概率（发射概率）**P(Et｜Xt)

$$P(X_1, \ldots, X_t, E_1, \ldots, E_t) = P(X_1) \prod_{i=2}^{t} P(X_i|X_{i-1}) \prod_{i=1}^{t} P(E_i|X_i)$$

推断：预测(Prediction)，滤波(Filtering)，平滑(Smoothing).

• **预测**：计算未来状态的概率分布 P(Xt+k ｜ E1:t)

$P(X_{t+1}|E_{1:t}) = T \cdot P(X_t|E_{1:t})$ $P(X_{t+k}|E_{1:t}) = T^k \cdot P(X_t|E_{1:t})$ 多步预测

• **滤波**：当前的 P(Xt｜E1:t) $P(X_t|E_{1:t}) = \alpha \cdot P(E_t|X_t) \cdot \sum_{X_{t-1}} P(X_t|X_{t-1}) P(X_{t-1}|E_{1:t-1})$

$P(X_t|E_{1:t}) = \alpha \cdot P(E_t|X_t) \cdot \sum_{X_{t-1}} P(X_t|X_{t-1}) P(X_{t-1}|E_{1:t-1})$ 用新信息(α 后)修正原预测

• **平滑**：过去 k 的 P(Xk｜E1:t)，k<t $P(X_k|E_{1:t}) = P(X_k|E_{1:k}) \cdot \beta(X_k)$

稳定分布 stationary：求解方程组 $\pi T = \pi$ 乘转移矩阵后概率仍不变

即稳定 $\begin{bmatrix} \pi_{\text{high}} & \pi_{\text{low}} \end{bmatrix} \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} = \begin{bmatrix} \pi_{\text{high}} & \pi_{\text{low}} \end{bmatrix}$ 叠加归一化 $\pi_{\text{high}} + \pi_{\text{low}} = 1$ 解方程

1. $\pi_{\text{high}} = 0.9\pi_{\text{high}} + 0.3\pi_{\text{low}}$
2. $\pi_{\text{low}} = 0.1\pi_{\text{high}} + 0.7\pi_{\text{low}}$

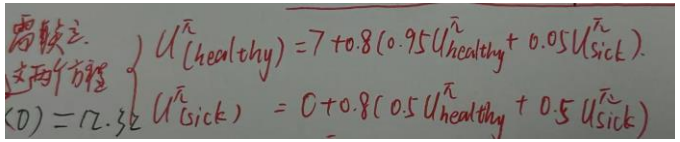
，解得 π=[0.75,0.25]

3. Markov Decision Process(MDP)

MDP 的基本组成：状态集合 (S)：描述系统的所有可能状态；**动作集合 (A)**：每个状态可采取的动作集合；**状态转移函数 P(s′｜s,a)**：描述在状态 s 下执行动作 a 转移到状态 s′ 的概率；**奖励函数 R(s,a,s′)**：描述从状态 s 经动作 a 转移到状态 s′ 所获得的即时奖励（不是效用 utility）；**折扣因子 (γ)**：介于 0 和 1 之间，用于权衡立即奖励与未来奖励的重要性。

预测问题：策略已知，算每个状态的效用 Utility➔联立求解线性方程组。方程：U(S) = SA 对应的 reward + γ *未来的 Utility（各自 S 的 Utility × 概率 - 即期望）

$$U^\pi(s) = \sum_{s'} P(s'|s,\pi(s)) \left[R(s,\pi(s),s') + \gamma U^\pi(s') \right]$$



控制问题：策略未知，找效用最大化策略，用 max 非线性优化，多次迭代直至收敛

$$\pi'(s) = \arg\max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma U^{\pi}(s')]$$

总策略的数量等于所有状态的动作选择数的乘积。策略数量=(S0 的选择数)×(S1 的选择数)×(S3 的选择数)×(S2 的选择数)

4. Machine Learning: Introduction

• **监督学习**：**回归**：预测连续值（如房价、温度）；**分类**：预测离散类别（如垃圾邮件检测）。• **无监督学习**：**聚类**：发现数据中的隐藏分组（如客户分群）；**降维**：简化数据维度（如 PCA）。• **强化学习**：基于奖励学习策略（如游戏 AI）

线性模型：**线性回归**：用于预测连续输出；**逻辑回归**：用于二分类问题；**目标**：找到特征和输出之间的最佳线性关系。

多分类问题：将分类扩展到多个类别，使用 **Softmax** 函数将模型输出转化为概率分布，常用于图像分类和文本分类。

模型优化：梯度下降：通过反复调整参数找到损失函数的最小值。正则化：防止过拟合，往损失函数中增加惩罚项，L1(Lasso Regularization 惩罚参数绝对值)/L2(Ridge Regularization 惩罚参数平方和)正则化常用。**Dropout** 训练时随机丢弃一部分神经元，防止过拟合。

模型评估：• **训练集、验证集、测试集**：用于训练和评估模型性能。• **偏差与方差**：**偏差（Bias）**：欠拟合，模型过于简单；**方差（Variance）**：过拟合，模型过于复杂。

5. Neural Networks

感知机：一种单层神经网络，用于二分类问题。使用符号函数 g(z)激活，输+1：如果 z≥0；−1：如果 z<0。对每个输入 xi，计算输出 y_hat = g(wT*xi+b)。

$$z = \sum w_i x_i + b = w^T x + b$$; $$w \leftarrow w + \eta \cdot (y - \hat{y}) \cdot x$$

多层感知机（MLP）:输入层、一个或多个隐藏层、输出层；每一层的输出通过权重和激活函数传递到下一层；可以处理非线性问题（区别于单层感知机）。

MLP 能够近似任意函数，通过多层结构实现非线性映射；使用激活函数引入非线性，使模型能够处理复杂问题。**激活函数**

激活函数引入非线性，常见激活函数：**ReLU** 处理梯度消失问题，适用于隐藏层。**Leaky ReLU** 改进 ReLU，允许小幅负梯度，解决死区问题。**Sigmoid** 将输出限制在(0,1)，适用于二分类。**Tanh** 输出范围(−1,1)，适合归一化数据。**Softmax**：将多分类问题的输出转为概率分布。

损失函数用于衡量模型预测值和实际值之间的差异，优化时会最小化损失。回归任务：均方误差（MSE）；平均绝对误差（MAE）。分类任务：二元交叉熵（Binary Cross-Entropy）：用于二分类；多类交叉熵（Multiclass Cross-Entropy）：用于多分类；Hinge Loss：用于支持向量机（SVM）分类。

优化算法通过最小化损失函数更新模型参数。**SGD**：简单但可能收敛慢。**Momentum**：加速收敛，减小震荡。**AdaGrad**：自适应学习率，但会快速衰减。**RMSProp**：改进 AdaGrad，减少学习率衰减。**Adam**：结合 Momentum 和 RMSProp，适用广泛。

单层的感知机，**前向传播**算权重 w $a = g(f(x)), f(x) = w \cdot x$ Single-Layer Perceptron

$$\underbrace{a^{[l]}}_{\text{Layer } l \text{ Activations}} = \overbrace{g^{[l]}}^{\text{Layer } l \text{ Activation Function}}\left(\left(\overbrace{W^{[l]}}^{\text{Layer } l \text{ Weights}}\right)^{\top} \underbrace{a^{[l-1]}}_{\text{Layer } l-1 \text{ Activations}}\right)$$

Layer l in Neural Network

神经网路中的一层**反向传播（Backpropagation）**根据损失函数计算每层的梯度，更新权重

$$\hat{y} = a^{[3]} = g^{[3]}\left(f^{[3]}\left(g^{[2]}\left(f^{[2]}\left(g^{[1]}\left(f^{[1]}\left(a^{[0]}\right)\right)\right)\right)\right)\right) = g_1^{[3]}\left(f_1^{[3]}\left(g^{[2]}\right)\right)$$

$$W^{[l]} := W^{[l]} - \eta \cdot \frac{\partial \mathcal{L}}{\partial W^{[l]}}$$

$$b^{[l]} := b^{[l]} - \eta \cdot \frac{\partial \mathcal{L}}{\partial b^{[l]}}$$

矩阵链式法则：$$\underbrace{\frac{\partial g(f)}{\partial x}}_{n\times m} = \underbrace{\frac{\partial f}{\partial x}}_{n\times d}\underbrace{\frac{\partial g(f)}{\partial f}}_{d\times m}$$

梯度消失问题：当网络层数较深时，梯度在反向传播中逐渐减小，导致前层权重更新缓慢。**解决方案**：使用 ReLU 激活函数代替 Sigmoid 或 Tanh。正则化或优化初始化权重。

• $g'(f) = \frac{e^{-f}}{(1+e^{-f})^2} = \frac{1}{1+e^{-f}}\left(1 - \frac{1}{1+e^{-f}}\right) = g(f)[1 - g(f)]$ Sigma