

作业 3

注: 使用 RStudio 作答的题目, 请在答案最后附上 R 语言代码。

1. (10 分) 在线性回归模型 $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$ 下, 假设 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 独立并都服从 $N(0, \sigma^2)$, 请使用极大似然法估计 β_0 和 β_1 , 得到 $\hat{\beta}_0$ 和 $\hat{\beta}_1$ 的形式, 并与最小二乘法的估计量比较, 它们是否有区别?

$$\text{似然函数: } L(\beta_0, \beta_1, S^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi S^2}} e^{-\frac{(Y_i - \beta_0 - \beta_1 X_i)^2}{2S^2}} = \prod_{i=1}^n \frac{1}{\sqrt{2\pi S^2}} e^{-\frac{(Y_i - \beta_0 - \beta_1 X_i)^2}{2S^2}}$$

$$\text{对数似然函数: } \ln L(\beta_0, \beta_1, S^2) = -\frac{n}{2} \ln(2\pi) - n \ln S - \sum_{i=1}^n \frac{(Y_i - \beta_0 - \beta_1 X_i)^2}{2S^2}$$

$$\text{即解 } \frac{\partial}{\partial \beta_0} \ln L(\beta_0, \beta_1, S^2) = 0 \text{ 和 } \frac{\partial}{\partial \beta_1} \ln L(\beta_0, \beta_1, S^2) = 0$$

$$\Rightarrow \begin{cases} \sum_{i=1}^n (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i) = 0 \\ \sum_{i=1}^n X_i (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i) = 0 \end{cases} \Rightarrow \begin{cases} \hat{\beta}_1 = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2} \\ \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X} \end{cases}$$

∴ $\hat{\beta}_0, \hat{\beta}_1$ 与最小二乘法的形式相同

2. (10 分) 为什么在逻辑回归模型的参数估计中, 不是去最小化 $\sum_{i=1}^n (y_i - \hat{p}_i)^2$? 其中 y_i 表示观测的类别, 取值为 0 或 1, \hat{p}_i 表示逻辑回归模型得到的对 $P(Y_i = 1)$ 的估计。

逻辑回归模型主要预测的是离散型目标变量, 而最小化 $\sum_{i=1}^n (y_i - \hat{p}_i)^2$ 是线性回归模型中的常用方法, 适用于连续型变量的预测。

3. (15 分) 我们在讲逻辑回归模型的参数估计时提到, $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$ 没有显式表达式, 需要借助梯度下降法、牛顿法等优化算法。请你自行学习这两个优化算法分别是什么, 有什么区别和联系。如果你是老师, 现在想要向学生讲述这两个优化算法, 你会怎么讲? 请把你的讲述内容用 PPT 呈现出来(提供 PPT 截图)。

梯度下降法原理

梯度下降法的数学公式:

梯度下降法的数学公式为: $\Delta w = -\alpha \nabla f(w)$, 其中 Δw 表示参数变化量, α 为学习率, $\nabla f(w)$ 为函数在当前点的梯度。

梯度下降法的基本概念

梯度下降法是一种迭代优化算法, 通过计算函数在当前点的梯度方向和大小, 来不断更新参数, 从而找到函数的最小值点。

梯度下降法的工作流程

梯度下降法首先设定一个初始点, 然后计算该点的梯度, 根据梯度的方向和大小, 更新参数, 再计算新的梯度, 如此反复迭代, 直到满足停止条件。

梯度下降法的优缺点

梯度下降法的优点是实现简单, 收敛速度快, 但缺点是可能会陷入局部最优解, 对初始点和学习率的选择敏感。

牛顿法原理

$$\text{梯度: } J_F = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}, \quad \text{海森矩阵: } Hf(a) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(a) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(a) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(a) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(a) \end{bmatrix},$$

牛顿法基本原理

牛顿法是一种迭代的优化算法, 通过求目标函数在当前点的梯度来更新参数, 使目标函数值逐步减小, 最终找到最小值点。

牛顿法迭代过程

牛顿法每次迭代时, 先计算目标函数在当前点的梯度和海森矩阵, 然后根据这些信息更新参数, 重复这个过程直到满足收敛条件。

牛顿法优缺点分析

牛顿法具有收敛速度快、精度高的优点, 但计算量较大, 且对初始点敏感, 可能会陷入局部最优解。因此, 在实际应用中需要权衡弊选合适的优化算法。

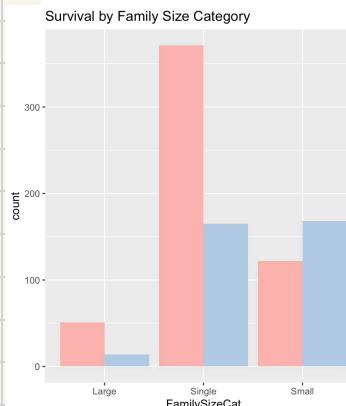
两种方法的比较分析

两种方法的计算复杂度对比
梯度下降法计算复杂度较低，适合处理大规模数据集；而牛顿法因为需要计算二阶导数，计算复杂度较高，更适合小规模数据。

- 1 梯度下降法与牛顿法的基本原理
梯度下降法属于函数在给定点处的梯度方向进行搜索，而牛顿法则通过函数的一阶和二阶导数来寻找函数的极值点。
- 2 两种方法的收敛速度分析
牛顿法由于利用了函数的二阶导数信息，收敛速度快于梯度下降法，但可能会陷入局部最优解。
- 3

4. (20 分) 在课程中讲到的 titanic 数据中，定义一个名为 FamilySize 的新变量，它等于 SibSp+Parch+1，即乘客随行的兄弟姐妹、配偶、父母、子女的数量加上自己。然后再基于 FamilySize 定义一个家庭大小的类别型变量 FamilySizeCat：如果 FamilySize = 1，则 FamilySizeCat = ‘Single’；如果 FamilySize 在 2 到 4 之间，则 FamilySizeCat = ‘Small’；如果 FamilySize 大于等于 5，则 FamilySizeCat = ‘Large’。注：本题请使用 RStudio 作答。

- (1) 使用 ggplot() 函数，按家庭大小类别画出乘客的幸存情况的柱状图。(5 分)
- (2) 把 FamilySizeCat 加入课程中拟合的逻辑回归模型，得到 FamilySizeCat 的两个回归系数（请提供模型拟合结果系数的截图），并对这两个回归系数作出具体解释。(10 分)
- (3) 画出(2)中所拟合模型的 ROC 曲线，与课程中所拟合模型的 ROC 曲线画在同一张图上（用不同的颜色，请提供你所画图形的截图），比较这两个模型的优劣。(5 分)

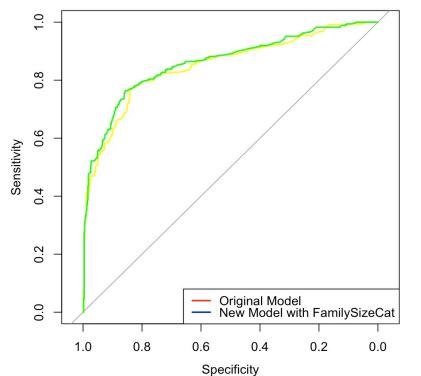


6(2)

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	2.37121	0.45780	5.180	2.22e-07 ***
Sexmale	-2.63534	0.20018	-13.165	< 2e-16 ***
as.factor(Pclass)2	-1.41685	0.26742	-5.298	1.17e-07 ***
as.factor(Pclass)3	-2.47383	0.25456	-9.718	< 2e-16 ***
Age	-0.04891	0.00773	-6.327	2.50e-10 ***
FamilySizeCatSingle	1.87336	0.40739	4.598	4.26e-06 ***
FamilySizeCatSmall	2.07805	0.40695	5.106	3.28e-07 ***

```
12 titanic <- read.csv('train_Titanic_complete.csv', header = TRUE)
13
14 #get FamilysizeCat
15 titanic$FamilySize <- titanic$SibSp + titanic$Parch + 1
16
17 titanic$FamilySizeCat[titanic$FamilySize == 1] <- 'Single'
18 titanic$FamilySizeCat[titanic$FamilySize > 2 & titanic$FamilySize <= 4] <- 'small'
19 titanic$FamilySizeCat[titanic$FamilySize >= 5] <- 'Large'
20
21 library(ggplot2)
22 ggplot(titanic, aes(x = FamilySizeCat, fill = as.factor(Survived))) +
23   geom_bar(position = position_dodge()) +
24   scale_fill_brewer(palette="Accent") +
25   guides(fill = guide_legend(title="Survived")) +
26   ggtitle("survival by Familysizecat")
```



回归系数 β_1 ：FamilySizeCat 变化 1 个单位，对生还率的影响为 β_1 个单位

反映了不同 FamilySize 对生还率的影响大小，其中 Small > Single

5. (15 分) k 近邻法虽然也是一个非常直观和简单的方法，但其在处理大规模数据集时的计算效率并不高，在对新实例进行分类预测时，需要遍历整个训练数据集以找到最近的 k 个邻居。为提高 k 近邻法的计算效率，有人提出可以利用 KD 树(k-dimensional tree)这一数据结构，在近邻搜索中快速排除与目标距离较远的区域，提高搜索效率。请点击[这里](#)阅读百度百科关于 KD 树的介绍，特别是结构实例、构建算法、查找算法部分。

- (1) 请仿照百度百科介绍中 KD 树的构建实例，给定 7 个二维数据点 $\{(1, 6), (2, 7), (3, 2), (4, 9), (5, 5), (7, 8), (8, 4)\}$ ，构建一棵 KD 树。给出空间划分的详细步骤，并绘制最后生成的 KD 树。(10 分)
- (2) 请仿照百度百科介绍中最近邻的查找算法，基于(1)中生成的 KD 树，以欧氏距离为距离量，搜索实例 $(8, 5, 5.2)$ 的最近邻点。给出搜索的详细步骤。(5 分)

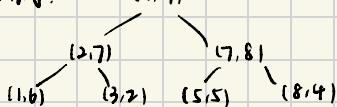
(1) 中位数根节点: $(4, 9)$

左子集: $\{(1, 6), (2, 7), (3, 2)\}$ 右子集: $\{(5, 5), (7, 8), (8, 4)\}$

对于左子集，子根节点为 $(2, 7)$ ，左子集: $\{(1, 6)\}$ ，右子集: $\{(3, 2)\}$

对于右子集，子根节点为 $(7, 8)$ ，左子集: $\{(5, 5)\}$ ，右子集: $\{(8, 4)\}$

∴ KD 树:



(2) $(8, 5, 5.2)$ 的查找路径: $(4, 9) \rightarrow (7, 8) \rightarrow (8, 4)$ ，距离 $d = \sqrt{0.5^2 + 1.2^2} = 1.3$

回溯至 father: $(7, 8)$ ，发现 $(7, 8) \notin B_r(8, 5, 5.2)$ ，即无更近的点

∴ 最近邻点为 $(8, 4)$

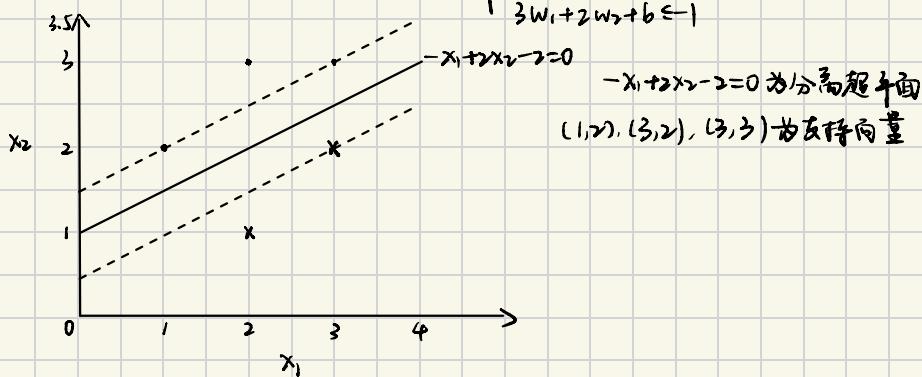
6. (10 分) 在课程 6.4 节中，我们使用 titanic 数据演示了朴素贝叶斯法的求解过程，其中仅考虑了性别、乘客等级、年龄段 3 个特征。利用上面第 4 题定义的乘客随行家庭大小的类别型变量 FamilySizeCat，使用朴素贝叶斯法对不同乘客幸存的概率进行估计，将结果填写至如下表格中：(答案保留 4 位小数)

Sex	Pclass	AgeCat	FamilySizeCat	幸存概率估计
Male	1	Child	Single	0.2887
Male	1	Child	Small	0.5569
Male	1	Child	Large	0.2004
Male	1	Senior	Single	0.1777
Male	1	Senior	Small	0.4008
Male	1	Senior	Large	0.1177
Male	3	Child	Single	0.2887
Male	3	Child	Small	0.5569
Male	3	Child	Large	0.2004
Female	3	Child	Single	0.8183
Female	3	Child	Small	0.9331
Female	3	Child	Large	0.7354

7. (10 分) 假设有观测数据 $\{(y_i, x_{i1}, x_{i2}), i = 1, 2, \dots, 5\} = \{(1, 1, 2), (1, 2, 3), (1, 3, 3), (-1, 2, 1), (-1, 3, 2)\}$. 试求出最大间隔分离超平面的表达式，并手动画一个二维散点图，在图上画出该分离超平面及支持向量.

经优化，即求 $\min \frac{1}{2} \|w\|^2$, 使得 $i=1, \dots, 5, y_i(w^T x_i + b) \geq 1$

$$\text{即 } y_i(w_1 x_{i1} + w_2 x_{i2} + b) \geq 1 \Rightarrow \begin{cases} w_1 + 2w_2 + b \geq 1 \\ 2w_1 + 3w_2 + b \geq 1 \\ 3w_1 + 3w_2 + b \geq 1 \\ 2w_1 + w_2 + b \leq -1 \\ 3w_1 + 2w_2 + b \leq -1 \end{cases} \Rightarrow w_1 = -1, w_2 = 2, b = -2 \text{ 此时 } \min \frac{1}{2} \|w\|^2 = \frac{5}{2}$$



8. (10 分) 请你出一道与第六章内容有关的作业题，并给出解答。题目和解答各占 5 分。评分会考虑题目的趣味性与难度(难度适中，不能太简单)。

题目：假设你是一名数据分析师，在一个电商平台工作，负责分析客户对新推出的智能家居产品的接受度。该产品在市场上的反响分为三类：正面 (Positive)，中立 (Neutral)，负面 (Negative)。你的目标是使用支持向量机 (SVM) 模型来预测用户的评价倾向。

提供的数据特征包括：

用户年龄 (User Age)

用户收入 (User Income)

购买频率 (Purchase Frequency)

产品评级 (Product Rating)

回购意向 (Repurchase Intent): 用户是否表示愿意回购 (1 表示是, 0 表示否)

虚拟数据样例：

年龄: 25, 34, 45, 23, 55

收入: 50, 80, 30, 25, 60

购买频率: 10, 20, 5, 3, 15

产品评级: 4, 5, 3, 2, 4

回购意向: 1, 1, 0, 0, 1

用户评价 (1 表示正面, 0 表示中立, -1 表示负面): 1, 1, 0, -1, 1

目标：

使用支持向量机 (SVM) 模型对给出的数据进行训练。

使用交叉验证来评估模型的性能。

解释 SVM 的优势在处理此类分类问题上的适用性。

```
# 加载 e1071包, 它包含 SVM 功能
library(e1071)

# 创建数据框
data <- data.frame(
  UserAge = c(25, 34, 45, 23, 55),
  UserIncome = c(50, 80, 30, 25, 60),
  PurchaseFrequency = c(10, 20, 5, 3, 15),
  ProductRating = c(4, 5, 3, 2, 4),
  RepurchaseIntent = c(1, 1, 0, 0, 1),
  UserReview = factor(c(1, 1, 0, -1, 1)) # 将用户评价转换为因子类型
)

# 使用 SVM 进行模型训练
svm_model <- svm(UserReview ~ ., data = data, kernel = "radial")

# 输出模型摘要
summary(svm_model)

# 使用交叉验证方法评估模型
set.seed(123) # 设置种子以确保结果可重复
cv_results <- tune(svm, train.x = UserReview ~ ., data = data, kernel = "radial", ranges = list(cost = 10^{(-1:2)}, gamma = 10^{(-2:1)}))

# 输出最佳模型的结果
print(cv_results$best.model)
```

步骤 3：模型解释和讨论

为什么选择 SVM：SVM 是一种有效的分类算法，通过最大化类之间的间隔来提高模型的泛化能力。对于具有明显分界的数据集，SVM 可以非常有效。

优势：SVM 在处理非线性边界问题时表现出色，通过使用不同的核函数，可以在原始特征空间中创建复杂的决策边界。

限制：对大数据集和缺失数据较敏感，参数调整和核函数选择可能复杂。