HW4

xw-zeng

2022-10-20

(1)

Derive the Maximum likelihood estimation function.

$$
\begin{aligned}
l(\beta) &= \sum_{i=1}^{N} \left\{ y_i \log p\left(x_i;\beta\right) + \left(1-y_i\right) \log\left(1-p\left(x_i;\beta\right)\right) \right\} \\
&= \sum_{i} \left\{ y_i x_i^\top \beta - \log\left(1+\exp\left(x_i^\top \beta\right)\right) \right\} \\
&= y^\top X\beta - b^\top 1
\end{aligned}
$$

$$
b = \left(b\left(x_1\right), b\left(x_2\right), \ldots, b\left(x_N\right)\right)^\top \quad \mathit{and} \quad b\left(x_i\right) = \log\left(1+\exp\left(x_i^\top \beta\right)\right)
$$

$$
l'(\beta) = \frac{\partial l(\beta)}{\partial \beta^\top} = X^\top y - \frac{\partial b^\top}{\partial \beta^\top} 1
$$

$$
\begin{aligned}
\frac{\partial b^\top}{\partial \beta^\top} &= \frac{\partial}{\partial \beta^\top} \left(\log\left(1+\exp\left(x_1^\top \beta\right)\right), \ldots, \log\left(1+\exp\left(x_N^\top \beta\right)\right)\right) \\
&= \left(\frac{x_1 \cdot \exp\left(x_1^\top \beta\right)}{1+\exp\left(x_1^\top \beta\right)}, \ldots, \frac{x_N \cdot \exp\left(x_N^\top \beta\right)}{1+\exp\left(x_N^\top \beta\right)}\right)
\end{aligned}
$$

$$
l'(\beta) = X^\top y - X^\top p
$$

$$
p = \left(p\left(x_1;\beta\right), \ldots, p\left(x_N;\beta\right)\right)^\top \quad \mathit{and} \quad p\left(x_i;\beta\right) = \frac{\exp\left(x_i^\top \beta\right)}{1+\exp\left(x_i^\top \beta\right)}
$$

(2)

Define the log likelihod function.

```
logL <- function(beta){
  b <- function(x, beta){
    out <- c()
    for (i in 1:nrow(x)){out <- c(out, log(1 + exp(x[i, ] %*% beta)))}
    return (out)
  }
  t(y) %*% X %*% beta - sum(b(X, beta))
}
```

Define the first derivative of log likelihood function.

```
logL_1 <- function(beta){
  p <- function(x, beta){
    out <- c()
    for (i in 1:nrow(x)){out <- c(out, 1 / (1 + exp(-1 * x[i, ] %*% beta)))}
    return (out)
  }
  t(X) %*% (y - p(X, beta))
}
```

Compute the second derivative of log likelihood function.

$$
\begin{aligned}
l''(\beta) &= \frac{\partial l'(\beta)}{\partial \beta^\top} = -X^\top \frac{\partial p}{\partial \beta^\top} \\
\frac{\partial p}{\partial \beta^\top} &= \frac{\partial}{\partial \beta^\top} \left(1 - \frac{1}{1+\exp\left(x_1^\top \beta\right)}, \dots, 1 - \frac{1}{1+\exp\left(x_N^\top \beta\right)}\right)^\top \\
&= \left(\frac{x_1 \exp\left(x_1^\top \beta\right)}{\left(1+\exp\left(x_1^\top \beta\right)\right)^2}, \dots, \frac{x_N \exp\left(x_N^\top \beta\right)}{\left(1+\exp\left(x_N^\top \beta\right)\right)^2}\right)^\top \\
l''(\beta) &= -X^\top \operatorname{diag}\left(\frac{\exp\left(x_i^\top \beta\right)}{\left(1+\exp\left(x_i^\top \beta\right)\right)^2}\right) X = -X^\top W X
\end{aligned}
$$

Define the second derivative of log likelihood function.

```
logL_2 <- function(beta){
  w <- function(x, beta){
    out <- matrix(0, nrow(x), nrow(x))
    for (i in 1:nrow(x)){
      out[i, i] <- exp(x[i, ] %*% beta) / (1 + exp(x[i, ] %*% beta)) ^ 2
```

```
    }
    return (out)
  }
  - t(X) %*% w(X, beta) %*% X
}
```

Define the **Newton-Raphson method** function.

- `start`: starting value of beta.
- `criterion`: absolute convergence criterion, by default set as $1e^{-5}$.

```
Newton_Raphson <- function(start, criterion = 1e-5){
    ##Initial values
    x <- start
    ##Updating function
    delta <- function(x){solve(logL_2(x)) %*% logL_1(x)}
    ##Main
    while (max(abs(delta(x))) >= criterion){x <- x - delta(x)}
    ##Output
    structure(list(N = nrow(X),
                   best_beta = x,
                   epsilon = x - beta,
                   best_logL = logL(x),
                   start = start))
}
```

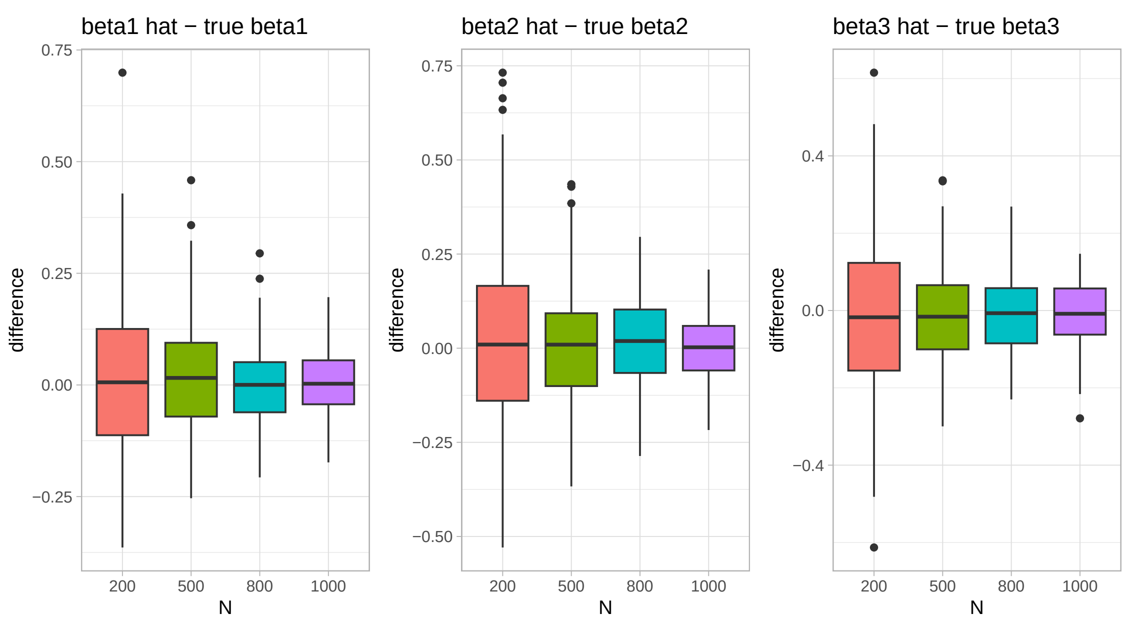
Compute results with N = 200, 500, 800, 1000.

```
##True beta
beta <- c(0.5, 1.2, -1)
df <- matrix(NA, 1, 4)
##Set a random seed to obtain repeatable results
set.seed(2022)
##N = 200, 500, 800, 1000
for (N in c(200, 500, 800, 1000)){
  ##Simulate for 200 rounds
  for (i in 1:200){
    ##Generate values of X
    X <- matrix(1, N, 3)
    for (j in 2:3){X[, j] <- rnorm(N)}
    ##Generate values of y
    p <- exp(X %*% beta) / (1 + exp(X %*% beta))
```

```
    y <- c()
    for (j in 1:N){y <- c(y, rbinom(1, 1, p[j]))}
    ##Track the difference between beta hat and true beta
    df <- rbind(na.omit(df), cbind(N, t(Newton_Raphson(c(0,0,0))$epsilon)))}}
##Transform the df matrix to a data frame
df <- data.frame(df)
df$N <- as.factor(df$N)
```

For each j, draw $\left(\hat{\beta}_j^{(r)} - \beta_j\right)$ in boxplot for N = 200, 500, 800, 1000.

```
p1 = ggplot(df, mapping = aes(x = N, y = V2, fill = N)) +
  geom_boxplot() + guides(fill = 'none') + theme_light() +
  labs(title = 'beta1 hat - true beta1', y = 'difference')
p2 = ggplot(df, mapping = aes(x = N, y = V3, fill = N)) +
  geom_boxplot() + guides(fill = 'none') + theme_light() +
  labs(title = 'beta2 hat - true beta2', y = 'difference')
p3 = ggplot(df, mapping = aes(x = N, y = V4, fill = N)) +
  geom_boxplot() + guides(fill = 'none') + theme_light() +
  labs(title = 'beta3 hat - true beta3', y = 'difference')
p1 + p2 + p3
```



从箱线图中我们可以看出，基本上随着样本量的增加，$\left(\hat{\beta}_j^{(r)} - \beta_j\right)$ 不断向 0 靠近，且方差也在不断减小。这说明增大样本量能够一定程度上减少对真实 β 的估计误差。

(3)

考虑一种简单的情况，当数据中不存在 $f(x^+) = f(x^-)$ 时，“排序” 损失的定义如下：

$$\ell_{rank} = \frac{1}{m^+m^-} \sum_{x^+\in D^+} \sum_{x^-\in D^-} \left(I\left(f\left(x^+\right) < f\left(x^-\right)\right)\right)$$

Proof 1

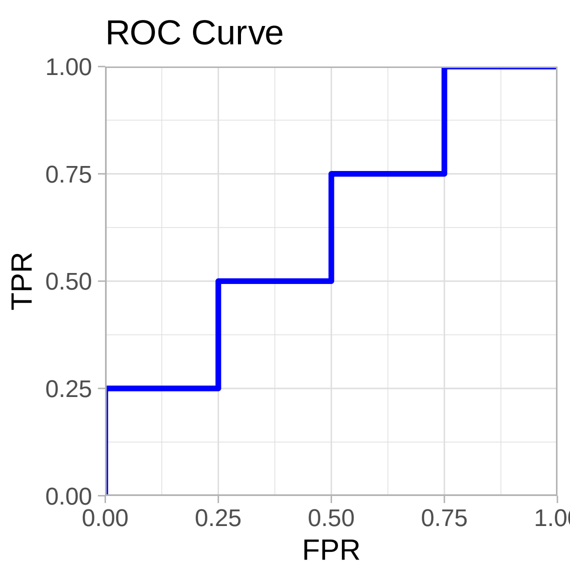
要证明 $AUC = 1 - \ell_{rank}$ 即为有限样本 ROC 曲线下方的面积，等价于证明 ℓ_{rank} 表示有限样本 ROC 曲线与 y 轴围成的面积。

首先生成一幅 ROC 曲线图作为例子。

```r
fpr <- c(0, 0, 1/4, 1/4, 2/4, 2/4, 3/4, 3/4, 1)
tpr <- c(0, 1/4, 1/4, 2/4, 2/4, 3/4, 3/4, 1, 1)
p <- ggplot(mapping = aes(x = fpr, y = tpr)) + theme_light() +
  geom_path(size = 1, color = 'blue') +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  labs(title = 'ROC Curve', x = 'FPR', y = 'TPR')
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```

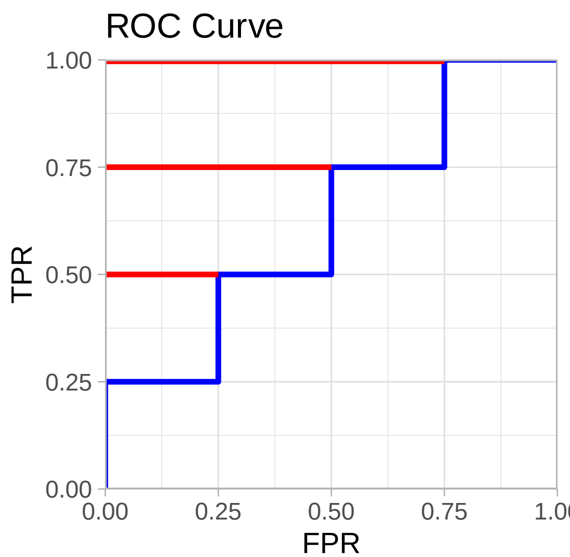
```r
p
```



我们想要求 ROC 与 y 轴围成的面积可以用矩形的面积公式长 × 宽进行计算（因为我们假设数据中不存在 $f(x^+) = f(x^-)$ 的情况，就不会出现梯形）。

我们首先把 ROC 曲线上方的多边形**水平**切分为几个矩形（垂直切也可以，原理都是一样的），如下图所示。

```
p + geom_line(aes(x = c(0, 1/4), y = c(2/4, 2/4)), color = 'red', size = 1) +
  geom_line(aes(x = c(0, 2/4), y = c(3/4, 3/4)), color = 'red', size = 1) +
  geom_line(aes(x = c(0, 3/4), y = c(1, 1)), color = 'red', size = 1.5)
```



水平的线段表示在分类阈值变动的过程中只新增了假正例，垂直的线段表示只新增了真正例。每新增一个假正例，x 的坐标就新增一个步长 $\frac{1}{m^-}$；每新增一个真正例，y 的坐标就新增一个步长 $\frac{1}{m^+}$。

因此，每一个矩形的宽都为 $\frac{1}{m^+}$，每一个矩形的长就表示在当前有多少个负例满足 $f(m^-)$ 大于分类阈值，即 $\sum_{x^-\in D^-}\frac{1}{m^-}\left(I\left(f\left(x^+\right)<f\left(x^-\right)\right)\right)$。

故每个矩形的面积 $S=\frac{1}{m^+}\sum_{x^-\in D^-}\frac{1}{m^-}\left(I\left(f\left(x^+\right)<f\left(x^-\right)\right)\right)$，最后遍历所有真正例对所有矩形求和 $\sum_{x^+\in D^+}S$，就能够得到 ROC 曲线与 y 轴围成的面积。

我们现在再回顾一下 ℓ_{rank} 的公式，发现该公式就等于我们推导出来的 ROC 曲线与 y 轴围成的面积，而 y 轴与 x 轴围成的正方形面积为 1，故 $1-\ell_{rank}$ 就是有限样本 ROC 曲线下方的面积，得证。

$$
\begin{aligned}
\ell_{rank} &= \frac{1}{m^+m^-}\sum_{x^+\in D^+}\sum_{x^-\in D^-}\left(I\left(f\left(x^+\right)<f\left(x^-\right)\right)\right)\\
&= \sum_{x^+\in D^+}\frac{1}{m^+}\sum_{x^-\in D^-}\frac{1}{m^-}\left(I\left(f\left(x^+\right)<f\left(x^-\right)\right)\right)
\end{aligned}
$$

Proof 2

现在我们换一种理解方式。ROC 曲线下的面积表示的其实是从数据集中随机抽取正负例对 $\{x^+,x^-\}$，模型 $f(x)$ 对正例的打分 $f(x^+)$ 大于对负例的打分 $f(x^-)$ 的概率，即正样本排序

在负样本之前的概率。先前我们了定义 $AUC = 1 - \ell_{rank}$，因此问题转化为证明 ℓ_{rank} 为正样本排序在负样本之后的概率，即排序错误的概率（我们假设数据中不存在 $f(x^+) = f(x^-)$ 的情况，因此正负样本的得分都不同）。

排序错误的正负例对的组合共有：

$$\sum_{x^+ \in D^+} \sum_{x^- \in D^-} \left(I\left(f\left(x^+\right) < f\left(x^-\right) \right) \right)$$

正负例对的组合共有 m^+m^- 个，故排序错误的概率为:

$$\frac{\sum_{x^+ \in D^+} \sum_{x^- \in D^-} \left(I\left(f\left(x^+\right) < f\left(x^-\right) \right) \right)}{m^+m^-}$$

该式就等于 ℓ_{rank}，问题得证。

$$\begin{aligned}\ell_{rank} =& \frac{1}{m^+m^-} \sum_{x^+ \in D^+} \sum_{x^- \in D^-} \left(I\left(f\left(x^+\right) < f\left(x^-\right) \right) \right) \\ =& \frac{\sum_{x^+ \in D^+} \sum_{x^- \in D^-} \left(I\left(f\left(x^+\right) < f\left(x^-\right) \right) \right)}{m^+m^-}\end{aligned}$$

THE END. THANKS! ^_^