

机器学习方法概论

1. 机器学习的对象是：具有一定的统计规律的数据。
2. 机器学习根据任务类型，可以划分为：
 - 监督学习任务：从已标记的训练数据来训练模型。主要分为：分类任务、回归任务、序列标注任务。
 - 无监督学习任务：从未标记的训练数据来训练模型。主要分为：聚类任务、降维任务。
 - 半监督学习任务：用大量的未标记训练数据和少量的已标记数据来训练模型。
 - 强化学习任务：从系统与环境的大量交互知识中训练模型。
3. 机器学习根据算法类型，可以划分为：
 - 传统统计学习：基于数学模型的机器学习方法。包括 `SVM`、逻辑回归、决策树等。
这一类算法基于严格的数学推理，具有可解释性强、运行速度快、可应用于小规模数据集的特点。
 - 深度学习：基于神经网络的机器学习方法。包括前馈神经网络、卷积神经网络、递归神经网络等。
这一类算法基于神经网络，可解释性较差，强烈依赖于数据集规模。但是这类算法在语音、视觉、自然语言等领域非常成功。
4. 没有免费的午餐定理(No Free Lunch Theorem:NFL)：对于一个学习算法 `A`，如果在某些问题上它比算法 `B` 好，那么必然存在另一些问题，在那些问题中 `B` 比 `A` 更好。
因此不存在这样的算法：它在所有的问题上都取得最佳的性能。因此要谈论算法的优劣必须基于具体的学习问题。

一、基本概念

1.1 特征空间

1. 输入空间：所有输入的可能取值；输出空间：所有输出的可能取值。
特征向量表示每个具体的输入，所有特征向量构成特征空间。
2. 特征空间的每一个维度对应一种特征。
3. 可以将输入空间等同于特征空间，但是也可以不同。绝大多数情况下，输入空间等于特征空间。
模型是定义在特征空间上的。

1.2 样本表示

1. 通常输入实例用 \vec{x} 表示，真实标记用 \hat{y} 表示，模型的预测值用 \hat{y} 表示。
具体的输入取值记作 $\vec{x}_1, \vec{x}_2, \dots$ ；具体的标记取值记作 $\tilde{y}_1, \tilde{y}_2, \dots$ ；具体的模型预测取值记作 $\hat{y}_1, \hat{y}_2, \dots$ 。
2. 所有的向量均为列向量，其中输入实例 \vec{x} 的特征向量记作（假设特征空间为 n 维）：

$$\vec{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix}$$

这里 $x^{(i)}$ 为 \vec{x} 的第 i 个特征的取值。第 i 个输入记作 \vec{x}_i ，它的意义不同于 $x^{(i)}$ 。

- 训练数据由输入、标记对组成。通常训练集表示为： $\mathbb{D} = \{(\vec{x}_1, \tilde{y}_1), (\vec{x}_2, \tilde{y}_2), \dots, (\vec{x}_N, \tilde{y}_N)\}$ 。
 - 输入、标记对又称作样本点。
 - 假设每对输入、标记对是独立同分布产生的。
- 输入 \vec{x} 和标记 \tilde{y} 可以是连续的，也可以是离散的。
 - \tilde{y} 为连续的：这一类问题称为回归问题。
 - \tilde{y} 为离散的，且是有限的：这一类问题称之为分类问题。
 - \vec{x} 和 \tilde{y} 均为序列：这一类问题称为序列标注问题。

二、监督学习

2.1 监督学习

- 监督学习中，训练数据的每个样本都含有标记，该标记由人工打标，所以称之为 **监督**。
- 监督学习假设输入 \vec{x} 与标记 \tilde{y} 遵循联合概率分布 $p(\vec{x}, y)$ ，训练数据和测试数据依联合概率分布 $p(\vec{x}, y)$ 独立同分布产生。

学习过程中，假定这个联合概率分布存在，但是具体定义未知。

- 监督学习的目的在于学习一个由输入到输出的映射，该映射由模型表示。

模型属于由输入空间到输出空间的映射的集合，该集合就是解空间。解空间的确定意味着学习范围的确定。

- 监督学习的模型可以为概率模型或者非概率模型：

- 概率模型由条件概率分布 $p(y | \vec{x})$ 表示。
- 非概率模型由决策函数 $y = f(\vec{x})$ 表示。

- 监督学习分为学习和预测两个过程。

给定训练集 $\mathbb{D} = \{(\vec{x}_1, \tilde{y}_1), (\vec{x}_2, \tilde{y}_2), \dots, (\vec{x}_N, \tilde{y}_N)\}$ ，其中 $\vec{x}_i \in \mathcal{X}$ 为输入值， $\tilde{y}_i \in \mathcal{Y}$ 是标记值。假设训练数据与测试数据是依据联合概率分布 $p(\vec{x}, y)$ 独立同分布的产生的。

- 学习过程：在给定的训练集 \mathbb{D} 上，通过学习训练得到一个模型。该模型表示为条件概率分布 $p(y | \vec{x})$ 或者决策函数 $y = f(\vec{x})$
- 预测过程：对给定的测试样本 \vec{x}_{test} ，给出其预测结果：
 - 对于概率模型，其预测值为： $\hat{y}_{test} = \arg_y \max p(y | \vec{x}_{test})$
 - 对于非概率模型，其预测值为： $\hat{y}_{test} = f(\vec{x}_{test})$

- 可以通过无监督学习来求解监督学习问题 $p(y | \vec{x})$ ：

- 首先求解无监督学习问题来学习联合概率分布 $p(\vec{x}, y)$
- 然后计算： $p(y | \vec{x}) = \frac{p(\vec{x}, y)}{\sum_{y'} p(\vec{x}, y')}$ 。

2.2 生成模型和判别模型

- 监督学习又分为生成方法和判别方法，所用到的模型分别称为生成模型和判别模型。
- 生成方法：通过数据学习联合概率分布 $p(\vec{x}, y)$ ，然后求出条件概率分布 $p(y | \vec{x})$ 作为预测的模型。
即生成模型为：

$$p(y | \vec{x}) = \frac{p(\vec{x}, y)}{p(\vec{x})}$$

- 生成方法的优点：能还原联合概率分布 $p(\vec{x}, y)$ ，收敛速度快，且当存在隐变量时只能用生成方法。
- 生成方法有：朴素贝叶斯法，隐马尔可夫链。

3. 判别方法：直接学习决策函数 $f(\vec{x})$ 或者条件概率分布 $p(y | \vec{x})$ 的模型。

- 判别方法的优点：直接预测，一般准确率更高，且一般比较简化问题。
- 判别方法有：逻辑回归，决策树。

三、机器学习三要素

1. 机器学习三要素：模型、策略、算法。

3.1 模型

1. 模型定义了解空间。监督学习中，模型就是要学习的条件概率分布或者决策函数。

模型的解空间包含了所有可能的条件概率分布或者决策函数，因此解空间中的模型有无穷多个。

- 模型为一个条件概率分布：

解空间为条件概率的集合： $\mathcal{F} = \{p | p(y | \vec{x})\}$ 。其中： $\vec{x} \in \mathcal{X}, y \in \mathcal{Y}$ 为随机变量， \mathcal{X} 为输入空间， \mathcal{Y} 为输出空间。

通常 \mathcal{F} 是由一个参数向量 $\vec{\theta} = (\theta_1, \dots, \theta_n)$ 决定的概率分布族： $\mathcal{F} = \{p | p_{\vec{\theta}}(y | \vec{x}), \vec{\theta} \in \mathbb{R}^n\}$ 。其中： $p_{\vec{\theta}}$ 只与 $\vec{\theta}$ 有关，称 $\vec{\theta}$ 为参数空间。

- 模型为一个决策函数：

解空间为决策函数的集合： $\mathcal{F} = \{f | y = f(\vec{x})\}$ 。其中： $\vec{x} \in \mathcal{X}, y \in \mathcal{Y}$ 为变量， \mathcal{X} 为输入空间， \mathcal{Y} 为输出空间。

通常 \mathcal{F} 是由一个参数向量 $\vec{\theta} = (\theta_1, \dots, \theta_n)$ 决定的函数族： $\mathcal{F} = \{f | y = f_{\vec{\theta}}(\vec{x}), \vec{\theta} \in \mathbb{R}^n\}$ 。其中： $f_{\vec{\theta}}$ 只与 $\vec{\theta}$ 有关，称 $\vec{\theta}$ 为参数空间。

2. 解的表示一旦确定，解空间以及解空间的规模大小就确定了。

如：一旦确定解的表示为： $f(y) = \sum \theta_i x_i = \vec{\theta} \cdot \vec{x}$ ，则解空间就是特征的所有可能的线性组合，其规模大小就是所有可能的线性组合的数量。

3. 将学习过程看作一个在解空间中进行搜索的过程，搜索目标就是找到与训练集匹配的解。

3.2 策略

1. 策略考虑的是按照什么样的准则学习，从而定义优化目标。

3.2.1 损失函数

1. 对于给定的输入 \vec{x} ，由模型预测的输出值 \hat{y} 与真实的标记值 \tilde{y} 可能不一致。此时，用损失函数度量错误的程度，记作 $L(\tilde{y}, \hat{y})$ ，也称作代价函数。

2. 常用损失函数：

- 0-1 损失函数：

$$L(\tilde{y}, \hat{y}) = \begin{cases} 1, & \text{if } \hat{y} \neq \tilde{y} \\ 0, & \text{if } \hat{y} = \tilde{y} \end{cases}$$

- 平方损失函数 MSE： $L(\tilde{y}, \hat{y}) = (\tilde{y} - \hat{y})^2$

- 绝对损失函数 MAE： $L(\tilde{y}, \hat{y}) = |\tilde{y} - \hat{y}|$

- 对数损失函数： $L(\tilde{y}, \hat{y}) = -\log p(\tilde{y} | \vec{x})$ 。

- 其物理意义是：二分类问题的真实分布与模型分布之间的交叉熵。
- 一个简单的解释：因为样本 (\vec{x}, \tilde{y}) 易经出现，所以理论上 $p(\tilde{y} | \vec{x}) = 1$ 。

如果它不为 1，则说明预测存在误差。越远离 1，说明误差越大。

3. 训练时采用的损失函数不一定是评估时的损失函数。但通常二者是一致的。

因为目标是需要预测未知数据的性能足够好，而不是对已知的训练数据拟合最好。

3.2.2 风险函数

1. 通常损失函数值越小，模型就越好。但是由于模型的输入、标记都是随机变量，遵从联合分布 $p(\vec{x}, y)$ ，因此定义风险函数为损失函数的期望：

$$R_{exp} = \mathbb{E}_P [L(\tilde{y}, \hat{y})] = \int_{\mathcal{X} \times \mathcal{Y}} L(\tilde{y}, \hat{y}) p(\vec{x}, y) d\vec{x} dy$$

其中 \mathcal{X}, \mathcal{Y} 分别为输入空间和输出空间。

2. 学习的目标是选择风险函数最小的模型。

3. 求 R_{exp} 的过程中要用到 $p(\vec{x}, y)$ ，但是 $p(\vec{x}, y)$ 是未知的。

实际上如果它已知，则可以轻而易举求得条件概率分布，也就不需要学习。

3.2.3 经验风险

1. 经验风险也叫经验损失。

给定训练集 $\mathbb{D} = \{(\vec{x}_1, \tilde{y}_1), (\vec{x}_2, \tilde{y}_2), \dots, (\vec{x}_N, \tilde{y}_N)\}$ ，模型关于 \mathbb{D} 的经验风险定义为：

$$R_{emp} = \frac{1}{N} \sum_{i=1}^N L(\tilde{y}_i, \hat{y}_i)$$

经验风险最小化 (empirical risk minimization:ERM) 策略认为：经验风险最小的模型就是最优的模型。即：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(\tilde{y}_i, f(\vec{x}_i))$$

2. 经验风险是模型在 \mathbb{D} 上的平均损失。根据大数定律，当 $N \rightarrow \infty$ 时 $R_{emp} \rightarrow R_{exp}$ 。

但是由于现实中训练集中样本数量有限，甚至很小，所以需要经验风险进行矫正。

3. 结构风险是在经验风险上叠加表示模型复杂度的正则化项（或者称之为罚项）。它是为了防止过拟合而提出的。

给定训练集 $\mathbb{D} = \{(\vec{x}_1, \tilde{y}_1), (\vec{x}_2, \tilde{y}_2), \dots, (\vec{x}_N, \tilde{y}_N)\}$ ，模型关于 \mathbb{D} 的结构风险定义为：

$$R_{srn} = \frac{1}{N} \sum_{i=1}^N L(\tilde{y}_i, \hat{y}_i) + \lambda J(f)$$

其中：

- $J(f)$ 为模型复杂度，是定义在解空间 \mathcal{F} 上的泛函。 f 越复杂，则 $J(f)$ 越大。
- $\lambda \geq 0$ 为系数，用于权衡经验风险和模型复杂度。

4. 结构风险最小化 (structurel risk minimization:SRM) 策略认为：结构风险最小的模型是最优的模型。即：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(\tilde{y}_i, f(\tilde{\mathbf{x}}_i)) + \lambda J(f)$$

5. 结构风险最小化策略符合奥卡姆剃刀原理：能够很好的解释已知数据，且十分简单才是最好的模型。

3.2.4 极大似然估计

1. 极大似然估计就是经验风险最小化的例子。

2. 已知训练集 $\mathbb{D} = \{(\tilde{\mathbf{x}}_1, \tilde{y}_1), (\tilde{\mathbf{x}}_2, \tilde{y}_2), \dots, (\tilde{\mathbf{x}}_N, \tilde{y}_N)\}$ ，则出现这种训练集的概率为： $\prod_{i=1}^N p(\tilde{y}_i | \tilde{\mathbf{x}}_i)$ 。

根据 \mathbb{D} 出现概率最大，有：

$$\max \prod_{i=1}^N p(\tilde{y}_i | \tilde{\mathbf{x}}_i) \rightarrow \max \sum_{i=1}^N \log p(\tilde{y}_i | \tilde{\mathbf{x}}_i) \rightarrow \min \sum_{i=1}^N (-\log p(\tilde{y}_i | \tilde{\mathbf{x}}_i))$$

定义损失函数为： $L(\tilde{y}, \hat{y}) = -\log p(\tilde{y} | \tilde{\mathbf{x}})$ ，则有：

$$\min \sum_{i=1}^N (-\log p(\tilde{y}_i | \tilde{\mathbf{x}}_i)) \rightarrow \min \sum_{i=1}^N L(\tilde{y}_i, \hat{y}_i) \rightarrow \min \frac{1}{N} \sum_{i=1}^N L(\tilde{y}_i, \hat{y}_i)$$

即：极大似然估计 = 经验风险最小化。

3.2.5 最大后验估计

1. 最大后验估计就是结构风险最小化的例子。

2. 已知训练集 $\mathbb{D} = \{(\tilde{\mathbf{x}}_1, \tilde{y}_1), (\tilde{\mathbf{x}}_2, \tilde{y}_2), \dots, (\tilde{\mathbf{x}}_N, \tilde{y}_N)\}$ ，假设已知参数 θ 的先验分布为 $g(\theta)$ ，则出现这种训练集的概率为： $\prod_{i=1}^N p(\tilde{y}_i | \tilde{\mathbf{x}}_i) g(\theta)$ 。

根据 \mathbb{D} 出现概率最大：

$$\begin{aligned} \max \prod_{i=1}^N p(\tilde{y}_i | \tilde{\mathbf{x}}_i) g(\theta) &\rightarrow \max \sum_{i=1}^N \log p(\tilde{y}_i | \tilde{\mathbf{x}}_i) + \log g(\theta) \\ &\rightarrow \min \sum_{i=1}^N (-\log p(\tilde{y}_i | \tilde{\mathbf{x}}_i)) + \log \frac{1}{g(\theta)} \end{aligned}$$

定义损失函数为： $L(\tilde{y}, \hat{y}) = -\log p(\tilde{y} | \tilde{\mathbf{x}})$ ；定义模型复杂度为 $J(f) = \log \frac{1}{g(\theta)}$ ；定义正则化系数为 $\lambda = \frac{1}{N}$ 。则有：

$$\begin{aligned} \min \sum_{i=1}^N (-\log p(\tilde{y}_i | \tilde{\mathbf{x}}_i)) + \log \frac{1}{g(\theta)} &\rightarrow \min \sum_{i=1}^N L(\tilde{y}_i, \hat{y}_i) + J(f) \\ &\rightarrow \min \frac{1}{N} \sum_{i=1}^N L(\tilde{y}_i, \hat{y}_i) + \lambda J(f) \end{aligned}$$

即：最大后验估计 = 结构风险最小化。

3.3 算法

1. 算法指学习模型的具体计算方法。通常采用数值计算的方法求解，如：梯度下降法。