Lec04 - Maximum Likelihood Learning

上节课已经学习了一些有关于自回归模型和自编码器的基本知识，接下来的问题是：**如何训练**

已有条件

1. 假设定义域是由一些底层概率分布P_{data}控制的。
2. 有一批含有m个元素的数据集$\mathcal{D}$，其中的元素都是从P_{data}采样得到的。
3. 标准假设是数据实例是**独立且相同分布的**(IID,independent and identically distributed)
4. 我们有一批模型$\mathcal{M}$，我们的任务是从中选取出最“好”的模型$\hat{\mathcal{M}} \in \mathcal{M}$，其对应的概率分布为$p_{\hat{\mathcal{M}}}$

学习目标

1. 得到一个最“好”的模型$\hat{\mathcal{M}}$，其能够捕捉并尽可能贴近底层概率分布P_{data}
2. 这一般来讲不太可能实现，因为：
    - 有限的数据只能提供真实底层分布的粗略近似值
    - 计算复杂度原因

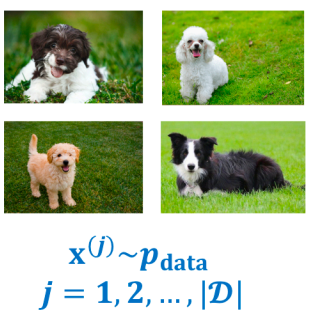
什么是最“好”的?

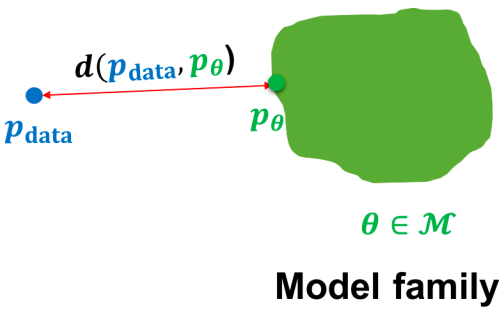
这实际上取决于我们的目的

1. **密度估计**：我们对**整个分布**感兴趣（这样以后我们可以计算任何我们想要的条件概率）
2. **指定的预测任务**：我们使用改分布进行预测。
3. **注重网络架构**：我们对模型本身更感兴趣。

主要针对**密度估计**任务。

密度估计中的学习





我们希望让P_θ尽可能“接近“P_{data}

什么是最“近”的？KL散度！

1. **KL散度**的定义与性质
   - **定义**：KL散度(Kullback-Lerbler Divergence)衡量两个概率分布p和q之间的差异：

$$
\begin{aligned}
D_{KL}(p \parallel q) &= \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \\
&= \mathbf{E}_{\mathbf{x} \sim p} \left[\log \left(\frac{p(\mathbf{x})}{q(\mathbf{x})} \right) \right] \\
&= \mathbf{E}_{\mathbf{x} \sim p}[\log p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim q}[\log q(\mathbf{x})]
\end{aligned}
$$

     其中$\mathbf{E}_{x \sim p}[\cdot]$表示对从真实数据分布$p_{data}(x)$中采样的随机变量$x$计算期望值。例如：

$$
\mathbf{E}_{x \sim p}[\log p(x)] = \sum_{x} p(x) \log p(x)
$$

     对于连续变量，求和替换为积分即可。

   - **解释**：用信息论中的观点解释，它在某种程度上告诉你与**压缩**相关的东西。它告诉你基于p和q的压缩方案将如何表现。换句话说：如果数据确实来自p，而你使用一个优化针对q的压缩方案，那么它会比基于真实数据分布的压缩方案差多少。更精准的表示方法是：如果真实数据来自p，而你使用针对q的优化压缩方案，则$D_{KL}(p \parallel q)$是你平均需要的额外的bits数量
   - **性质**：
     - 非负性：$D_{KL}(p \parallel q) \geq 0$，当且仅当$p = q$时等号成立。
     - 不对称性：$D_KL(p \parallel q) \neq D_{KL}(q \parallel p)$
2. **KL散度**在密度估计中的作用
   - **目标**：学习一个模型分布$p_\theta(x)$，使其尽可能接近真实数据分布$p_{data}(x)$
   - **优化目标**：最小化$D_{KL}(p_{data} \parallel p_\theta)$，即：

$$
\min_\theta D_{KL}(p_{data} \parallel p_\theta) = \min_\theta \left(\mathbf{E}_{x \sim p_{data}} \left[\log p_{data}(x) \right] - \mathbf{E}_{x \sim p_{data}} \left[\log p_\theta(x) \right] \right)
$$

     公式中的第一项是真实分布下数据的平均对数概率。第二项是模型分布下数据的平均对数概率。

     由于$\mathbf{E}_{x \sim p_{data}} \left[\log p_{data}(x) \right]$是常数，因此优化目标等价于最大化**期望对数似然**，也就是模型分布下数据的平均对数概率：

$$
\max_\theta \mathbf{E}_{x \sim p_{data}} \left[\log p_\theta(x) \right]
$$

     由于偏置项永远存在，所以我们无法准确地知道模型分布和真实分布之间有多“远”，但是我们能够知道的是模型A的分布和模型B的分布哪个更接近真实分布。
3. 从**KL散度**到**最大似然估计**

   由于真实分布$p_{data}(x)$是未知的，我们通过训练数据集$\mathcal{D}$近似计算对数似然期望值：

$$
\mathbf{E}_{x \sim p_{data}}[\log p_\theta(x)] = \sum_x p_{data}(x) \log p_\theta(x) \approx \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log p_\theta(\mathbf{x})
$$

   所以优化目标变为**最大化数据的对数似然**：

$$
\max_{p_\theta} \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log p_\theta(\mathbf{x})
$$

蒙特卡洛估计的核心思想

蒙特卡洛估计是一种通过**随机采样**来近似复杂数学期望或积分的方法。它的核心思想可以分解为以下几步：

1. **将问题转化为期望值计算**：假设需要计算某个函数$g(x)$在分布$P(x)$下的期望值：

   $$\mathbf{E}_{x\sim P}[g(x)] = \sum_x P(x)g(x)$$

   直接计算可能因高维或分布复杂而难以实现。

2. **用样本均值近似期望值**
   1. **生成样本**：从分布$P(x)$中独立抽取T个样本$x^1, x^2, \ldots, x^T$
   2. **计算样本均值**：用这些样本的平均值近似期望：

      $$\hat{g} = \frac{1}{T}\sum_{t=1}^{T} g(x^t)$$

关键性质：

- **无偏性**：估计值的期望等于真实期望：

  $$\mathbf{E}[\hat{g}] = \mathbf{E}_{x\sim P}[g(x)]$$

- **收敛性**：随着样本量$T \to \infty$，估计值依概率收敛到真实值（大数定律）：

  $$\hat{g} \to \mathbf{E}_{x\sim P}[g(x)]$$

- **方差降低**：估计的方差与样本量成反比，增加样本量$T可减少估计的波动。

  $$Var(\hat{g}) = \frac{Var(g(x))}{T}$$

在密度估计中的应用：在生成模型中，蒙特卡洛估计常用于近似**对数似然的期望**，即(5)式。

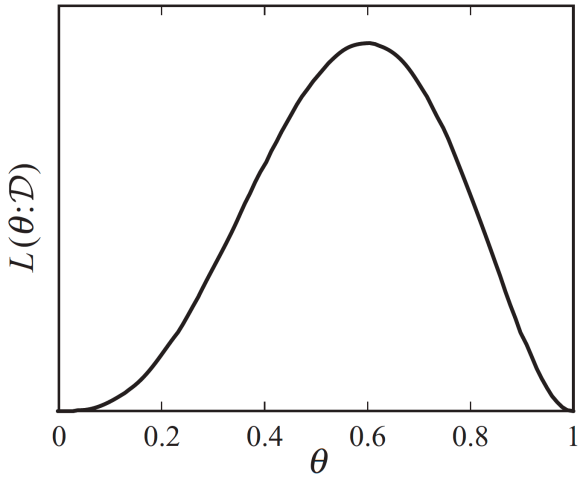
以抛硬币为例

1. **问题设定**
   - 目标：估计一枚偏执硬币的正面概率θ ($\theta \in [0,1]$)
   - 观测数据：通过抛硬币实验得到数据集$\mathcal{D} = \{H, H, T, H, T\}$（3次正面，两次反面）
   - 模型假设：硬币的每次抛掷独立并服从伯努利分布
2. **构建似然函数**
   - **似然函数**：表示在一种底层概率分布下，采样得到如同数据集这样的样本分布的可能性。

     数据集的联合概率（假设独立同分布）为：

     $$\mathcal{L}(\theta;\mathcal{D}) = \prod_{x\in\mathcal{D}} P(x;\theta) = \theta^{\#Heads}\cdot\theta^{\#Teils}$$

     对于示例数据，$\mathcal{L}(\theta;\mathcal{D}) = \theta^3\cdot(1-\theta)^2$



- **对数似然函数**：取对数简化计算

$$\log \mathcal{L}(\theta;\mathcal{D}) = 3\log\theta + 2\log(1-\theta)$$

3. **最大似然估计(MLE)的推导**
   - 优化目标：找到使对数似然最大的θ，因为我们只有有限的数据集，因此我们需要找到在什么分布下采样得到这种数据集的可能性最大，得到的分布就是最好的分布。

$$\theta^* = arg\max_{\theta} \log \mathcal{L}(\theta;\mathcal{D})$$

   - 求导并解方程：

$$\frac{d}{d\theta}\log \mathcal{L}(\theta;\mathcal{D}) = \frac{3}{\theta} - \frac{2}{1-\theta} = 0$$

     解得：$\theta^* = 0.6$

4. **推广至贝叶斯网络**
   - 假设我们有一个含有n个变量的自回归模型和贝叶斯网络中的分解公式：

$$P_\theta(\mathbf{x}) = \prod_{i=1}^{n} p_{neural}\left(x_i|pa(x_i);\theta_i\right)$$

   - 训练数据集为$\mathcal{D} = \left\{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)}\right\}$，于是我们可以根据这个分解公式来分解似然函数：

$$\mathcal{L}(\theta;\mathcal{D}) = \prod_{j=1}^{m} P_\theta(\mathbf{x}^{(j)}) = \prod_{j=1}^{m}\prod_{i=1}^{n} p_{neural}\left(x_i^{(j)}|pa(x_i)^{(j)};\theta_i\right)$$

   - 目标：最大化似然函数，也就是最大化对数似然函数

$$\max_{\theta} l(\theta;\mathcal{D}) = \max_{\theta} \log \mathcal{L}(\theta;\mathcal{D}) = \max_{\theta} \sum_{j=1}^{m}\sum_{i=1}^{n} \log p_{neural}\left(x_i^{(j)}|pa(x_i)^{(j)};\theta_i\right)$$

梯度下降(Gradient Descent)

在最大似然估计(MLE)框架下，目标是通过优化模型参数θ以最大化数据的对数似然函数。

优化步骤：

1. 初始化参数：随机初始化θ^0
2. 计算梯度：通过反向传播计算对数似然的梯度$\nabla_\theta l(\theta)$
3. 参数更新：$\theta^{t+1} = \theta^t + \alpha_t \nabla_\theta l(\theta)$

由于目标函数是非凸的，梯度下降可能陷入局部最优，但在实践中通常表现良好。

随机梯度下降(Stochastic Gradient Descent,SGD)

当数据集规模m极大时，计算全体样本的梯度计算量过大。SGD通过采样小批量数据近似梯度

梯度近似：梯度可表示为：

$$\nabla_\theta l(\theta) = m \cdot \mathbf{E}_{x^{(j)} \sim \mathcal{D}} \left[\sum_{i=1}^{n} \log p_{neural} \left(x_i^{(j)} | pa(x_i)^{(j)}; \theta_i \right) \right]$$

通过蒙特卡洛模拟，每次随机采样一个样本$x^{(j)}$，用其梯度近似整体期望即可。

过拟合与泛化

1. 过拟合风险：
    - 模型可能记住训练数据（如“数据即模型”），导致在未见数据上表现差。
    - 需通过限制假设空间（Hypothesis Space）或正则化提升泛化能力。
2. **偏差-方差权衡**：
    - **偏差**：模型假设空间过于简单，无法逼近真实分布P_{data}（如线性模型拟合非线性关系）。
    - **方差**：模型过于复杂，对训练数据的小扰动敏感（如高阶多项式过拟合）。
    - **平衡方法**：选择中等复杂度模型（如低阶多项式），或加入正则化项。

避免过拟合的策略：

- **硬约束**：限制模型复杂度（如贝叶斯网络的最大父节点数、减少神经网络参数量）。
- **正则化**：在目标函数中增加惩罚项：

$$Objective = Loss(x, \mathcal{M}) + \lambda R(\mathcal{M})$$

    常见的正则化方法包括L1/L2正则化
- **验证集**：使用独立验证集评估模型泛化能力。