

西瓜书复习笔记02

- **错误率：**

分类错误的样本数占样本总数的比例称为错误率 ($E = a/m$)

- **精度：**

精度=1-错误率

- **训练误差：**

学习器在训练集上的误差

- **泛化误差：**

学习器在测试集上的误差

- **过拟合：**

- 什么是过拟合：

学习能力过于强大，把训练样本所包含的不太一般的特性也学到了。

- 过拟合的具体表现：

随着训练过程进行，模型复杂度增加，在训练集上的错误率减小，但在验证集上的错误率增大。

- 什么原因导致过拟合：

- 模型太复杂

- 数据噪声大

(大到模型过分记住了噪声，反而忽略了输入输出间的关系。)

- 训练数据太少

(训练集与测试集特征分布不一致。)

- 权值学习迭代次数过多

- 过拟合解决方法：

- 增加训练集

(例如数据增广)

- 调小模型复杂度 (例如减小树的深度)

- 正则化

(L2: 目标函数增加所有权重参数的平方和，迫使所有尽可能趋向于0。因为过拟合时某些w非常大，L2的加入惩罚了权重变大的趋势。)

(L1: 目标函数增加所有权重参数的绝对值之和，迫使更多w为0。它能实现特征选择，将无用的特征去除，也就是对应的权值为0。但防止过拟合的效果不如L2好。)

- dropout

(让神经元以p的概率激活，1-p的概率失活，使得每一个w都随机参与训练。)

- 逐层归一化

(给神经网络的每一层输出都做一次归一化，使下一层输入接近高斯分布，避免了下一层

w过大导致的以偏概全。)

- 提前终止

(设置迭代次数, 或者设置阈值高于某个精度就停止。)

- 集成学习

(bagging 平均、投票多个模型的结果降低方差, boosting能减少方差与偏差。)

- 欠拟合:

学习能力不足, 导致模型在训练集和测试集上表现都很差。

- 评估方法与模型选择:

- 验证集:

在评估和模型选择中, 用于评估测试的数据集。

- 留出法:

使用分层采样(保持样本类别比例相似)的方法把数据集划分两个互斥子集, 一个作为训练集(训练模型), 另一个作为测试集(验证集, 评估泛化误差)。

- 交叉验证法:

交叉验证先将数据集划分为k个大小相似的互斥子集, 每个自己都尽可能保持数据分布的一致性(分层采样), 然后每次用k-1个子集的并集作为训练集, 余下的那个子集作为测试集(验证集), 这样就可以获得k组训练/测试集(验证集), 训练k个模型, 最终返回k个测试结果的均值。

缺点: 数据集比较大的时候, 计算开销很大。

- 自助法:

自助采样(有放回的随机采样): 假设要从数据集中抽取m个样本, 每次抽一个放在新的数据集中, 在将该样本放回原有数据集, 重复m次。这样当m足够大时, 约有36.8%的样本没有抽到, 没抽到的样本可以作为测试集(验证集)。

$$\lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m = \frac{1}{e} \approx 0.368$$

好处: 使得数据保持一定的联系, 又有一定的差异。

- 调参:

在进行模型评估与选择时, 除了要对适用学习算法进行选择, 还需要对算法参数进行设定, 这就是参数调节或者调参。

- 性能度量:

衡量模型泛化能力的评价标准

- 均方误差:

回归任务最常用的性能度量。

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

- 错误率与精度：
分类任务最常用的性能度量。
- 查准率、查全率和F1：
分类任务中，错误率和精度往往不能满足所有任务需求
 - 混淆矩阵：

真实情况	预测为正例	预测为反例
正例	TP（真正例）	FN（假反例）
反例	FP（假正例）	TN（真反例）

- 查准率（准确率）：
模型认为对的样例里，有多少真是对的。

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

TP比混淆矩阵的第一列的和

- 查全率（召回率）：
在所有对的样例里面，模型找出了真是多少对的。

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

TP比混淆矩阵的第一行的和

- 查准率和查全率的关系：
查准率和查全率会反向变动。查准率高时，查全率往往低；查全率高，查准率往往偏低。
随着阈值的增长，查准率上升，查全率下降。
- 查准率和查全率的应用：
在推荐系统中，为了尽可能少的打扰用户，更希望推荐内容是用户感兴趣的，此时查准率更重要；
在风险检测系统中，为了尽可能的检测潜在的风险，查全率更重要。
- F1度量：
查准率和查全率的调和平均数，更重视两者的最小值。

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

- 宏F1：
如果进行多次训练/测试，或是在多个数据集上进行训练/测试，每次得到一个混淆矩阵。

可以在n个混淆矩阵上综合考察查准率和查全率，估算全局性能。

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

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

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

◦ ROC与AUC:

▪ 真正率:

$$\text{TPR} = \frac{TP}{TP + FN}$$

TP比混淆矩阵第一行的和

▪ 假正率:

$$\text{FPR} = \frac{FP}{TN + FP}$$

TP比混淆矩阵第二行的和

▪ ROC曲线:

根据学习器的预测结果，把阈值从0变到最大，随着阈值的增大，在这一过程中，每次计算出两个重要量的值（TPR和FPR）。构成横轴为FPR，纵轴为TPR的曲线。

意义：ROC曲线能很容易的查出任意阈值对学习器的泛化性能的影响。

▪ AUC:

AUC是ROC曲线面积下的和，用于解决两ROC曲线相交的问题，只能用于二分类模型评价。

$$\text{AUC} = \frac{1}{2} \sum_{i=1}^{m-1} (x_{i+1} - x_i) \cdot (y_i + y_{i+1})$$

(梯形面积上底加下底乘高除以2)

意义：AUC表示预测的正例排在负例前面的概率。所以AUC反映的是分类器对样本的排序能力。

- 归一化：

- 什么是归一化：

将数值规约到[0,1]区间内。

- 怎么归一化：

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- 归一化的好处：

- 归一化后加快了梯度下降求最优解的速度。

（假设有两个特征一个范围（0，1）另一个范围（0，10000），它们的损失等高线是椭圆形，归一化后等高线就显得很圆，梯度下降会较快收敛。）

- 归一化可能提高精度。

（一些需要计算样本之间距离的分类器（例如KNN）。如果一个特征范围特别大的话，那么距离会主要取决于这个特征。）

- 线性模型（例如LR）有时为什么要归一化？

同上，有利于梯度下降。

- 树状结构为什么不需要归一化：

因为数值缩放并不会影响分裂点的位置，对模型结构不会产生影响。

树模型归一化没有意义，因为树模型不能进行梯度下降。

- 归一化不是标准化，两者之间有很大的差异

- 偏差-方差分解

- 偏差：

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

期望输出与真实标记之间的差别。

- 方差：

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

不同训练集产生的预测输出与期望输出得差别

- 对学习算法的期望泛化错误率进行拆解，有：

$$E(f; D) = \text{bias}^2(\mathbf{x}) + \text{var}(\mathbf{x}) + \epsilon^2$$