机器学习导论
习题一参考答案

2020 年 3 月 22 日

Problem 1

若数据包含噪声，则假设空间中有可能不存在与所有训练样本都一致的假设，此时的版本空间是什么？在此情形下，试设计一种归纳偏好用于假设选择。

Solution. 此处用于写解答 (中英文均可)

此时的版本空间是空集。

归纳偏好: 采用与训练样本一致数量最多的假设 (言之有理即可)。批改中遇到的错误: 本题很多同学回答的答案是修改原数据以去除特征相同标签不同的样本，修改原始数据并不是一种归纳偏好。

Problem 2 [编程]

现有 500 个测试样例,其对应的真实标记和学习器的输出值如表1所示 (完整数据见 data.csv 文件)。该任务是一个二分类任务，1 表示正例，0 表示负例。学习器的输出越接近 1 表明学习器认为该样例越可能是正例，越接近 0 表明学习器认为该样例越可能是负例。

表 1: 测试样例表

样本	x_1	x_2	x_3	x_4	x_5	...	x_{496}	x_{497}	x_{498}	x_{499}	x_{500}
标记	1	1	0	0	0	...	0	1	0	1	1
输出值	0.206	0.662	0.219	0.126	0.450	...	0.184	0.505	0.445	0.994	0.602

(1) 请编程绘制 P-R 曲线

(2) 请编程绘制 ROC 曲线，并计算 AUC

本题需结合关键代码说明思路，并贴上最终绘制的曲线。建议使用 Python 语言编程实现。(预计代码行数小于 100 行)

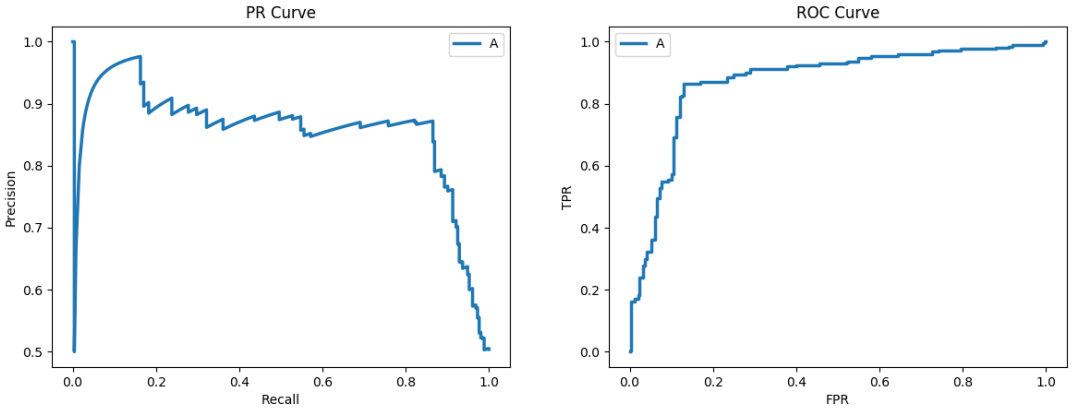
提示:

- 需要注意数据中存在输出值相同的样例。

- 在 Python 中，数值计算通常使用 Numpy, 表格数据操作通常使用 Pandas, 画图可以使用 Matplotlib (Seaborn), 同学们可以通过上网查找相关资料学习使用这些工具。未来同学们会接触到更多的 Python 扩展库，如集成了众多机器学习方法的 Sklearn, 深度学习工具包 Tensorflow, Pytorch 等。

Solution. 此处用于写解答 (中英文均可)

本题主要考察对 P-R 曲线和 ROC 曲线的理解与代码实现能力 (参考代码见 py 文件)。结果曲线如图所示。AUC = 0.873719918。



(a) P-R curve (b) ROC curve

图 1: P-R 曲线与 ROC 曲线

数据中存在输出值相同的样例，这些样例的 label 都是相同的，所以在本题中不会出现斜线。但是为了考虑一般性，代码实现中应考虑出现斜线的情况。核心代码见图 2, 代码中第八行的 if 语句目的是处理输出值相同的样例。批改中遇到的错误: 1. 有同学删除了多的输出值相同的样例 2. 绝大多数同学代码中未考虑输出值相同情况

```
    for i, y in enumerate(reversed(label)):
        if y == 1:
            TP += 1.
            FN -= 1.
        else:
            FP += 1.
            TN -= 1.
        if i < len(output) - 1 and output[i+1] == output[i]:
            continue
        precision = TP/(TP + FP)
        recall = TP/(TP+FN)
        TPR = TP/(TP+FN)
        FPR = FP/(TN+FP)
        PR_list.append((recall, precision))
        ROC_list.append((FPR, TPR))
```

图 2: 核心代码

Problem 3

对于有限样例，请证明

$$\mathrm{AUC} = \frac{1}{m^+m^-}\sum_{x^+\in D^+}\sum_{x^-\in D^-}\left(\mathbb{I}(f(x^+) > f(x^-)) + \frac{1}{2}\mathbb{I}(f(x^+) = f(x^-))\right)$$

Proof. 此处用于写证明 (中英文均可)
考虑 ROC 曲线绘制过程。设前一个样例在 ROC 曲线上的坐标为 (x, y),

(1) 若当前样例为真正例，则对应在 ROC 曲线上的坐标为 $(x, y + \frac{1}{m^+})$;

(2) 若当前样例为假正例，则对应在 ROC 曲线上的坐标为 $(x + \frac{1}{m^-}, y)$。

由此可知，考虑任何一对正例和负例对，

(1) 若其中正例预测值小于反例，则 x 先增加，y 后增加，曲线下方的面积 (即 AUC) 将不会因此而增加;

(2) 若其中正例预测值大于反例，则 y 值会先增加，x 后增加，曲线下方的面积 (即 AUC) 将增加一个矩形格子，其面积为 $\frac{1}{m^+m^-}$;

(3) 若一个正例预测值等于反例，对应标记点 x, y 坐标值同时增加，曲线下方的面积 (即 AUC) 将增加一个三角形，其面积为 $\frac{1}{2}\frac{1}{m^+m^-}$.

考虑所有正例和负例对，AUC 的面积即为曲线下方的面积，根据上述情况进行累加，则有

$$\mathrm{AUC} = \frac{1}{m^+m^-}\sum_{x^+\in D^+}\sum_{x^-\in D^-}\left(\mathbb{I}(f(x^+) > f(x^-)) + \frac{1}{2}\mathbb{I}(f(x^+) = f(x^-))\right)$$

西瓜书的辅助参考资料南瓜书[1]中也有对该题详细的解释，可供参考。

□

[1] https://datawhalechina.github.io/pumpkin-book/#/chapter2/chapter2?id=227

Problem 4 [编程]

在数据集 D_1, D_2, D_3, D_4, D_5 运行了 A, B, C, D, E 五种算法，算法比较序值表如表2所示:

表 2: 算法比较序值表

数据集	算法 A	算法 B	算法 C	算法 D	算法 E
D_1	2	3	1	5	4
D_2	5	4	2	3	1
D_3	4	5	1	2	3
D_4	2	3	1	5	4
D_5	3	4	1	5	2
平均序值	3.2	3.8	1.2	4	2.8

使用 Friedman 检验 ($\alpha = 0.05$) 判断这些算法是否性能都相同。若不相同，进行 Nemenyi 后续检验 ($\alpha = 0.05$)，并说明性能最好的算法与哪些算法有显著差别。本题需编程实现 Friedman 检验和 Nemenyi 后续检验。(预计代码行数小于 50 行)

Solution. 此处用于写解答 (中英文均可)
按照书本内容实现 Friedman 检验和 Nemenyi 检验即可。
Friedman 检验核心代码见图 3
$k = 5, N = 5$。由 42 页式 (2.34) 与 (2.35) 可得:

```python
def f(df: pd.DataFrame):
    k = len(df.columns)
    N = len(df.index)
    S = 0
    for i in range(k):
        performance = df.iloc[:, i].values
        mean_performance = np.mean(performance)
        S += mean_performance ** 2
    forigin = 12 * N / (k * (k+1)) * (S - k * ((k+1)**2)/4)
    F = (N-1) * forigin / (N * (k-1) - forigin)
    return F
```

图 3: Friedman 检验核心代码

$\tau_{\chi^2}2 = 9.92, \tau_F = 3.9365 > 3.007$。因此拒绝 “所有算法性能相同” 假设。
$CD = 2.728, 1.2 + 2.728 = 3.928 < 4$。因此 C 与 D 有显著区别。C 与其余算法之间没有显著区别。