Machine Learning 2019

袁欣

2019 年 3 月 3 日

1 模型评估与选择

1.1 构造数据集

构造一个包含 1000 个样本的数据集，按照某种模型对样本排序，前 500 个样本中正例（取值 1）占 90%，后 500 个样本中反例（取值 0）占 80%。

- 代码如下：

```
pred <- c(round(runif(500) / 2 + 0.45),
          round(runif(500) / 2 + 0.10))
```

- 数据展示：

```
head(pred)
```

```
## [1] 1 1 1 1 1 1
```

```
tail(pred)
```

```
## [1] 0 0 0 0 1 0
```

```
mean(pred)
```

```
## [1] 0.547
```

- 真实均值：

$$\bar{pred} = (500 \times 0.9 + 500 \times 0.2) \div 1000 = 0.55$$

1.2 绘制曲线

试给出该模型的 P-R 曲线和 ROC 曲线的代码。

1.2.1 P-R 曲线

- 理论基础：

对于二分类问题，可将样例根据其真实类别与学习器预测类别的组合划分为真正例 (true positive)、假正例 (false positive)、真反例 (true negative)、假反例 (false negative) 四种情形，对应的混淆矩阵如下所示[1]。

```
knitr::kable(da)
```

	Predict1	Predict0
Act1	TP	FN
Act0	FP	TN

查准率 P 与查全率 R 的定义分别为：

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$F1$ 统计量的定义为：

$$F1 = \frac{2 \times P \times R}{P + R}$$

我们根据学习器的预测结果对样例进行排序，排在最前面的是学习器认为“最可能”是正例的样本，排在最后面的是学习器认为“最不可能”的正例样本。按此顺序逐个把样本作为正例预测，每次可计算出查全率与查准率。以查准率为纵轴，查全率为横轴作图，就可以得到“P-R 曲线”

注意：在 1 中我们构建的 `pred` 已经是按照预测概率排序后的数据集 (**真实数据的标签**)，所以在接下来我们只需要利用 `for` 循环计算每一次的 P、R 即可。

- 构建函数计算 P-R：

```
PRCurve <- function(pred){
  m <- length(pred)
  P <- R <- rep(0, m)
  for(i in 1 : m){
    predi <- c(rep(1, i), rep(0, m - i))
    tab <- table(predi, pred)
    if(i != m){
      P[i] <- tab[2, 2] / (tab[2, 1] + tab[2, 2])
      R[i] <- tab[2, 2] / (tab[1, 2] + tab[2, 2])
    }else{
      P[i] <- tab[1, 2] / (tab[1, 1] + tab[1, 2])
```

```
      R[i] <- tab[1, 2] / tab[1, 2]
    }
  }
  F1 <- 2 * P * R / (P + R)
  bound <- which(F1 == max(F1))
  F1 <- max(F1)
  return(list(P = P, R = R, F1 = F1, bound = bound))
}
PR <- PRCurve(pred)
P <- PR$P
R <- PR$R
F1 <- PR$F1
bound <- PR$bound
```

- 绘制 P-R 曲线：

```
library(ggplot2)
da1 <- data.frame(P = P, R = R)
da2 <- data.frame(x = seq(0, 1, 0.01), y = seq(0, 1, 0.01))
ggplot(data = da1, aes(x = R, y = P)) +
  geom_line(colour = "red") + xlim(0, 1) + ylim(0, 1) +
  geom_line(data = da2, aes(x = x, y = y), colour = "blue") +
  geom_text(data = data.frame(x = 0.5, y = 0.5), aes(x = x,
            y = y, label = paste("F1=", round(F1, 3))))
```

1.2.2 *ROC* 曲线

- 理论基础：

ROC 曲线与 P-R 曲线相似，只不过 *ROC* 曲线的纵轴为 “真正例率”(True Positive Rate，简称 TPR)，横轴是 “假正例率”(False Positive Rate，简称 FPR)，两者的定义如下。

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{TP}{TN + FP}$$

若一个机器学习的 *ROC* 曲线被另一个机器学习的曲线完全 “包住”，则可断言后者的性能优于前者。当两条曲线发生交叉时，可利用 *ROC* 曲线下的面积 *AUC*(Area Under *ROC* Curve) 判断哪个机器学习的性能更好。

$$AUC = \frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1} - x_i)(y_i + y_{i+1})$$

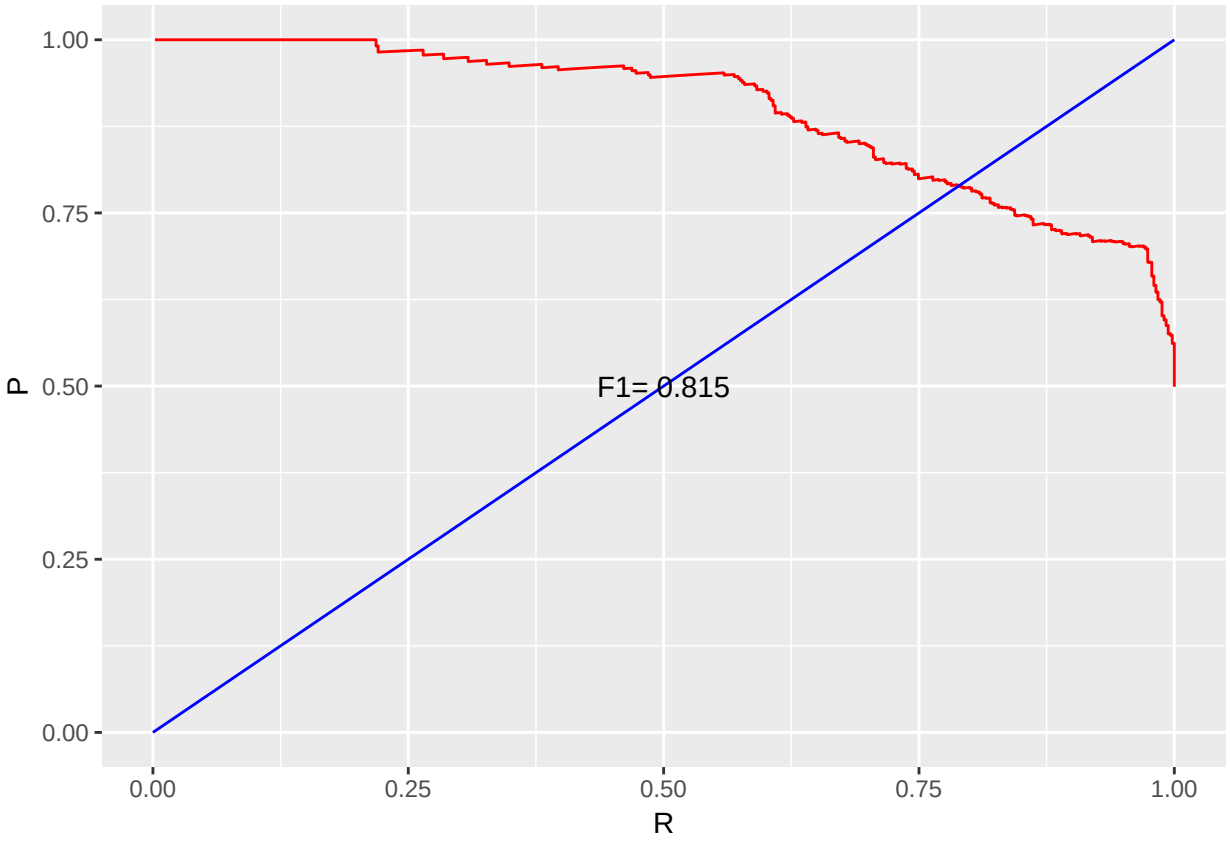

图 1: P-R 曲线

- 构建函数计算 *ROC*：

```
ROCCurve <- function(pred){
  m <- length(pred)
  TPR <- FPR <- rep(0, m + 1)
  AUC <- 0
  for(i in 1 : (m - 1)){
    predi <- c(rep(1, i), rep(0, m - i))
    tab <- table(predi, pred)
    TPR[i + 1] <- tab[2, 2] / (tab[1, 2] + tab[2, 2])
    FPR[i + 1] <- tab[2, 1] / (tab[1, 1] + tab[2, 1])
    AUC <- AUC + (1/2) * (TPR[i + 1] + TPR[i]) *
      (FPR[i + 1] - FPR[i])
  }
  TPR[m + 1] <- 1
  FPR[m + 1] <- 1
  AUC <- AUC + (1/2) * (TPR[m + 1] + TPR[m]) *
    (FPR[m + 1] - FPR[m])
  return(list(TPR = TPR, FPR = FPR, AUC = AUC))
}
ROC <- ROCCurve(pred)
TPR <- ROC$TPR
FPR <- ROC$FPR
AUC <- ROC$AUC
```

- 绘制 *ROC* 曲线：

```
library(ggplot2)
da1 <- data.frame(TPR = TPR, FPR = FPR)
da2 <- data.frame(x = seq(0, 1, 0.01), y = seq(0, 1, 0.01))
ggplot(data = da1, aes(x = FPR, y = TPR)) +
  geom_line(colour = "red") + xlim(0, 1) + ylim(0, 1) +
  geom_line(data = da2, aes(x = x, y = y), colour = "blue") +
  geom_text(data = data.frame(x = 0.5, y = 0.5), aes(x = x,
            y = y, label = paste("AUC=", round(AUC, 3))))
```

1.3 小结

模型评估与选择的方法还有很多种，如错误率与精度、代价敏感错误率与代价曲线、比较检验、偏差与方差等。在模型评估过程中应因地制宜，根据模型本身制定合适的评价标准。当负例样本占样本集比例较小时（如 5%）就不能使用错误率衡量模型的好坏。

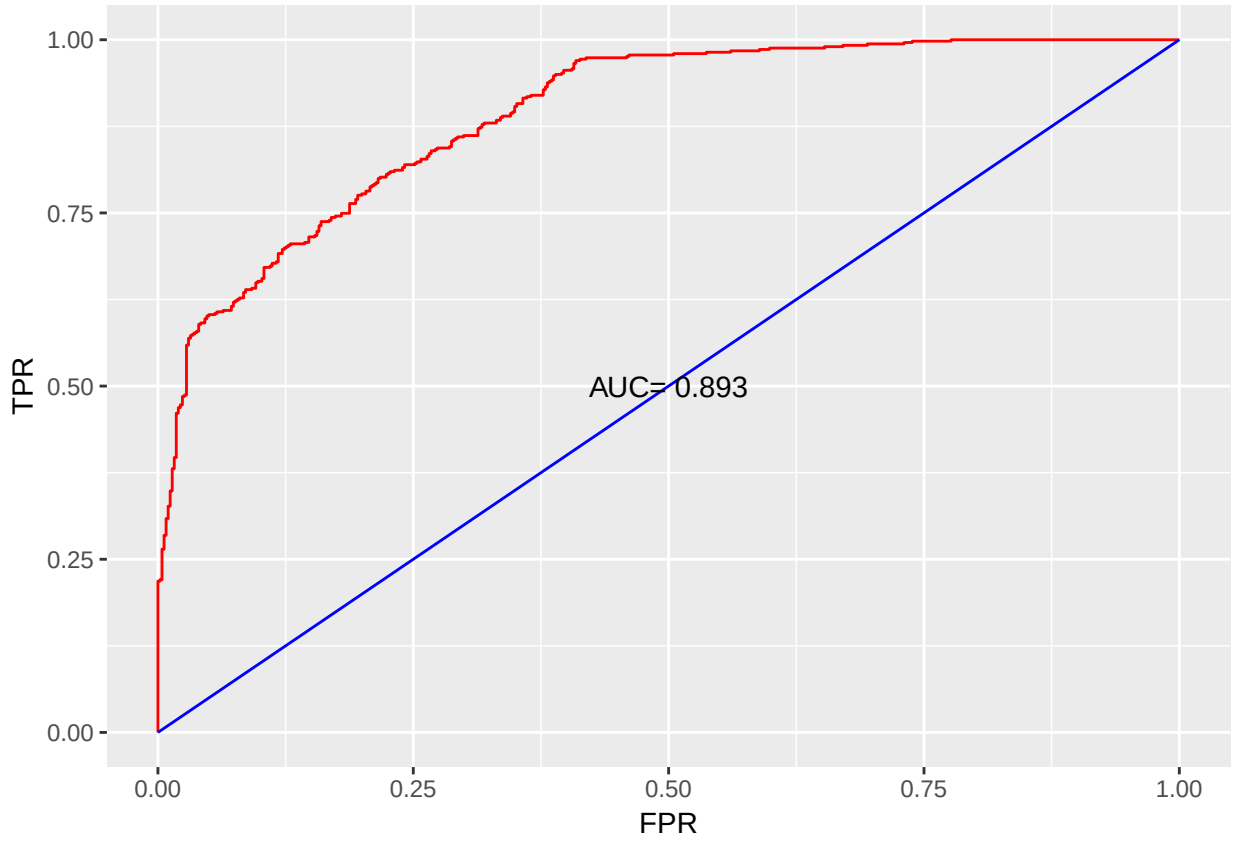

图 2: ROC 曲线