第三章　模型评估

张皓

https://haomood.github.io/homepage/

zhangh0214@gmail.com

摘要

上一章提出了学习算法的目标是获得泛化性能好的学习器, 本章介绍如何对模型的泛化性能进行近似, 以及在不同任务下对应的不同性能度量.

本系列文章有以下特点: *(a).* 为了减轻读者的负担并能使尽可能多的读者从中收益, 本文试图尽可能少地使用数学知识, 只要求读者有基本的微积分、线性代数和概率论基础, 并在第一节对关键的数学知识进行回顾和介绍. *(b).* 本文不省略任何推导步骤, 适时补充背景知识, 力图使本节内容是自足的, 使机器学习的初学者也能理解本文内容. *(c).* 机器学习近年来发展极其迅速, 已成为一个非常广袤的领域. 本文无法涵盖机器学习领域的方方面面, 仅就一些关键的机器学习流派的方法进行介绍. *(d).* 为了帮助读者巩固本文内容, 或引导读者扩展相关知识, 文中穿插了许多问题, 并在最后一节进行问题的 "快问快答".

1　模型评估

1.1　归纳偏好

定义 1 (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. 假设偏好可看做学习算法自身在一个可能很庞大的假设空间中对假设进行选择的启发式或价值观.

任何一个有效的学习算法必有其归纳偏好, 否则它将被假设空间中看似在训练集上等效的假设所迷惑, 而无法产生确定的学习结果. 归纳偏好是否与问题本身匹配, 大多数时候直接决定了算法能否取得好的性能.

定义 2 (奥卡姆剃刀 (Occam's razor) 原则 [2])**.** 若有多个假设与经验观察一致, 则选最简单的那个.

奥卡姆剃刀原则是在自然科学如物理、天文等领域广为沿用的基础性原则. 但在机器学习领域, 什么是 "更简单的" 这个问题一直困扰着研究者们 [5, 8] .

定义 3 (多释原则 (principle of multiple explanations [1]))**.** 保留与经验观察一致的所有假设. 多释原则与集成学习方面的研究更加吻合.

1.2　测试集与验证集

在现实任务中, 我们往往有多种学习算法可供选择, 甚至对同一个学习算法, 当使用没有免费的午餐定理不同的参数配置时, 也会产生不同的模型. 这是机器学习中的模型选择问题. 理想的解决方案是对候选模型的泛化误差进行评估, 然后选择泛化误差最小的模型. 但是, 我们无法直接获得泛化误差, 而训练误差又由于过拟合现象的存在而不适合作为标准.

为什么要使用测试集? 通常, 我们使用测试集 (testing set) 来测试学习器对新样本的判别能力, 然后以测试集上的测试误差作为泛化误差的近似. 通常我们假设测试样本也是从样本真实分布 $\mathcal{D}$ 中独立同分布采样而得. 测试集需要和训练集互斥.

为什么要使用验证集? 大多数学习算法都有些超参数需要设定. 我们在研究对比不同算法的泛化性能时, 用测试集上的判别效果来估计模型在实际使用时的泛化能力, 而把训练数据另外划分为训练集和验证集 (validation set), 基于验证集上的性能来进行模型选择和调参. 验证集要尽可能和测试的环境相一致.

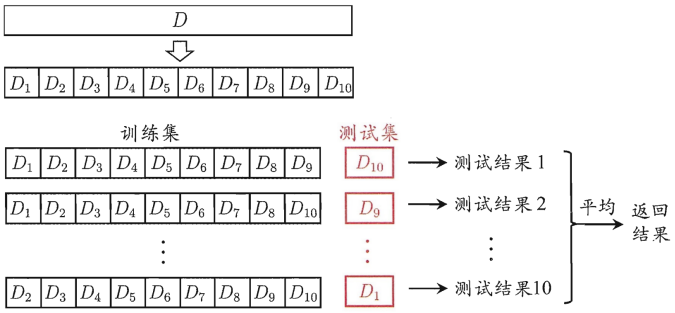


Figure 1: 10 折交叉验证示意图. 本图源于 [9] .

1.3 留出法、交叉验证法和留一法

留出法的基本思路. 留出法 (hold-out) 直接将训练数据划分为两个互斥的集合, 一个作为训练集, 另一个作为验证集.

使用留出法的注意事项. (1). 训练/验证集的划分要尽可能保持数据分布的一致性 (例如在分类任务中使用分层采样), 避免因数据划分过程引入额外的偏差而对最终结果产生影响. (2). 由于划分的随机性, 单次使用留出法得到的估计结果往往不够稳定可靠. 因此, 一般采用若干次随机划分, 每次产生一对训练/验证集用于实验评估, 最后报告所有结果的平均值和标准差. (3). 验证集过小时, 评估结果的方差较大, 不够稳定准确. 训练集小时, 评估结果的偏差较大, 降低了评估结果的保真性 (fidelity). 这个问题没有完美的解决方案, 常见做法是将大约 $\frac{2}{3} \sim \frac{4}{5}$ 的训练数据用于训练, 剩余数据作为验证.

交叉验证法的基本思路. 如图 1 所示, K 折交叉验证 (K-fold cross validation) 先将训练数据划分为 K 个大小相似的互斥子集, 然后, 每次用一个子集作为验证集, 其余 $K-1$ 个子集的并集作为训练集. 这样可进行 K 次训练和验证, 最终返回这 K 个验证结果的均值. K 最常用取值是 10, 其他常用取值是 5 和 20.

使用交叉验证的注意事项. (1). 每个子集都尽可能保持数据分布的一致性. (2). 为减小因样本划分不同而引入的差别, 一般采用若干次交叉验证, 如 10 次 10 折交叉验证 [4] .

留一法的基本思路. 留一法 (leave-one-out) 是 K 折交叉验证在 $K = m$ 时的特例.

留一法的优缺点. 由于每个子集只有一个样本, 留一法不受随机样本划分方式的影响. 由于每个训练集和原始训练数据相比只少了一个样本, 在绝大多数情况下, 留一法中被实际评估的模型和期望评估的用全部训练数据训练出的模型很相似. 因此, 留一法的评估结果往往被认为比较准确. 但是, 在数据集比较大时, 训练 m 个模型的计算开销是难以忍受的.

应该使用哪一个评估方法? 没有哪一个评估方法永远比其他方法准确, 没有免费的午餐定理对实验评估方法同样适用.

Table 1: 分类结果混淆矩阵.

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

2 性能度量

由于回归任务的性能度量比较简单, 在这里, 我们介绍二分类和多分类任务的性能度量.

2.1 二分类任务性能度量

在二分类问题中, 将样例根据其真实类别和学习器预测类别的组合可分为四种情况, 如表 1 所示. 我们有 $TP + FP + TN + FN = m$. 常用二分类问题的性能度量如表 2 所示.

F_1 和 F_β 度量. F_1 度量是查准率和查全率的调和平均

$$\frac{1}{F_1} := \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right). \tag{1}$$

F_β 则是加权调和平均

$$\frac{1}{F_1} := \frac{1}{1+\beta^2}\left(\frac{1}{P} + \frac{\beta^2}{R}\right). \tag{2}$$

相比算术平均和几何平均, 调和平均更重视较小值. 通过 $\beta > 0$, 能让我们表达出对查准率/查全率的不同偏好 [7] . $\beta > 1$ 时查全率有更大的影响, $\beta < 1$ 时查准率有更大的影响.

很多学习器为样本产生一个实值或概率预测, 然后将这个预测值与一个分类阈值进行比较, 若大于阈值则分为正类, 否则为反类. 根据这个实值或概率预测结果, 我们可将样本进行排序, 最可能是正例的排在最前面, 最不可能是正例的排在最后面. 这样, 分类过程就相当于在这个排序中以某个截断点将样本分为两部分, 前一

Table 2: 常用二分类问题的性能度量.

名称	定义	等价式
错误率	$\frac{\sum_{i=1}^{m}\mathbb{I}(\boldsymbol{x}_i\neq y_i)}{m}$	$\frac{FP+FN}{m}$
精度	$\frac{\sum_{i=1}^{m}\mathbb{I}(\boldsymbol{x}_i= y_i)}{m}$	$\frac{TP+TN}{m}$
查准率 (precision)	$\frac{\sum_{i=1}^{m}\mathbb{I}(h(\boldsymbol{x}_i)=1\wedge y_i=1)}{\sum_{i=1}^{m}\mathbb{I}(h(\boldsymbol{x}_i)=1)}$	$\frac{TP}{TP+FP}$
查全率 (recall)	$\frac{\sum_{i=1}^{m}\mathbb{I}(h(\boldsymbol{x}_i)=1\wedge y_i=1)}{\sum_{i=1}^{m}\mathbb{I}(y_i=1)}$	$\frac{TP}{TP+FN}$
F_1 度量	$\frac{2\cdot P\cdot R}{P+R}$	$\frac{2TP}{m+TP-TN}$
F_β 度量	$\frac{(1+\beta^2)\cdot P\cdot R}{\beta^2\cdot P+R}$	
真正例率 (true positive rate, TPR)	$\frac{\sum_{i=1}^{m}\mathbb{I}(y_i=1\wedge h(\boldsymbol{x}_i)=1)}{\sum_{i=1}^{m}\mathbb{I}(y_i=1)}$	$\frac{TP}{TP+FN}$
假正例率 (false positive rate, FPR)	$\frac{\sum_{i=1}^{m}\mathbb{I}(y_i=0\wedge h(\boldsymbol{x}_i)=1)}{\sum_{i=1}^{m}\mathbb{I}(y_i=0)}$	$\frac{FP}{TN+FP}$

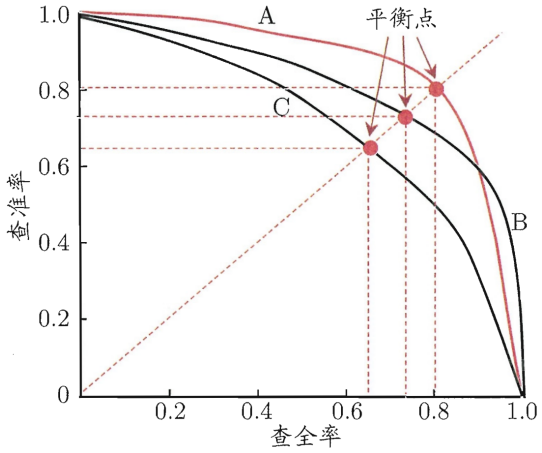


Figure 2: P-R 曲线与平衡点示意图. 本图源于 [9] .

部分判作正例, 后一部分则判作反例. 排序质量的好坏, 直接决定了学习器的泛化能力.

查准率-查全率矛盾. 查准率-查全率是一对矛盾的度量. 如果选择在这个实值或概率预测结果排序中靠后的位置进行截断, 使真实正例的样本尽可能多的被选出来, 这样查全率高, 但是预测为正例的样本变多, 查准率会降低. 如果在排序中靠前的位置进行截断, 只有最有把握的样例才被预测为正例, 这样查准率高, 但是会漏掉不少真实正例, 查全率会降低.

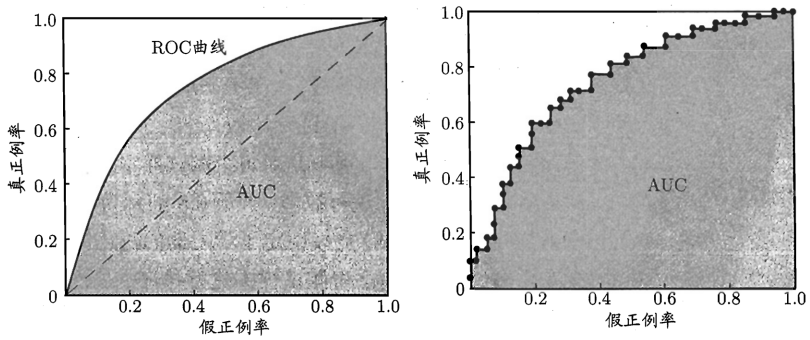
P-R 曲线绘制. 根据学习器的实值或概率预测结果对样本进行排序, 按此顺序逐个把样本作为正例进行预测, 每次可以计算出当前的查全率和查准率, 这样可以得到 P-R 曲线, 如图 2 所示. 现实任务中的 P-R 曲线通常是非单调、不平滑的, 在很多局部有上下波动.

比较两学习器的 *P-R* 曲线. 若一个学习器的 P-R 曲线被另一个学习器的曲线完全 “包住”, 则可断言后者的性能优于前者. 若两个学习器的 P-R 曲线发生交叉, 则一般难以断言孰优孰劣, 只能在具体的查准率或查全率条件下进行比较. 如果一定要进行比较, 这时一个比较合理的判断依据是比较 P-R 曲线下面积的大小, 它在一定程度上表征了学习器在查准率和查全率上取得相对 “双高” 的比例, 但是这个值不容易估算. 平衡点 (break-even point, BEP) 是查准率 = 查全率时的取值, 综合考虑了查准率和查全率.

ROC 曲线绘制. 根据学习器的实值或概率预测结果对样本进行排序, 按此顺序逐个把样本作为正例进行预测, 每次可以计算出当前的 TPR 和 FPR, 这样可得到 ROC 曲线, 如图 3 所示. 对角线对应于随机猜测. 现实任务中的 ROC 曲线是单调的、不平滑的 [6] .

比较两学习器的 *ROC* 曲线. 若一个学习器的 ROC 曲线被另一个学习器的曲线完全 “包住”, 则可断言后者的性能优于前者. 若两个学习器的 ROC 曲线发生交叉, 则一般难以断言孰优孰劣. 如果一定要进行比较, 这时一个比较合理的判断依据是比较 ROC 曲线下面积的大小, 即 AUC (area under ROC curve) [3] . 假设 ROC 是由坐标 $(FPR_1=0,TPR_1),(FPR_1=0,TPR_1),\ldots,(FPR_m=1,TPR_m)$ 的点按序连接而而形成, AUC 可估算为

$$
\begin{aligned}
&AUC\\
&\approx \frac{1}{2}\sum_{i=1}^{m-1}(FPR_{i+1}-FPR_i)(TPR_i+TPR_{i+1})\\
&=1-\frac{1}{m_+m_-}\sum_{\boldsymbol{x}_i:y_i=1}\sum_{\boldsymbol{x}_j:y_j=0}\left(\mathbb{I}(h(\boldsymbol{x}_i)<h(\boldsymbol{x}_j))+\frac{1}{2}\mathbb{I}(h(\boldsymbol{x}_i)=h(\boldsymbol{x}_j))\right)\\
&=1-\ell_{\text{rank}}.
\end{aligned}
\tag{3}
$$



(a) ROC 曲线与 AUC　　(b) 基于有限样例绘制的 ROC 曲线与 AUC

Figure 3: ROC 曲线与 AUC 示意图. 本图源于 [9] .

其中

$$m_+ := \sum_{i=1}^{m} \mathbb{I}(y_i = 1)\,, \tag{4}$$

$$m_- := \sum_{i=1}^{m} \mathbb{I}(y_i = 0)\,. \tag{5}$$

AUC 和排序损失 ℓ 有紧密联系: 考虑每一对正反例, 若正例的预测值小于反例, 则记一个罚分, 若相等, 则记 0.5 个罚分. ℓ 是 ROC 曲线之上的面积.

应该选择哪一个性能度量? 模型的好坏是相对的, 什么样的模型是好的, 不仅取决于算法和数据, 还决定于任务需求. 性能度量反映了任务需求.

2.2 多分类任务性能度量

多分类问题, 我们可以得到多个二分类的混淆矩阵. 除了沿用二分类的错误率和精度的性能度量外, 我们可以对二分类的查准率和查全率进行推广.

宏 F_1 和微 F_1. 宏 F_1 (marco-F_1) 先在各混淆矩阵上分别计算出查准率和查全率, 记为 $(P_1, R_1), (P_2, R_2), \ldots, (P_n, R_n)$. 再计算

$$\text{macro-}P := \frac{1}{n}\sum_{i=1}^{n} P_i = \frac{1}{n}\sum_{i=1}^{n} \frac{TP_i}{TP_i + FP_i}\,; \tag{6}$$

$$\text{macro-}R := \frac{1}{n}\sum_{i=1}^{n} R_i = \frac{1}{n}\sum_{i=1}^{n} \frac{TP_i}{TP_i + FN_i}\,. \tag{7}$$

最后计算相应的宏 F_1

$$\text{macro-}F_1 := \frac{2 \cdot \text{macro-}P \cdot \text{macro-}R}{\text{macro-}P + \text{macro-}R}\,. \tag{8}$$

微 F_1 (micro-F_1) 先在各混淆矩阵上元素进行平均, 得到 TP、FP、TN、FN 的平均值, 分别记为 $\overline{TP}$、$\overline{FP}$、$\overline{TN}$、$\overline{FN}$. 再计算

$$\text{micro-}P := \frac{\overline{TP}}{\overline{TP} + \overline{FP}} = \frac{\frac{1}{n}\sum_{i=1}^{n} TP_i}{\frac{1}{n}\sum_{i=1}^{n} TP_i + \frac{1}{n}\sum_{i=1}^{n} FP_i}\,; \tag{9}$$

$$\text{micro-}R := \frac{\overline{TP}}{\overline{TP} + \overline{FN}} = \frac{\frac{1}{n}\sum_{i=1}^{n} TP_i}{\frac{1}{n}\sum_{i=1}^{n} TP_i + \frac{1}{n}\sum_{i=1}^{n} FN_i}\,. \tag{10}$$

最后计算相应的宏 F_1

$$\text{micro-}F_1 := \frac{2 \cdot \text{micro-}P \cdot \text{micro-}R}{\text{micro-}P + \text{micro-}R}\,. \tag{11}$$

2.3 代价敏感错误率与代价曲线

在现实任务中, 有时不同类型的错误所造成的后果不同. 为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). 以二分类任务为例, 我们可根据任务的领域知识设定一个代价矩阵 $\boldsymbol{C} \in \mathbb{R}^{2\times 2}$, 其中 C_{ij} 表示将真实标记为 i 的样本预测为标记 j 的代价. 一般来说, $c_{ii} := 0$, 重要的是代价比值 $\frac{C_{01}}{C_{10}}$ 而非绝对值. 前面介绍的一些性能度量大都隐式地假设了均等代价. 在非均等代价下, 我们最小化总体代价 (total cost). 代价敏感错误率为

$$E := \frac{1}{m}\sum_{i=1}^{m}(\mathbb{I}(y_i = 0 \wedge h(\boldsymbol{x}_i) = 1)C_{01} + \mathbb{I}(y_i = 1 \wedge h(\boldsymbol{x}_i) = 0)C_{10})\,. \tag{12}$$

3 快问快答

应当选择什么评估方法和性能度量? 答案见上文.

如何看待学术论文中汇报的性能? 虽然每篇学术论文都只用了公共数据集的训练集进行训练, 用测试集作为泛化误差的近似. 但是, 这些学术论文都反复使用了相同的公共数据集用来进行算法性能的比较. 后人在设计新的学习算法时, 会将之前所有人工作中最好的学习算法作为基准, 这已经在训练过程间接接触到了测试数据. 因此, 随着这些公共数据集使用的越多, 后来的学术论文汇报的性能数字将渐渐成为泛化性能过于乐观的估计, "如果被折磨得足够久, 数据将招供". 我们在分析这些学术论文汇报的性能时要持有审慎的态度.

References

[1] E. Asmis. *Epicurus' Scientific Method*. 1984. 1

[2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, 1987. 1

[3] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. 3

[4] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural CComputation*, 10(7):1895–1923, 1998. 2

[5] P. M. Domingos. The role of occam's razor in knowledge discovery. *Data Mining and Knowledge Discovery*, 3(4):409–425, 1999. 1

[6] K. A. Spackman. Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the Sixth International Workshop on Machine Learning (IWML)*, pages 160–163, 1989. 3

[7] C. J. van Rijsbergen. *Information Retrieval, 2nd Edition*. Butterworth, 1979. 2

[8] G. I. Webb. Further experimental evidence against the utility of occam's razor. *J. Artif. Intell. Res.*, 4:397–417, 1996. 1

[9] 周志华. 机器学习. 清华大学出版社, 2016. 2, 3, 4