

Chapter.1 - 绪论

1.1 引言

什么是机器学习?

机器学习致力于研究图和通过计算的手段, 利用经验来改善系统自身的性能.

机器学习的主要内容, 是关于在计算机上从数据中产生 **模型(model)** 的算法, 即 **学习算法(learning algorithm)**. 有了学习算法, 我们把经验数据提供给它, 它就能基于这些数据产生模型; 在面对新的情况时, 它会给我们提供相应的判断.

1.2 基本术语

在之后会经常出现的名词, 最好能记住其英文. 忘记可以随时回来翻阅.

名词解释

- **数据集(data set)**: 一组记录的集合;
- **示例(instance)**, **样本(sample)**: 每条记录关于一个事件或对象的描述;
- **属性(attribute)**, **特征(feature)**: 反映事件或对象在某方面的表现或性质的事项;
- **属性空间(attribute space)**, **样本空间(sample space)**, **输入空间**: 属性张成的空间;
- **特征向量(feature vector)**: 一个示例也称为一个特征向量;
- **学习(learning)**, **训练(training)**: 从数据中学得模型的过程;
- **训练数据(training data)**: 训练过程中使用的数据;
- **训练样本(training sample)**: 每个样本;
- **训练集(training set)**: 训练样本组成的集合;
- **假设(hypothesis)**: 学得模型对应了关于数据的某种潜在的规律;
- **真相(ground-truth)**: 这种潜在规律自身;
- **学习器(learner)**: 学习过程就是为了找出或逼近真相. 本书有时将模型称为“学习器”(learner);
- **标记(label)**: 这里关于示例结果的信息, 例如“好瓜”;
- **样例(example)**: 拥有了标记信息的示例;
- **测试样本(testing sample)**: 被预测的样本;
- **假设空间**: 假设(hypothesis)组成的空间;
- **版本空间(version space)**: 学习过程是基于有限样本训练集进行的, 因此, 可能有多个假设与训练集一致, 即存在着一个与训练集一致的“假设集合”, 我们称之为版本空间.

通常约定

一般的:

- 我们用 $D = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_m\}$ 表示一个含有 m 个样本的数据集;
- 每个样本由 d 个属性来描述, 则每个样本 $\mathbf{x}_i = (x_{i1}; x_{i2}; \dots; x_{id})$ 是 d 维样本空间 $\mathcal{X}$ 中的一个向量, 我们称 d 为样本 $\mathbf{x}_i$ 的 维数(dimensionality), x_{ij} 为 $\mathbf{x}_i$ 在第 j 个属性上的取值;
 - 注意, 样本空间 $\mathcal{X}$ 是所有可能出现情况的集合, 而数据集 D 可以称为是样本空间 $\mathcal{X}$ 的子集;
 - 注意, 加粗字体变量 $\mathbf{x}_i$ 是一个向量, 手写时可写为 $\vec{x}_i$. 同时, 在未特殊声明的情况下, 我们认为 $(\cdot, \cdot, \cdot)$ 这种以逗号为间隔的向量看作行向量; 而 $(\cdot; \cdot; \cdot)$ 这种以分号为间隔的向量看作列向量.

分类与回归

如果预测的是一个离散值, 此类学习任务称为 分类(classification); 而若预测的是一个连续值, 则此类学习任务称为 回归(regression);

- 其中, 分类又分为 二分类任务(binary classification) 和 多分类任务(multi-class classification);
- 二分类任务(binary classification) 中, 通常称一个类为 正类(positive class), 另一个为 负类(negative class). 一般将重要的分类称为 正类;
- 通常 二分类任务 将标记空间定义为 $\mathcal{Y} = \{-1, +1\}$ 或 $\mathcal{Y} = \{0, 1\}$;
多分类任务 将标记空间定义为 $|\mathcal{Y}| > 2$; 而 回归任务 将标记空间定义为 $\mathcal{Y} = \mathbb{R}$, $\mathbb{R}$ 为实数集.

聚类任务

聚类(clustering): 将训练集中的数据分成若干组, 每组称为一个簇(cluster).

- 在聚类学习中, 通常我们不会对样本进行标记, 也不会知道会出先多少簇或哪些簇, 这些都是由机器学习而生成的;
- 根据训练数据是否拥有标记信息, 可以分为 监督学习(supervised learning) (分类问题, 回归问题) 和 非监督学习(unsupervised learning) (聚类问题).

泛化能力

机器学习的目标是使学得模型能很好地适用于新样本, 而不是仅仅在训练样本上工作得很好.

即使对聚类这样的无监督学习任务, 我们也希望学得簇划分能适用于没有在训练集中出现的样本, 学习模型适用于新样本的能力, 称为泛化能力.

1.3 假设空间

归纳(induction) 与 演绎(deduction) 是科学推断的两大基本手段. 其中, 归纳是从特殊到一般的 泛化(generalization) 过程; 演绎则是从一般到特殊的 特化(specialization) 过程. 从数据样本中学习显然是一种归纳的过程, 因此称为 归纳学习(inductive learning).

归纳学习 分为广义和狭义两种解释:

- 广义来说, 大体相当于从样例中学习;
- 狭义来说, 归纳学习要从训练数据中获取 概念 (concept), 因此也称为 概念学习 或者 概念形成.

假设空间

监督学习 的任务是学习一个模型, 使模型能够对任意给定的输入, 对其相应的输出做出一个好的预测. 模型属于由输入空间到输出空间的映射的集合, 这个集合就是 假设空间(hypothesis space).

简单来讲, 假设空间的概念, 即由输入空间到输出空间的映射的集合, 即输入空间 X 到输出空间 Y 的映射 $f: X \rightarrow Y$ 所构成的集合 (ref: 统计学习方法 P.7).

假设有3个特征, 每个特征可能取值个数分别为3, 2, 2.

此时, 该假设空间的大小为 $(3 + 1) * (2 + 1) * (2 + 1) + 1 = 37$.

要把每个属性取什么值都可以通配符 * 考虑进去, 还要考虑概念不成立 $\emptyset$.

版本空间

现实问题中我们常面临很大的假设空间, 单学习过程中是基于有限样本训练集进行的.

因此, 可能有多个假设与训练集一致, 即存在着一个与训练集一致的假设集合, 我们称之为 版本空间(version space)

1.4 归纳偏好

归纳偏好(inductive bias): 机器学习算法在学习过程中对某种类型假设的偏好. 简单讲就是什么样的模型更好这一问题.

奥卡姆剃刀”(Occam's razor): 常用的, 自然科学中最基本的规原则 - 若有多个假设与观察一致, 则选最简单的那个.

没有免费的午餐定理(NFL)

设一个样本空间 $\mathcal{X}$ 和假设空间 $\mathcal{H}$ 都是离散的. 令 $P(h|X, \mathcal{L}_a)$ 代表算法 $\mathcal{L}_{ote}$ 基于训练数据 X 产生假设 h 的概率, f 表示我们希望学习的真实目标函数.

则算法 $\mathcal{L}_a$ 在 训练集外误差 为:

$$E_{ote}(\mathcal{L}_a|X, f) = \sum_h \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \mathbb{I}(h(\mathbf{x}) \neq f(\mathbf{x})) P(h|X, \mathcal{L}_a) \quad (1.1)$$

其中:

- $E_{ote}(\mathcal{L}_a|X, f)$ 为训练集外的误差期望;

- $\mathbf{x} \in \mathcal{X} - X$ 为训练集外的样本空间;
- $P(\mathbf{x})$ 为 $\mathbf{x}$ 的常见程度;
- $\mathbb{I}(\cdot)$ 为指示函数, 若 $\cdot$ 为真取 1; 反之, $\cdot$ 为假取 0;
- $P(h|X, \mathcal{L}_a)$: 算法 $\mathcal{L}_a$ 基于数据集 X 产生 h 概率.

二分类问题, 且真实目标函数可以是任何函数 $\mathcal{X} \mapsto \{0, 1\}$, 函数空间为 $\{0, 1\}^{|\mathcal{X}|}$. 根据式1.1, 对所有可能的 f 按照均匀分布对误差求和:

$$\sum_f E_{ote}(\mathcal{L}_a|X, f) = \sum_f \sum_h \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \mathbb{I}(h(\mathbf{x}) \neq f(\mathbf{x})) P(h|X, \mathcal{L}_a) \quad (1.2)$$

若 f 均匀分布, 则一半的 f 对 $\mathbf{x}$ 的预测与 $h(\mathbf{x})$ 不一致.

接下来计算式1.2:

交换 Σ 的顺序,

比如: 只有 $\mathbb{I}(f(\mathbf{x}) \neq h(\mathbf{x}))$ 存在 f , 则将其和 $\sum_f$ 移动至最后, 其余同理.

$$\begin{aligned} \sum_f E_{ote}(\mathcal{L}_a|X, f) &= \sum_f \sum_h \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \mathbb{I}(h(\mathbf{x}) \neq f(\mathbf{x})) P(h|X, \mathcal{L}_a) \\ &= \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \sum_h P(h|X, \mathcal{L}_a) \sum_f \mathbb{I}(f(\mathbf{x}) \neq h(\mathbf{x})) \end{aligned}$$

可以发现 $\sum_f \mathbb{I}(f(\mathbf{x}) \neq h(\mathbf{x}))$ 可以直接进行计算:

- 因为函数空间为 $\{0, 1\}^{|\mathcal{X}|}$, 因此特征空间中有 $2^{|\mathcal{X}|}$ 个点.
- 同时, 已知 f 为均匀分布, 有一半的 f 对 $\mathbf{x}$ 的预测与 $h(\mathbf{x})$ 不一致. 因此存在 $\frac{1}{2}$ 的点是正确, 另 $\frac{1}{2}$ 是错误的.

得到: $\sum_f \mathbb{I}(f(\mathbf{x}) \neq h(\mathbf{x})) = \frac{1}{2} \cdot 2^{|\mathcal{X}|} = 2^{|\mathcal{X}|-1}$

$$\begin{aligned} \sum_f E_{ote}(\mathcal{L}_a|X, f) &= \sum_f \sum_h \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \mathbb{I}(h(\mathbf{x}) \neq f(\mathbf{x})) P(h|X, \mathcal{L}_a) \\ &= \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \sum_h P(h|X, \mathcal{L}_a) \sum_f \mathbb{I}(f(\mathbf{x}) \neq h(\mathbf{x})) \\ &= 2^{|\mathcal{X}|-1} \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \sum_h P(h|X, \mathcal{L}_a) \end{aligned}$$

根据性质, $\sum_h P(h|X, \mathcal{L}_a)$, 所有通过算法 $\mathcal{L}_a$ 基于数据集 X 产生的 h 概率之和为 1. 得到: $\sum_h P(h|X, \mathcal{L}_a) = 1$

$$\begin{aligned}
\sum_f E_{ote}(\mathcal{L}_a|X, f) &= \sum_f \sum_h \sum_{x \in \mathcal{X}-X} P(x) \mathbb{I}(h(x) \neq f(x)) P(h|X, \mathcal{L}_a) \\
&= \sum_{x \in \mathcal{X}-X} P(x) \sum_h P(h|X, \mathcal{L}_a) \sum_f \mathbb{I}(f(x) \neq h(x)) \\
&= 2^{|\mathcal{X}|-1} \sum_{x \in \mathcal{X}-X} P(x) \sum_h P(h|X, \mathcal{L}_a) \\
&= 2^{|\mathcal{X}|-1} \sum_{x \in \mathcal{X}-X} P(x) \cdot 1 \\
&= 2^{|\mathcal{X}|-1} \sum_{x \in \mathcal{X}-X} P(x)
\end{aligned}$$

综上, 通过计算式1.2, 我们发现误差期望与学习算法 $\mathcal{L}_a$ 是无关的. 因此对于两个不同的算法 $\mathcal{L}_a$ 与 $\mathcal{L}_b$, 可以得到结论:

$$\sum_f E_{ote}(\mathcal{L}_a|X, f) = \sum_f E_{ote}(\mathcal{L}_b|X, f)$$

即: 无论一个算法多么笨拙, 无论一个算法多么聪明, 他们的期望性能相同.

但是, 现实中并不是这样的, 因为NFL有一个重要前提: 所有问题出现的机会相同(或所有问题都同等重要). 因此通过NFL所了解的, 是需要认识到脱离具体问题, 空泛讨论什么学习算法最好是毫无意义的事情.