Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 27 (2006) 861–874
Pattern Recognition Letters
www.elsevier.com/locate/patrec

ROC 分析简介

Tom Fawcett

美国加利福尼亚州帕洛阿尔托市斯汤顿路 2164 号 学习与专长研究所 邮编：94306

2005 年 12 月 19 日在线发布

摘要

受试者工作特征（ROC）曲线有助于对分类器进行组织并直观展示其性能。ROC 曲线在医疗决策中应用广泛，近年来在机器学习和数据挖掘研究中的使用也日益增多。尽管 ROC 曲线看似简单，但在实际应用中仍存在一些常见的误解和陷阱。本文旨在介绍 ROC 曲线，并为在研究中使用它们提供指导。

关键词：ROC 分析；分类器评估；评估指标

1. 简介

受试者工作特征（ROC）图是一种基于分类器性能进行可视化、组织和选择分类器的技术。长期以来，ROC 图在信号检测理论中一直用于描绘分类器的命中率和虚警率之间的权衡（Egan，1975；Swets 等人，2000）。ROC 分析已扩展用于可视化和分析诊断系统的行为（Swets，1988）。医学决策领域有大量的文献介绍了 ROC 图在诊断测试中的应用（Zou，2002）。Swets 等人（2000）在《科学美国人》杂志上发表的文章使 ROC 曲线引起了更广泛的关注。

在机器学习领域，最早采用 ROC 图的学者之一是斯帕克曼（1989 年），他展示了 ROC 曲线在评估和比较算法方面的价值。近年来，随着人们逐渐认识到简单的分类准确率往往不是衡量性能的良好指标（普罗沃斯特和法塞特，1997 年；普罗沃斯特等人，1998 年），机器学习社区对 ROC 图的使用有所增加。除了作为一种普遍适用的性能绘图方法外，它们还具有一些特性，使其在某些方面特别有用。

具有类别分布不均衡和分类错误代价不等的领域。随着对成本敏感学习以及在类别不平衡情况下学习的研究不断深入，这些特征变得越来越重要。

ROC 曲线在概念上很简单，但在研究中使用时会出现一些不明显的复杂情况。在实际应用中使用它们时也存在一些常见的误解和陷阱。本文旨在作为 ROC 曲线的基本介绍，并为在研究中使用它们提供指导。本文的目标是增进人们对 ROC 曲线的普遍了解，从而促进该领域更好的评估实践。

2. 分类器性能

我们首先考虑仅使用两类的分类问题。形式上，每个实例 I 都被映射到正负两类标签集合 {p，n} 中的一个元素。分类模型（或分类器）是从实例到预测类别的映射。有些分类模型会产生连续的输出（例如，对实例属于某一类别的概率估计），可以应用不同的阈值来预测类别归属。而其他模型则直接产生一个离散的类别标签，仅表明实例的预测类别。为了区分这两种情况，

E-mail addresses: tfawcett@acm.org, tom.fawcett@gmail.com

0167-8655/$ - see front matter doi:10.1016/j.patrec.2005.10.010

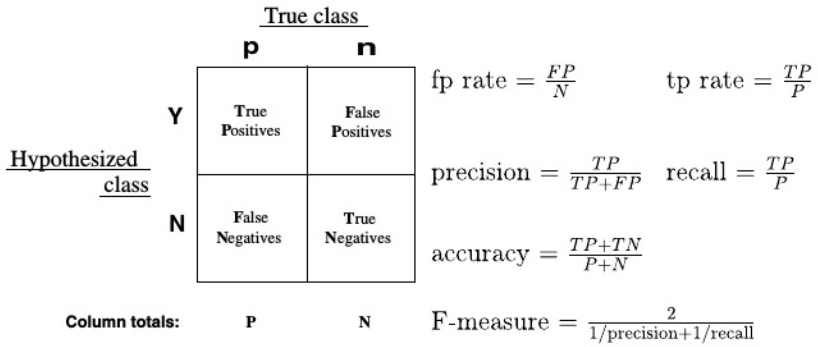


图 1. 混淆矩阵及其计算得出的常见性能指标。

对于实际类别和预测类别，我们使用标签 {Y，N} 来表示模型生成的类别预测结果。

给定一个分类器和一个实例，会有四种可能的结果。如果实例为正类且被分类为正类，则计为真阳性；如果被分类为负类，则计为假阴性。如果实例为负类且被分类为负类，则计为真阴性；如果被分类为正类，则计为假阳性。给定一个分类器和一组实例（测试集），可以构建一个 2×2 的混淆矩阵（也称为列联表），表示该组实例的分布情况。此矩阵是许多常见度量的基础。

图 1 展示了一个混淆矩阵以及可由其计算出的几个常见指标的公式。主对角线上的数字代表正确决策的数量，而该对角线以外的数字则代表错误——即各类之间的混淆情况。分类器的真正例率 1 [1]（也称为命中率和召回率）估计为

$$tp\ rate \approx \frac{\text{Positives correctly classified}}{\text{Total positives}}$$

该分类器的误报率（也称为虚报率）是

$$fp\ rate \approx \frac{\text{Negatives incorrectly classified}}{\text{Total negatives}}$$

与 ROC 曲线相关的其他术语有

$$\text{sensitivity} = \text{recall}$$

$$\text{specificity} = \frac{\text{True negatives}}{\text{False positives} + \text{True negatives}} = 1 - fp\ rate$$

阳性预测值 ¼ 精确度

3. ROC 空间

ROC 曲线是二维图形，其中 Y 轴表示真阳性率，X 轴表示假阳性率。ROC 曲线描绘了收益（真阳性）与成本（假阳性）之间的相对权衡。图 2 展示了一个带有五个分类器（分别标记为 A 至 E）的 ROC 曲线。

离散分类器仅输出类别标签。每个离散分类器都会生成一个（假阳性率，真阳性率）对，对应于 ROC 空间中的一个点。图 2 中的所有分类器均为离散分类器。

在 ROC 空间中有几个重要的点需要注意。左下角的点（0,0）表示从不给出正类别的策略；这样的分类器不会出现假阳性错误，但也不会获得任何真阳性结果。相反的策略，即无条件给出正类别，则由右上角的点（1,1）表示。

点（0,1）表示完美分类。如图所示，D 的性能是完美的。

通俗地说，在 ROC 空间中，如果一个点位于另一个点的西北方向（即真阳性率更高、假阳性率更低，或者两者兼而有之），那么这个点就优于另一个点。在 ROC 曲线图中，位于左侧靠近 X 轴的分类器可能

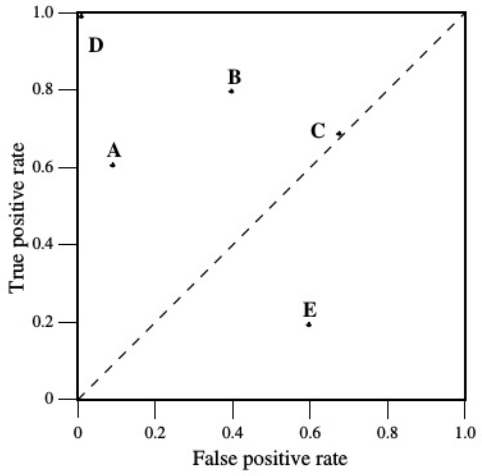


图 2. 一个基本的 ROC 图，展示了五个离散分类器。

[1] For clarity, counts such as TP and FP will be denoted with upper-case letters and rates such as tp rate will be denoted with lower-case.

被视为“保守型”的分类器：只有在有充分证据的情况下才会做出肯定分类，因此它们的误报率很低，但其真正例率也往往较低。位于 ROC 曲线右上角的分类器可被视为“自由型”：它们在证据不足的情况下也会做出肯定分类，因此几乎能正确分类所有正例，但其误报率通常较高。在图 2 中，A 比 B 更保守。许多现实世界领域中负例的数量远远多于正例，因此 ROC 曲线左下角的性能表现就变得更有意义。

3.1. 随机性能

对角线 y = x 代表随机猜测类别的策略。例如，如果一个分类器随机猜测正类别的概率为 50%，那么可以预期它将正确识别 50% 的正样本和 50% 的负样本；这在 ROC 空间中对应于点 （0.5, 0.5）。如果它 90% 的时间都猜测正类别，那么可以预期它将正确识别 90% 的正样本，但其假正率也会增加到 90%，在 ROC 空间中对应于点 （0.9, 0.9）。因此，一个随机分类器在 ROC 空间中的点会根据其猜测正类别的频率在对角线上来回移动。为了从这条对角线移动到上三角区域，分类器必须利用数据中的某些信息。在图 2 中，C 的表现几乎完全是随机的。在点 （0.7, 0.7） 处，可以说 C 有 70% 的时间猜测正类别。

在 ROC 图中，出现在右下角三角形区域内的任何分类器的表现都比随机猜测还要差，因此这个三角形区域通常是空的。如果对某个分类器进行取反操作，即对其对每个实例的分类决策进行反转，那么其真阳性分类就会变成假阴性错误，其假阳性就会变成真阴性。因此，任何在右下角三角形区域产生点的分类器，都可以通过取反操作在左上角三角形区域产生一个点。在图 2 中，E 的表现比随机猜测差很多，实际上它是 B 的取反。位于对角线上的任何分类器都可以说是没有关于类别的信息。位于对角线下方的分类器可以说是有用的信息，但它对这些信息的应用是错误的（Flach 和 Wu, 2003）。

给定一个 ROC 图，在其中某个分类器的性能似乎略好于随机水平，人们自然会问：“这个分类器的性能是否真的具有显著性，还是仅仅偶然好于随机水平？”对于这个问题，目前没有定论性的检验方法，但福尔曼（2002 年）展示了一种利用 ROC 曲线来解决此问题的方法。

4. ROC 空间中的曲线

许多分类器，例如决策树或规则集，其设计初衷仅是为了对每个实例做出类别判定，即给出“是”或“否”。当这种离散分类器应用于测试集时，它会产生一个单一的混淆矩阵，而这又对应于一个 ROC 点。因此，离散分类器在 ROC 空间中仅产生一个点。

某些分类器，例如朴素贝叶斯分类器或神经网络，自然会得出实例概率或得分，这是一个数值，表示某个实例属于某个类别的程度。这些值可以是严格的概率，在这种情况下，它们遵循标准的概率定理；也可以是一般的、未校准的得分，在这种情况下，唯一成立的性质是得分越高，概率越大。尽管输出可能不是严格意义上的概率，我们仍将这两种分类器统称为概率分类器。

这样的排名或评分分类器可以与阈值结合使用，以生成一个离散（二元）分类器：如果分类器的输出高于阈值，则分类器输出 Y，否则输出 N。每个阈值都会在 ROC 空间中产生一个不同的点。从概念上讲，我们可以想象将阈值从 1 变化到 +1，并在 ROC 空间中描绘出一条曲线。但从计算角度来看，这是生成 ROC 曲线的一种效率低下且不够严谨的方法，下一节将介绍一种更高效且更谨慎的方法。

图 3 展示了一个在 20 个实例的测试集上生成的 ROC 曲线示例。这些实例包括 10 个正例和 10 个负例，它们在图旁的表格中列出。由有限数量的实例生成的任何 ROC 曲线实际上都是阶梯函数，随着实例数量趋于无穷大，该函数会逐渐逼近一条真正的曲线。图 3 中的阶梯函数取自一个非常小的实例集，以便能够理解每个点的推导过程。在图 3 的表格中，实例按照其得分排序，ROC 曲线上的每个点都用产生该点的得分阈值进行标注。阈值为 +1 时，得到点 （0,0）。当我们将阈值降低到 0.9 时，第一个正例被分类为正例，从而得到点 （0,0.1）。随着阈值进一步降低，曲线向右上方攀升，最终在阈值为 0.1 时到达点 （1,1）。请注意，降低此阈值相当于从图的“保守”区域向“宽松”区域移动。

尽管测试集规模很小，但我们仍能对分类器做出一些初步观察。它在图表中较为保守的区域表现更佳；在 （0.1,0.5） 这个 ROC 点上，其准确率最高（70%）。这相当于说，该分类器在识别可能的正例方面比识别可能的负例更出色。还需注意的是，分类器的最佳准确率出现在 P0.54 这个阈值上，而非我们预期的平衡分布下的 P0.5。下一节将讨论这一现象。

4.1. 相对分数与绝对分数

关于 ROC 曲线的一个重要点在于，它们衡量的是分类器产生良好相对结果的能力。

[2] Techniques exist for converting an uncalibrated score into a proper probability but this conversion is unnecessary for ROC curves.

指令#	班级	分数	指令#	班级	分数
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

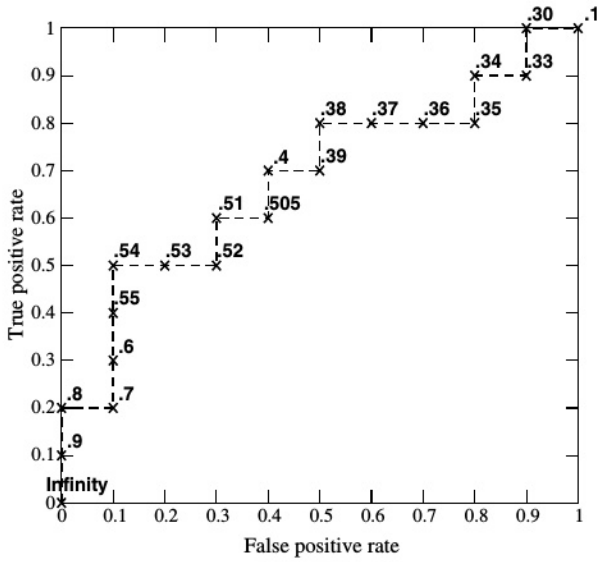


图 3.通过阈值处理测试集所生成的 ROC（受试者工作特征）曲线。表格展示了 20 个数据以及由评分分类器为每个数据分配的分数。图表展示了相应的 ROC 曲线，每个点都标注了生成该点的阈值。

实例得分。分类器无需生成准确的、经过校准的概率估计值；它只需生成相对准确的得分，以区分正例和负例即可。

考虑图 4 中来自朴素贝叶斯分类器的简单实例得分。将假设的类别（如果得分大于 0.5 则为 Y，否则为 N）与真实类别进行比较，我们可以看到分类器将实例 7 和 8 分类错误，准确率为 80%。然而，再看图左侧的 ROC 曲线。该曲线从 （0,0） 垂直上升至 （0,1），然后水平延伸至 （1,1）。这表明在该测试集上分类性能完美。为何存在这种差异?

其原因在于两者所衡量的内容不同。ROC 曲线展示了分类器将正例与负例进行排序的能力，而在这方面它确实是完美的。准确率指标则设定了一个阈值（分数 > 0.5），并根据分数来衡量分类结果。如果分数是真正的概率值，那么准确率指标是合适的，但实际情况并非如此。换句话说，这些分数没有像真正的概率那样进行恰当的校准。在 ROC 空间中，设定 0.5 的阈值会导致图 4 中圈出的“准确率点”所代表的性能表现。这个操作点是次优的。我们可以通过训练集估计先验概率 p(p) = 6/10 = 0.6，并将其作为阈值，但这仍会产生次优的性能表现（准确率为 90%）。

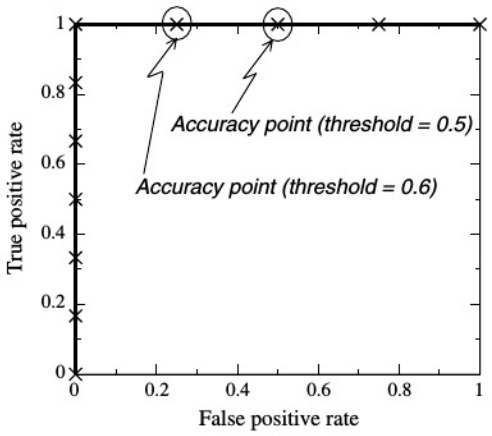
消除这种现象的一种方法是对分类器的得分进行校准。有一些方法可以做到这一点（Zadrozny 和 Elkan，2001）。另一种方法是使用一种 ROC 方法，根据操作点的相对性能来选择它们，也有相关的方法可以实现（Provost 和 Fawcett，1998，2001）。这些后一种方法在第 6 节中会简要讨论。

相对评分的一个结果是，不同模型类别的分类器得分不应相互比较。一种模型类别可能被设计为产生 [0,1] 范围内的得分，而另一种则产生 [−1，+1] 或 [1,100] 范围内的得分。在共同的阈值下比较模型性能将毫无意义。

4.2. 类别偏差

ROC 曲线具有一个吸引人的特性：它们不受类别分布变化的影响。如果测试集中正例与负例的比例发生变化，ROC 曲线不会改变。要理解其中的原因，可以考虑图 1 中的混淆矩阵。请注意，类别分布——正例与负例的比例——是左列（+）与右列（-）之间的关系。任何使用混淆矩阵中两列值的性能度量都会对类别偏斜敏感。诸如准确率、精确率、提升率和 F 值等度量都使用混淆矩阵中两列的值。随着类别分布的变化，这些度量也会发生变化，即使基本的分类器性能没有改变。ROC 图基于真阳性率和假阳性率，其中每个维度都是严格的列比，因此不受类别分布的影响。

对于一些研究人员来说，较大的类别偏差以及类别分布的大幅变化可能看起来是人为设计且不切实际的。然而，在现实世界中，类别偏差达到 10^{-1} 和 10^{-2} 的情况非常普遍，甚至在某些领域观察到了高达 10^{-6} 的偏差（Clearwater 和 Stern，1991；Fawcett 和 Provost，1996；Kubat 等人，1998；Saitta 和 Neri，1998）。类别分布的显著变化也并非不切实际。例如，在医疗决策中，传染病可能会导致某种疾病的发病率随时间上升。在欺诈检测领域，欺诈的比例在不同月份和不同地点之间存在显著差异（Fawcett 和 Provost，1997）。生产实践的变化可能会导致不合格产品的比例发生变化。



Inst no.	Class True	Class Hyp	Score
1	p	Y	0.99999
2	p	Y	0.99999
3	p	Y	0.99993
4	p	Y	0.99986
5	p	Y	0.99964
6	p	Y	0.99955
7	n	Y	0.68139
8	n	Y	0.50961
9	n	N	0.48880
10	n	N	0.44951

图 4. 10 个实例的得分和分类结果，以及由此得出的 ROC 曲线。

由一条生产线生产的产品数量可能会增加或减少。在上述每个例子中，某一类别的普遍程度可能会发生巨大变化，但不会改变该类别的基本特征，即目标概念。

在信息检索中，准确率和召回率是评估检索（分类）性能的常用指标（刘易斯，1990 年，1991 年）。在静态文档集有时可以被假定的情况下，通常会使用准确率 - 召回率曲线图；然而，在诸如网页检索这样的动态环境中，与查询无关的网页数量（N）可能比相关网页数量（P）多几个数量级，并且随着时间的推移，N 可能会持续增加，因为网页在不断生成。

为了观察类别不平衡的影响，可参考图 5 中的曲线，该图展示了使用 ROC 曲线和精确率 - 召回率曲线评估的两个分类器。在图 5a 和 5b 中，测试

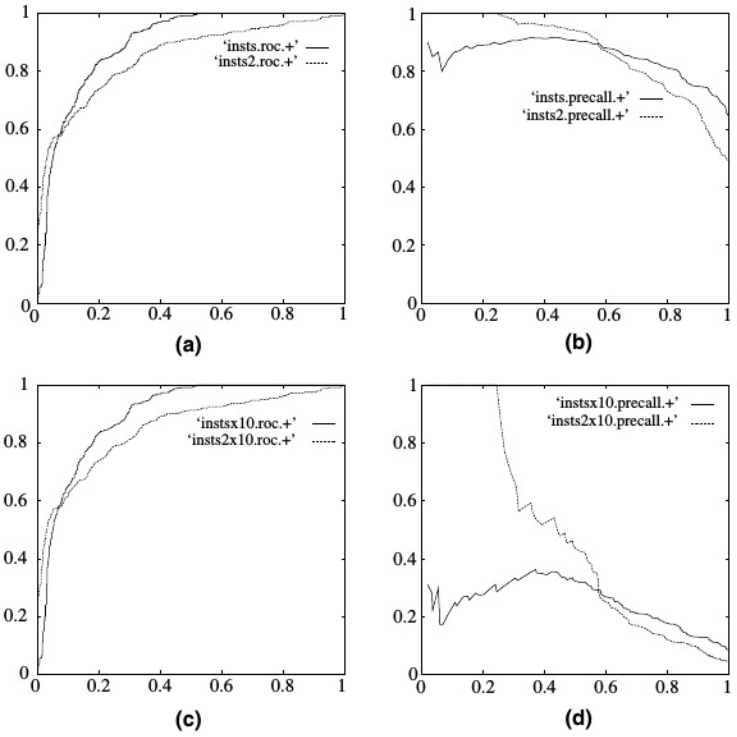


图 5.类别不平衡情况下的 ROC 曲线和精确率 - 召回率曲线。（a）ROC 曲线，类别比例 1:1；（b）精确率 - 召回率曲线，类别比例 1:1；（c）ROC 曲线，类别比例 1:10；（d）精确率 - 召回率曲线，类别比例 1:10。

该数据集具有平衡的 1:1 类别分布。图 5c 和 5d 展示了在相同领域中的同一分类器，但负例的数量增加了 10 倍。请注意，分类器和潜在概念未发生变化，只有类别分布不同。观察图 5a 和 5c 中的 ROC 曲线是相同的，而图 5b 和 5d 中的精确率 - 召回率曲线则有显著差异。在某些情况下，随着分布的变化，关于哪个分类器性能更优的结论可能会改变。

4.3. 创建评分分类器

许多分类器模型是离散型的：它们被设计成仅从每个测试实例中生成一个类别标签。然而，我们通常希望从分类器生成完整的 ROC 曲线，而不仅仅是单个点。为此，我们希望从分类器生成分数，而不仅仅是类别标签。有几种方法可以生成这样的分数。

许多离散分类器模型可以通过“查看内部”它们所保存的实例统计信息轻松地转换为评分分类器。例如，决策树根据节点处实例的比例确定叶节点的类别标签；类别决策只是最常见的类别。这些类别比例可以用作评分（普罗沃斯特和多明戈斯，2001 年）。规则学习器在规则置信度方面保存类似的统计信息，与实例匹配的规则的置信度可以用作评分（费塞特，2001 年）。

即使分类器仅产生类别标签，但对它们的集合进行处理也能生成一个分数。MetaCost（多明戈斯，1999 年）采用装袋法生成一组离散分类器的集成，每个分类器都投出一票。这些票的集合可用于生成一个分数。3

最后，可以采用评分和投票相结合的方式。例如，规则可以提供基本的概率估计值，然后这些估计值可用于加权投票（Fawcett，2001）。

5. 高效生成 ROC 曲线

给定一个测试集，我们通常希望从中高效地生成一条 ROC 曲线。我们可以利用阈值分类的单调性：对于给定的阈值被分类为正的任何实例，在所有更低的阈值下也会被分类为正。因此，我们可以简单地按照 f 分数从高到低对测试实例进行排序，然后逐个处理列表中的实例，同时不断更新真阳性（TP）和假阳性（FP）的值。通过这种方式，可以对列表进行一次线性扫描来创建 ROC 曲线。

算法如算法 1 所示。TP 和 FP 均从 0 开始。对于每个正例，我们将 TP 加 1；对于每个负例，我们将 FP 加 1。我们维护一个 ROC 点的栈 R，在处理完每个实例后将新的点压入 R 中。最终输出为栈 R，其中包含 ROC 曲线上的点。

设 n 为测试集中的点数。此算法需要进行 O（nlogn）的排序，然后沿列表向下进行 O(n) 的扫描，因此总复杂度为 O（nlogn）。

陈述 7 至 10 需要一些解释。这些陈述对于正确处理由 f 得分相同的实例序列是必要的。考虑图 6 中所示的 ROC 曲线。假设我们有一个测试集，其中存在一个实例序列，包含 4 个负例和 6 个正例，它们的 f 得分相同。算法 1 第 1 行的排序不会对这些实例施加任何特定的顺序，因为它们的 f 得分相等。那么当我们创建 ROC 曲线时会发生什么？在一种极端情况下，所有正例都出现在序列的开头，从而生成图 6 中所示的“乐观”的上 L 段。在相反的情况下

Algorithm 1. Efficient method for generating ROC points
Inputs: L, the set of test examples; $f(i)$, the probabilistic classifier's estimate that example i is positive; P and N, the number of positive and negative examples.
Outputs: R, a list of ROC points increasing by *fp rate*.
Require: $P > 0$ and $N > 0$

```
1: L_sorted ← L sorted decreasing by f scores
2: FP ← TP ← 0
3: R ← ⟨⟩
4: f_prev ← −∞
5: i ← 1
6: while i ⩽ |L_sorted| do
7:    if f(i) ≠ f_prev then
8:       push (FP/N, TP/P) onto R
9:       f_prev ← f(i)
10:   end if
11:   if L_sorted[i] is a positive example then
12:      TP ← TP + 1
13:   else      /* i is a negative example */
14:      FP ← FP + 1
15:   end if
16:   i ← i + 1
17: end while
18: push (FP/N, TP/P) onto R   /* This is (1,1) */
19: end
```

[3] MetaCost actually works in the opposite direction because its goal is to generate a discrete classifier. It first creates a probabilistic classifier, then applies knowledge of the error costs and class skews to relabel the instances so as to ‘‘optimize’’ their classifications. Finally, it learns a specific discrete classifier from this new instance set. Thus, MetaCost is not a good method for creating a scoring classifier, though its bagging method may be.

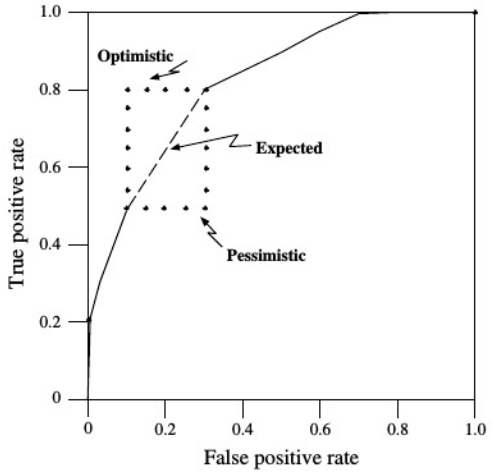


图 6. 由 10 个得分相同的实例序列得出的乐观、悲观和预期的 ROC 段。

极端情况下，所有负例都出现在序列的开头，于是我们得到图 6 中的“悲观”下限 L。实例的任何混合排序都会在由这两个极端形成的矩形内产生不同的阶梯段。然而，ROC 曲线应当代表分类器的预期性能，在缺乏其他信息的情况下，这是悲观和乐观段的平均值。该平均值是矩形的对角线，这可以通过在处理完所有 f 值相等的实例之前不输出 ROC 点来实现，这正是 f_{prev} 变量和第 7 行的 if 语句所完成的工作。

得分相同的实例可能看起来不寻常，但在某些分类器模型中却很常见。例如，如果我们使用决策树中节点的实例数量来对实例进行评分，那么一个大的、高熵的叶节点可能会产生许多两类得分相同的实例。如果不对这些实例进行平均处理，那么生成的 ROC 曲线将对测试集的排序敏感，不同的排序可能会产生非常具有误导性的曲线。这在计算 ROC 曲线下面积时尤其关键，这在第 7 节中有所讨论。考虑一个决策树中包含一个叶节点，其中包含 n 个正例和 m 个负例。所有被分类到这个叶节点的实例都将被赋予相同的分数。图 6 中的矩形大小为 PN。纳米

而且，如果不对这些情况取平均值，那么这一个叶片就可能导致 ROC 曲线下面积的误差高达 2PN。米

6. ROC 凸包

ROC 曲线的一个优点在于，它能够直观地展示和组织分类器的性能，而不受类别分布或错误成本的影响。这种能力在研究类别分布不均衡或成本敏感学习时变得非常重要。研究人员可以绘制一组分类器的性能曲线，而该曲线对于操作条件（类别分布倾斜和错误成本）的变化保持不变。尽管这些条件发生变化，感兴趣的区域可能会改变，但曲线本身不会改变。

普罗沃斯特和法塞特（1998 年，2001 年）表明，一组操作条件可以很容易地在 ROC 空间中转换为所谓的等性能线。ROC 空间中的两点 (FP_1,TP_1) 和 (FP_2,TP_2) 具有相同的性能，如果

$$\frac{TP_2 - TP_1}{FP_2 - FP_1} = \frac{c(\boldsymbol{Y},\boldsymbol{n})p(\boldsymbol{n})}{c(\boldsymbol{N},\boldsymbol{p})p(\boldsymbol{p})} = m \tag{1}$$

此方程定义了等性能线的斜率。在斜率为 m 的直线上所有点对应的分类器具有相同的预期成本。每组类别和成本分布都定义了一组等性能线。斜率更靠西北（具有更大的真阳性截距）的线更好，因为它们对应的分类器具有更低的预期成本。更一般地说，一个分类器是潜在最优的，当且仅当它位于 ROC 空间中点集的凸包上。ROC 空间中点集的凸包被称为相应分类器集合的 ROC 凸包（ROCCH）。

图 7a 展示了四条 ROC 曲线（A 至 D）及其凸包（标注为 CH）。D 不在凸包上，显然是次优的。B 也不在凸包上，因此对于任何条件而言都不是最优的。凸包仅由 A 和 C 曲线上的点构成。因此，如果我们寻求最优的分类性能，可以完全排除 B 和 D 分类器。此外，我们还可以从 A 和 C 中移除不在凸包上的任何离散点。

图 7b 再次展示了 A 和 C 曲线，并且有两条明确的等性能线 a 和 b。假设负样本数量是正样本数量的 10 倍，但误报和漏报的成本相同。根据公式（1），m = 10，斜率为 m = 10 的最西北的线是 a，它与分类器 A 相切，因此在这种情况下，分类器 A 将是性能最佳的分类器。

再考虑另一种情形，其中正例和负例的数量是均衡的，但假阴性的代价是假阳性的 10 倍。根据公式（1），m = 1/10。斜率为 1/10 的最西北的直线会是直线 b，与分类器 C 相切。对于这些条件，C 是最优分类器。

如果我们想在 A 和 C 之间的凸包上某处生成一个分类器，可以在两者之间进行插值。第 10 节解释了如何生成这样的分类器。

这种 ROCCH 表达式具有若干有用的含义。由于只有凸包上的分类器才有可能是最优的，所以无需保留其他分类器。分类器的操作条件可以转换为一条等性能线，进而可用于识别 ROCCH 的一部分。随着条件的变化，凸包本身不会改变，只有感兴趣的部分会发生变化。

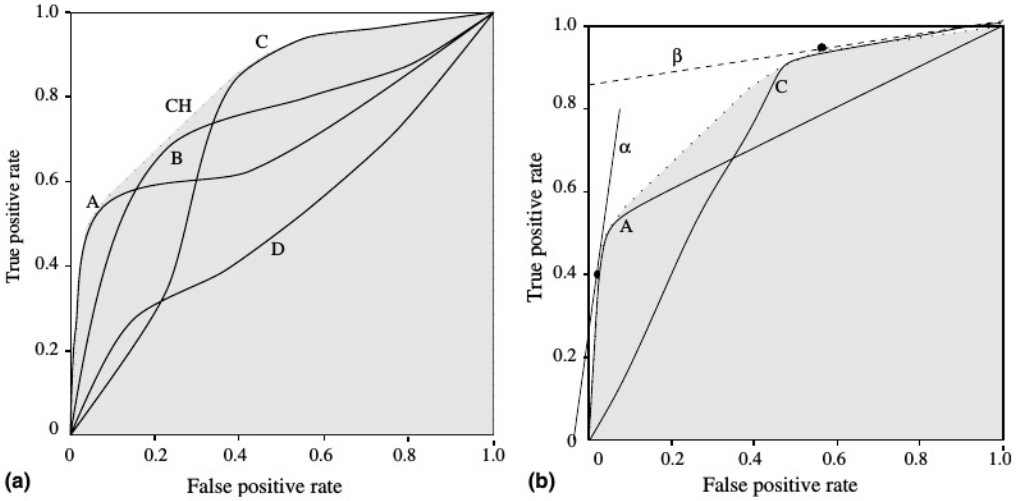


图 7.（a）ROC 凸包确定了潜在的最优分类器。（b）直线 a 和 b 分别展示了在不同条件下最优的分类器。

7. 受试者工作特征曲线下的面积（AUC）

ROC 曲线是分类器性能的二维描述。为了比较分类器，我们可能希望将 ROC 性能缩减为一个代表预期性能的单一标量值。一种常见的方法是计算 ROC 曲线下方的面积，简称为 AUC（布拉德利，1997 年；汉利和麦克尼尔，1982 年）。由于 AUC 是单位正方形面积的一部分，其值始终在 0 到 1.0 之间。然而，由于随机猜测会产生从（0,0）到（1,1）的对角线，其面积为 0.5，所以任何现实中的分类器的 AUC 值都不应低于 0.5。

AUC 具有一个重要的统计特性：分类器的 AUC 等同于该分类器将随机选取的正例排在随机选取的负例之前的概率。这与 Wilcoxon 秩和检验（Hanley 和 McNeil，1982 年）等价。AUC 还与基尼系数（Breiman 等人，1984 年）密切相关，基尼系数是 ROC 曲线与对角线之间面积的两倍。Hand 和 Till（2001 年）指出，$\text{Gini} + 1 = 2 \cdot \text{AUC}$。

图 8a 展示了两条 ROC 曲线 A 和 B 下的面积。曲线 B 的面积更大，因此平均性能更优。图 8b 展示了二分类器 A 和评分分类器 B 的曲线下面积。分类器 A 表示当 B 使用单一固定阈值时的性能。尽管在固定点（A 的阈值）处两者的性能相同，但随着远离该点，A 的性能会逊于 B。

在 ROC 空间中的特定区域，高 AUC 值的分类器的表现可能不如低 AUC 值的分类器。图 8a 展示了一个这样的例子：分类器 B 通常优于 A，但在 FPrate（假阳性率）大于 0.6 时，A 的表现更佳。

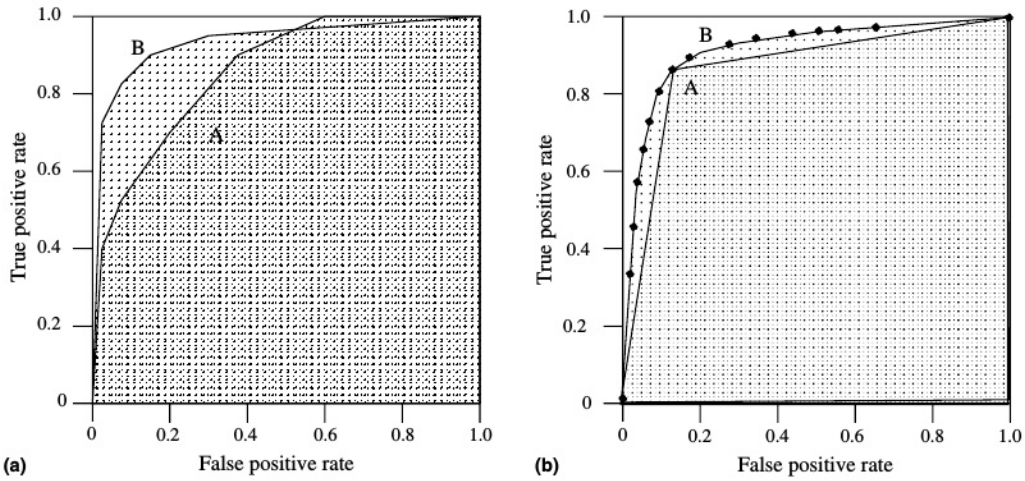


图 8. 两张 ROC 曲线图。左边的图展示了两条 ROC 曲线下方的面积。右边的图展示了离散分类器（A）和概率分类器（B）的 ROC 曲线下方的面积。

Algorithm 2. Calculating the area under an ROC curve
Inputs: L, the set of test examples; $f(i)$, the probabilistic classifier's estimate that example i is positive; P and N, the number of positive and negative examples.
Outputs: A, the area under the ROC curve.
Require: $P > 0$ and $N > 0$

```
1: L_sorted ← L sorted decreasing by f scores
2: FP ← TP ← 0
3: FP_prev ← TP_prev ← 0
4: A ← 0
5: f_prev ← −∞
6: i ← 1
7: while i ⩽ |L_sorted| do
8:    if f(i) ≠ f_prev then
9:       A ← A + TRAPEZOID_AREA(FP, FP_prev,
               TP, TP_prev)
10:      f_prev ← f(i)
11:      FP_prev ← FP
12:      TP_prev ← TP
13:   end if
14:   if i is a positive example then
15:      TP ← TP + 1
16:   else      /* i is a negative example */
17:      FP ← FP + 1
18:   end if
19:   i ← i + 1
20: end while
21: A ← A + TRAPEZOID_AREA(N, FP_prev, N, TP_prev)
22: A ← A/(P × N)  /* scale from P × N onto the unit
    square */
23: end
1: function TRAPEZOID_AREA(X1, X2, Y1, Y2)
2: Base ← |X1 − X2|
3: Height_avg ← (Y1 + Y2)/2
4: return Base × Height_avg
5: end function
```

微弱的优势。但在实际应用中，AUC 表现得非常出色，当需要一个通用的预测性度量时，它常常被采用。

可以对算法 1 进行细微修改，如算法 2 所示，轻松计算出 AUC。该算法不是收集 ROC 点，而是将梯形的连续面积累加到 A 中。使用梯形而非矩形是为了在各点之间取平均值，如图 6 所示。最后，算法将 A 除以单位正方形的总面积，以将值缩放到单位正方形。

8. 平均 ROC 曲线

尽管可以使用 ROC 曲线来评估分类器，但在用其得出关于分类器优劣的结论时应谨慎行事。一些研究人员认为，只需将分类器在 ROC 空间中绘图，然后观察哪个分类器占优，就可以据此选择最佳分类器。这种观点具有误导性；这就好比仅根据单个测试集得出的一组准确率数值来取最大值。若没有方差的度量，我们就无法对这些分类器进行比较。

如果原始实例可用，那么平均 ROC 曲线就很简单。给定通过交叉验证或自助法生成的测试集 $T_1, T_2, \ldots, T_n$，我们可以简单地将这些实例按照其分配的分数进行合并排序，形成一个大的测试集 T_M。然后，我们可以在 T_M 上运行诸如算法 1 这样的 ROC 曲线生成算法，并绘制结果。然而，使用多个测试集的主要原因是得出方差的度量，而这种简单的合并方法无法提供。我们需要一种更复杂的方法，即在不同点对各个曲线进行采样，并对样本进行平均。

ROC 空间是二维的，而任何平均值必然是单维的。ROC 曲线可以投影到单个维度上并进行常规平均，但这会引发投影是否恰当的问题，或者更确切地说，是否保留了感兴趣的特征。答案取决于平均曲线的原因。本节介绍了两种平均 ROC 曲线的方法：垂直平均和阈值平均。

图 9a 展示了五条要进行平均的 ROC 曲线。每条曲线包含一千个点，并且存在一些凹陷。图 9b 展示了通过合并这五个测试集并计算其综合 ROC 曲线所形成的曲线。图 9c 和 9d 展示了通过对五条单独的 ROC 曲线进行抽样而形成的平均曲线。误差条为 95% 的置信区间。

8.1. 垂直平均

垂直平均法是在固定假阳性率的情况下对 ROC 曲线进行垂直采样，并对相应的真阳性率进行平均。当研究者确实能够固定假阳性率，或者需要单一维度的变异性度量时，这种平均方法是合适的。普罗沃斯特等人（1998 年）在其关于 k 折交叉验证分类器的 ROC 曲线平均的研究中使用了这种方法。

在这种方法中，每条 ROC 曲线都被视为一个函数 R_i，使得真阳性率 = R_i(fp 假阳性率）。这是通过为每个假阳性率选择最大的真阳性率，并在必要时在各点之间进行插值来实现的。平均 ROC 曲线即为函数 $\bar{R}$。$\bar{R}(\text{fp rate}) = \text{mean}[R_i(\text{fp rate})]$。绘制一个我们可以从 $\bar{R}$ 中抽取平均 ROC 曲线 $\bar{R}$ 在某些时候定期在假阳性率轴上等距分布。假阳性率均值的置信区间是基于常见的二项分布假设来计算的。

算法 3 计算一组 ROC 点的垂直平均值。它将平均值保存在数组 TPavg 中。

为了清晰起见，此处省略了几个扩展内容。该算法很容易扩展到

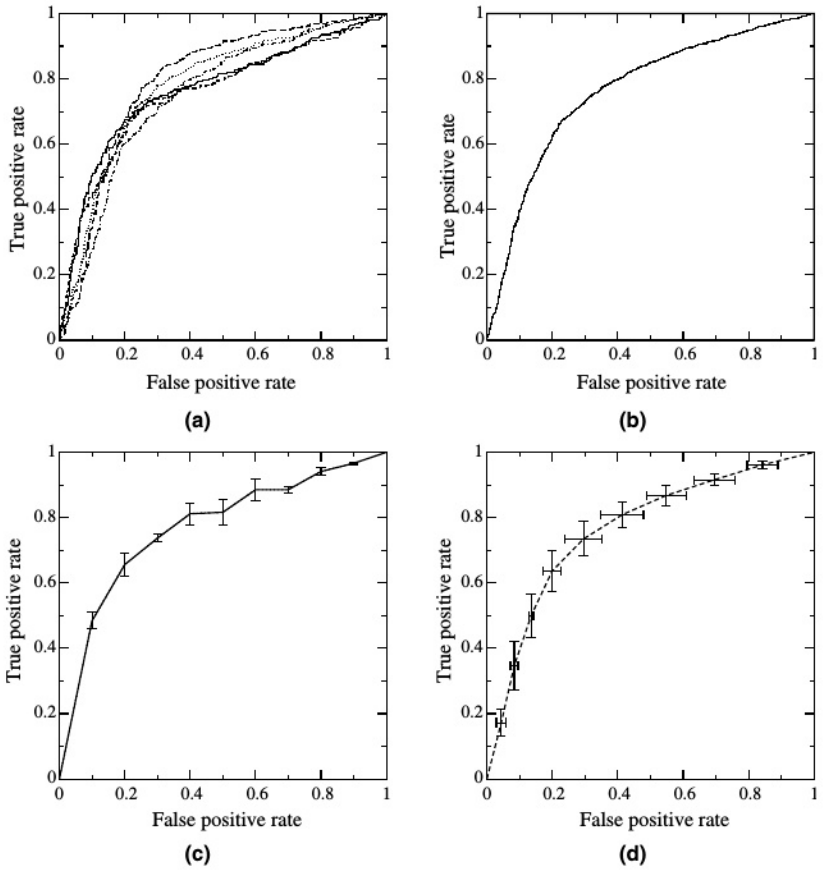


图 9. ROC 曲线的平均值。 (a) 五个实例样本的 ROC 曲线，(b) 由五个样本合并而成的 ROC 曲线，(c) 垂直方向上的平均曲线，(d) 按阈值平均的曲线。

为了绘制置信区间，需要计算样本的标准差。此外，函数 TP_FOR_FP 还可以进行一定程度的优化。由于它仅在 FP 值单调递增时被调用，因此不必每次从头开始扫描每个 ROC 数组；它可以记录上次看到的点，并从该数组中的该点开始初始化 i 。

图 9c 展示了图 9a 中五条曲线的垂直平均值。曲线上的垂直条形表示 ROC 均值的 95%置信区间。对于此平均曲线，曲线在从 0 到 1 的 FP 率范围内以 0.1 为间隔进行采样。可以对曲线进行更精细的采样，但置信区间条形可能会难以阅读。

8.2. 阈值平均

垂直平均法的优势在于，其平均值是基于单一的因变量——真正例率，这简化了置信区间的计算。然而，霍尔特（2002 年）指出，自变量——假正例率，通常不受研究人员的直接控制。或许更可取的做法是使用一个研究人员能够直接控制其值的自变量（比如分类器得分的阈值）来对 ROC 点进行平均。

阈值平均法实现了这一点。与垂直平均法基于 ROC 空间中各点的位置进行采样不同，阈值平均法是基于生成这些点的阈值进行采样。该方法必须生成一组阈值用于采样，然后对于每个阈值，找到每条 ROC 曲线对应的点并取其平均值。

算法 4 展示了实现此目的的基本方法。它生成一个分类器得分数组 T，该数组按从大到小的顺序排序，并用作阈值集。这些阈值以由 samples（所需样本数）确定的固定间隔进行采样。对于给定的阈值，算法从每条 ROC 曲线中选择得分小于或等于该阈值的最大点。然后分别沿 X 轴和 Y 轴对这些点求平均值，并将中心点返回到 Avg 数组中。

图 9d 展示了对图 9a 中的五条曲线按阈值进行平均后的结果。所得曲线在 X 和 Y 方向上都有平均值点和置信区间。图中所示的区间为 95% 的置信水平。

阈值平均相对于垂直平均存在一些较小的局限性。要执行阈值平均，我们需要为每个点分配分类器得分。此外，第 4.1 节指出，分类器得分

[4] We assume the ROC points have been generated by an algorithm like 1 that deals correctly with equally scored instances.

Algorithm 3. Vertical averaging of ROC curves
Inputs: *samples*, the number of FP samples; *nrocs*, the number of ROC curves to be sampled, *ROCS*[*nrocs*], an array of *nrocs* ROC curves; *npts*[*m*], the number of points in ROC curve *m*. Each ROC point is a structure of two members, the rates *fpr* and *tpr*.
Output: Array *tpravg*[*samples* + 1], containing the vertical averages.

```
1: s ← 1
2: for fpr_sample = 0 to 1 by 1/samples do
3:    tprsum ← 0
4:    for i = 1 to nrocs do
5:       tprsum ← tprsum + TPR_FOR_FPR(fpr_sample,
                    ROCS[i], npts[i])
6:    end for
7:    tpravg[s] ← tprsum/nrocs
8:    s ← s + 1
9: end for
10: end
1: function TPR_FOR_FPR(fpr_sample, ROC, npts)
2: i ← 1
3: while i < npts and ROC [i + 1].fpr ⩽ fpr_sample do
4:    i ← i + 1
5: end while
6: if ROC[i].fpr = fpr_sample then
7:    return ROC[i].tpr
8: else
9:    return INTERPOLATE(ROC[i], ROC [i + 1], fpr_sample)
10: end if
11: end function
1: function INTERPOLATE(ROCP1, ROCP2, X)
2: slope = (ROCP2.tpr − ROCP1.tpr)/(ROCP2.fpr −
          ROCP1.fpr)
3: return ROCP1.tpr + slope · (X − ROCP1.fpr)
4: end function
```

Algorithm 4. Threshold averaging of ROC curves
Inputs: *samples*, the number of threshold samples; *nrocs*, the number of ROC curves to be sampled; *ROCS*[*nrocs*], an array of *nrocs* ROC curves sorted by score; *npts*[*m*], the number of points in ROC curve *m*. Each ROC point is a structure of three members, *fpr*, *tpr* and score.
Output: *Avg*[*samples* + 1], an array of (*X*, *Y*) points constituting the average ROC curve.
Require: *samples* > 1

```
1: initialize array T to contain all scores of all ROC
   points
2: sort T in descending order
3: s ← 1
4: for tidx = 1 to length(T) by int(length(T)/samples) do
5:    fprsum ← 0
6:    tprsum ← 0
7:    for i = 1 to nrocs do
8:       p ← ROC_POINT_AT_THRESHOLD(ROCS[i], npts[i],
             T[tidx])
9:       fprsum ← fprsum + p.fpr
10:      tprsum ← tprsum + p.tpr
11:   end for
12:   Avg[s] ← (fprsum/nrocs, tprsum/nrocs)
13:   s ← s + 1
14: end for
15: end
1: function ROC_POINT_AT_THRESHOLD(ROC, npts, thresh)
2: i ← 1
3: while i ⩽ npts and ROC[i]. score > thresh do
4:    i ← i + 1
5: end while
6: return ROC[i]
7: end function
```

不应在不同模型类别之间进行比较。由于这一原因，从不同模型类别平均得出的 ROC 曲线可能会产生误导，因为这些分数可能无法直接比较。

最后，马克斯卡西和普罗沃斯特（2004 年）研究了生成 ROC 曲线置信带的不同技术。他们研究了来自垂直和阈值平均的置信区间，以及来自医学领域的三种生成置信带的方法（同时联合置信区域、基于 Working-Hotelling 的置信带和固定宽度置信带）。关于这些技术、其假设以及实证研究的更详细讨论，请参阅他们的论文。

9. 具有多于两类的决策问题

到目前为止的讨论仅涉及两类情况，而 ROC 文献中的很多内容也一直基于这一假设。ROC 分析在医学领域中应用广泛，在二分类诊断问题（存在或不存在异常状况）中，这种决策过程很常见。两个坐标轴代表了分类器在两类之间所做决策的误差（假阳性）与收益（真阳性）之间的权衡。由于二分类问题存在对称性，所以很多分析都很直接。最终的性能可以在二维空间中绘图，这便于直观理解。

9.1. 多类别 ROC 曲线图

当类别超过两类时，若要对整个空间进行管理，情况就会变得复杂得多。对于 n 个类别，混淆矩阵会变成一个 n×n 的矩阵，其中包含 n 个正确的分类（主对角线上的元素）以及 n^2 n 种可能的错误（非对角线上的元素）。此时，我们面对的不再是真阳性（TP）和假阳性（FP）之间的权衡，而是 n 项收益和 n^2 n 种错误。仅对于三个类别，其表面就变成了一个 3^{-2} 3 = 6 维的多面体。莱恩（2000 年）概述了其中涉及的问题以及解决这些问题的前景。斯里尼瓦桑（1999 年）已经证明

ROC 凸包背后的分析可扩展到多类别和多维凸包。

处理 n 个类别的一种方法是生成 n 个不同的 ROC 图，每个类别对应一个。称此为类别参考公式。具体来说，如果 C 是所有类别的集合，ROC 图 i 绘制的是将类别 c_i 作为正类，而将所有其他类别作为负类的分类性能，即

$$P_i = c_i \tag{2}$$

$$N_i = \bigcup_{j \neq i} c_j \in C \tag{3}$$

虽然这种表述方式很方便，但它削弱了 ROC 曲线的一个优点，即它们对类别不平衡不敏感（见第 4.2 节）。因为每个 N_i 由 n 1 个类别组成，这些类别内的流行率变化可能会改变 c_is ROC 曲线。例如，假设某个类别 c_k 2 N 特别容易识别。针对类别 c_i，i 5 k 的分类器可能会利用 c_k 的某些特征来为 c_k 实例生成低分。增加 c_k 的流行率可能会改变分类器的性能，这相当于通过增加其析取项之一的流行率来改变目标概念。这反过来又会改变 ROC 曲线。不过，只要注意到这一点，这种方法在实践中通常效果不错，并且在评估方面提供了相当的灵活性。

9.2. 多类别 AUC

AUC 是衡量两类之间区分度的指标。在二分类问题中，AUC 是一个单一的标量值，但在多分类问题中，会引入如何组合多个成对区分度值的问题。关于这些问题的精彩讨论，读者可参阅 Hand 和 Till（2001）的文章。

普罗沃斯特和多明戈斯（2001 年）在其关于概率估计树的研究中提出了一种计算多类别 AUC 的方法。他们通过依次生成每个类别的参考 ROC 曲线，测量曲线下面积，然后将 AUC 加权求和，权重为该参考类别在数据中的流行率来计算多类别问题的 AUC。更确切地说，他们定义了

$$\mathrm{AUC}_{\mathrm{total}} = \sum_{c_i \in C} \mathrm{AUC}(c_i) \cdot p(c_i)$$

其中 AUC(c_i) 是 c_i 的类别参考 ROC 曲线下面积，如公式（3）所示。此定义仅需进行 jCj 次 AUC 计算，因此其总体复杂度为 O（jCjnlogn）。

普罗沃斯特和多明戈斯的 AUC 表达式的优点在于 $\mathrm{AUC}_{\mathrm{total}}$ 是直接从类别参考 ROC 曲线生成的，而且这些曲线易于生成和可视化。缺点是类别参考 ROC 曲线对类别分布和错误成本很敏感，因此这种 $\mathrm{AUC}_{\mathrm{total}}$ 的表达式也是如此。

Hand 和 Till（2001 年）在推导多类别 AUC 的泛化形式时采用了不同的方法。他们希望得到一种不受类别分布和错误成本影响的度量标准。其推导过程过于复杂，这里无法详述，但它是基于这样一个事实：AUC 等同于分类器将随机选取的正例排在随机选取的负例之前的概率。基于这种概率形式，他们推导出了一种衡量类别间无权配对可分性的公式。他们将这种度量标准称为 M，其等同于：

$$\mathrm{AUC}_{\mathrm{total}} = \frac{2}{|C|(|C|-1)} \sum_{\{c_i, c_j\} \in C} \mathrm{AUC}(c_i, c_j)$$

其中 n 表示类别数量，AUC(c_i,c_j) 表示涉及类别 c_i 和 c_j 的二分类 ROC 曲线下面积。该求和运算涵盖所有类别对（不论顺序），共有 jCj(jCj - 1)/2 个这样的类别对，因此其度量的时间复杂度为 O(jCj^{2}nlogn)。尽管 Hand 和 Till 的公式推导合理且不受类别分布变化的影响，但要直观地展示出所计算面积的曲面却并非易事。

10.插值分类器

有时，所期望的分类器性能并非由任何现有的分类器直接产生，而是介于两个现有分类器之间。可以通过对每个分类器的决策进行采样来获得期望的性能。采样比例将决定最终分类性能的位置。

以一个具体的例子来说，考虑一下 2000 年 CoIL 挑战赛中的决策问题（范德普滕和索梅伦，2000 年）。在这个挑战中，有一组 4000 名客户，我们希望向他们推销一种新的保险政策。我们的预算规定，我们只能向其中 800 人进行推销，所以我们希望挑选出最有可能对推销做出回应的 800 人。预计响应者的类别先验概率为 6%，所以在 4000 人的群体中，我们预计会有 240 名响应者（正例）和 3760 名非响应者（反例）。

假设我们生成了两个分类器 A 和 B，它们根据客户购买保险的概率对客户进行评分。在 ROC 空间中，A 位于 （0.1, 0.2） 处，B 位于 （0.25, 0.6） 处，如图 10 所示。我们希望向恰好 800 人进行营销，因此我们的解决方案约束条件为假阳性率×3760 + 真阳性率×240 = 800。如果使用 A，我们预计会有 0.1×3760 + 0.2×240 = 424 名候选人，这太少了。如果使用 B，我们预计会有 0.25×3760 + 0.6×240 = 1084 名候选人，这又太多了。我们想要一个介于 A 和 B 之间的分类器。

如图 10 所示，解的约束条件以虚线表示。它与 A 和 B 之间的连线在 C 点相交，C 点的坐标约为（0.18，0.42）。位于 C 点的分类器能给出我们期望的性能，我们可以通过线性插值来实现。计算 C 点在 A 和 B 连线上的比例距离 k：

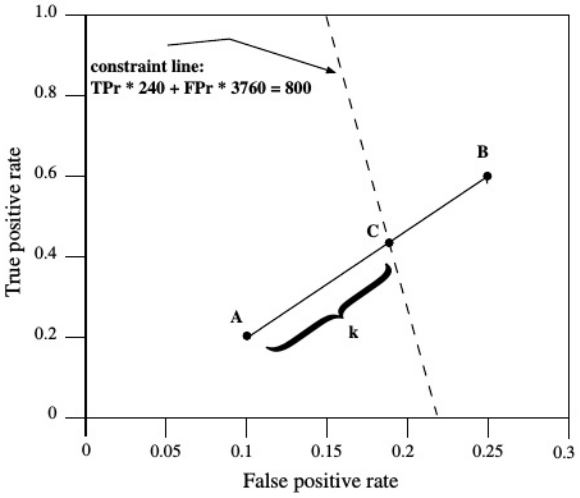


图 10. 插值分类器。

$$k = \frac{0.18 - 0.1}{0.25 - 0.1} \approx 0.53$$

因此，如果我们以 0.53 的概率对 B 的决策进行采样，以 1 0.53 = 0.47 的概率对 A 的决策进行采样，那么我们就能达到 C 的性能。在实际操作中，这种分数采样可以通过从每个分类器的决策中随机采样来实现：对于每个实例，生成一个 0 到 1 之间的随机数。如果随机数大于 k，则对实例应用分类器 A 并报告其决策，否则将实例传递给 B。

11.结论

ROC 曲线是一种非常有用的工具，可用于可视化和评估分类器。与准确率、错误率或错误成本等标量度量相比，它们能够提供更丰富的分类性能度量。由于它们将分类器的性能与类别偏差和错误成本分离开来，因此相较于其他评估指标（如精确率 - 召回率曲线和提升曲线）具有优势。然而，与任何评估指标一样，明智地使用它们需要了解其特点和局限性。希望本文能够增进人们对 ROC 曲线的普遍了解，并有助于在模式识别领域推广更好的评估实践。

参考文献

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn. 30 (7), 1145–1159.

Breiman, L., Friedman, J., Olshen, R., Stone, C., 1984. Classification and Regression Trees. Wadsworth International Group, Belmont, CA.

Clearwater, S., Stern, E., 1991. A rule-learning program in high energy physics event classification. Comput. Phys. Commun. 67, 159–182.

Domingos, P., 1999. MetaCost: A general method for making classifiers cost-sensitive. In: Proc. Fifth ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining, pp. 155–164.

Egan, J.P., 1975. Signal detection theory and ROC analysis, Series in Cognition and Perception. Academic Press, New York.

Fawcett, T., 2001. Using rule sets to maximize ROC performance. In: Proc. IEEE Internat. Conf. on Data Mining (ICDM-2001), pp. 131– 138.

Fawcett, T., Provost, F., 1996. Combining data mining and machine learning for effective user profiling. In: Simoudis, E., Han, J., Fayyad, U. (Eds.), Proc. Second Internat. Conf. on Knowledge Discovery and Data Mining. AAAI Press, Menlo Park, CA, pp. 8–13.

Fawcett, T., Provost, F., 1997. Adaptive fraud detection. Data Mining and Knowledge Discovery 1 (3), 291–316.

Flach, P., Wu, S., 2003. Repairing concavities in ROC curves. In: Proc. 2003 UK Workshop on Computational Intelligence. University of Bristol, pp. 38–44.

Forman, G., 2002. A method for discovering the insignificance of ones best classifier and the unlearnability of a classification task. In: Lavrac, N., Motoda, H., Fawcett, T. (Eds.), Proc. First Internat. Workshop on Data Mining Lessons Learned (DMLL-2002). Available from: http://www.purl.org/NET/tfawcett/DMLL-2002/Forman.pdf.

Hand, D.J., Till, R.J., 2001. A simple generalization of the area under the ROC curve to multiple class classification problems. Mach. Learning 45 (2), 171–186.

Hanley, J.A., McNeil, B.J., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143, 29– 36.

Holte, R., 2002. Personal communication.

Kubat, M., Holte, R.C., Matwin, S., 1998. Machine learning for the detection of oil spills in satellite radar images. Machine Learning 30 (2–3), 195–215.

Lane, T., 2000. Extensions of ROC analysis to multi-class domains. In: Dietterich, T., Margineantu, D., Provost, F., Turney, P. (Eds.), ICML-2000 Workshop on Cost-Sensitive Learning.

Lewis, D., 1990. Representation quality in text classification: An intro-duction and experiment. In: Proc. Workshop on Speech and Natu-ral Language. Morgan Kaufmann, Hidden Valley, PA, pp. 288– 295.

Lewis, D., 1991. Evaluating text categorization. In: Proc. Speech and Natural Language Workshop. Morgan Kaufmann, pp. 312–318.

Macskassy, S., Provost, F., 2004. Confidence bands for ROC curves: Methods and an empirical study. In: Proc. First Workshop on ROC Analysis in AI (ROCAI-04).

Provost, F., Domingos, P., 2001. Well-trained PETs: Improving prob-ability estimation trees, CeDER Working Paper #IS-00-04, Stern School of Business, New York University, NY, NY 10012.

Provost, F., Fawcett, T., 1997. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proc. Third Internat. Conf. on Knowledge Discovery and Data Mining (KDD-97). AAAI Press, Menlo Park, CA, pp. 43–48.

Provost, F., Fawcett, T., 1998. Robust classification systems for imprecise environments. In: Proc. AAAI-98. AAAI Press, Menlo Park, CA, pp. 706–713. Available from: <http://www.purl.org/NET/tfawcett/papers/aaai98-dist.ps.gz>.

Provost, F., Fawcett, T., 2001. Robust classification for imprecise environments. Mach. Learning 42 (3), 203–231.

Provost, F., Fawcett, T., Kohavi, R., 1998. The case against accuracy estimation for comparing induction algorithms. In: Shavlik, J. (Ed.), Proc. ICML-98. Morgan Kaufmann, San Francisco, CA, pp. 445–453. Available from: <http://www.purl.org/NET/tfawcett/papers/ICML98-final.ps.gz>.

Saitta, L., Neri, F., 1998. Learning in the ‘‘real world’’. Mach. Learning 30, 133–163.

Spackman, K.A., 1989. Signal detection theory: Valuable tools for evaluating inductive learning. In: Proc. Sixth Internat. Workshop on Machine Learning. Morgan Kaufman, San Mateo, CA, pp. 160–163.

Srinivasan, A., 1999. Note on the location of optimal classifiers in n-dimensional ROC space. Technical Report PRG-TR-2-99, Oxford University Computing Laboratory, Oxford, England. Available from: <http://citeseer.nj.nec.com/srinivasan99note.html>.

Swets, J., 1988. Measuring the accuracy of diagnostic systems. Science 240, 1285–1293.

Swets, J.A., Dawes, R.M., Monahan, J., 2000. Better decisions through science. Scientific American 283, 82–87.

van der Putten, P., van Someren, M., 2000. CoIL challenge 2000: The insurance company case. Technical Report 2000–09, Leiden Institute of Advanced Computer Science, Universiteit van Leiden. Available from: <http://www.liacs.nl/putten/library/cc2000>.

Zadrozny, B., Elkan, C., 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classiers. In: Proc. Eighteenth Internat. Conf. on Machine Learning, pp. 609–616.

Zou, K.H., 2002. Receiver operating characteristic (ROC) literature research. Online bibliography available from: <http://splweb.bwh.harvard.edu:8000/pages/ppl/zou/roc.html>.