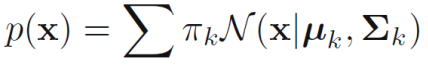
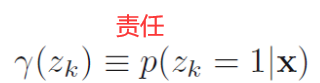
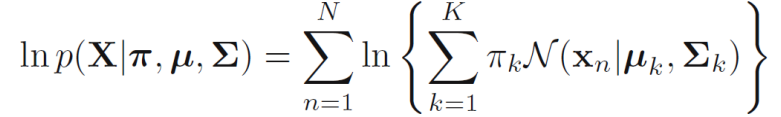
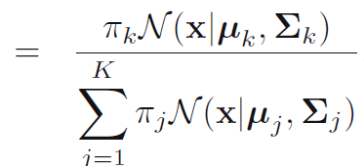
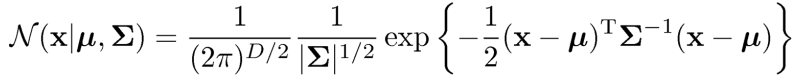
GMM:



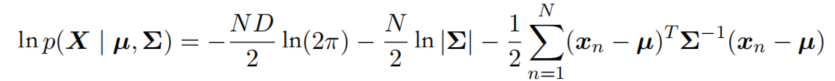


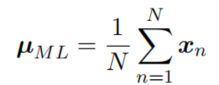
 

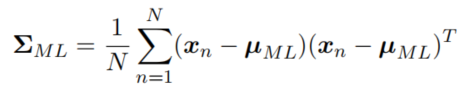
高斯分布：

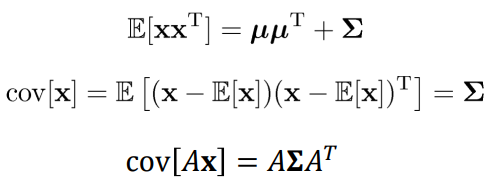


最大似然估计



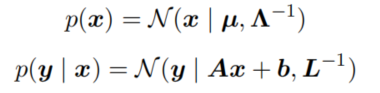




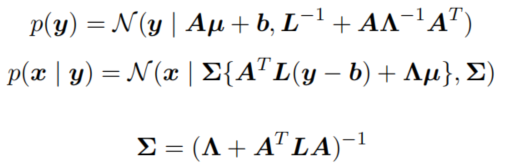


⾼斯变量的贝叶斯定理

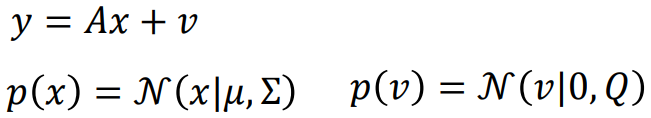








给定

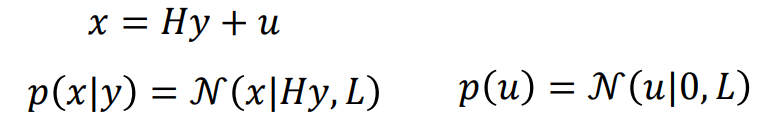


有

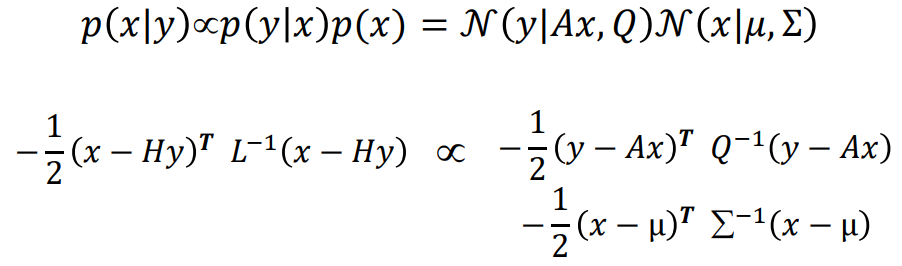


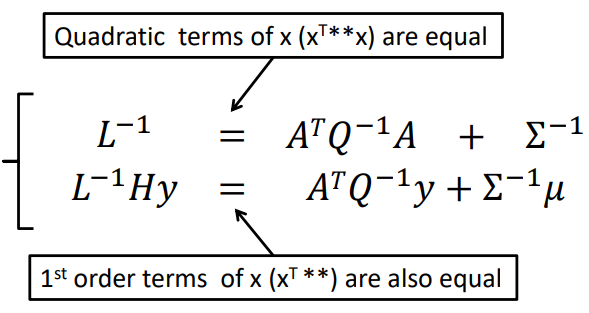


给定

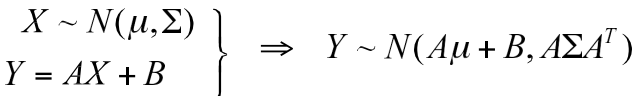


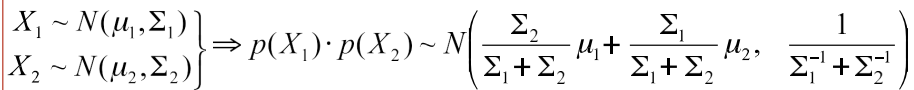
有



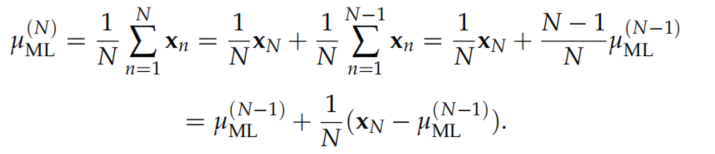


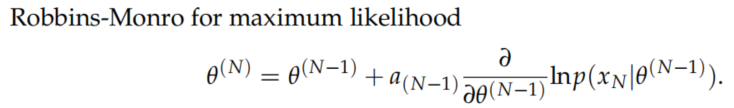
Multivariate Gaussians:

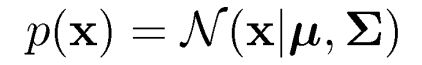


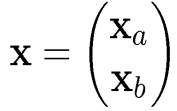
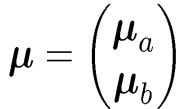
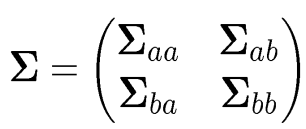


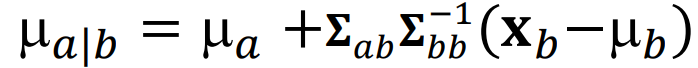
顺序估计（Robbins-Monro）:

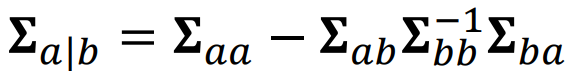




分块高斯分布：





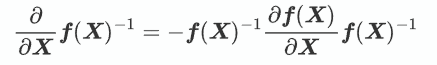
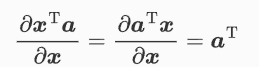
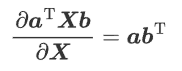
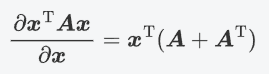
平方误差函数：



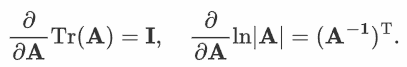


加上正则化，就在逆中填加

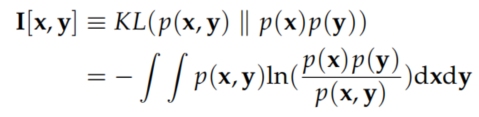
矩阵求导：



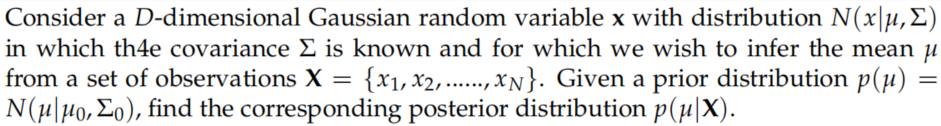


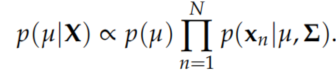
互信息：

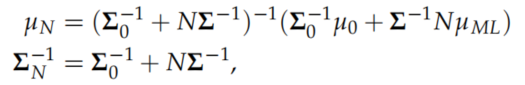




先验后验1：





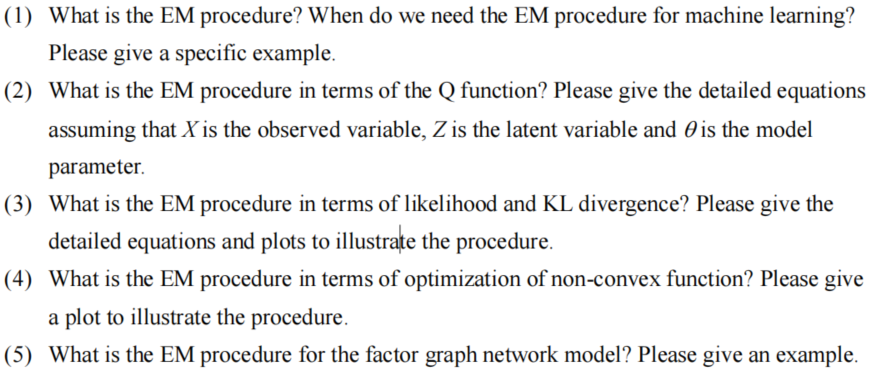


EM相关概念：

1. 2. E: 计算

3. M: 计算

4. 收敛或者



（1）EM算法用于含有隐变量（latent variables）的统计模型的最大似然估计。当模型的似然函数因为存在隐变量而难以直接最大化时，EM算法提供了一种有效的迭代方法来估计模型参数。

EM算法分为两个步骤：期望步骤（E步）和最大化步骤（M步）。

E: 估计隐变量的分布，即计算隐变量的期望。

M: 最大化E步中得到的期望，以更新模型参数。

例如，在混合高斯模型中，观测数据是由多个高斯分布混合生成的，每个高斯分布对应一个隐变量，代表了数据点属于该分布的概率。EM算法可以用来估计每个高斯分布的均值和协方差以及混合比例。

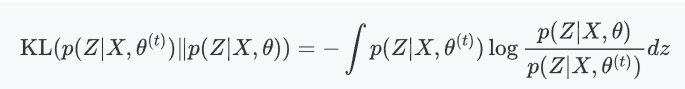
(2)Q函数是在E步骤中计算的，代表了给定当前参数估计下隐变量分布的对数似然的期望。Q函数的定义为：



其中是当前迭代下的参数估计，而是新的参数值。E步涉及计算，M步涉及找到使得

最大化的。

(3) EM算法可以视为最大化似然和最小化KL散度的过程。在EM算法的上下文中，KL散度衡量了当前参数下隐变量真实分布和估计分布之间的差异。

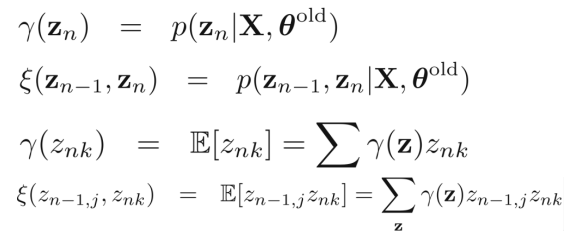


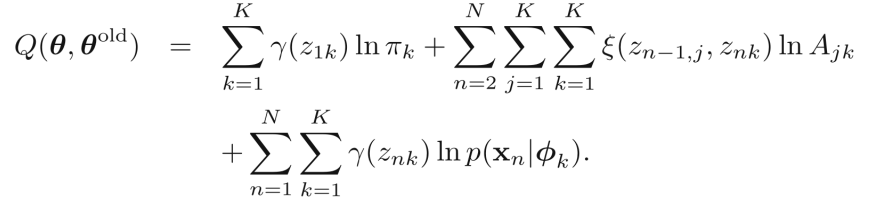
在E步，我们寻找一个新的隐变量分布来减少这个KL散度，而在M步，我们更新参数 来最大化似然。

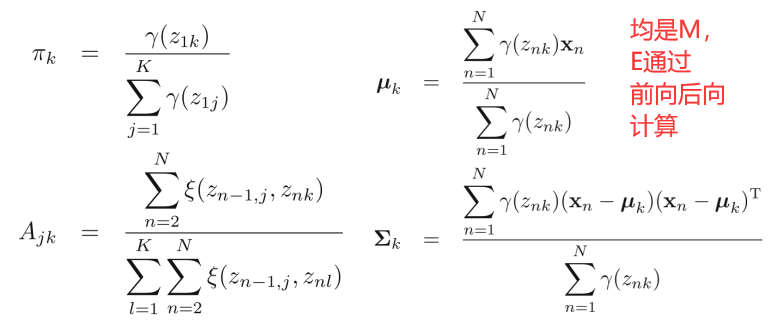
(4) EM算法是非凸优化问题中的一种方法，可以找到局部最优解。在每个M步，我们找到使得似然函数局部最大化的参数值。这个过程可以用非凸函数的优化曲线来可视化，其中EM迭代在概率分布空间中“攀爬”以寻找一个局部最高点。

HMM:

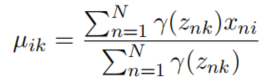
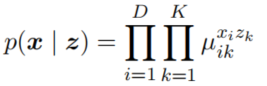
EM For HMM



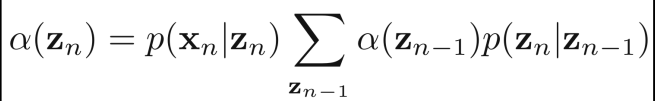


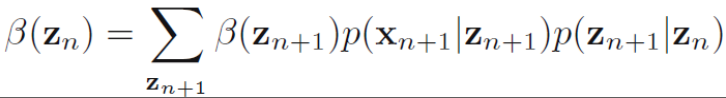


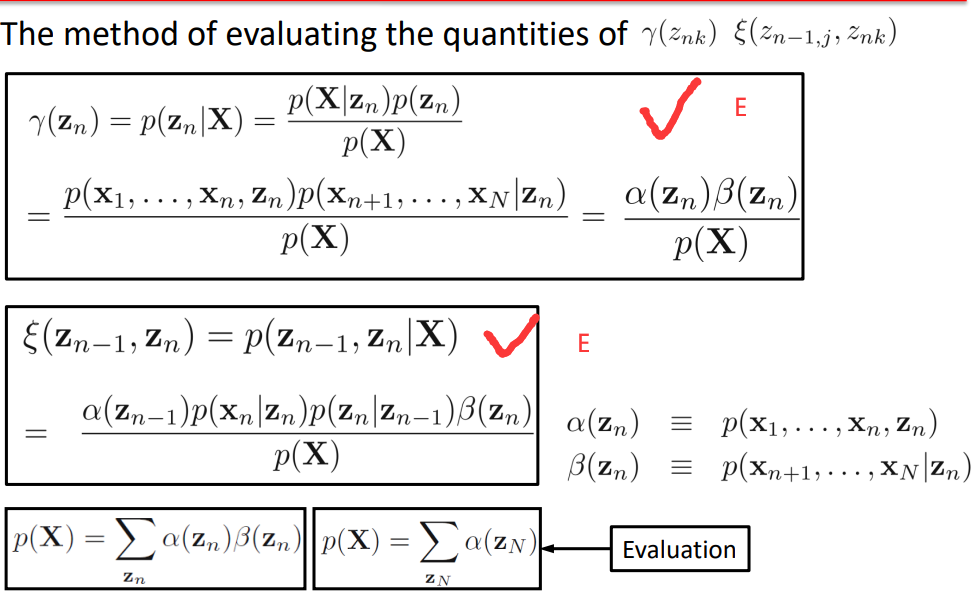
对于观测变量服从离散多项式分布的情形，观测变量的条件概率分布和M步骤方程分别为



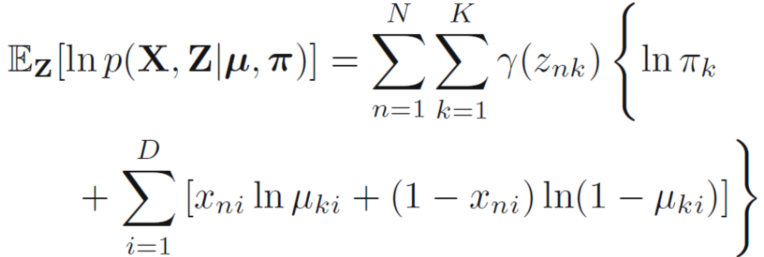
前向后向：

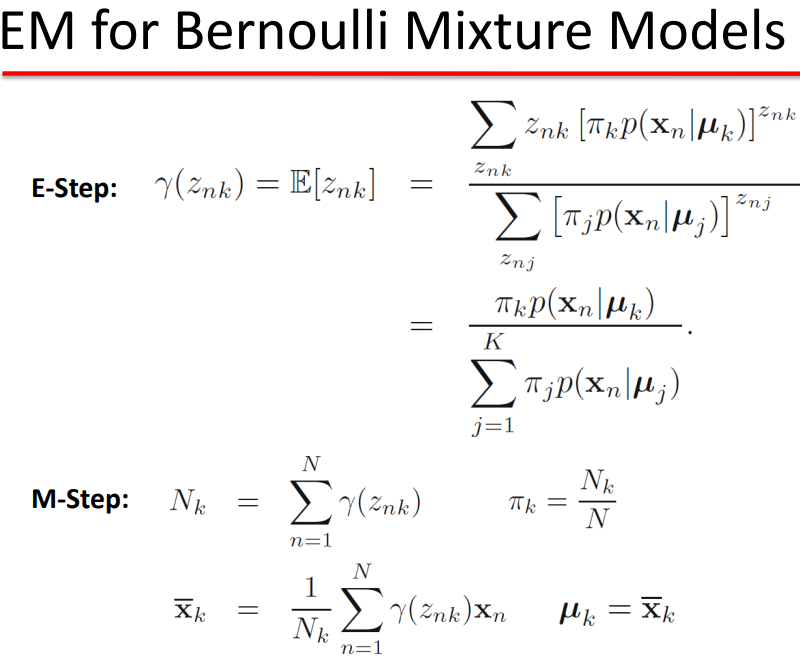




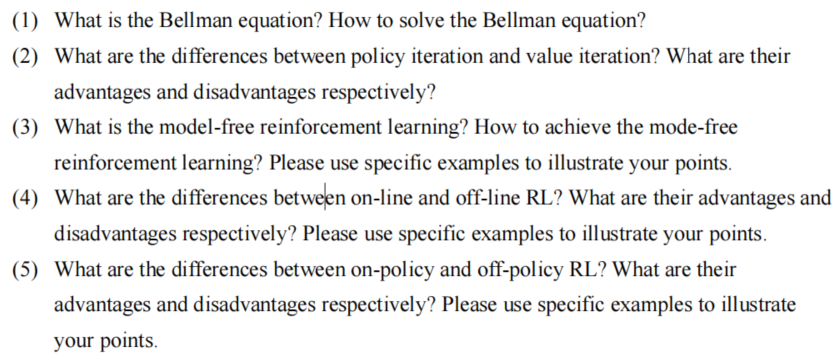


EM for Bernoulli MM:

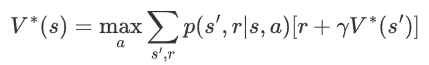


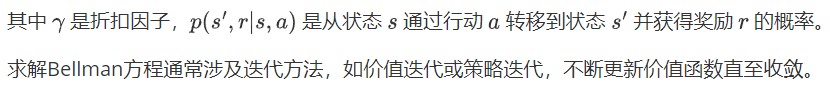


强化学习相关概念：



(1)Bellman方程是强化学习中的一个基本方程，它提供了一种递归的方式来表达最优策略下的价值函数。对于状态价值函数，Bellman最优方程为：





不需要模型。

(2) 策略迭代和价值迭代是求解Bellman方程的两种常见方法。

策略迭代包括两个步骤：策略评估（计算当前策略下的价值函数）和策略改进（基于价值函数改进策略）。它的优点是通常比价值迭代更快收敛，但每次迭代都需要求解整个MDP，这可能计算上更昂贵。

价值迭代直接迭代更新价值函数，直到收敛，然后从最终的价值函数导出策略。它的优点是实现简单，计算上通常比策略迭代更省资源，但可能需要更多的迭代次数才能收敛。

(3)无模型强化学习指的是智能体不需要学习或知道环境的模型（即转移概率和奖励函数），而是直接从与环境的交互中学习策略。这可以通过方法如Q学习或Sarsa来实现，这些方法通过采样的方式直接估计价值函数或动作价值函数。

(4)在线强化学习是指智能体在学习的同时与环境交互。这意味着智能体的每一步行动都是基于当前学到的知识，这可以加速学习但也可能导致在学习过程中做出不好的决策。

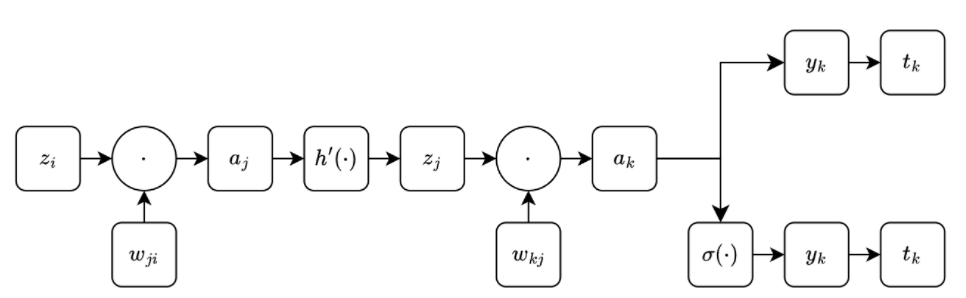
离线强化学习是指智能体从已经收集好的数据（例如通过早期的在线交互）中学习，不直接与环境交互。这允许智能体从大量数据中学习，但可能导致策略过于依赖于数据集中的分布。

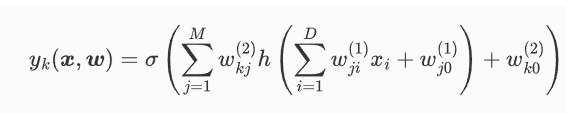
(5)在策（On-policy）强化学习是指智能体学习和评估基于当前策略采取的行动。Sarsa是一个典型的在策学习算法。

离策（Off-policy）强化学习是指智能体学习一个与它用来探索环境的策略不同的策略。Q学习是一个典型的离策学习算法。

在策学习的优点是策略始终与智能体的行为保持一致，而离策学习的优点是智能体可以学习一个最优策略，而不必将其用于探索。但离策算法可能更难收敛，因为它涉及到从可能与目标策略有很大不同的行为策略中学习。

神经网络：

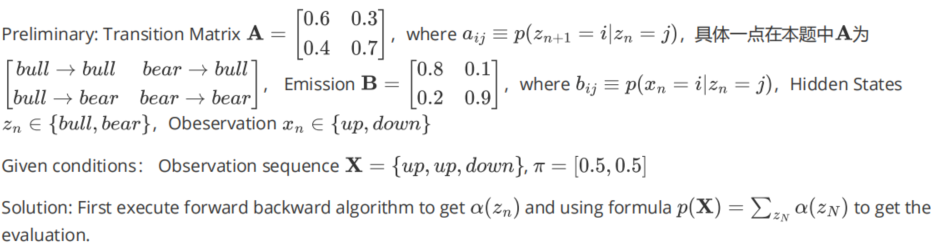


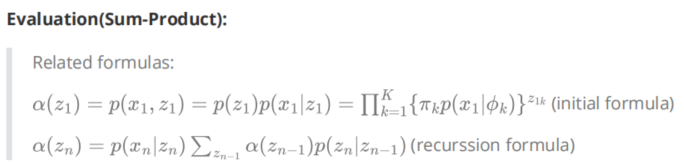


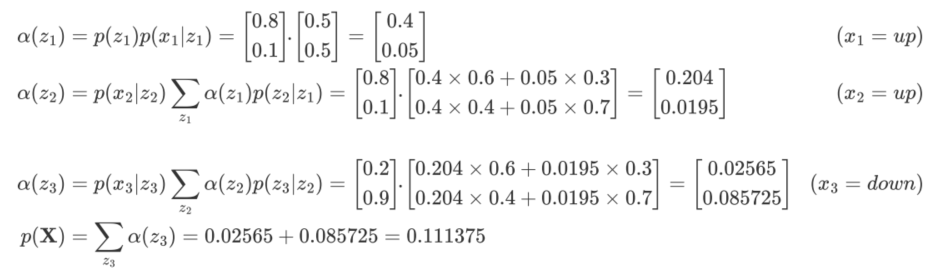
生成模型（generative approaches）和判别模型（discriminative approaches）的主要区别在于：

生成模型旨在学习输入数据的整体分布，包括特征和标签，从而能够生成新的数据实例。

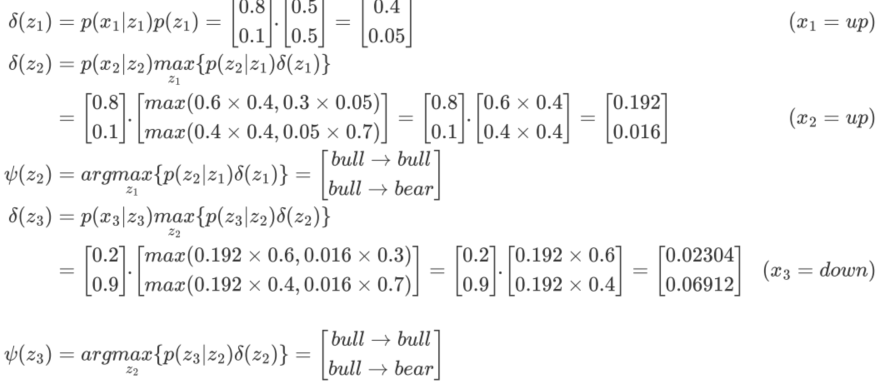
判别模型则专注于学习从输入特征到输出标签的映射，即直接学习不同类别之间的边界。

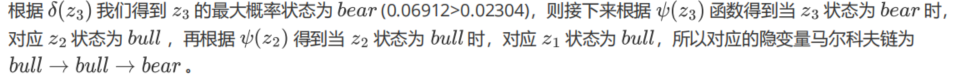


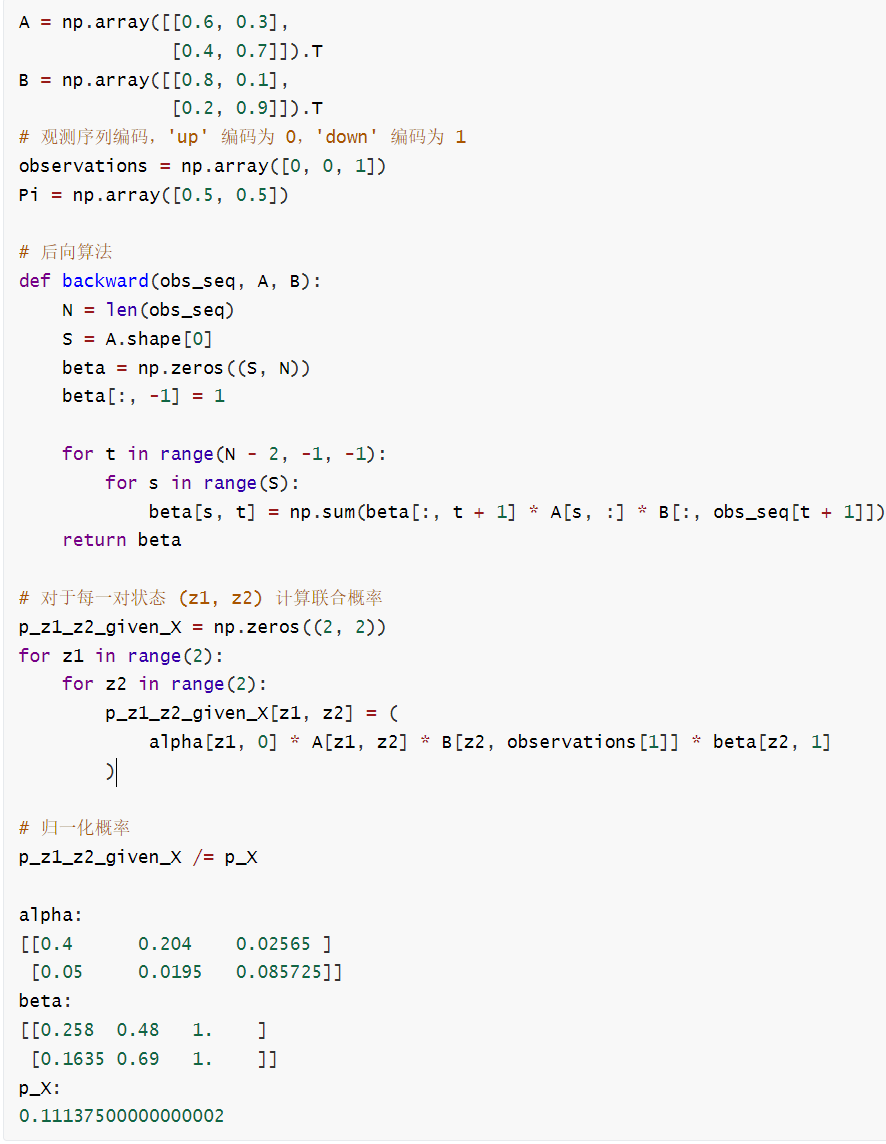




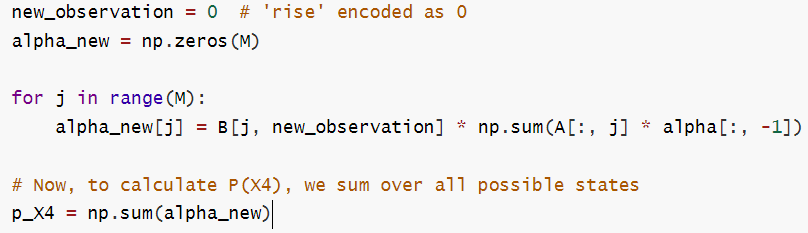




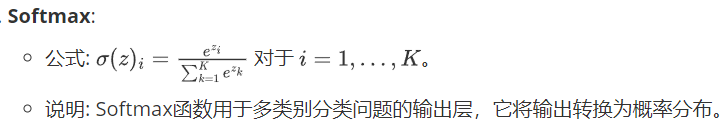




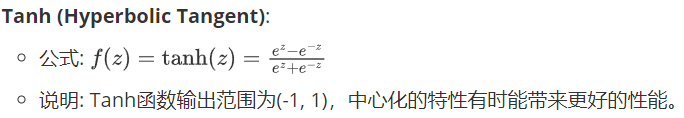
预测p(X4):



损失函数：

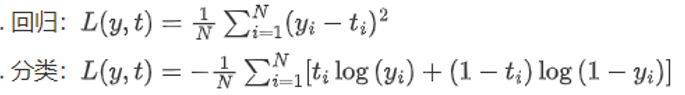


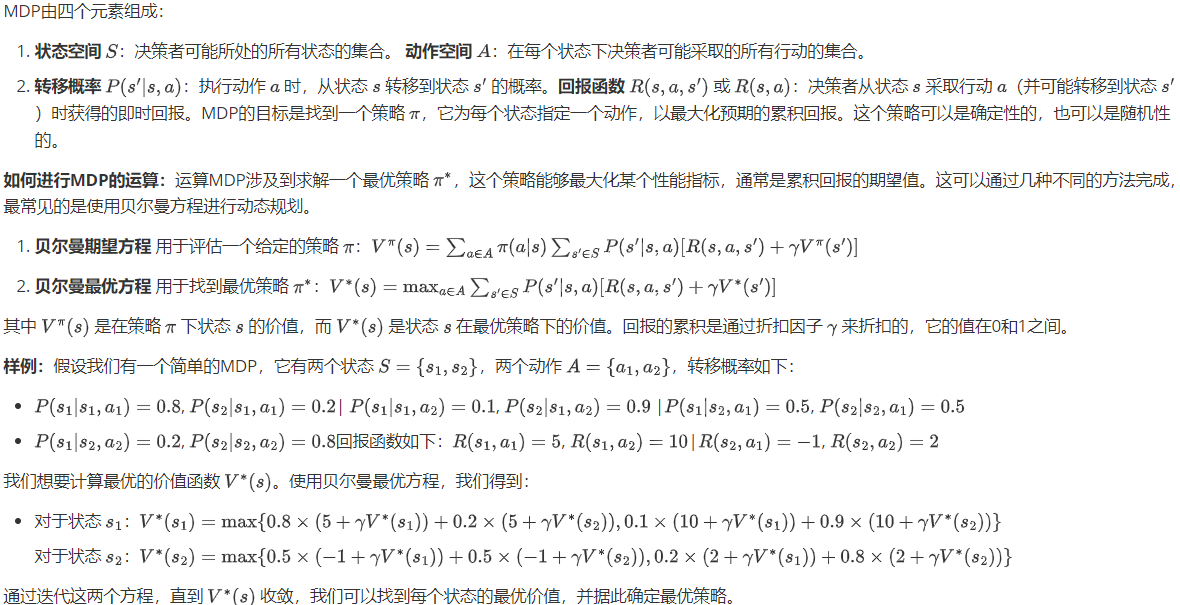






ReLU





强化学习题目：

动态规划，它适用于已知奖励和转移概率的情况。

蒙特卡洛方法不需要转移概率，只需要奖励函数即可。

时序差分方法适用于在线学习，它结合了动态规划和蒙特卡洛方法的特点，可以在不知道完整环境模型的情况下学习。

SVM:

铰链损失函数（Hinge Loss Function）：用于SVM的标准分类问题，特别是在最大间隔分类器的情景下。

-不敏感损失函数(Insensitive Loss Function）：用于支持向量回归（SVR），允许在范围内的误差不被惩罚。