第2章：模型评估

蒋良孝 |  中国地质大学（武汉）

机器学习与数据挖掘团队

ljiang@cug.edu.cn

http://www.escience.cn/people/jlx/

本章内容

一、评估方法

二、评估指标

三、比较检验

一、评估方法

- 在学习得到的模型投放使用之前，通常需要对其进行性能评估。为此， 需使用一个“测试集”（testing set）来测试模型对新样本的泛化能力，然后以测试集上的“测试误差”（testing error）作为泛化误差的近似。

- 我们假设测试集是从样本真实分布中独立采样获得，所以测试集要和训练集中的样本尽量互斥。

- 给定一个已知的数据集，将数据集拆分成训练集S和测试集T，通常的做法包括**留出法、交叉验证法、自助法**。

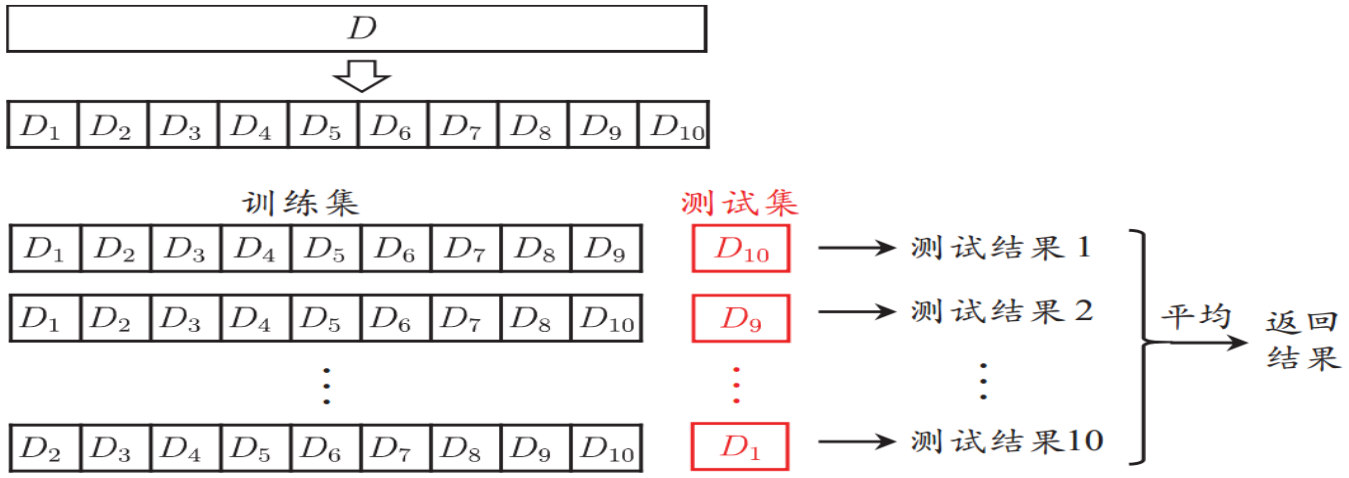
一、评估方法

- 留出法：
  - ✓ 直接将数据集划分为两个互斥集合
  - ✓ 训练/测试集划分要尽可能保持数据分布的一致性
  - ✓ 一般若干次随机划分、重复实验取平均值
  - ✓ 训练/测试样本比例通常为2:1~4:1

一、评估方法

- 交叉验证法：

将数据集分层采样划分为k个大小相似的互斥子集，每次用k-1个子集的并集作为训练集，余下的子集作为测试集，最终返回k个测试结果的均值，k最常用的取值是10。



10 折交叉验证示意图

一、评估方法

- 与留出法类似，将数据集D划分为k个子集同样存在多种划分方式，为了减小因样本划分不同而引入的差别，k折交叉验证通常随机使用不同的划分重复p次，最终的评估结果是这p次k折交叉验证结果的均值，例如常见的“10次10折交叉验证”。
- 假设数据集D包含m个样本，若令k=m，则得到留一法：
  - ✓ 不受随机样本划分方式的影响
  - ✓ 结果往往比较准确
  - ✓ 当数据集比较大时，计算开销难以忍受

一、评估方法

- 自助法：

以自助采样法为基础，对数据集D有放回采样m次得到训练集 D', $D \setminus D'$ 用做测试集

- ✓ 实际模型与预期模型都使用m个训练样本
- ✓ 约有1/3的样本没在训练集中出现，用作测试集
- ✓ 从初始数据集中产生多个不同的训练集，对集成学习有很大的好处
- ✓ 自助法在数据集较小、难以有效划分训练/测试集时很有用；由于改变了数据集分布可能引入估计偏差，在数据量足够时，留出法和交叉验证法更常用。

二、评估指标

要评估模型的好坏光有评估方法还不行，还得确定评估指标。

所谓评估指标就是衡量模型泛化能力好坏的评估标准，反映了任务需求；使用不同的评估指标往往会导致不同的评估结果。

在分类预测任务中，给定测试样例集，评估分类模型的性能就是把对每一个待测样本的分类结果和它的真实标记比较。

因此，准确率和错误率是最常用的两种评估指标：

✓准确率就是分对样本占测试样本总数的比例；

✓错误率就是分错样本占测试样本总数的比例。

二、评估指标

由于准确率和错误率将每个类看的同等重要，因此不适合用来分析类不平衡数据集。在类不平衡数据集中，正确分类稀有类比正确分类多数类更有意义。此时查准率和查全率比准确率和错误率更适合。对于二分类问题，稀有类样本通常记为正例，而多数类样本记为负例。统计真实标记和预测结果的组合可以得到如下所示的混淆矩阵：

真实情况	预测结果	
	正例	负例
正例	**TP（真正例）**	**FN（假负例）**
负例	**FP（假正例）**	**TN（真负例）**

查准率（P）就是被分为正类的样本中实际为正类的样本比例：

$P=TP/(TP+FP)$

查全率（R）就是实际为正类的样本中被分为正类的样本比例：

$R=TP/(TP+FN)$

二、评估指标

可见，查准率是被分类器分为正类的样本中实际为正类的比例；而查全率是被分类器正确分类为正类的比例。二者通常是矛盾的。查准率高时，查全率往往偏低；而查全率高时，查准率往往偏低。为综合考虑查准率和查全率，它们的调和均值**F1度量**被提出：

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

比F1更一般的形式 F_β，

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

$\beta = 1$：标准的F1
$\beta > 1$：偏重查全率
$\beta < 1$：偏重查准率

二、评估指标

很多分类器可以为测试样例产生一个概率预测，因此可以根据预测的概率将测试样例进行排序，把最可能是正例的排在最前面，把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，前一部分判为正例，后一部分判为反例。在不同的应用任务下，用户可以根据不同的任务需求来选择不同的截断点。因此，排序本身的质量好坏体现了分类器在不同任务下的泛化性能。

ROC曲线，全称是“受试者工作特征”（Receiver Operating Characteristics）曲线。根据分类器的预测结果对样例排序，并按此顺序依次选择不同的“截断点”逐个把样例作为正例进行预测，每次计算出当前分类器的“真正率”和“假正率”，然后分别以它们为纵轴和横轴绘图，就可得到ROC曲线。

二、评估指标

真正率（TPR）就是被分为正类的正样本比例：TPR=TP/(TP+FN)

假正率（FPR）就是被分为正类的负样本比例：FPR=FP/(FP+TN)

待测样例	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
样本标记	-	+	-	-	+	-	+	+	-
$P(+\|x_i)$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1

| -+ - - + - + + - TPR=0/4; FPR=0/5
- |+ - - + - + + - TPR=0/4; FPR=1/5
- +| - - + - + + - TPR=1/4; FPR=1/5
- + -| - + - + + - TPR=1/4; FPR=2/5
……
- + - - + - + +| - TPR=4/4; FPR=4/5
- + - - + - + +- | TPR=4/4; FPR=5/5

二、评估指标

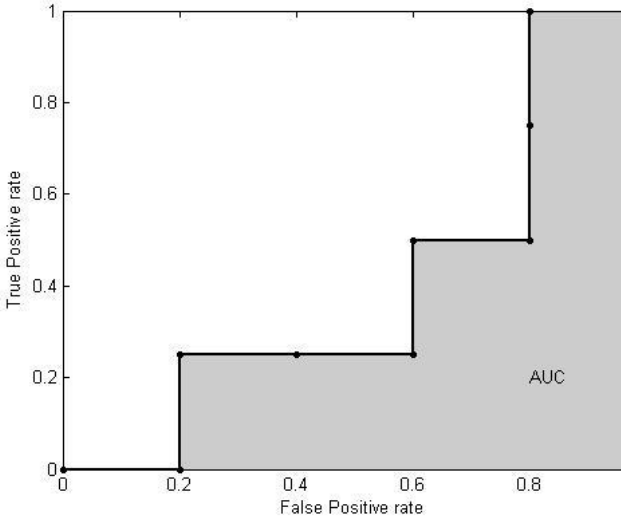


图1

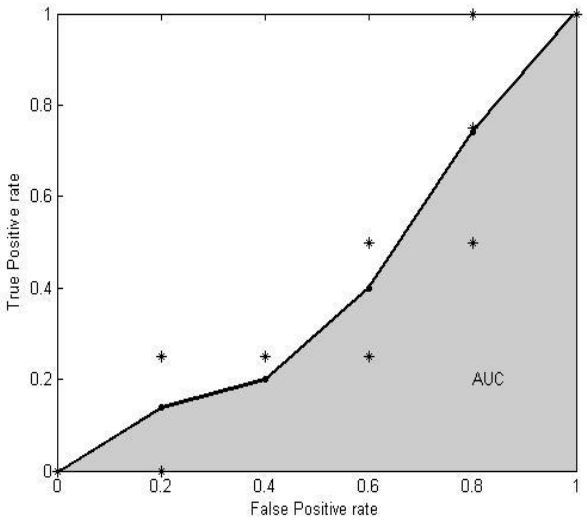


图2

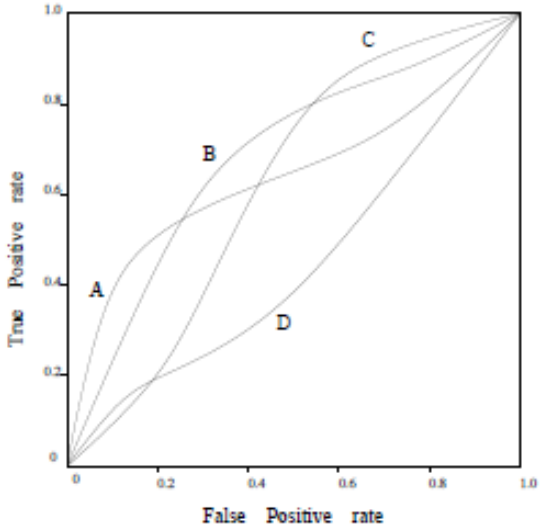


图3

若某个分类器的ROC曲线被另一个分类器的曲线“包住”，则后者性能优于前者；否则如果曲线交叉，可以根据ROC曲线下面积的大小进行比较，即AUC（Area Under ROC Curve）.

二、评估指标

AUC的计算：

$$AUC = \frac{\sum_{i=1}^{n_0} r_i - n_0 \times (n_0 + 1)/2}{n_0 \times n_1}$$

其中，n_0 和 n_1分别表示反例和正例的个数，r_i 分别为第 i 个反例（-）在整个测试样例中的排序。

AUC度量了分类器预测样本排序的性能。

二、评估指标

对于刚才的例子，具体如表（1）所示，

表（1）：一个计算 AUC 的例子

Ranking	−	+	−	−	+	−	+	+	−
i	1		2	3		4			5
r_i	1		3	4		6			9

表（2）：AUC 取最大值的特例

Ranking	+	+	+	+	−	−	−	−	−
i					1	2	3	4	5
r_i					5	6	7	8	9

$$AUC = \frac{(1+3+4+6+9) - 5(5+1)/2}{5 \times 4} = 0.4$$

显然，如表（2）所示的特例，当所有的反例（-）都排在正例（ + ） 的后面时， AUC 的值达到最大值1 。

$$AUC \frac{(5+6+7+8+9) - 5(5+1)/2}{5 \times 4} = 1$$

二、评估指标

CLL (Conditional Log Likelihood) 的计算：

给定分类器G和一个测试样本集$T = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_i \cdots, \boldsymbol{x}_t\}$

假设每一个测试样本$\boldsymbol{x}_i$ 的真实类标记是 y_i

那么分类器G预测测试样本集T的CLL定义如下：

$$CLL(G|T) = \sum_{i=1}^{t} log\hat{P}_G(y_i|\boldsymbol{x}_i)$$

CLL度量了分类器预测样本类成员概率的性能。

三、比较检验

- 有了实验评估方法和评估指标，看似可以对分类器的性能进行评估比较了：先使用某种实验评估方法测得分类器的某个评估指标结果，然后对这些结果进行比较。但怎么来做这个“比较”呢？是直接比较不同分类器的评估指标结果吗?
- 关于性能比较：
  - ✓ 测试性能并不等于泛化性能
  - ✓ 测试性能会随着测试集的变化而变化
  - ✓ 很多机器学习算法本身有一定的随机性

  直接选取相应评估方法在相应度量下比大小的方法不可取！
- 假设检验为分类器的性能比较提供了重要依据，基于其结果我们可以推断出，若在测试集上观察到分类器A比B好，则A的泛化性能是否在统计意义上优于B，以及这个结论的把握有多大。
- 下面我们将介绍几种常用的机器学习性能比较方法来对不同分类器的性能进行比较。

三、比较检验

- **成对双边t检验**

  对两个分类器A和B，若k折交叉验证得到的测试错误率分别为$\epsilon_1^A, ..., \epsilon_k^A$和$\epsilon_1^B, ..., \epsilon_k^B$，可用“成对t检验”进行比较检验。若两个分类器的性能相同，则他们使用相同的训练/测试集得到的测试错误率应相同，即$\epsilon_i^A = \epsilon_i^B$.

  具体来说，对k折交叉验证产生的k对测试错误率：先对每对结果求差，$\Delta_i = \epsilon_i^A - \epsilon_i^B$；然后根据差值$\Delta_1, ..., \Delta_k$来对“分类器A与B性能相同”这个假设做t检验，计算出差值的均值μ和方差σ^2，以及t统计量：

$$\mu = \frac{1}{k}\sum_{i=1}^{k}\Delta_i \quad , \quad \sigma^2 = \frac{1}{k-1}\sum_{i=1}^{k}(\Delta_i - \mu)^2 \quad , \quad \tau_t = \left|\frac{\mu\sqrt{k}}{\sigma}\right| \quad 。$$

三、比较检验

因为计算得到的t统计量服从自由度为k-1的t分布，如果t值小于双边t检验在显著度α下的临界值，则认为这两个分类器的性能没有显著差别；否则可认为这两个分类器的性能有显著差别，且平均错误率较小的那个分类器的性能较优。

在不同自由度v和显著度α下的临界值可通过查找t分布的临界值表得到。

单侧 双侧	α=0.10 α=0.20	0.05 0.10	0.025 0.05	0.01 0.02	0.005 0.01
V=1	3.078	6.314	12.706	31.821	63.657
2	1.886	2.920	4.303	6.965	9.925
3	1.638	2.353	3.182	4.541	5.841
4	1.533	2.132	2.776	3.747	4.604
5	1.476	2.015	2.571	3.365	4.032
6	1.440	1.943	2.447	3.143	3.707
7	1.415	1.895	2.365	2.998	3.499
8	1.397	1.860	2.306	2.896	2.355
9	1.383	1.833	2.262	2.821	3.250
10	1.372	1.812	2.228	2.764	3.169

三、比较检验

- **Friedman检验与Nemenyi后续检验**

  成对双边t检验是在一个数据集上比较两个分类器的性能，而在很多时候，我们需要在一组数据集上比较多个分类器的性能，这就需要使用基于排序的Friedman 检验。

  假定我们要在N个数据集上比较k个算法，首先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，然后在每个数据集上根据性能好坏排序，并赋序值1,2,…；若算法性能相同则平分序值，继而得到每个算法在所有数据集上的平均序值。

三、比较检验

令 r_i 表示第 i 个算法的平均序值，则变量 τ_{χ^2} 服从自由度为k-1的 χ^2 分布：

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

其中N，k表示数据集和算法的个数。

基于计算得到的 τ_{χ^2}，可以计算出F检验的统计量：

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

三、比较检验

因为τ_F服从自由度为k-1和(k-1)(N-1)的F分布，如果计算得到的 τ_F 值小于 F检验的常用临界值（F检验的常用临界值可通过查表得到），则认为所有比较的算法是相同的；否则可认为所有比较的算法是显著不同的，这时需进行“后续检验”来进一步区分各算法，常用的有Nemenyi后续检验。

F 检验的常用临界值

$\alpha = 0.05$									
数据集	算法个数 k								
个数 N	2	3	4	5	6	7	8	9	10
4	10.128	5.143	3.863	3.259	2.901	2.661	2.488	2.355	2.250
5	7.709	4.459	3.490	3.007	2.711	2.508	2.359	2.244	2.153
8	5.591	3.739	3.072	2.714	2.485	2.324	2.203	2.109	2.032
10	5.117	3.555	2.960	2.634	2.422	2.272	2.159	2.070	1.998
15	4.600	3.340	2.827	2.537	2.346	2.209	2.104	2.022	1.955
20	4.381	3.245	2.766	2.492	2.310	2.179	2.079	2.000	1.935

三、比较检验

Nemenyi后续检验计算平均序值差别的临界阈值

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

其中q_α是Tukey分布的临界值，不同置信度下的临界值可通过查找Tukey分布的临界值表得到。

如果两个算法的平均序值之差小于临界阈值CD，则认为这两个算法的性能在相应的置信度下没有显著差别；否则可认为这两个算法的性能在相应的置信度下有显著差别，且平均序值较小的那个算法的性能较优。

Nemenyi 检验中常用的 q_α 值

α	算法个数 k								
	2	3	4	5	6	7	8	9	10
0.05	1.960	2.344	2.569	2.728	2.850	2.949	3.031	3.102	3.164
0.1	1.645	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920

三、比较检验

假定用 D_1, D_2, D_3, D_4 四个数据集对算法 A, B, C 进行比较。得到表格如下所示。首先计算出 τ_F=24.429，查表可知，它大于α=0.05时的F检验临界值5.143，因此拒绝“所有算法性能相同”这个假设。然后使用Nemenyi后续检验，计算出临界值域CD=1.657。因为算法A与B以及算法B与C的平均序值之差均小于临界值域，而算法A与C的平均序值之差大于临界值域，因此，检验结果认为算法A与C的性能显著不同，而算法A与B以及算法B与C的性能没有显著差别。

算法比较序值表

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875

三、比较检验

上述检验比较可以直观地用Friedman检验图显示。图中纵轴显示各个算法，横轴是平均序值。对每个算法用一个圆点显示其平均序值，以圆点为中心的横线段表示临界值域的大小。若两个算法有交叠(A和B以及B和C)，则说明没有显著差别；否则有显著差别(A和C)，算法A的性能明显优于算法C。