不平衡类问题

赵海臣

背景

- 具有不平衡类分布的数据集在许多实际应用中都会见到，属于不同类的实例数量不成比例：
  - 生产线上，不合格产品的数量远远低于合格产品的数量
  - 信用卡欺诈中，合法交易远远多于欺诈交易

- 准确率经常用来比较分类器性能，但不合适评价从不平衡数据集得到的模型：
  - 如果1%信用卡交易时欺骗行为，则分类器将所有交易都预测成合法的情况下模型具有99%的准确率，但它检测不到任何欺诈。

度量方法

- 在不平衡数据集中，稀有类比多数类更有意义。
  - 对于二元分类，稀有类通常记为正类，多数类被认为是负类
- 混淆矩阵：

		预测的类		
		+	-	TOTAL
实际的类	+	f++(TP)	f+-(FN)	Acutal Positive TP+FN
	-	f-+(FP)	f--(TN)	Actual Negative FP+TN
	TOTAL	Predicted Positve TP+FP	Predicted Negative FN+TN	Total TP+TN+ FP+FN

混淆矩阵术语

- 真正(True Positive, TP)
  - 正确预测的正样本数
- 假负(False Negative, FN)
  - 错误预测的负样本数
- 假正(False Positive, FP)
  - 错误预测的正样本数
- 真负(True Negative, TN)
  - 正确预测的负样本数

- 所有正样本数 $= TP + FN$
- 所有负样本数 $= TN + FP$
- 样本总数 $= TP + TN + FP + FN$

混淆矩阵术语

- 真正率TPR = TP / (TP + FN)
- 真负率TNR = TN / (TN + FP)
- 假正率FPR = FP / (TN + FP)
- 假负率FNR = FN / (TP + FN)

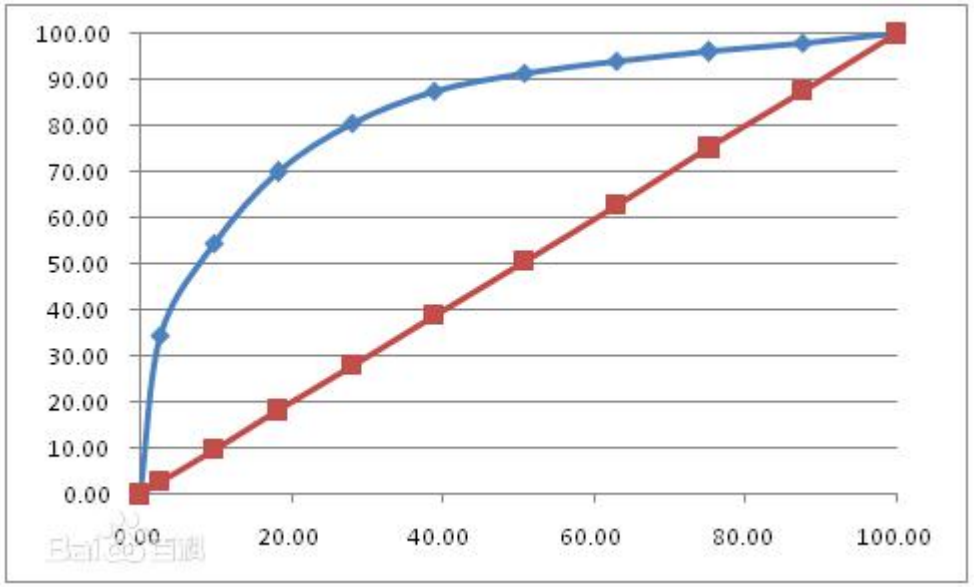
- 召回率Recall = 真正率TPR = TP / (TP + FN)
- 精确度Precise = TP / (TP + FP)

精确度/召回率/F-score

- 召回率最大化：
  - 所有样本预测为正类，有100%的召回率，但精确度差
- 精确度最大化：
  - 只将已知的正类预测为正类，有100%的精确度，但召回率差
- 构建最大化精度与召回率的模型是分类算法主要任务之一：
  - 精确度和召回率可以合并成调和均值，F1度量：
    - $F\text{-score} = 2*r*p/(r+p)$
      $= 2*TP / (2*TP+FP+FN)$

ROC (Receiver Operating Characteristic)

- 接受者操作特征ROC曲线是实现分类器真正率TPR和假正率FPR之间折中的一种图形化方法。
  - x轴：FPR
  - y轴：TPR



ROC绘制步骤

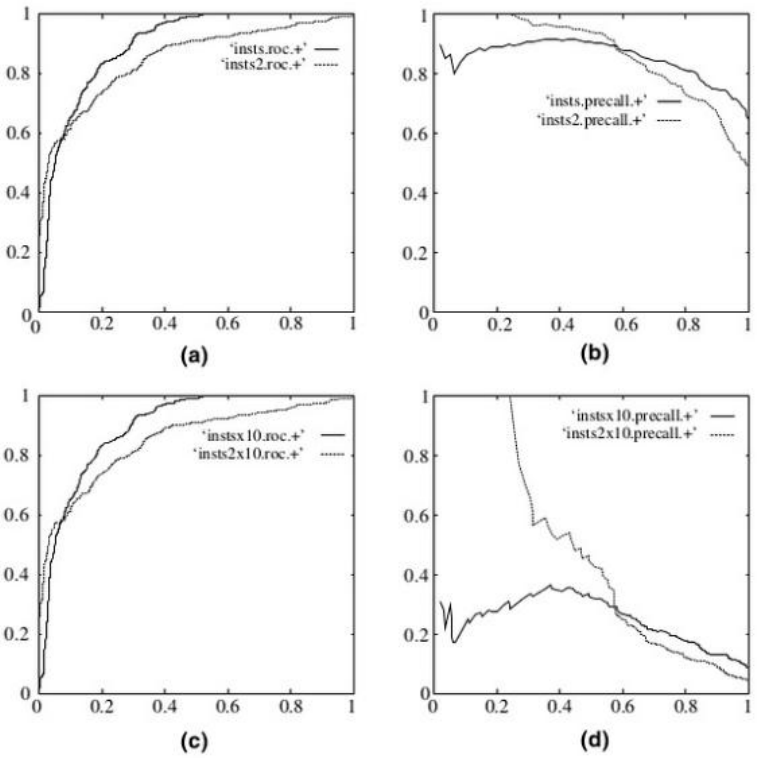
- 1. 假定正类定义了连续值输出，对检验记录按照它们的输出值递增排序
- 2. 选择排序列表中值最小的记录，把大于/等于该值的类指派为正类
- 3. 选择排序列表中下一个检验记录，把大于/等于该值的类指派为正类，小于该值得指派为负类
- 4. 重复步骤3并相应更新TP与FP计数，直到最大值记录被选择
- 5. 根据分类器的TP与FP画出TPR曲线

ROC结果解析

- 一个好的分类器应该：
  - TPR＞50%
  - FPR＜50%
- 左上方是TPR＞50%与FPR＜50%的区域，因此曲线越偏向左上方分类效果越好：
  - ROC下面的面积AUC(Area Under Curve)提供了评价模型性能的一种办法。
    - 如果模型是完美的，则ROC曲线下的面积AUC＝1
    - 如果模型是随机猜的，则ROC曲线下的面积AUC＝0.5
    - 如果模型A好于模型B，则模型A的AUC ＞ 模型B的AUC

为什么使用ROC曲线

- ROC曲线有个很好的特性：ROC曲线对正负样本的分布比例不敏感。
- ROC曲线和Precision-Recall曲线的对比：



(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。

对不平衡类问题的处理

- 抽样是处理不平衡类的广泛使用方法：
  - 主要思想是改变实例的分布，从而帮助稀有类在训练数据集中得到很好的表示
- 100个正样本与1000个负样本抽样例子：
  - 取全部100个正样本与随机抽样100个负样本
    - 多次抽样形成多个分类器，多分类器ensemble
  - 过分抽样100个正样本到1000个(直到训练集中正样本与负样本一样多)
    - 过分抽样提供了需要的额外样本，确保围绕小群正样本的决策边界不被剪除
    - 过分抽样将可能放大正样本的噪声数据，造成过拟合
    - 对正样本的过分抽样并没有添加新的信息，仅仅是为了阻止学习算法剪掉很少训练样本的部分。

多类问题

- 1-r方法：
  - 将多类问题分解成K个二类问题，为每一个类创建一个二类问题，其中所有属于yi的被看作正类，而其他样本作为负类。
  - 最后进行投票方法来确定实例的类。

The end