

硕 士 研 究 生 读 书 报 告



题目 异常检测算法

作者姓名 黄闯

作者学号 21860451

指导教师 才振功

学科专业 计算机技术

所在学院 工程师学院

提交日期 2019.03.10

Algorithm model evaluation method

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: CaiZhenGong

By

HuangChuang

Zhejiang University, P.R. China

2019

算法模型评估方法

摘要

当我们针对一个问题，需要去进行特征提取，数据清洗，模型选择，参数配置，模型训练，参数调整，验证模型的过程中我们需要多方面的评估指标来对我们的模型进行估计，而不同场景，不同要求的模型，我们往往要用不同的评估方法，要用不同的角度去对模型进行分析。本部分内容就是针对模型评估的内容总结。

关键词：算法评估，查准率，召回率，ROC，AUC，F1

Abstract

When we are dealing with a problem, we need to perform feature extraction, data cleaning, model selection, parameter configuration, model training, parameter adjustment, and verification of the model. We need multiple evaluation indicators to estimate our model, and different scenarios. For different requirements, we often use different evaluation methods to analyze the model from different angles. This section is a summary of the content of the model assessment.

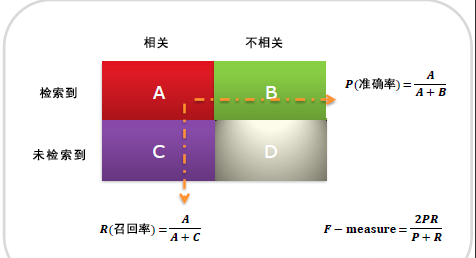
Keywords: algorithm evaluation, precision, recall, ROC, AUC, F1

前言

通常来说，评价一个分类器要比评价一个回归算法困难得多。著名的 MNIST 数据集是一个很好的例子，它包含多张从 0 到 9 的手写数字图片。如果我们想要构建一个分类器来判断数值是否为 6，构建一个算法将所有的输入分类为非 6，然后你将在 MNIST 数据集中获得 90% 的准确率，因为数据集中只有大约 10% 的图像是 6。这是机器学习中一个主要的问题，也是你需要多用几个评价指标测试你的分类器的原因。

一、混淆矩阵

首先，你可以了解一下混淆矩阵，它也被称为误差矩阵。它是一个描述监督学习模型在测试数据上的性能的表格，其中真实的值是未知的。矩阵的每一行表示预测出的类中的实例，而每一列则表示实际类别中的实例（反之亦然）。它被称之为「混淆矩阵」的原因是，利用它你很容易看出系统在哪些地方将两个类别相混淆了。



二、精确率和召回率

一个混淆矩阵可以给你很多关于你的（分类）模型做的有多好的信息，但是有一种方法可以让你得到更多的信息，比如计算分类的精确（precision）。说白了，它就是预测为正的样本的准确率（accuracy），并且它经常是和召回率（recall，即正确检测到的正实例在所有正实例中的比例）一起看的。

**三、F-值**

你可以把精确率和召回率融合到一个单独的评价指标中，它被称为「F-值」（也被称为「F1-值」）。如果你想要比较两个分类器，F-值会很有用。它是利用精确率和召回率的调和平均数计算的，并且它将给低的数值更大的权重。这样一来，只有精确率和召回率都很高的时候，分类器才会得到高 F-1 值。

不过 F-值并不是万能的「圣杯」，精确率和召回率接近的分类器会有更好的 F-1 分数。这是一个问题，因为有时你希望精确率高，而有时又希望召回率高。事实上，精确率越高会导致召回率越低，反之亦然。这被称为精确率和召回率的折衷，我们将在下一个章节讨论。

3.1精确率和召回率的折衷

为了更好地解释，我将举一些例子，来说明何时希望得到高精确率，何时希望得到高召回率。

**高精确率：**

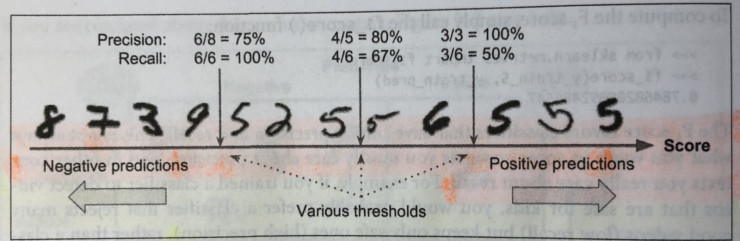
如果你训练了一个用于检测视频是否适合孩子看的分类器，你可能希望它有高的精确率。这意味着，这个你希望得到的分类器可能会拒绝掉很多适合孩子的视频，但是不会给你包含成人内容的视频，因此它会更加保险。（换句话说，精确率很高）

**高召回率：**

如果你想训练一个分类器来检测试图闯入大楼的人，这就需要高召回率了。可能分类器只有 25% 的精确率（因此会导致一些错误的警报），只要这个分类器有 99% 的召回率并且几乎每次有人试图闯入时都会向你报警，但看来是一个不错的分类器。

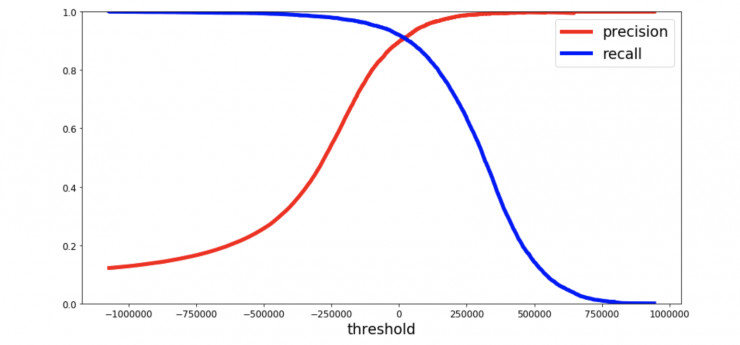
为了更好地理解这种折衷，我们来看看随机梯度下降（SGD）的分类器如何在 MNIST 数据集上做出分类决策。对于每一个需要分类的图像，它根据一个决策函数计算出分数，并将图像分类为一个数值（当分数大于阈值）或另一个数值（当分数小于阈值）。

下图显示了分数从低（左侧）到高（右侧）排列的手写数字。假设你有一个分类器，它被用于检测出「5」，并且阈值位于图片的中间（在中央的箭头所指的地方）。接着，你会在这个阈值右边看到 4 个真正类（真正为「5」的实例）和 1 个假正类（实际上是一个「6」）。这一阈值会有 80% 的精确率（五分之四），但是它仅仅只能从图片中所有的 6 个真正的「5」中找出 4 个来，因此召回率为 67%（六分之四）。如果你现在将阈值移到右侧的那个箭头处，这将导致更高的精确率，但召回率更低，反之亦然（如果你将阈值移动到左侧的箭头处）。

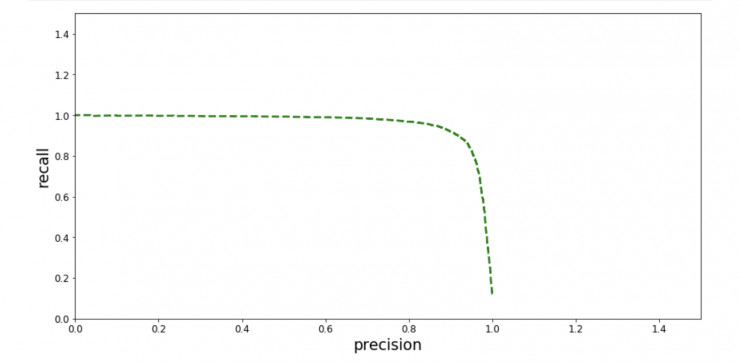


3.2精确率/召回率曲线

精确率和召回率之间的折衷可以用精确率-召回率曲线观察到，它能够让你看到哪个阈值最佳。



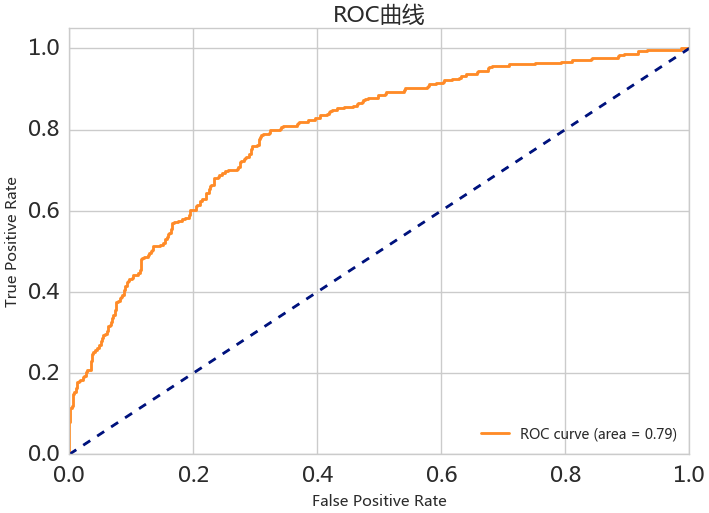
另一种方法是将精确率和召回率以一条曲线画出来：



在上图中，可以清晰地看到，当精确率大约为 95% 时，精准率升高，召回率迅速下降。根据上面的两张图，你可以选择一个为你当前的机器学习任务提供最佳精确率/召回率折衷的阈值。如果你想得到 85% 的精确率，可以查看第一张图，阈值大约为 50000。

四、ROC曲线和AUC 值

ROC 曲线是另一种用于评价和比较二分类器的工具。它和精确率/召回率曲线有着很多的相似之处，当然它们也有所不同。它将真正类率（true positive rate，即recall）和假正类率（被错误分类的负实例的比例）对应着绘制在一张图中，而非使用精确率和召回率。



|  |  |  |
| --- | --- | --- |
| FPR | TPR | 备注 |
| 0 | 1 | 完美分类器 |
| 1 | 0 | 糟糕分类器 |
| 0 | 0 | 预测都是负样本 |
| 1 | 1 | 预测都是正样本 |

当然，在这里也有所折衷。分类器产生越多的假正类，真正类率就会越高。中间的对角线是一个完全随机的分类器，分类器的曲线应该尽可能地远离它。

通过测量 ROC 曲线下方的面积（ AUC），ROC 曲线为比较两个分类器的性能提供了一种方法。这个面积被称作 ROC-AUC值，100% 正确的分类器的 ROC-AUC 值为 1。

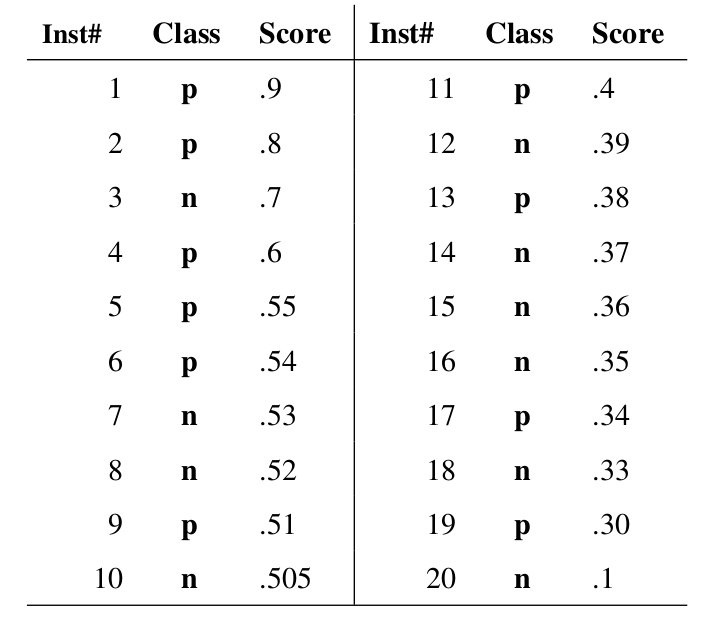
一个完全随机的分类器 ROC-AUC 为 0.5。

如果评价分类器，以及用哪些工具去评价。此外，还能学到如何对精确率和召回率进行折衷，以及如何通过 ROC AUC 曲线比较不同分类器的性能。

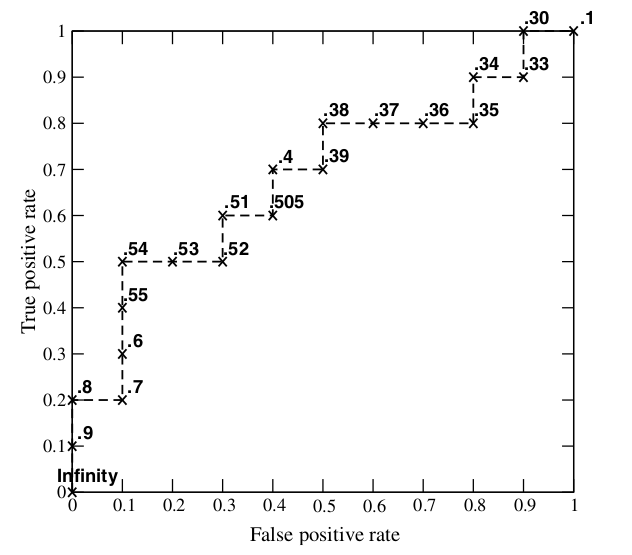
我们还了解到，精确率高的分类器并不像听起来那么令人满意：因为高精确率意味着低召回率。

4.1 如何画ROC曲线

假如我们已经得到了所有样本的概率输出（属于正样本的概率），现在的问题是如何改变“discrimination threashold”？我们根据每个测试样本属于正样本的概率值从大到小排序。下图是一个示例，图中共有20个测试样本，“Class”一栏表示每个测试样本真正的标签（p表示正样本，n表示负样本），“Score”表示每个测试样本属于正样本的概率。



接下来，我们从高到低，依次将“Score”值作为阈值threshold，当测试样本属于正样本的概率大于或等于这个threshold时，我们认为它为正样本，否则为负样本。举例来说，对于图中的第4个样本，其“Score”值为0.6，那么样本1，2，3，4都被认为是正样本，因为它们的“Score”值都大于等于0.6，而其他样本则都认为是负样本。每次选取一个不同的threshold，我们就可以得到一组FPR和TPR，即ROC曲线上的一点。这样一来，我们一共得到了20组FPR和TPR的值，将它们画在ROC曲线的结果如下图：



当我们将threshold设置为1和0时，分别可以得到ROC曲线上的(0,0)和(1,1)两个点。将这些(FPR,TPR)对连接起来，就得到了ROC曲线。当threshold取值越多，ROC曲线越平滑。

其实，我们并不一定要得到每个测试样本是正样本的概率值，只要得到这个分类器对该测试样本的“评分值”即可（评分值并不一定在(0,1)区间）。评分越高，表示分类器越肯定地认为这个测试样本是正样本，而且同时使用各个评分值作为threshold。我认为将评分值转化为概率更易于理解一些。

4.4 AUC值的计算

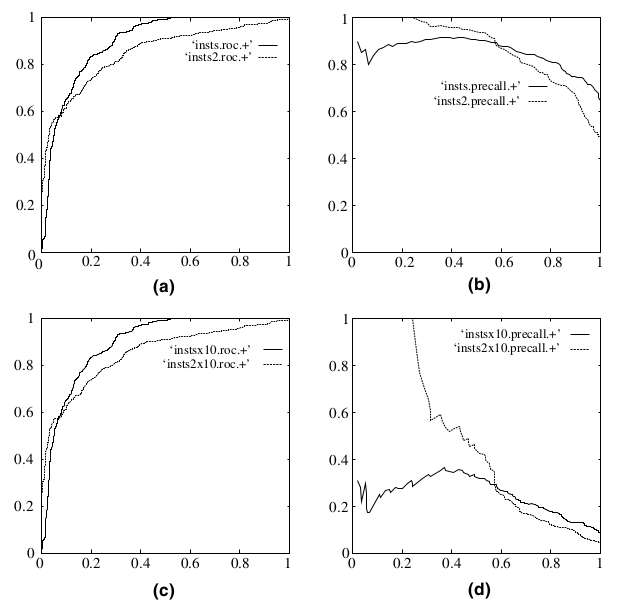
AUC（Area Under Curve）被定义为ROC曲线下的面积，显然这个面积的数值不会大于1。又由于ROC曲线一般都处于y=x这条直线的上方，所以AUC的取值范围在0.5和1之间。使用AUC值作为评价标准是因为很多时候ROC曲线并不能清晰的说明哪个分类器的效果更好，而作为一个数值，对应AUC更大的分类器效果更好。

4.2 AUC值意味着什么

首先AUC值是一个概率值，当你随机挑选一个正样本以及一个负样本，当前的分类算法根据计算得到的Score值将这个正样本排在负样本前面的概率就是AUC值。当然，AUC值越大，当前的分类算法越有可能将正样本排在负样本前面，即能够更好的分类。

4.3 为什么使用ROC曲线

既然已经这么多评价标准，为什么还要使用ROC和AUC呢？因为ROC曲线有个很好的特性：当测试集中的正负样本的分布变化的时候，ROC曲线能够保持不变。在实际的数据集中经常会出现类不平衡（class imbalance）现象，即负样本比正样本多很多（或者相反），而且测试数据中的正负样本的分布也可能随着时间变化。下图是ROC曲线和Precision-Recall曲线[5](http://alexkong.net/2013/06/introduction-to-auc-and-roc/" \l "fn:5)的对比：



在上图中，(a)和(c)为ROC曲线，(b)和(d)为Precision-Recall曲线。(a)和(b)展示的是分类其在原始测试集（正负样本分布平衡）的结果，(c)和(d)是将测试集中负样本的数量增加到原来的10倍后，分类器的结果。可以明显的看出，ROC曲线基本保持原貌，而Precision-Recall曲线则变化较大。

1. 总结

综上所述，我们知道其实针对模型的评估的方法依然还有很多，但是限于篇幅，我们并不能在此把所有的内容全部总结下来，但是对于查准率与召回率以及F1值是我们在机器学习建模过程中经常使用的参数，通过这些参数能够让我们很好的了解模型的训练效果以及测试效果，还有模型的性能，通过这些数据指标，能够帮助我们去改进我们所训练的模型，以达到更好的预测或者分类效果，做到最小化误差。

参考文献：

[1]Bradley A P. The use of the area under the ROC curve in the evaluation of machine learning algorithms[M]. 1997.

[2]Friedman J H. On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality[J]. Data Mining & Knowledge Discovery, 1997, 1(1):55-77.

[3]Drummond C, Holte R C. Cost curves: An improved method for visualizing classifier performance[J]. Machine Learning, 2006, 65(1):95-130.

[4]Bischoff E, Bischoff H, Giuliano F. An Introduction to the Bootstrap / B. Efron, R.J. Tibshirani.[J]. 2005.

[5]Johnson R W. An Introduction to the Bootstrap[J]. Teaching Statistics, 2010, 23(2):49-54.

[6]Fawcett T. An introduction to ROC analysis[J]. Pattern Recognition Letters, 2005, 27(8):861-874.

[7]Streiner D L, Cairney J. What's under the ROC? An introduction to receiver operating characteristics curves.[J]. Can J Psychiatry, 2007, 52(2):121-128.

[8] Dytham C . Choosing and using statistics a biologist's guide[J]. Austral Ecology, 1999, 87(4):734–735.