机器学习

第一章 绪论

1.1.引言

- 计算机科学是研究关于算法的学问，机器学习是研究关于学习算法的学问

1.2.基本术语

- 数据集(data set) 是一组记录的集合
- 示例(instance)样本(sample) 是对一个事件或对象的描述
- 属性(attribute)特征(feature) 是反映事件或对象在某方面的表现或性质的事项
- 属性值(attribute value) 属性上的取值
- 属性空间(attribute space)样本空间(sample space) 属性张成的空间
- 特征向量(feature vector) 每个示例都可以在样本空间里找到对应的坐标位置，由于空间的每个点也对应一个坐标向量，一个实例也可以叫做一个特征向量
- 维数(dimensionality) 属性的个数
- 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) 训练数据中的具体一个样本
- 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规律
- 真相、真实(ground-truth) 潜在规律本身
- 学习器(learner) 具体学习算法在给定数据和参数空间上的实例化
- 预测(prediction) 模型基于训练样本的得到的某种结果
- 标记(label) 关于样本的结果信息
- 样例(example) 拥有标记的样本
- 标记空间、输出空间(label space) 所有标记的集合
- 有监督任务(supervised learning) 有标记信息的数据集
    1. 分类(classification) 预测结果是离散值
    2. 回归(regression) 预测结果是连续值
- 无监督任务(unsupervised learning) 没有标记信息的数据集
    1. 聚类(clustering) 将训练集分组，每组为一个簇(cluster)，簇可能存在潜在的概念划分
- 测试(testing) 学习模型后使用模型预测的过程
- 测试样本(testing sample) 被预测的样本

- 泛化(generalization) 学得模型适用于新样本的能力

1.3.假设空间

- 假设空间(hypothesis space) 所有假设组成的空间
- 版本空间(version space) 与训练集一致的假设集合
    1. 假设空间按照删除与正例不一致（与反例相同）的假设的搜索方式在某个训练集上得到的结果就是版本空间

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
    1. 例如，在回归学习中，一般认为相似的样本具有相似的输出，我们应该让算法偏向于归纳这种假设
- 没有免费的午餐定理(No Free Lunch Theorem) 所有算法的期望性能和随机胡猜一样

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

> NFL定理证明

第二章 模型评估与选择

2.1.经验误差与过拟合

- 错误率(error rate) 分类错误的样本占样本总数的比例
- 精度(accuracy) 1-错误率
- 误差(error) 学习器在训练集的实际预测输出与样本的真实输出之间的差异
- 训练误差(training error)经验误差(empirical error) 学习器在训练集上的误差
- 泛化误差(generalization error) 学习器在新样本上的误差
    1. 我们希望得到一个泛化误差小的学习器，然而，我们根据训练样本只能得到一个经验误差很小的学习器
- 过拟合(overfitting) 把训练样本自身的一些特点当作了所有潜在样本具有的某种一般性质，导致泛化性能下降
- 欠拟合(underfitting) 对训练样本的一般性质尚未学习好
    1. 一般情况下，学习能力高低分别会导致过拟合和欠拟合，欠拟合可以通过增加扩展分支（决策树）、训练轮数（神经网络），过拟合则是机器学习的主要障碍，过拟合是无法避免的

2.2.评估方法

- 测试集(testing set) 测试学习器对新样本的判别能力
- 测试误差(testing error) 用作泛化误差的近似
    1. 我们一般假设测试样本是从真实样本分布中独立同分布采样而得。测试集应该和训练集互斥

2.2.1.留出法(hold-out)

- 将数据集D划分为互斥的两个集合，其中一个为训练集S，另一个为测试集T，即$D = S \cup T$, $S \cap T = \varnothing$,基于S训练出的模型，用T来评估测试误差，作为泛化误差的估计
    1. 按照采样(sampling)的角度看待数据集的划分过程，则保留类别比例的采样方式通常为分层采样(stratified sampling)
    2. 训练集S过大会导致接近数据集D、测试集T较小，评估结果不够准确；S过小会导致基于S和D训练出模型的差别过大，降低评估结果的保真性(fidelity)
    3. 通常取2/3~4/5的样本作为训练集

2.2.2.交叉验证法(cross validation)

- 将数据集D划分为k个大小相同的互斥子集，即 $D = D_1 \cup D_2 \cup \ldots \cup D_k, D_i \cup D_j = \varnothing (i \neq j)$ 。每个子集D_i 都保持数据分布的一致性，即按照分层采样得到。每次取k -1个作为训练集，其余作为测试集，进行k次训练和测试，返回k个结果的均值
    1. 交叉验证法评估结果依赖于k的取值，因此也叫做“k折交叉验证”(k-fold cross validation)
    2. k通常取值为10
    3. 若数据集D包含m个样本，令k=m，得到交叉验证法的一个特例：留一法(Leave-One-Out)

2.2.3.自助法(bootstrapping)

- 自助法以自助采样法(bootstrap sampling)为基础，在一个含有m个样本的数据集D中，每次随机抽取一个样本拷贝入D' (D' 中有重复的样本)，重复执行m次，使得D' 中也有m个样本
    1. 样本在m次采样始终采样不到的概率为$(1-\frac{1}{m})^m$，极限是 $\lim\limits_{m \to \infty}(1-\frac{1}{m})^m = \frac{1}{e} \approx 0.368$
    2. 使用始终没有被采样的部分作为测试集
    3. 这样的测试结果，叫包外估计(out-of-bagestimate)
    4. 自助法适用于较小和难以划分的数据集，对集成学习有很大好处

2.2.4.调参与最终模型

- 参数的取值范围是在实数范围，参数的选择一般是按照取值范围和步长
    1. 超参数:数目通常在10以内，采用人工设定
    2. 模型的参数:比如神经网络中的参数，采用学习（比如训练轮数）
- 包含m个样本的数据集D，在模型评估选择时，选用了部分数据做训练，另一部分做评估，在学习算法和参数配置选定以后，需要重新将所有的样本，即数据集D进行训练，得到的模型交给用户
- 模型在实际使用的过程中遇到的数据称为测试数据
- 模型评估与选择中用于评估测试的数据叫做验证集(validation set)，比如，研究算法泛化性能时，我们测试集估计实际使用的泛化能力，而把训练数据划分为训练集和验证集，基于验证集进行模型选择和调餐参数

2.3.性能度量(performance measure)

- 性能度量(performance measure) 衡量模型泛化性能的评价标准

---

1. 回归任务的性能度量是均方误差(mean squared error)

$$E(f;D) = \frac{1}{m}\sum_{i=1}^{m}(f(\mathbf{x\_i}) - y_i)^2$$

2. 对于数据分布$\mathcal{D}$和概率密度函数$p(\cdot)$，均方误差的描述为

$$E(f;D)=\int_{\mathbf{x}\sim\mathcal{D}}(f(\mathbf{x})-y)^2p(\mathbf{x})d\mathbf{x}$$

2.3.1.错误率与精度

- 错误率是分类错误的样本数占样本总数的比例
- 精度是分类正确的样本数占样本总数的比例

---

1. 样例集D分类错误率定义为

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\mathbf{x\_i})\neq y_i)$$

精度定义为

$$acc(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\mathbf{x\_i})=y_i)=1-E(f;D)$$

2. 对于数据分布$\mathcal{D}$和概率密度函数$p(\cdot)$，错误率定义为

$$E(f;D)=\int_{\mathbf{x}\sim\mathcal{D}}\mathbb{I}(f(\mathbf{x})\neq y)p(\mathbf{x})d\mathbf{x}$$

精度定义为

$$acc(f;D)=\int_{\mathbf{x}\sim\mathcal{D}}\mathbb{I}(f(\mathbf{x})=y)p(\mathbf{x})d\mathbf{x}=1-E(f;D)$$

2.3.2.查准度(precision)、查全率(recall)与F1

- 样例根据其真实类别和学习器预测类别的组合为真正例(true positive)、假正例(false positive)、真反例(true negative)、假反例(false negative)

---

1. 查准率定义为

$$P=\frac{TP}{TP+FP}$$

2. 查全率定义为

$$R=\frac{TP}{TP+FN}$$

- 查准率P和查全率R是一对矛盾的度量
- 以查准率为纵轴、以查全率为横轴作图，得到查准率-查全率曲线，简称“P-R曲线”，显示曲线的的图称为“P-R图”
- 一个学习器P-R曲线完全“包住”另一个的，则说明前者性能好与后者性能
- 当查准率P=查全率R时的取值，叫做平衡点(Break-Even Point)

- 由于BEP过于简化，更常用的是$F1$度量

---

1. $F1$度量定义为

$$F1=\frac{2\times P\times R}{P+R}=\frac{2\times TP}{\text{样例总数}+TP-TN}$$

2. $F1$是基于查准率和查全率的平均调和平均定义的

$$\frac{1}{F1}=\frac{1}{2}\cdot(\frac{1}{P}+\frac{1}{R})$$

3. F1的一般形式是F_β，F_β定义为

$$F_\beta=\frac{(1+\beta^2)\times P\times R}{(\beta^2\times P)+R}$$

4. 当$\beta=1$时，退化为$F1$；$\beta>1$时，查全率有更大的影响；$\beta<1$时，查准率有更大的影响

- 在n个二分类混淆矩阵上宏查准率、宏查全率和宏F1定义为

---

1. 宏查准率

$$macro-P=\frac{1}{n}\sum_{i=1}^{n}P_i$$

2. 宏查全率

$$macro-R=\frac{1}{n}\sum_{i=1}^{n}R_i$$

3. 宏F1

$$macro-F1=\frac{2\times macro-P\times macro-R}{macro-P+macro-R}$$

- 在n个二分类混淆矩阵上将TP、FP、TN、FN进行平均$\overline{TP}$、$\overline{FP}$、$\overline{TN}$、$\overline{FN}$进而得到微查准率、微查全率和微F1

---

1. 微查准率

$$micro-P=\frac{\overline{TP}}{\overline{TP}+\overline{FP}}$$

2. 微查全率

$$micro-R=\frac{\overline{TP}}{\overline{TP}+\overline{FR}}$$

3. 微F1

$$micro-F1=\frac{2\times micro-P\times micro-R}{micro-P+micro-R}$$

2.3.3.ROC与AUC

- ROC曲线是衡量学习器泛化性能的有力工具，其中ROC曲线的纵轴是真正例率$TPR=\frac{TP}{TP+FN}$、横轴是假正例率$FPR=\frac{FP}{TN+FP}$

---

1. AUC(Area Under ROC Curve)面积可以反映学习器性能

$$\begin{aligned} AUC &= \frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1}-x_i)\cdot(y_i+y_{i+1}) \\ &= 1-\ell_{rank} \end{aligned}$$

2. m^+个正例、m^-个反例，D^+为正例集合、D^-为反例集合排序的损失，AUC和Mann-Whitney U检验等价

$$\ell_{rank}=\frac{1}{m^+m^-}\sum_{m^+\in D^+}\sum_{m^-\in D^-}\left(\mathbb{I}(f(x^+)<f(x^-)+\frac{1}{2}\mathbb{I}(f(x^+)=f(x^-))\right)$$

> ROC曲线画法

2.3.4.代价敏感错误率与代价曲线

- 为了权衡不同类型错误所造成的不同损失，可为错误赋予非均等代价(unequal cost)
- 在考虑非均等代价下，我们希望最小化总体代价(total cost)，以二分类问题为例，

  此时错误率为

$$E(f;D;cost)=\frac{1}{m}\left(\sum_{x_i\in D^+}\mathbb{I}(f(\mathbf{x\_i})\neq y_i)\times cost_{01}+\sum_{x_i\in D^-}\mathbb{I}(f(\mathbf{x\_i})\neq y_i)\times cost_{10}\right)$$

- 在非均等代价下，ROC不能直接反映学习器的期望总体代价，而需要通过代价曲线(cost curve)

---

1. 代价曲线横轴是取值[0,1]的正例概率代价

$$P(+)cost=\frac{p\times cost_{01}}{p\times cost_{01}+(1-p)\times cost_{10}}$$

2. 纵轴是取值为[0,1]归一化代价

$$cost_{norm}=\frac{FNR\times p\times cost_{01}+FPR\times(1-p)\times cost_{10}}{p\times cost_{01}+(1-p)\times cost_{10}}$$

- 每个ROC曲线上的一点转化为代价平面上的一条线段，取所有线段的下界围成的面积为期望的总体代价

2.4.比较检验

- 使用统计假设检验(hypotheis test)进行学习器性能的比较

2.4.1.假设检验

---

1. 在包含m个样本的测试集上，泛化错误率为ϵ的学习器被测得测试错误率$\hat{\epsilon}$的概率是

$$P(\hat{\epsilon};\epsilon)=\begin{pmatrix}m\\ \hat{\epsilon}\times m\end{pmatrix}\epsilon^{\hat{\epsilon}\times m}(1-\epsilon)^{m-\hat{\epsilon}\times m}$$

2. 考虑$\epsilon\leq\epsilon_0$，则在$1-\alpha$的概率内所观测到最大错误率

$$\bar{\epsilon}=max\epsilon\ \ s.t.\ \ \sum_{i=\epsilon_0\times m+1}^{m}\binom{m}{i}\epsilon^i(1-\epsilon)^{m-i}<\alpha$$

3. k个测试错误率的平测试错误率μ和方差σ^2分别为

$$\mu=\frac{1}{k}\sum_{i=1}^{k}\hat{\epsilon}_i\qquad\sigma^2=\frac{1}{k-1}\sum_{i=1}^{k}(\hat{\epsilon}_i-\mu)^2$$

4. k个测试错误率看作泛化错误率ϵ_0的独立采样，变量$\tau_t=\frac{\sqrt{k}(\mu-\epsilon_0)}{\sigma}$服从自由度为$k-1$的$t$分布

2.4.2.交叉验证t检验

- 用于比较不同学习器的性能
- 学习器A和学习器B使用k折交叉验证法得到测试错误率分别为$\epsilon_1^A,\epsilon_2^A,.....,\epsilon_k^A$和$\epsilon_1^B,\epsilon_2^B,.....,\epsilon_k^B$，可以使用k折交叉验证“成对t检验”(paired t-tests)进行比较检验，基本思想是如果两个学习器性能相同，则使用相同数据集测试错误率应该相同，即$\epsilon_i^A=\epsilon_i^B$分别对每对测试错误率求差$\Delta_i=\epsilon_i^A-\epsilon_i^B$，（如果两个学习器性能相同它们的差值应该为零），计算均值μ和方差σ^2，在显著度α下，变量$\tau_t=\left|\frac{\sqrt{k}\mu}{\sigma}\right|$小于临界值$t_{\alpha/2,\,k-1}$则假设不能被拒绝，说明学习器A和学习器B的性能相当；如果不一样，则认为平均错误率小的性能更好
- 以上需要满足测试错误率均为泛化错误率的独立采样，但是通常样本有限，训练集会有重叠，即测试错误率不独立，会过高估计假设成立的概率，因此，可以采用“5×2交叉验证”法5×2交叉验证是作5次2折交叉验证，每次2折交叉验证之前随机将数据打乱，使得5次交叉验证数据划分不重复，对学习器A和B的第i次的两个测试错误率求差，计算平均值$\mu=0.5(\Delta_1^1+\Delta_1^2)$，对每次2折实验的结果都计算出其方差$\sigma_i^2=(\Delta_i^1-\frac{\Delta_i^1+\Delta_i^2}{2})^2+(\Delta_i^2-\frac{\Delta_i^1+\Delta_i^2}{2})^2$，变量$\tau_t=\mu/\sqrt{0.2\sum_{i=1}^{5}\sigma_i^2}$

2.4.3.McNemar检验

- 对于二分类问题，使用留出法可以得到分类器分类结果的差别，如果两个学习器性能相同，则$e_{01}=e_{10}$，那么变量$|e_{01}-e_{10}|$应当服从正态分布。McNemar检验考虑变量$\tau_{\chi^2}=\frac{(|e_{01}-e_{10}|-1)^2}{e_{01}+e_{10}}$在给定显著度$\alpha$，如果变量小于临界值$\chi_\alpha^2$则假设不能被拒绝，说明学习器A和学习器B的性能相当；如果不一样，则认为平均错误率小的性能更好

2.4.4.Friedman检验与Nemenyi后续检验

- 在一组数据集上对多个算法性能比较，既可以在每个数据集上分别列出不同算法两两比较，也可以使用基于算法排序的Friedman检验就可以满足我们的要求

---

1. 使用留出法或者交叉验证法得到每个算法在每个数据集上的测试结果，然后根据测试的性能进行排序，并计算平均序值
2. 使用Friedman检验算法性能假设N个数据集，k个算法，令r_i表示第i个算法的平均序值，变量

$$\tau_{\chi^2} = \frac{k-1}{k} \cdot \frac{12N}{k^2-1} \sum_{i=1}^{k} \Big(r_i - \frac{k+1}{2} \Big)^2$$
$$= \frac{12N}{k(k+1)} \Big(\sum_{i=1}^{k} r_i^2 - \frac{k(k+1)^2}{4} \Big)$$

但是“原始Friedman检验”过于保守，通常使用变量

$$\tau_F = \frac{(N-1)\tau_{\chi^2}}{N(k-1) - \tau_{\chi^2}}$$

3. 若性能显著不同，此时需要进行后续检验(post-hoc test)进行进一步检验，这里我们使用Nemenyi后续检验，Nemenyi检验计算出平均序值差别的临界值域为

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

4. 如果两个算法的平均序值之差超出临界值域CD，则认为两个算法的性能有显著差异

2.5.偏差与方差

- 偏差-方差分解(bias-variance decomposition)是解释学习算法泛化性能的一种工具

---

1. 回归任务的算法期望为

$$\bar{f}(x) = \mathbb{E}_D[f(x;D)]$$

2. 使用样本相同的不同算法产生的方差为

$$var(x) = \mathbb{E}_D\Big[\big(f(x;D) - \bar{f}(x) \big)^2 \Big]$$

3. 噪声为

$$\varepsilon^2 = \mathbb{E}_D\Big[(y_D - y)^2 \Big]$$

4. 期望输出和真实标记输出的偏差

$$bias^2(x) = (\bar{f}(x) - y)^2$$

5. 算法的期望泛化误差（假定噪声期望为0）

$$E(f;D) = \mathbb{E}_D\big[(f(x;D) - y_D)^2 \big] = bias^2(x) + var(x) + \varepsilon^2$$

- 偏差度量了学习算法期望预测与真实结果的偏离程度，即刻画了学习算法本身的拟合能力
- 方差度量了同样大小训练集的变动导致的学习性能的变化，即刻画数据扰动所造成的影响
- 噪声则表达了当前任务学习算法所能达到的期望泛化误差的下界，即刻画问题本身的难度
- 偏差和方差是有冲突的，称为偏差-方差窘境(bias-variance dilemma)