习题一

141210016, 刘冰楠

2017 年 3 月 15 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.

The *version space* under such circumstance is an **empty set** $\emptyset$.

归纳偏好一：从训练样本中去除尽可能少 的示例使得版本空间不为空，若版本空间刚好只有一个假设，则使用该假设；若在去除尽可能少示例的前提下，版本空间存在多个假设，则使用奥卡姆剃刀作为进一步选择的归纳偏好。

归纳偏好二：在接受训练示例之前，给假设空间中每个假设赋一个惩罚计数，初始值为 0。每出现一个与之不一致的训练示例，惩罚计数 +1. 训练结束后，选择惩罚计数最少的归纳偏好；若存在多个技术相同且最小的惩罚计数，则使用奥卡姆剃刀作为进一步选择的归纳偏好。这一想法是收到支持向量机中 “软” 边界的启发。

以上两个偏好都假设：训练样本中噪音是少数，正确的数据是大多数。

Problem 2

对于有限样例，请证明

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

Proof.

Notation

The sample size is m, with m^+ positive and m^- negative. Use D, D^+, D^- to denote the whole sample set, positive sample set, negative sample set respectively.

$$D = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_m\}. \tag{1}$$

Let $f(\vec{x})$ be our learner's prediction value for $\vec{x}$. Let $f(D) = R$. Sort R in descending order, we have:

$$R = \{r_1, r_2, \ldots, r_k\}, \tag{2}$$

where k is usually less than m because there will be **identical prediction values**.

Use n_{i+} to denote the number of $\vec{x}$ s.t.

$$f(\vec{x}) = r_i \vec{x} \in D^+. \tag{3}$$

Similarly use n_{i-} to denote the number of $\vec{x}$ s.t.

$$f(\vec{x}) = r_i \vec{x} \in D^-. \tag{4}$$

Derivation of Coordinates of ROC

To avoid duplication, we use (a, b) to denote the coordinates of a ROC curve, where $(a_0, b_0) = (0, 0)$ and $(a_1, b_1) = (1, 1)$. Now we can write the formula for a_i and b_i:

$$a_i = a_{i-1} + \frac{n_{i-}}{m^-}, \tag{5}$$

$$b_i = b_{i-1} + \frac{n_{i+}}{m^+} \tag{6}$$

and

$$\begin{aligned} b_i =& b_0 + \sum_{j=1}^{i} \frac{n_{j+}}{m^+} \\ =& \sum_{j=1}^{i} \frac{n_{j+}}{m^+} \end{aligned} \tag{7}$$

Derivation of AUC

$$\begin{aligned} \text{AUC} =& \frac{1}{2} \sum_{i=1}^{m} (a_i - a_{i-1})(b_i + b_{i_1}) \\ =& \frac{1}{2} \sum_{i=1}^{m} \frac{n_{i-}}{m^-} (2b_{i-1} + \frac{n_{i+}}{m^+}) \\ =& \frac{1}{2m^+m^-} \sum_{i=1}^{m} n_{i-} \left(2 \sum_{j=1}^{i-1} n_{j+} + n_{i+} \right) \\ =& \frac{1}{m^+m^-} \left(\sum_{i=1}^{m} \sum_{j=1}^{i-1} n_{i-} n_{j+} + \frac{1}{2} \sum_{i=1}^{m} n_{i-} n_{i+} \right) \\ =& \frac{1}{m^+m^-} \left(\sum_{1 \leq j < i \leq m} n_{i-} n_{j+} + \frac{1}{2} \sum_{i=1}^{m} n_{i-} n_{i+} \right) \\ =& \frac{1}{m^+m^-} \sum_{\mathbf{x}^+ \in D^+} \sum_{\mathbf{x}^- \in D^-} \left(\mathbb{1}(f(\mathbf{x}^+) > f(\mathbf{x}^-)) + \frac{1}{2} \mathbb{1}(f(\mathbf{x}^+) = f(\mathbf{x}^-)) \right) \end{aligned} \tag{8}$$

Note that

$$l_{\text{rank}} = \frac{1}{m^+m^-} \sum_{\mathbf{x}^+ \in D^+} \sum_{\mathbf{x}^- \in D^-} \left(\mathbb{1}(f(\mathbf{x}^+) < f(\mathbf{x}^-)) + \frac{1}{2} \mathbb{1}(f(\mathbf{x}^+) = f(\mathbf{x}^-)) \right), \tag{9}$$

therefore we have

$$\mathrm{AUC} = 1 - l_{\mathrm{rank}} \tag{10}$$

Further Interpretation

Actually, AUC and l_{rank} are all possibilities. We can easily understand the formula from the perspective of *classical models of probability (古典概型)*: If we randomly pick x from D^+ and y from D^-, then $\mathrm{AUC} = P(f(x) > f(y))$.

Problem 3

在某个西瓜分类任务的验证集中，共有 10 个示例，其中有 3 个类别标记为“1”，表示该示例是好瓜；有 7 个类别标记为“0”，表示该示例不是好瓜。由于学习方法能力有限，我们只能产生在验证集上精度 (accuracy) 为 0.8 的分类器。

(a) 如果想要在验证集上得到最佳查准率 (precision)，该分类器应该作出何种预测？
此时的查全率 (recall) 和 F1 分别是多少？

(b) 如果想要在验证集上得到最佳查全率 (recall)，该分类器应该作出何种预测？
此时的查准率 (precision) 和 F1 分别是多少？

Solution.

(a) Predict 2 of positive instances wrongly and predict other ones correctly. Then TP=1, TN=7, FP=0, FN=2. Recall=1/3. Precision=1/1=1. F1=2/4=1/2.

(b) Predict 2 of negative instances wrongly and predict other ones correctly. Then TP=3, TN=5, FP=2, FN=0. Recall=3/3=1. Precision=3/5. F1=6/8=3/4.

Problem 4

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1所示:

表 1: 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。

Solution.

I write a small Python program (The **source code is attached**. See *Friedman_test.py*) to do all these things. The result is shown below.

Friedman Test statistics τ_F=3.9365, P value = 0.0207. The critical range in *Nemenyi Test* is: 2.7278. These are pairs that have different performance:Performance of these 5 algorithms are different. Using *Nemenyi test*, I find that Algo.C and Algo.D is statistically different. See Figure1

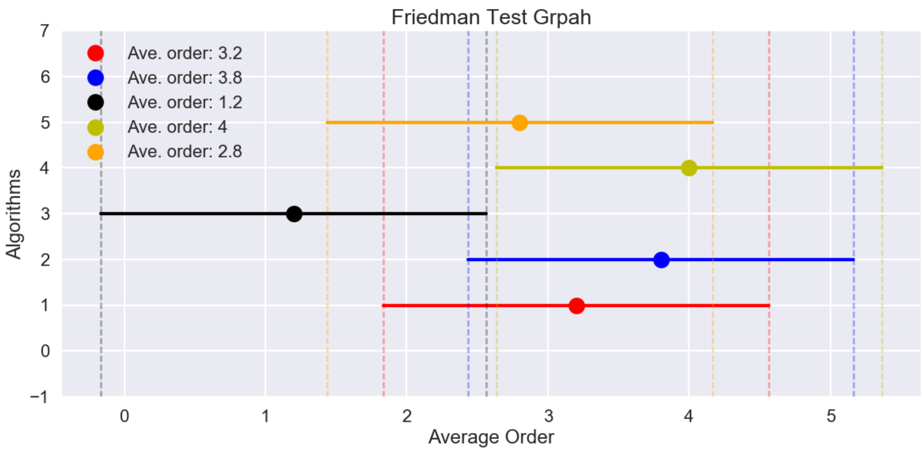

图 1: Friedman Test Grpah