第九章 概率模型

第 27 讲 机器学习中的概率模型

黄定江

DaSE @ ECNU

djhuang@dase.ecnu.edu.cn

1. 27.1 机器学习的概率思路
2. 27.2 统计机器学习中的概率模型
3. 27.3 深度学习中的概率模型
4. 27.4 强化学习中的概率模型

监督学习的数据假设：服从联合概率分布

联合概率分布

- 在统计机器学习中，通常假设输入与输出的随机变量 X 和 Y 遵循联合概率分布 $P(X, Y)$，$P(X, Y)$ 表示分布函数或分布密度函数。
- 在学习过程中，假定这些联合概率分布存在，但对学习系统来说，联合概率分布的具体定义是未知的。
- 训练数据与测试数据被看作是依联合概率分布 $P(X, Y)$ 独立同分布产生的。

统计机器学习中的概率模型：条件概率分布

- 监督学习的概率模型通常由条件概率分布 $P(Y|X)$ 来表示。对具体的输入进行相应的输出预测时，写作 $P(y|x)$。
- 无监督学习的概率模型取条件概率分布形式 $P(z|x)$ 或者 $P(x|z)$，$x \in \mathcal{X}$ 是输入，$z \in \mathcal{Z}$ 是输出，$\mathcal{X}$ 是输入空间，$\mathcal{Z}$ 是输出空间。

监督学习中的生成模型与判别模型

监督学习方法可以分为生成方法和判别方法，所学到的模型分别称为：

1. 生成模型
   - 生成方法由数据学习联合概率分布 $P(X, Y)$，然后求出条件概率分布 $P(Y|X)$ 作为预测的模型，即生成模型：

$$P(Y|X) = \frac{P(X, Y)}{P(X)}$$

     这样的方法之所以称为生成方法，是因为模型表示了给定输入 X 产生输出 Y 的生成关系。典型的概率型生成模型有朴素贝叶斯法和隐马尔可夫模型等等。
2. 判别模型
   - 判别方法由数据直接学习决策函数 $f(X)$ 或者条件概率分布 $P(Y|X)$ 作为预测的模型，即判别模型。判别方法关心的是对给定的输入 X，应该预测什么样的 Y。典型的概率型判别模型有决策树、逻辑斯蒂回归模型、最大熵模型和条件随机场等等。

概率模型的假设空间

概率模型的假设空间可以定义为条件概率的集合：

$$\mathcal{F} = \{P|P(Y|X)\}, \tag{1}$$

其中，X 和 Y 是定义在输入空间 $\mathcal{X}$ 和输出空间 $\mathcal{Y}$ 上的随机变量。这时 $\mathcal{F}$ 通常是由一个参数向量决定的条件概率分布族：

$$\mathcal{F} = \{P|P_\theta(Y|X), \theta \in R^n\}, \tag{2}$$

参数向量 θ 取值于 n 维欧氏空间 R^n，也称为参数空间。

概率模型学习的过程

- 监督学习分为学习和预测两个过程，由学习系统和预测系统完成。在学习过程中，学习系统利用给定的训练数据集，通过学习（或训练）得到一个模型，表示为条件概率分布 $\hat{P}(Y|X)$，描述了输入与输出随机变量之间的映射关系。在预测过程中，预测系统对于给定的测试样本集中的输入 x_{N+1}，由模型 $y_{N+1} = \arg\max_y \hat{P}(y|x_{N+1})$ 给出相应的输出 y_{N+1}。
- 无监督学习也由学习系统和预测系统完成。在学习过程中，学习系统从训练数据集学习，得到一个最优模型，表示条件概率分布 $\hat{P}(z|x)$ 条件概率分布 $\hat{P}(x|z)$。在预测过程中，预测系统对于给定的输入 x_{N+1}，由模型 $z_{N+1} = \arg\max_z \hat{P}(z|x_{N+1})$ 给出相应的输出 z_{N+1}，进行聚类或降维，或者由模型 $\hat{P}(x|z)$ 给出输入的概率 $\hat{P}(x_{N+1}|z_{N+1})$，进行概率估计。

概率模型学习的常用损失函数：对数似然函数

常用的损失函数：

- 对数损失函数或对数似然损失函数

$$L(Y, P(Y|X)) = -logP(Y|X)$$

模型求解

- 监督学习的本质是学习输入到输出的映射的统计规律。对于监督概率模型的学习，本质与概率模型的参数估计相关。
- 无监督学习的本质是学习数据中的统计规律或潜在结构。对于无监督概率模型的学习，本质与概率模型的非参数估计相关。

注：在概率模型的学习和推理中还经常使用贝叶斯技巧，其主要想法是：利用贝叶斯定理，计算在给定数据条件下模型的条件概率，即后验概率，并应用这个原理进行模型的估计，以及对数据的预测。将模型、未观测要素及其参数用变量表示，使用模型的先验分布是贝叶斯学习的特点。估计的方法包括最大后验估计等各种贝叶斯估计方法。

27.2.1 决策树模型

决策树是一类常见的机器学习方法。顾名思义，它是基于树结构来进行决策（比如分类或回归），这恰是人类在面临决策问题时一种很自然的处理机制。

定义 1

决策树模型是一种对实例进行某种决策（比如分类或回归）的树形结构，其由一个根结点、若干个内部结点和若干个叶结点以及有向边组成；其中根结点包含样本全集，内部结点对应一个特征或属性，叶结点对应于决策结果（比如在分类中，对应某个具体的类）；每个结点包含的样本集合根据特征选择或属性测试的结果被划分到子结点中去。

决策树学习的目的是为了生成一颗泛化能力强，即处理未见实例能力强的决策树。其基本流程遵循“分而治之”的策略：以用决策树分类为例，从根结点开始对实例的某一个特征进行测试，根据测试结果将实例分配到其子结点；这时，每一个子结点对应着该特征的一个取值。如此递归地对实例进行测试并分配，直到达到叶结点，最后将实例分到叶结点的类中。所以在分类问题中，决策树模型表示基于特征对实例进行分类的过程。

决策树模型的表示：条件概率分布

决策树模型可以通过两种方式来表示：看成是 if-then 规则的集合或看成是定义在特征空间与类空间上的条件概率分布。下面我们主要介绍基于条件概率分布的决策树表示与学习。

- 决策树可看成在给定特征条件下类的条件概率分布。这一条件概率分布定义在特征空间的一个划分上。将特征空间划分为互不相交的单元或区域，并在每个单元定义一个类的概率分布就构成了一个条件概率分布。决策树的一条路径对应于划分中的一个单元。决策树所表示的条件概率分布由各个单元给定条件下类的条件概率分布组成。
- 假设 X 为表示特征的随机向量，Y 表示类的随机向量，那么这个条件概率分布就可以表示为 $P(Y|X)$。X 取值于给定划分下单元的集合，Y 取值于类的集合。各叶结点（单元）上的条件概率往往偏向于某一个类，即属于某一类的概率较大。决策树分类时将该结点的实例强行分到条件概率大的那一类去。

决策树学习的三个步骤和生成策略

决策树学习通常包括三个步骤：特征选择、决策树的生成、决策树的修剪。

- 由于决策树表示一个条件概率分布，所以深浅不同的决策树对应着不同复杂度的概率模型。
- 决策树的生成对应于模型的局部选择，决策树的剪枝对于模型的全局选择，决策树的生成只考虑局部最优，相对的决策树的剪枝则考虑全局最优。

决策树学习的关键是如何选择最优划分属性或最优的特征来划分特征空间。一般而言，随着划分过程不断进行，我们希望决策树的分支结点所包含的类别尽可能属于同一类别，即结点的“纯度”越来越高。目前主要有三类属性划分或特征选择的准则：

- 信息增益
- 增益比
- 基尼系数

1. 基于信息增益的决策树生成

现假设训练数据集为 D，$|D|$ 表示其样本容量。设有 K 个类 C_k，$k=1,2,\cdots,K$，$|C_k|$ 为属于类 C_k 的样本个数。显然 $\sum_{k=1}^{K}|C_k|=|D|$。设某特征 A 取值为 $\{a_1,a_2,\cdots,a_n\}$，根据特征 A 的取值将 D 划分为 n 个子集 $D_1,D_2,\cdots,D_n$，$|D_i|$ 为 D_i 的样本个数，$\sum_{i=1}^{n}|D_i|=|D|$。记子集 D_i 中属于类 C_k 的样本的集合为 D_{ik}，即 $D_{ik}=D_i\cap C_k$，$|D_{ik}|$ 为 D_{ik} 的样本个数。因此，可以给出信息增益定义如下：

定义 2

(信息增益) 特征 A 对训练数据集 D 的信息增益 $g(D,A)$，定义为集合 D 的经验熵 $H(D)$ 与特征 A 给定条件下 D 的经验条件熵 $H(D\mid A)$ 之差，即

$$g(D,A)=H(D)-H(D\mid A)$$

其中

$$H(D)=-\sum_{k=1}^{K}\frac{|C_k|}{|D|}\log_2\frac{|C_k|}{|D|},$$

$$H(D\mid A)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\sum_{k=1}^{K}\frac{|D_{ik}|}{|D_i|}\log_2\frac{|D_{ik}|}{|D_i|}.$$

可以看出，决策树学习中的信息增益等价于训练数据集中类与特征的互信息。一般而言，信息增益越大，则意味着使用特征 A 来进行划分所获得的“纯度提升”越大。因此，我们可用信息增益来进行决策树的特征选择或划分属性选择，也即求解如下最优化问题为：

$$\max_{A_i} \ g(D, A_i) = H(D) - H(D \mid A_i),$$

其中 A_i 表示样本空间的第 i 个特征。

2. 基于信息增益比的决策树生成

以信息增益准则作为划分训练数据集的特征, 存在偏向于选择取值较多的特征的问题。为了减少这种偏好可能带来的不利影响，可以使用信息增益比 (information gain ratio) 来选择最优特征。

定义 3

(信息增益比) 特征 A 对训练数据集 D 的信息增益比 $g_R(D,A)$，定义为其信息增益 $g(D,A)$ 与训练数据集 D 关于特征 A 的值的熵 $H_A(D)$ 之比, 即

$$g_R(D,A) = \frac{g(D,A)}{H_A(D)}$$

其中,

$$H_A(D) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|},$$

n 是特征 A 取值的个数。

3. 基于基尼系数的决策树生成

还可使用基尼指数对分类决策树进行最优特征选择，基尼指数也可衡量分布或数据的不确定性，其定义如下：

定义 4

(基尼指数) 分类问题中, 假设有 K 个类, 样本点属于第 k 类的概率为 p_k, 则概率分布的基尼指数定义为

$$\mathrm{Gini}(p)=\sum_{k=1}^{K} p_k\left(1-p_k\right)=1-\sum_{k=1}^{K} p_k^2$$

对于给定的样本集合 D, 其基尼指数为

$$\mathrm{Gini}(D)=1-\sum_{k=1}^{K}\left(\frac{|C_k|}{|D|}\right)^2$$

这里, C_k 是 D 中属于第 k 类的样本子集, K 是类的个数。

一般地，基尼指数值越大，样本集合的不确定性也就越大，这一点与熵相似。

基于基尼系数的决策树生成

在决策树生成过程中，我们需要考虑某一特征划分下数据集的基尼指数。如果样本集合 D 根据特征 A 是否取某一可能值 a 被分割成 D_1 和 D_2 两部分, 即

$$D_1 = \{(x, y) \in D \mid A(x) = a\}, \quad D_2 = D - D_1$$

则在特征 A 的条件下, 集合 D 的基尼指数定义为

$$\text{Gini}(D, A) = \frac{|D_1|}{|D|} \text{Gini}\,(D_1) + \frac{|D_2|}{|D|} \text{Gini}\,(D_2)\,.$$

显然，基尼指数 $\text{Gini}(D, A)$ 表示经 $A = a$ 分割后集合 D 的不确定性。

因此，在决策树生成过程中，我们需要在所有可能的特征以及它所有可能的切分点 a 中，选择基尼指数最小的特征及其对应的切分点，作为最优特征与最优切分点。此时，最优特征和对应切分点选择的优化问题可表示为：

$$\min_{A_i, a_{ij}} \text{Gini}(D, A_i = a_{ij}),$$

其中 A_i 表示样本空间的第 i 个特征，a_{ij} 表示特征 A_i 的第 j 个可能的取值。

4. 决策树的剪枝优化：减少过拟合

决策树生成算法递归地产生决策树，直到不能继续下去为止，这样产生的决策树由于在学习时过多地考虑如何提高对训练数据的正确分类，从而构建出分支过多、过于复杂的决策树，因而出现过拟合的现象。剪枝是决策树学习算法对付过拟合的主要手段。

决策树的剪枝一般通过最小化决策树整体的损失函数或代价函数来实现，也即：

$$\min_{T} \sum_{t=1}^{|T|} N_t H_t(T) + \alpha |T|$$

其中 t 是树 T 的叶结点，$|T|$ 代表树 T 的叶结点个数，N_t 表示具体某个叶结点的样本数，$\alpha \geq 0$ 为参数，$H_t(T)$ 表示叶节点经验熵为

$$H_t(T) = -\sum_{k=1}^{K} \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t}$$

其中，N_{tk} 表示 k 类样本点的个数，$k = 1, 2, \cdots, K$。

决策树的剪枝优化

由于上述优化问题中目标函数第一项实际上就是负对数似然函数：

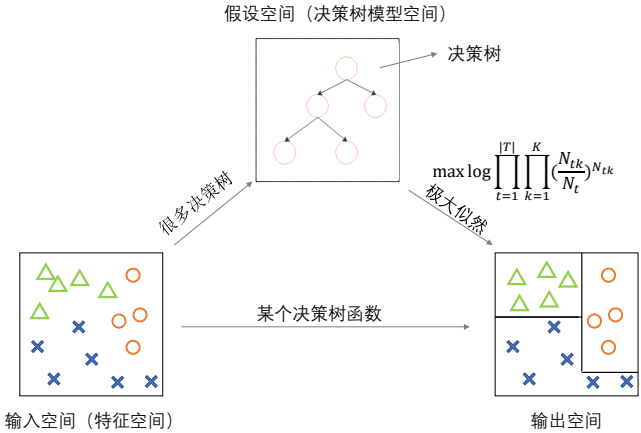
$$\sum_{t=1}^{|T|} N_t H_t(T) = -\sum_{t=1}^{|T|}\sum_{k=1}^{K} N_{tk} \log \frac{N_{tk}}{N_t}$$

$$= -\log \prod_{t=1}^{|T|}\prod_{k=1}^{K} \left(\frac{N_{tk}}{N_t}\right)^{N_{tk}}$$

因此，上述优化问题等价于极大对数似然函数，即**优化问题**:

$$\max \quad \log \prod_{t=1}^{|T|}\prod_{k=1}^{K} \left(\frac{N_{tk}}{N_t}\right)^{N_{tk}}$$

从空间的角度理解决策树模型



27.2.2 逻辑斯谛回归模型

逻辑斯谛回归是统计学习中的经典分类方法，它依赖于逻辑斯谛分布。

定义 5

(逻辑斯谛分布) 设 X 是连续随机变量，X 服从逻辑分布是指 X 具有下列分布函数和密度函数：

$$F(x) = P(X \leq x) = \frac{1}{1 + e^{-(x-\mu)/\gamma}} \tag{3}$$

$$f(x) = F'(x) = \frac{e^{-(x-\mu)/\gamma}}{\gamma(1 + e^{-(x-\mu)/\gamma})^2} \tag{4}$$

式中，μ 为位置参数，$\gamma > 0$ 为形状参数。

逻辑斯谛分布的分布函数属于逻辑函数，其图形是一条关于点 $(\mu, \frac{1}{2})$ 为中心对称的 sigmoid 曲线。该曲线在中心附近增长速度较快，在两端增长速度较慢。

二项逻辑斯谛回归模型

二项逻辑斯谛回归模型是一种分类模型，由条件概率分布 $P(Y|X)$ 表示，形式为参数化的逻辑斯谛分布。这里，随机变量 X 取值为实数，随机变量 Y 的取值为 1 和 0。

例 1

二项逻辑斯谛回归模型是如下的条件概率分布：

$$P(Y=1|x)=\frac{\exp(w\cdot x+b)}{1+\exp(w\cdot x+b)} \tag{5}$$

$$P(Y=0|x)=\frac{1}{1+\exp(w\cdot x+b)} \tag{6}$$

其中，$x\in R^n$ 是输入，$Y\in\{0,1\}$ 是输出，$w\in R^n$ 和 $b\in R$ 是参数，w 称为权值向量，b 称为偏置。

逻辑斯谛回归模型的变体

为了更简洁地表示，有时会将权值向量和输入向量加以扩充，记作 $w = (w^{(1)}, w^{(2)}, ..., w^{(n)}, b)^T$，$x = (x^{(1)}, x^{(2)}, ..., x^{(n)}, 1)^T$。此时逻辑回归模型如下：

$$P(Y = 1|x) = \frac{\exp(w \cdot x)}{1 + \exp(w \cdot x)} \tag{7}$$

$$P(Y = 0|x) = \frac{1}{1 + \exp(w \cdot x)} \tag{8}$$

- 在逻辑斯谛回归模型中，输出 $Y = 1$ 的对数概率是输入 x 的线性函数。
- 线性函数的值越接近正无穷，概率值就越接近 1；线性函数的值越接近负无穷，概率值就越接近 0。

分类策略和模型求解

- 分类策略。对于给定的输入实例 x，按照式(5)和(6)可以求得 $P(Y=1|x)$ 和 $P(Y=0|x)$。逻辑斯谛回归比较两个条件概率值的大小，将实例 x 分到概率值较大的那一类。
- 模型求解。对于给定的训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，其中，$x_i\in\mathbf{R}^n, y_i\in\{0,1\}$，可以应用极大似然估计法估计模型参数。设

$$P(Y=1|x)=\pi(x), \qquad P(Y=0|x)=1-\pi(x)$$

则似然函数为

$$\prod_{i=1}^{N}\left[\pi\left(x_i\right)\right]^{y_i}\left[1-\pi\left(x_i\right)\right]^{1-y_i}$$

模型求解

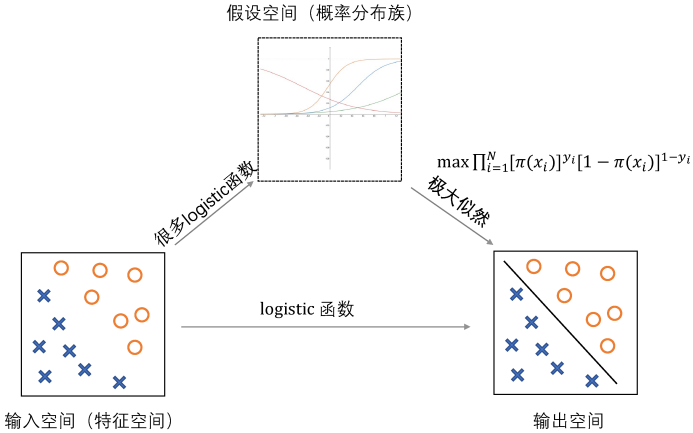
对数似然函数为

$$
\begin{aligned}
L(w) &= \sum_{i=1}^{N} \left[y_i \log \pi(x_i) + (1 - y_i) \log (1 - \pi(x_i)) \right] \\
&= \sum_{i=1}^{N} \left[y_i \log \frac{\pi(x_i)}{1 - \pi(x_i)} + \log (1 - \pi(x_i)) \right] \\
&= \sum_{i=1}^{N} \left[y_i (w \cdot x_i) - \log (1 + \exp(w \cdot x_i)) \right]
\end{aligned}
$$

因此，得到**优化问题**：

$$
\max_{w} : \sum_{i=1}^{N} \left[y_i (w \cdot x_i) - \log (1 + \exp(w \cdot x_i)) \right]
$$

从空间的角度理解 Logistic 模型



27.2.3 最大熵模型

模型原理：最大熵模型 (Maximum Entropy Model) 由最大熵原理推导实现。最大熵原理是概率模型学习的一个准则，它是指在学习概率模型时，在所有可能的概率模型 (分布) 中，熵最大的模型是最好的模型。通常用约束条件来确定概率模型的集合，因此最大熵原理也可以被表述为在满足约束条件的模型集合中选取熵最大的模型。

例 2

假设离散随机变量 X 的概率分布是 $P(X)$，则其熵是

$$H(P) = -\sum_{x} P(x) \log P(x) \tag{9}$$

熵满足下列不等式：

$$0 \leq H(P) \leq \log |X| \tag{10}$$

式中，$|X|$ 是 X 的取值个数，当且仅当 X 的分布是均匀分布时右边等号成立，即当 X 服从均匀分布时熵最大。

最大熵模型

将最大熵原理应用到分类得到最大熵分类模型，它是一个判别模型。假设分类模型是一个条件概率分布 $P(Y \mid X)$，$X \in \mathcal{X} \subseteq \mathbb{R}^n$ 表示输入，$Y \in \mathcal{Y}$ 表示输出，$\mathcal{X}$ 和 $\mathcal{Y}$ 分别是输入和输出的集合。这个模型表示的是对于给定的输入 X，以条件概率 $P(Y \mid X)$ 输出 Y。

模型约束条件：给定一个训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$，可以确定联合分布 $P(X, Y)$ 的经验分布 $\tilde{P}(x, y)$ 和边缘分布 $P(X)$ 的经验分布 $\tilde{P}(x)$，这里分别用训练数据中样本出现的频率和输入数据的频率来表示。用特征函数 $f(x, y)$ 描述输入 x 和输出 y 之间的某一个事实。如果模型能够获取训练数据中的信息，那么就可以假设特征函数关于经验分布 $\tilde{P}(X, Y)$ 的期望值 $E_{\tilde{P}}(f)$ 与特征函数关于模型 $P(Y|X)$ 与经验分布 $\tilde{P}(X)$ 的期望值 $E_P(f)$ 相等，即

$$\sum_{x,y} \tilde{P}(x, y) f(x, y) = \sum_{x,y} \tilde{P}(x) P(y|x) f(x, y), \tag{11}$$

以此作为模型学习的约束条件。假如有 n 个特征函数 $f_i(x, y), i = 1, 2, \cdots, n$，那么就有 n 个约束条件。

最大熵模型

定义 6

假设满足所有约束条件的模型集合为

$$\mathcal{C} \equiv \{P \in \mathcal{P} \mid E_P(f_i) = E_{\tilde{P}}(f_i), \quad i = 1, 2, \cdots, n\} \tag{12}$$

定义在条件概率分布 $P(Y|X)$ 上的条件熵为

$$H(P) = -\sum_{x,y} \tilde{P}(x) P(y|x) \log P(y|x) \tag{13}$$

则模型集合 $\mathcal{C}$ 中条件熵 $H(P)$ 最大的模型称为最大熵模型，式中的对数为自然对数。

最大熵模型的学习可以形式化为约束优化问题。

最大熵模型

模型学习：对于给定的训练数据集 $T=(x_1,y_1),(x_2,y_2),...,(x_N,y_N)$ 以及特征函数 $f_i(x,y),i=1,2,...,n$，最大熵模型的学习等价于约束最优化问题：

$$\max_{P\in C} H(P)=-\sum_{x,y}\tilde{P}(x)P(y|x)\log P(y|x) \tag{14}$$

$$s.t.\ E_p(f_i)=E_{\tilde{P}}(f_i),i=1,2,...,n \tag{15}$$

$$\sum_y P(y|x)=1 \tag{16}$$

通常将求最大值问题改写为等价的求最小值问题：

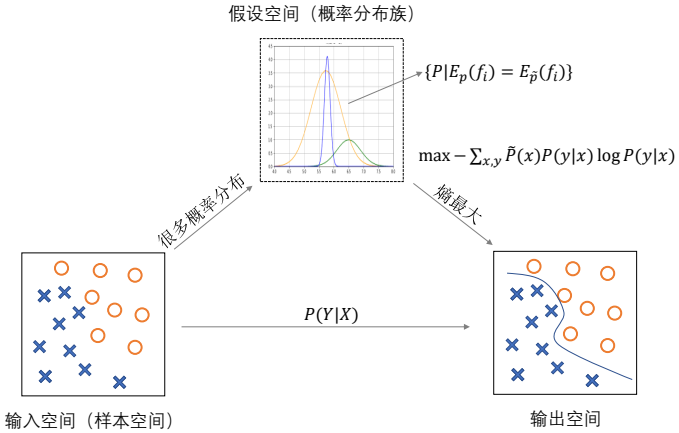
$$\min_{P\in C} -H(P)=\sum_{x,y}\tilde{P}(x)P(y|x)\log P(y|x) \tag{17}$$

$$s.t.\ E_p(f_i)-E_{\tilde{P}}(f_i)=0,i=1,2,...,n \tag{18}$$

$$\sum_y P(y|x)=1 \tag{19}$$

所得解即为最大熵模型学习的解。

从空间的角度理解最大熵模型



27.3.1 受限玻尔兹曼机

受限玻尔兹曼机是一种借助隐变量来描述复杂数据分布的概率图模型，在对复杂数据分布进行建模时，可以有效挖掘和学习出可观测变量之间复杂的依赖关系。

例 3

受限玻尔兹曼机（*Restricted Boltzmann Machine*，*RBM*）是一个二分图结构的无向图模型，如图所示．分别用可观测层和隐藏层来表示这两组变量．同一层中的节点之间没有连接，而不同层一个层中的节点与另一层中的所有节点连接，这和两层的全连接神经网络的结构相同。

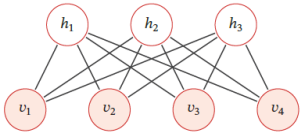


图 4

受限玻尔兹曼机

概率模型：受限玻尔兹曼机模型是一个生成模型，用于生成联合分布 $p(\boldsymbol{v}, \boldsymbol{h})$，这里的 $\boldsymbol{v}$ 和 $\boldsymbol{h}$ 分别表示可观测的随机向量和隐藏的随机向量。若一个受限玻尔兹曼机由 K_v 个可观测变量和 K_h 个隐变量组成, 权重矩阵为 $\boldsymbol{W} \in \mathbb{R}^{K_u \times K_h}$, 其中每个元素 w_{ij} 为可观测变量 v_i 和隐变量 h_j 之间边的权重. 偏置为 $\boldsymbol{a} \in \mathbb{R}^{K_v}$ 和 $\boldsymbol{b} \in \mathbb{R}^{K_h}$, 其中 a_i 为每个可观测的变量 v_i 的偏置, b_j 为每个隐变量 h_j 的偏置. 因此，受限玻尔兹曼机的能量函数定义为

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{a}^\top \boldsymbol{v} - \boldsymbol{b}^\top \boldsymbol{h} - \boldsymbol{v}^\top \boldsymbol{W}\boldsymbol{h}$$

对应的联合概率分布 $p(\boldsymbol{v}, \boldsymbol{h})$ 定义为

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} \exp(-E(\boldsymbol{v}, \boldsymbol{h})),$$

其中 $Z = \sum \exp(-E(\boldsymbol{v}, \boldsymbol{h}))$ 为配分函数.

在给定受限玻尔兹曼机的联合分布 $p(\boldsymbol{v}, \boldsymbol{h})$ 后，通常可以使用吉布斯采样方法生成服从该分布的样本。

受限玻尔兹曼机

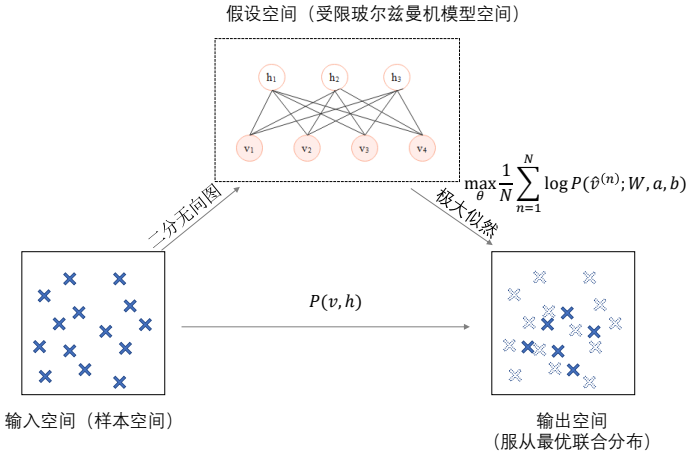
给出了模型的表示之后，作为概率图模型，受限玻尔兹曼机主要涉及推断和学习两类问题。其中，对于参数学习，受限玻尔兹曼机是通过最大化似然函数来找到最优的参数 $\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}$. 给定一组训练样本 $\mathcal{D} = \left\{\hat{\boldsymbol{v}}^{(1)}, \hat{\boldsymbol{v}}^{(2)}, \cdots, \hat{\boldsymbol{v}}^{(N)}\right\}$, 其对数似然函数为

$$\mathcal{L}(\mathcal{D}; \boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}) = \frac{1}{N}\sum_{n=1}^{N} \log p\left(\hat{\boldsymbol{v}}^{(n)}; \boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}\right).$$

因此，得到**优化问题**：

$$\max \quad \frac{1}{N}\sum_{n=1}^{N} \log p\left(\hat{\boldsymbol{v}}^{(n)}; \boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}\right).$$

从空间的角度理解受限玻尔兹曼机模型



吉布斯采样

对于生成模型，一般可以借助吉布斯采样的方法，生成服从对应联合分布的样本。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方法。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 1。对于一个 M 维的随机向量 $\boldsymbol{X} = [X_1, X_2, \cdots, X_M]^\top$, 其第 m 个变量 X_m 的全条件概率为

$$p\left(x_m \mid \boldsymbol{x}_{\backslash m}\right),$$

其中 $\boldsymbol{x}_{\backslash m} = [x_1, x_2, \cdots, x_{m-1}, x_{m+1}, \cdots, x_M]^\top$ 表示除 X_m 外其他变量的取值. 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.

吉布斯采样

假设从一个随机的初始化状态 $x^{(0)} = \left[x_1^{(0)}, x_2^{(0)}, \cdots, x_M^{(0)}\right]^\top$ 开始, 按照下标顺序依次采样：

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

其中 $x_m^{(t)}$ 是第 t 次迭代时变量 X_m 的采样.

吉布斯采样

吉布斯采样的每单步采样也构成一个马尔可夫链. 假设每个单步（采样维度为第 m 维) 的状态转移概率 $q(\boldsymbol{x} \mid \boldsymbol{x}')$ 为

$$q(\boldsymbol{x} \mid \boldsymbol{x}') = \begin{cases} \frac{p(\boldsymbol{x})}{p\left(\boldsymbol{x}'_{1m}\right)} & \text{if} \quad \boldsymbol{x}_{\setminus m} = \boldsymbol{x}'_{\setminus m} \\ 0 & \text{otherwise,} \end{cases}$$

其中边际分布 $p\left(\boldsymbol{x}'_{\setminus m}\right) = \sum_{x'_m} p\left(\boldsymbol{x}'\right)$。因此有 $p\left(x'_{\setminus m}\right) = p\left(x_{\setminus m}\right)$, 并可以得到

$$p\left(\boldsymbol{x}'\right) q\left(\boldsymbol{x} \mid \boldsymbol{x}'\right) = p\left(\boldsymbol{x}'\right) \frac{p(\boldsymbol{x})}{p\left(\boldsymbol{x}'_{\setminus m}\right)} = p(\boldsymbol{x}) \frac{p\left(\boldsymbol{x}'\right)}{p\left(\boldsymbol{x}_{\setminus m}\right)} = p(\boldsymbol{x}) q\left(\boldsymbol{x}' \mid \boldsymbol{x}\right).$$

根据第七章细致平衡条件可知, 该采样构成的马尔可夫链的平稳分布为 $p(\boldsymbol{x})$。

27.3.2 深度信念网络

例 4

深度信念网络（*Deep Belief Network*，*DBN*）是一种深层的概率有向图模型，其图结构由多层的节点构成．每层节点的内部没有连接，相邻两层的节点之间为全连接．网络的最底层为可观测变量，其他层节点都为隐变量．最顶部的两层间的连接是无向的，其他层之间的连接是有向的。

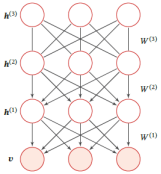


图 6

深度信念网络

概率模型：深度信念网络也是一种生成模型，它所有变量的联合概率可以分解为

$$p\left(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \cdots, \boldsymbol{h}^{(L)}\right) = p\left(\boldsymbol{v} \mid \boldsymbol{h}^{(1)}\right)\left(\prod_{l=1}^{L-2} p\left(\boldsymbol{h}^{(l)} \mid \boldsymbol{h}^{(l+1)}\right)\right) p\left(\boldsymbol{h}^{(L-1)}, \boldsymbol{h}^{(L)}\right)$$

$$= \left(\prod_{l=0}^{L-2} p\left(\boldsymbol{h}^{(l)} \mid \boldsymbol{h}^{(l+1)}\right)\right) p\left(\boldsymbol{h}^{(L-1)}, \boldsymbol{h}^{(L)}\right)$$

其中 $\boldsymbol{h}^{(0)} = \boldsymbol{v}$，$p\left(\boldsymbol{h}^{(l)} \mid \boldsymbol{h}^{(l+1)}\right)$ 为 Sigmoid 型条件概率分布, 定义为

$$p\left(\boldsymbol{h}^{(l)} \mid \boldsymbol{h}^{(l+1)}\right) = \sigma\left(\boldsymbol{a}^{(l)} + \boldsymbol{W}^{(l+1)} \boldsymbol{h}^{(l+1)}\right),$$

其中 $\sigma(\cdot)$ 为按位计算的 Logistic 函数, $\boldsymbol{a}^{(l)}$ 为偏置参数, $\boldsymbol{W}^{(l+1)}$ 为权重参数. 这样, 每一个层都可以看作一个 Sigmoid 信念网络.

深度信念网络

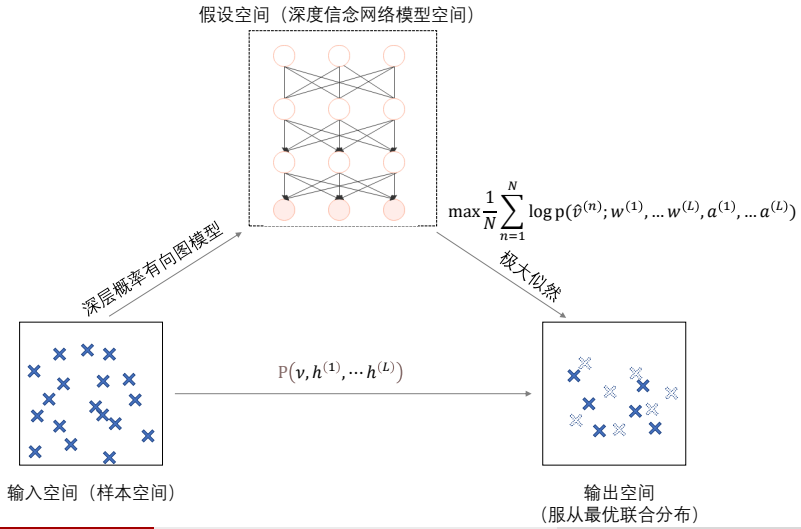
深度信念网络也是通过最大化似然函数来找到最优的参数 $\boldsymbol{W}^{(1)}, \cdots, \boldsymbol{W}^{(L)}, \boldsymbol{a}^{(1)}, \cdots, \boldsymbol{a}^{(L)}$.
给定一组训练样本 $\mathcal{D} = \left\{\hat{\boldsymbol{v}}^{(1)}, \hat{\boldsymbol{v}}^{(2)}, \cdots, \hat{\boldsymbol{v}}^{(N)}\right\}$, 其对数似然函数为

$$\mathcal{L}(\mathcal{D}; \boldsymbol{W}^{(1)}, \cdots, \boldsymbol{W}^{(L)}, \boldsymbol{a}^{(1)}, \cdots, \boldsymbol{a}^{(L)}) = \frac{1}{N}\sum_{n=1}^{N} \log p\left(\hat{\boldsymbol{v}}^{(n)}; \boldsymbol{W}^{(1)}, \cdots, \boldsymbol{W}^{(L)}, \boldsymbol{a}^{(1)}, \cdots, \boldsymbol{a}^{(L)}\right).$$

因此，得到**优化问题**：

$$\max \quad \frac{1}{N}\sum_{n=1}^{N} \log p\left(\hat{\boldsymbol{v}}^{(n)}; \boldsymbol{W}^{(1)}, \cdots, \boldsymbol{W}^{(L)}, \boldsymbol{a}^{(1)}, \cdots, \boldsymbol{a}^{(L)}\right).$$

从空间的角度理解深度信念网络模型



27.3.3 变分自编码器

例 5

变分自编码器（*Variational AutoEncoder*，*VAE*）是一种深度生成模型，其思想是利用神经网络来分别建模两个复杂的条件概率密度函数。

概率模型：变分自编码器其模型结构可以分为两个部分：

1、推断网络：用神经网络来产生变分分布 $q(z;\phi)$，也记为 $q(z \mid x;\phi)$（用简单的分布 q 去近似复杂的分 $p(z|x;\theta)$）

2、生成网络：用神经网络来产生概率分布，估计更好的分布 $p(x|z;\theta)$

变分自编码器

将推断网络和生成网络合并就得到了变分自编码器的整个网络结构：

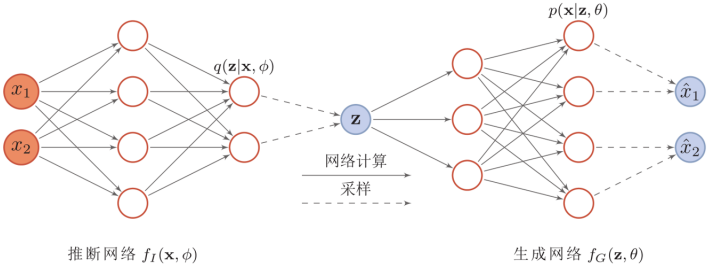


图 8

变分自编码器

推断网络的目标是使得 $q(z \mid x : \phi)$ 能接近真实的后验 $p(z \mid x : \theta)$，需要找到一组网络参数 ϕ^* 来最小化两个分布的 KL 散度，即:

$$\phi^* = \arg\min_{\phi} \mathrm{KL}(q(\boldsymbol{z} \mid \boldsymbol{x}; \phi), p(\boldsymbol{z} \mid \boldsymbol{x}; \theta)).$$

这实际上，等价于

$$\arg\max_{\phi} \mathrm{ELBO}(q, \boldsymbol{x}; \theta, \phi),$$

其中

$$\mathrm{ELBO}(q, \boldsymbol{x}; \theta, \phi) = \mathbb{E}_{\boldsymbol{z} \sim q(z;\phi)} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}; \theta)}{q(\boldsymbol{z}; \phi)} \right]$$

为证据下界。

变分自编码器

上述等价性是因为，对数似然函数 $\log p(\boldsymbol{x};\theta)$ 可以分解为：

$$\begin{aligned}\log p(\boldsymbol{x};\theta) &= \sum_{\boldsymbol{z}} q(\boldsymbol{z};\phi)\log p(\boldsymbol{x};\theta)\\ &= \sum_{\boldsymbol{z}} q(\boldsymbol{z};\phi)(\log p(\boldsymbol{x},\boldsymbol{z};\theta) - \log p(\boldsymbol{z}\mid\boldsymbol{x};\theta))\\ &= \sum_{\boldsymbol{z}} q(\boldsymbol{z};\phi)\log\frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q(\boldsymbol{z};\phi)} - \sum_{\boldsymbol{z}} q(\boldsymbol{z};\phi)\log\frac{p(\boldsymbol{z}\mid\boldsymbol{x};\theta)}{q(\boldsymbol{z};\phi)}\\ &= \mathrm{ELBO}(q,\boldsymbol{x};\theta,\phi) + \mathrm{KL}(q(\boldsymbol{z};\phi)||p(\boldsymbol{z}\mid\boldsymbol{x};\theta)).\end{aligned}$$

因此，推断网络的目标函数可以转换为

$$\begin{aligned}\phi^* &= \arg\min_{\phi}\mathrm{KL}(q(\boldsymbol{z}\mid\boldsymbol{x};\phi),p(\boldsymbol{z}\mid\boldsymbol{x};\theta)) = \arg\min_{\phi}\log p(\boldsymbol{x};\theta) - \mathrm{ELBO}(q,\boldsymbol{x};\theta,\phi)\\ &= \arg\max_{\phi}\mathrm{ELBO}(q,\boldsymbol{x};\theta,\phi),\end{aligned}$$

变分自编码器

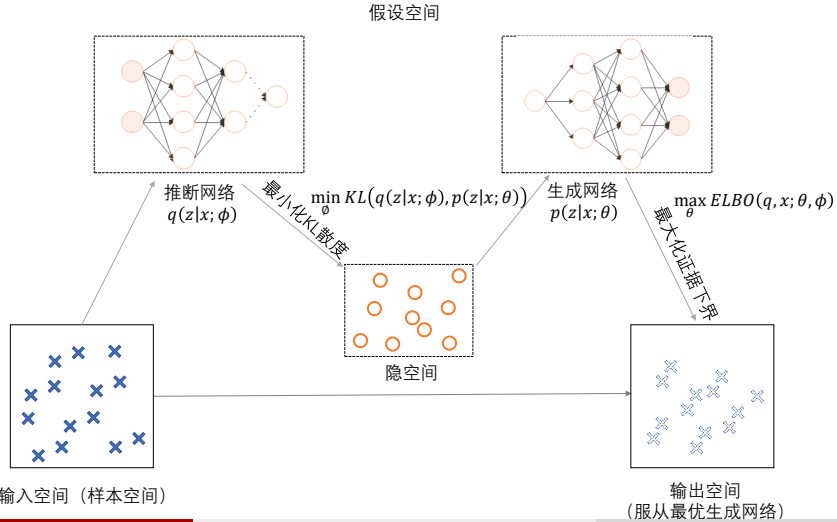
生成网络的目标：生成网络 $f_G(z;\theta)$ 的目标是找到一组网络参数 θ^* 来最大化证据下界 $\mathrm{ELBO}(q,\boldsymbol{x};\theta,\phi)$, 从而最大化对数似然，即:

$$\theta^* = \arg\max_{\theta} \mathrm{ELBO}(q,\boldsymbol{x};\theta,\phi)$$

结合上述公式，推断网络和生成网络的目标都为最大化证据下界 $\mathrm{ELBO}(q,\boldsymbol{x};\theta,\phi)$ 因此，变分自编码器的**优化问题**：

$$\max_{\theta,\phi} \mathrm{ELBO}(q,\boldsymbol{x};\theta,\phi) = \max_{\theta,\phi} \mathbb{E}_{\boldsymbol{z}\sim q(z;\phi)}\left[\log \frac{p(\boldsymbol{x}\mid \boldsymbol{z};\theta)p(\boldsymbol{z};\theta)}{q(\boldsymbol{z};\phi)}\right]$$

从空间的角度理解变分自编码器模型



27.3.4 生成对抗网络

例 6

生成对抗网络（*Generative Adversarial Networks*，*GAN*）是通过对抗训练的方式来使得生成网络产生的样本服从真实数据分布．在生成对抗网络中，有两个网络进行对抗训练．一个是判别网络，目标是尽量准确地判断一个样本是来自于真实数据还是由生成网络产生；另一个是生成网络，目标是尽量生成判别网络无法区分来源的样本.

判别网络和生成网络

判别网络（Discriminator Network）$D(x;\phi)$ 的目标是区分出一个样本 x 是来自于真实分布 $p_r(x)$ 是来自于生成模型 $p_\theta(x)$, 因此判别网络实际上是一个二分类的分类器．用标签 $y=1$ 来表示样本来自真实分布，$y=0$ 表示样本来自生成模型，判别网络 $D(x;\phi)$ 的输出为 x 属于真实数据分布的概率. 因此，判别网络的目标函数可以建模为最小化交叉熵，即

$$\min_{\phi} -\left(\mathbb{E}_x[y\log p(y=1\mid \boldsymbol{x})+(1-y)\log p(y=0\mid \boldsymbol{x})]\right)$$

生成网络（Generator Network）$G(\boldsymbol{z};\theta)$ 的目标刚好和判别网络相反，即让判别网络将自己生成的样本判别为真实样本。因此，

$$\max_{\theta}\left(\mathbb{E}_{\boldsymbol{z}\sim p(z)}[\log D(G(\boldsymbol{z};\theta);\phi)]\right)=\min_{\theta}\left(\mathbb{E}_{\boldsymbol{z}\sim p(z)}[\log(1-D(G(\boldsymbol{z};\theta);\phi))]\right).$$

上面的这两个目标函数是等价的．但是在实际训练时，一般使用前者，因为其梯度性质更好。

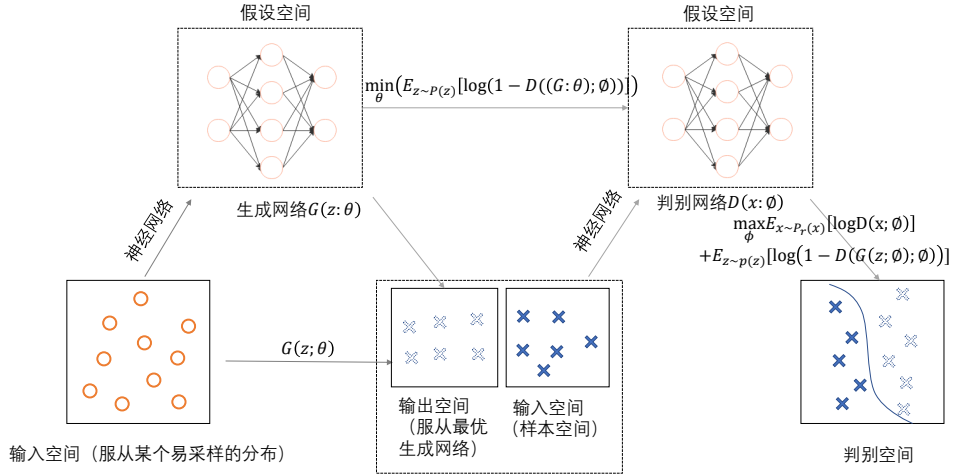
生成对抗网络

把判别网络和生成网络合并为一个整体, 将整个生成对抗网络的目标函数看作**最小化最大优化问题**:

$$\min_{\theta}\max_{\phi}\left(\mathbb{E}_{\boldsymbol{x}\sim p_r(\boldsymbol{x})}[\log D(\boldsymbol{x};\phi)]+\mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})}[\log(1-D(\boldsymbol{x};\phi))]\right)$$

$$=\min_{\theta}\max_{\phi}\left(\mathbb{E}_{\boldsymbol{x}\sim p_r(\boldsymbol{x})}[\log D(\boldsymbol{x};\phi)]+\mathbb{E}_{\boldsymbol{z}\sim p(z)}[\log(1-D(G(\boldsymbol{z};\theta);\phi))]\right).$$

从空间的角度理解生成对抗网络模型



27.3.5 自回归生成模型

许多数据是以序列的形式存在，如声音、语言、视频、DNA 序列或其他的时序数据等。序列数据有两个特点：（1）样本是变长的；（2）样本空间非常大。

例 7

给定一个序列样本 $x_{1:T} = x_1, x_2, \cdots, x_T$，其概率为 $p(x_{1:T})$，若在序列建模中，每一步都需要将前面的输出作为当前步的输入，也即是一种自回归的方式，称这样的序列概率模型为自回归生成模型。

概率模型：根据概率乘法公式，序列 $x_{1:T}$ 的概率可以写为

$$p(x_{1:T}) = p(x_1)p(x_2|x_1)p(x_3|x_{1:2}) \cdots p(x_T|x_{1:(T-1)}) = \prod_{t=1}^{T} p(x_t|x_{1:(t-1)}) \tag{20}$$

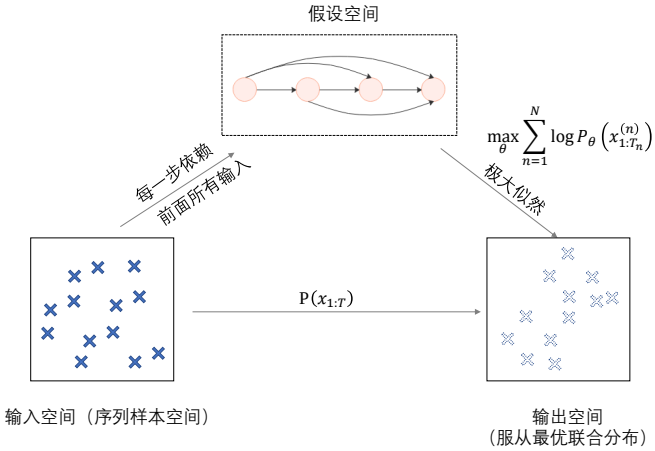
其中 $x_t \in \mathbb{V}, t \in (1, ..., T)$ 为词表 $\mathbb{V}$ 中的一个词，$p(x_1|x_0) = p(x_1)$。序列数据的概率密度估计问题可变为单变量条件概率估计问题，即给定 $x_{1:(t-1)}$ 时 x_t 的条件概率 $p(x_t|x_{1:(t-1)})$。

自回归生成模型

给定 N 个序列数据 $\{x_{1:T_n}^{(n)}\}_{n=1}^{N}$，序列概率模型需要学习一个模型 $p_\theta(x|x_{1:(t-1)})$ 来最大化整个数据集的对数似然函数，即为如下**优化问题**:

$$\max_{\theta} \sum_{n=1}^{N} \log p_\theta(x_{1:T_n}^{(n)}) = \max_{\theta} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \log p_\theta(x_t^{(n)}|x_{1:(t-1)}^{(n)}) \tag{21}$$

从空间的角度理解自回归模型



27.4.1 强化学习

例 8

强化学习既不是监督学习，也不是无监督学习。一般会借助于“智能体”和“环境”两个概念对其进行表述，即智能体通过与环境的交互，根据获得的奖励信息进行学习。

在交互的过程中，智能体通常能观察到环境的信息，这里简记为状态 s_t。然后，它会根据当前的策略执行动作 a_t。环境根据其内在的规律，达到一个新的状态 s_{t+1}，并且给出奖励 r_{t+1}。整个系统以这样的过程不断地持续进行，智能体也在不断地优化其执行策略。

强化学习

概率模型：与监督学习不同的是，在强化学习中，并非学习条件概率分布 $P(Y \mid X)$ 或者联合概率分布 $P(X, Y)$。因此，这并非大家所探讨的一般意义上的概率模型。若假定状态空间为 $\mathcal{S} = \{1, 2, \cdots, S\}$ 和动作空间 $\mathcal{A} = \{1, 2, \cdots, A\}$。现用 $\Delta_{\mathcal{A}}$ 表示在集合 $\mathcal{A}$ 上的概率分布构成的集合，则强化学习的目标是学习出的概率模型（策略）为 $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$。

强化学习

损失函数：通常在强化学习里面称之为**收益或回报**，即累积的奖励。若假定智能体与环境交互的一条轨迹为:

$$\tau = \{s_0, a_0, r_1, s_1, a_1, r_2, s_2, \cdots\}.$$

则累计奖励（带折扣因子 γ）可表示为：

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t.$$

由于在同一策略下的轨迹并不是一成不变的，它是一个随机时序序列。因此，回报函数为一个期望值：$E_{\tau\sim\pi}[R_(\tau)]$。

因此，强化学习即为求解如下**优化问题**：

$$\max_{\pi} E_{\tau\sim\pi}[R(\tau)].$$

本讲小结

机器学习中的概率模型

- 生成模型：联合分布
- 判别模型：条件概率分布
- 贝叶斯技巧和图模型

机器学习中的概率模型

- 浅层的概率模型
- 深层的概率模型
- 强化学习中概率模型

浅层模型：决策树模型、逻辑斯蒂回归模型、最大熵模型。

深层模型：受限玻尔兹曼机、深度信念网络、变分自编码器、生成对抗网络、自回归生成模型。

求解思路：极大似然估计为主。