

机器学习之类别不平衡问题 (1) —— 各种评估指标



wdmad

8 人赞同了该文章

在二分类问题中，通常假设正负类别相对均衡，然而实际应用中类别不平衡的问题，如100, 1000, 10000倍的数据偏斜是非常常见的，比如疾病检测中未患病的人数远超患病的人数，产品质量检测中合格产品数量远超不合格产品等。在检测信用卡欺诈问题中，同样正例的数目稀少，而且正例的数量会随着时间和地点的改变而不断变化，分类器要想在不断变化的正负样本中达到好的检测效果是非常困难的。

由于类别不平衡问题的特性使然，一般常使用于评估分类器性能的**准确率**和**错误率**可能就不再适用了。因为在类别不平衡问题中我们主要关心数目少的那一类能否被正确分类，而如果分类器将所有样例都划分为数目多的那一类，就能轻松达到很高的准确率，但实际上该分类器并没有任何效果。

所以在这种时候学习的前提往往是采用不同的评估指标。学习机器学习的过程中总不免碰到各种评估指标，刚开始很容易被五花八门的术语绕晕了，所以类别不平衡问题的第一篇先对这些指标进行梳理。毕竟评估指标不明确的话，后面模型的效果好坏也就无从谈起。

在二分类问题中，一般将数目少的类别视为正例，数目多的类别视为负例，下面先用matplotlib画张混淆矩阵图来直观地感受一下：

```
plt.figure(figsize=(10,6))
plt.text(0.5,2.25,'True Positive (TP)',size=20,horizontalalignment="center",verticalal
plt.text(1.5,2.4,'False Positive (FP)',size=20,horizontalalignment="center",verticalal
plt.text(0.5,0.9,'False Negative (FN)',size=20,horizontalalignment="center",verticalal
plt.text(1.5,0.75,'True Negative (TN)',size=20,horizontalalignment="center",verticalal
plt.text(1,3.4,'$True\ Class$',size=25,horizontalalignment="center")
plt.text(-0.5,1.5,'$Predicted$\n$Class$',size=23,verticalalignment="center")
plt.text(0.5,3.1,'$P$',size=20,horizontalalignment="center")
plt.text(1.5,3.1,'$N$',size=20,horizontalalignment="center")
plt.text(-0.1,2.25,'$Y$',size=20,va="center")
plt.text(-0.1,0.75,'$N$',size=20,va="center")
plt.text(2.4,2.25,'Precision = $\frac{TP}{Y}$ = $\frac{TP}{TP+FP}$ ',size=18,ha="cent
plt.text(0.5,-0.3,'Recall, Sensitivity, TPR = ',size=16,ha="center",va="center")
plt.text(0.5,-0.6,'$\frac{TP}{P}$ = $\frac{TP}{TP+FN}$ ',size=18,ha="center",va="cent
plt.text(1.5,-0.3,'FPR = $\frac{FP}{N}$ = $\frac{FP}{FP+TN}$ ',size=16,ha="center",va
plt.text(1.5,-0.7,'TNR, Specificity = $\frac{TN}{N}$ = $\frac{TN}{FP+TN}$ ',size=16,h
plt.text(1.5,2.1,'Type I Error',size=20,horizontalalignment="center",verticalalignment
plt.text(0.5,0.6,'Type II Error',size=20,horizontalalignment="center",verticalalignmen
plt.xticks([])
plt.yticks([])
plt.plot([1,1],[0,3], 'k--')
plt.plot([0,3],[1.5,1.5], 'k:')
plt.axis([0,2,0,3])
```

plt.fill_be

▲ 赞同 8 ▼

● 添加评论

➦ 分享

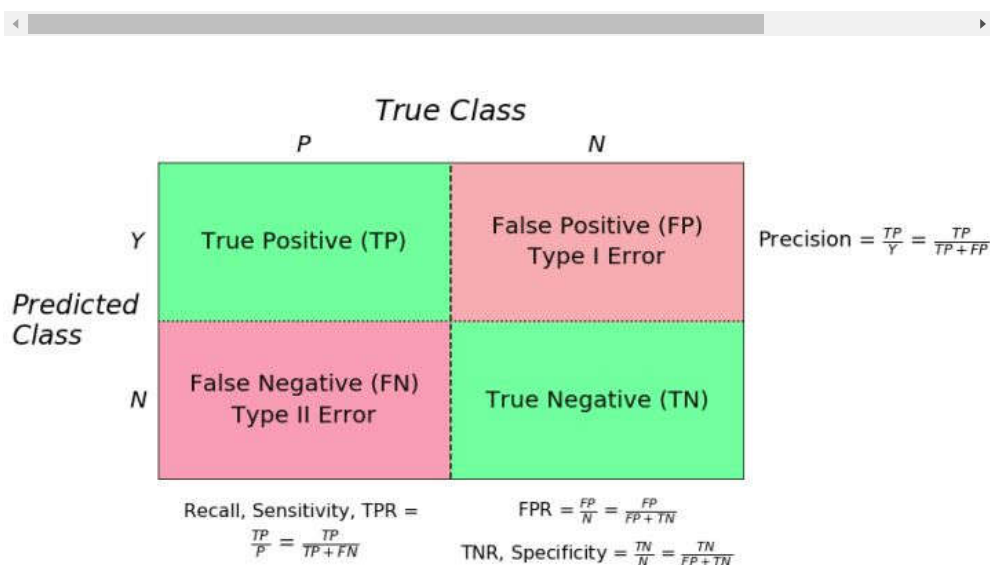
♥ 喜欢

★ 收藏

...

🏠

```
plt.fill_between([1,2],[1.5,1.5],[3,3],color='#EEB4B4')
```



True Positive (真正例, TP): 实际为正例, 预测为正例。

False Negative (假负例, FN): 实际为正例, 预测为负例。

True Negative (真负例, TN): 实际为负例, 预测为负例。

False Positive (假正例, FP): 实际为负例, 预测为正例。

Precision (查准率) = $\frac{TP}{TP+FP}$, Precision衡量的是所有被预测为正例的样本中有多少是真正例。但Precision并没有表现有多少正例是被错判为了负例(即FN), 举个极端的例子, 分类器只将一个样本判为正例, 其他所有都判为负例, 这种情况下Precision为100%, 但其实遗漏了很多正例, 所以Precision常和下面的Recall (TPR) 相结合。

True Positive Rate (TPR, 真正例率) = $\frac{TP}{TP+FN}$, 又称**Recall**(查全率), **Sensitivity**(灵敏性)。Recall (TPR)衡量的是所有的正例中有多少是被正确分类了, 也可以看作是为了避免假负例(FN)的发生, 因为TPR高意味着FN低。Recall的问题和Precision正相反, 没有表现出有多少负例被错判为正例(即FP), 若将所有样本全划为正例, 则Recall为100%, 但这样也没多大用。

False Negative Rate (FNR, 假负例率) = $\frac{FN}{TP+FN} = 1 - TPR$, 由混淆矩阵可以看出该指标的着眼点在于正例, 意为有多少正例被错判成了负例。

True Negative Rate (TNR, 真负例率) = $\frac{TN}{TN+FP}$, 又称**Specificity**(特异性)。Specificity衡量的所有的负例中有多少是被正确分类了, 由于类别不平衡问题中通常关注正例能否正确被识别, Specificity高则FP低, 意味着很少将负例错判为正例, 即该分类器对正例的判别具有“特异性”, 在预测为正例的样本中很少有负例混入。

的假设为正例，否则为负例。在ROC曲线下方以(0,0)和(1,1)为端点画一条对角线，以横轴作图，显示出一种正例与负例之间的“博弈”，在下篇文章中详解。

$$F1\ score = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = \frac{2 \times precision \times recall}{precision + recall}$$

，是一个综合指标，为Precision和Recall的调和平均 (harmonic mean)，数值上一般接近于二者中的较小值，因此如果F1 score比较高的话，意味着Precision和Recall都较高。

FP和FN还有个与之相关的概念，那就是统计假设检验中的**第一类错误 (Type I error)** 和**第二类错误 (Type II error)**。由于我们比较关心正例，所以将负例视为零假设，正例视为备选假设，则第一类错误为错误地拒绝零假设 (负例)，选择备选假设，则为FP；第二类错误为错误地接受零假设，则为FN。

上面介绍的这些指标都没有考虑检索结果的先后顺序，而像搜索问题中我们通常希望第一个结果是与查询最相关的，第二个则是次相关的，以此类推，因而有时候不仅要预测准确，对于相关性的顺序也非常看重。所以最后介绍两个广泛应用的排序指标。

Mean Average Precision (MAP, 平均准确率均值)，对于单个信息需求，返回结果中在每篇相关文档上 Precision 的平均值被称为 Average Precision (AP)，然后对所有查询取平均得到 MAP。

$$AP = \frac{\sum_{k=1}^n P(k) \times rel(k)}{M}$$

$$MAP = \sum_{q=1}^Q \frac{AP_q}{Q}$$

其中 $P(k)$ 为前 k 个结果的 Precision，又可写为 $P@k$ 。 $rel(k)$ 表示第 k 个结果是否为相关文档，相关为1不相关为0， M 表示所有相关文档的数量， n 表示所有文档数量。如果只关心前 K 个查询的情况，则是下式：

$$AP@K = \frac{\sum_{k=1}^K P(k) \times rel(k)}{M_K}$$

$$MAP@K = \sum_{q=1}^Q \frac{AP_q@K}{Q}$$

这里的 M_K 为前 K 个结果中相关文档的数量。

对于单个信息需求来说，Average Precision 是 PR 曲线下面积的近似值，因此 MAP 可粗略地认为是某个查询

Normalized Discounted Cumulative Gain (NDCG, 归一化折扣累计增益)。如果说 MAP 是基于 0/1 二值描述相关性, 那么 NDCG 则是可将相关性分为多个等级的指标。

对于信息检索和推荐之类的问题, 每一个返回的结果都被赋予一个相关性分数 rel , 则 NDCG 中的 CG 表示前 k 个结果的分数之和, 即累计增益:

$$CG_k = \sum_{i=1}^k rel_i$$

CG 没有考虑推荐的次序, 所以在此基础上引入对结果顺序的考虑, 即相关性高的结果若排在后面则会受更多的惩罚, 于是就有了 DCG (discounted CG), 折扣累积增益。公式如下:

$$DCG_k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)}.$$

i 表示一个结果在结果集中的顺序, 如果该结果 rel 很高, 但排在后面, 意味着分母 $\log_2(i + 1)$ 会变大, 则相应的总体 DCG 会变小 (注意这里的 \log 是以 2 为底的)。

对于不同的查询, 往往会返回不同的结果集, 而不同结果集之间因为大小不同难以直接用 DCG 进行比较, 所以需要进行归一化, 这其实和机器学习中不同特征因量纲不同要进行归一化差不多意思。这个归一化后的指标就是 NDCG:

$$NDCG_k = \frac{DCG_k}{IDCG_k}$$

其中 IDCG 表示 Ideal DCG, 指某个查询所能返回的最好结果集, IDCG 的值也是结果集中最大的。将所有结果按相关性大小排序, 计算出的 DCG 即为前 k 个结果的 IDCG:

$$IDCG_k = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i + 1)}.$$

其中 $|REL|$ 表示按相关性顺序排列的结果集。因此 DCG 的值介于 $(0, IDCG]$, 故 NDCG 的值介于 $(0, 1]$, 这样就起到了归一化的效果。不同查询或用户的 NDCG 平均起来可以用以评估一个搜索引擎或推荐系统的整体效果。

NDCG 的缺点是需要预先指定每一个返回结果的相关性, 这个超参数需要人为指定。

编辑于 2019-01-24

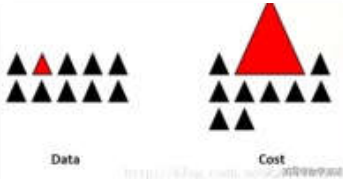
机器学习

文章被以下专栏收录

▲ 赞同 8 ▼ ● 添加评论 ➦ 分享 ♥ 喜欢 ★ 收藏 ...




推荐阅读



机器学习中解决数据不平衡问题

胡卫雄 发表于机器学习入...



机器学习模型性能评估二：代价曲线与性能评估方法总结

胡卫雄 发表于机器学习入...



机器学习篇-指标：AUC

眼睛流产

**机器学习之分类
ROC曲线、AUC**

查阅更为简洁方便
最新的课程、产品
全新呈现的http://
在分类任务中，人
错误率来衡量分类
度。错误率指的是
人工智能L... 发

还没有评论

写下你的评论...

