# 线性回归模型教案

## 课程基本信息

* **课程名称**：机器学习
* **课程主题**：线性回归模型
* **授课教师**：李老师
* **授课班级**：计算机科学与技术2201班
* **授课时间**：2025-04-29 08:00:00
* **课次**：第1次
* **课时**：2学时

## 教学目标

|  |  |
| --- | --- |
| **维度** | **具体目标** |
| **知识目标** | 1. 说出线性回归的模型最小二乘估计思想。<br>2. 理解正规方程与梯度下降两类求解方式及其数值特点。<br>3. 解释欠拟合、过拟合与正则化（岭回归 / Lasso）在回归中的作用。 |
| **技能目标** | 1. 能够独立使用 Python/NumPy 或 scikit-learn 训练一元和多元线性回归模型，并完成模型评估。<br>2. 能对给定数据集选择合适的特征缩放、交叉验证和正则化策略。 |
| **情感与态度目标** | 1. 形成"模型简单性 vs. 表达能力"平衡的工程思维。<br>2. 培养用数据与实验而非直觉作决策的科学态度。 |

## 教学重点

* 损失函数 → 最优化：均方误差的由来、几何含义与凸优化特性
* 解析解 vs. 数值解：正规方程与梯度下降的条件、效率、数值稳定性比较
* 正则化：λ 的物理意义，岭回归与 L1 正则对参数的不同影响

## 教学难点

|  |  |
| --- | --- |
| **难点** | **化解策略** |
| **梯度下降的收敛性与学习率选择** | 动画记录 loss 曲线并评价收敛速度。 |
| **多重共线性引发的矩阵不可逆** | 通过可视化列空间、演示正规方程求解失败，再引入岭回归作为解决方案。 |
| **正则化超参数 λ 的调优** | 课堂现场用交叉验证网格搜索，展示"U 形"验证误差曲线，帮助学生直观体会。 |

## 教学准备

### 教学资源

1. PPT讲义
2. Jupyter Notebook 演示代码
3. 线性回归可视化工具/动画
4. 学生练习数据集：波士顿房价数据集、加州房价数据集

### 教学工具

1. 多媒体教室
2. 演示用笔记本电脑
3. 投影设备
4. 白板/黑板

## 教学过程

### 1. 复习旧知（10分钟）

#### 教师活动

* 回顾机器学习基本概念：监督学习与无监督学习的区别
* 复习回归与分类问题的基本特点
* 简述特征、标签、损失函数等基本术语

#### 学生活动

* 回答教师提问
* 复习笔记
* 积极参与课堂讨论

### 2. 导入新课（10分钟）

#### 教师活动

* 展示实际案例：房价预测问题
* 提问：如何基于已知数据预测未知房价？
* 引入线性回归的直观理解：找到一条直线/平面拟合数据点

#### 学生活动

* 思考问题
* 提出初步解决方案
* 记录关键概念

#### 案例展示

python

*# 展示房价预测数据可视化*import matplotlib.pyplot as plt

import numpy as np

*# 示例数据*size = np.array([50, 65, 80, 95, 110, 125, 140, 155])

*# 房屋面积*price = np.array([150, 175, 210, 230, 270, 290, 330, 350])

*# 房价（万元）*plt.figure(figsize=(10, 6))

plt.scatter(size, price, color='blue', label='数据点')

plt.xlabel('房屋面积（平方米）')

plt.ylabel('房价（万元）')

plt.title('房屋面积与房价关系')

plt.grid(True)

plt.legend()

plt.show()

### 3. 讲授（60分钟）

#### 3.1 线性回归模型基础（15分钟）

##### 教师活动

* 讲解一元线性回归与多元线性回归模型表达式
* 推导模型假设：$y = w^T x + b + \epsilon$
* 解释参数的几何含义：斜率与截距

##### 学生活动

* 记录笔记
* 理解模型
* 尝试用自己的话解释模型含义

##### 核心内容

* 一元线性回归：$y = wx + b$
* 多元线性回归：$y = w\_1 x\_1 + w\_2 x\_2 + ... + w\_n x\_n + b$
* 矩阵形式：$y = Xw$（其中$X$包含截距项）

#### 3.2 损失函数与最小二乘法（15分钟）

##### 教师活动

* 引入损失函数概念：评估模型预测误差
* 推导均方误差（MSE）损失函数
* 解释最小二乘法的几何意义：垂直距离最小化
* 分析MSE为何是一个良好的损失函数：凸函数特性

##### 学生活动

* 记录公式推导
* 思考问题：为什么选择平方误差而非绝对误差？
* 尝试自行推导简单例子的最小二乘解

##### 核心内容

* 均方误差：$MSE = \frac{1}{n}\sum\_{i=1}^{n}(y\_i - \hat{y}\_i)^2$
* 损失函数：$J(w) = \frac{1}{n}\sum\_{i=1}^{n}(y\_i - w^T x\_i)^2$
* 矩阵形式：$J(w) = \frac{1}{n}(y - Xw)^T(y - Xw)$

#### 3.3 求解方法一：正规方程（15分钟）

##### 教师活动

* 推导正规方程：对损失函数求导并令导数为零
* 解析解：$w = (X^T X)^{-1}X^T y$
* 分析正规方程的计算复杂度与适用条件
* 讨论正规方程的局限性：特征数量大时的计算效率问题、多重共线性问题

##### 学生活动

* 跟随推导过程
* 记录关键步骤
* 思考问题：什么情况下$(X^T X)$不可逆？

##### 核心内容

* 正规方程推导：$\frac{\partial J}{\partial w} = \frac{2}{n}X^T(Xw-y) = 0$
* 解析解：$w = (X^T X)^{-1}X^T y$
* 正规方程的优势：一步求解，无需迭代
* 正规方程的局限：计算复杂度$O(n^3)$，特征数量大时效率低

#### 3.4 求解方法二：梯度下降（15分钟）

##### 教师活动

* 引入梯度下降算法基本思想
* 推导线性回归的梯度表达式
* 演示批量梯度下降、随机梯度下降和小批量梯度下降三种变体
* 讨论学习率选择及其对收敛性的影响
* 可视化梯度下降过程

##### 学生活动

* 记录算法步骤
* 思考不同变体的优缺点
* 观察并分析梯度下降的收敛过程

##### 核心内容

* 梯度计算：$\nabla\_w J(w) = \frac{2}{n}X^T(Xw-y)$
* 参数更新规则：$w := w - \alpha \nabla\_w J(w)$
* 批量梯度下降：使用所有样本计算梯度
* 随机梯度下降：每次使用单个样本
* 小批量梯度下降：使用一批样本
* 学习率选择：过大导致发散，过小导致收敛慢

### 4. 练习（30分钟）

#### 4.1 课堂练习（15分钟）

##### 教师活动

* 提供实际数据集（如波士顿房价数据）
* 指导学生用Python实现线性回归
* 巡视指导，解答问题

##### 学生活动

* 应用所学知识实现线性回归
* 尝试用正规方程和梯度下降两种方法求解
* 比较两种方法的结果和效率

##### 练习内容示例

python

*# 线性回归实现练习*import numpy as np

from sklearn.datasets import load\_boston

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

*# 1. 加载数据*boston = load\_boston()

X = boston.data

y = boston.target

*# 2. 划分训练集和测试集*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

*# 3. 特征缩放*

scaler = StandardScaler()X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

*# 4. 添加截距项*

X\_train\_b = np.c\_[np.ones((X\_train\_scaled.shape[0], 1)), X\_train\_scaled]

X\_test\_b = np.c\_[np.ones((X\_test\_scaled.shape[0], 1)), X\_test\_scaled]

*# 5. 使用正规方程求解*

def normal\_equation(X, y):

return np.linalg.inv(X.T @ X) @ X.T @ y

theta\_best = normal\_equation(X\_train\_b, y\_train)

*# 6. 预测和评估*

y\_pred = X\_test\_b @ theta\_best

mse = np.mean((y\_test - y\_pred) \*\* 2)

print(f"测试集MSE: {mse:.2f}")

#### 4.2 小组讨论（15分钟）

##### 教师活动

* 设置讨论主题：面对大规模数据，如何选择合适的线性回归求解方法？
* 引导小组讨论，确保参与度
* 组织代表分享讨论结果

##### 学生活动

* 小组内讨论问题
* 分析两种求解方法的适用场景
* 小组代表总结发言

##### 讨论问题

1. 当特征数量极大（如百万级）时，应选择哪种求解方法？为什么？
2. 如何判断数据是否存在多重共线性？这会对模型产生什么影响？
3. 在实际应用中，如何选择合适的学习率？

### 5. 总结（10分钟）

#### 教师活动

* 归纳本节课核心知识点
* 强调线性回归在机器学习中的基础地位
* 简要介绍下节课内容：正则化线性回归
* 布置课后任务

#### 学生活动

* 记录总结要点
* 提出疑问和补充
* 明确课后任务

#### 课程要点总结

1. 线性回归模型是机器学习最基础的模型之一，用于预测连续型目标变量
2. 最小二乘法是线性回归的核心思想，通过最小化均方误差来找到最优参数
3. 求解线性回归有两种主要方法：
   * 正规方程：直接求解，适用于特征较少的情况
   * 梯度下降：迭代求解，适用于大规模数据集
4. 模型评估通常使用均方误差(MSE)、决定系数(R²)等指标

#### 课后任务

1. 完成教材第3章线性回归部分的习题（P56-58）
2. 使用Scikit-learn库实现一个多元线性回归，并尝试调整参数
3. 阅读岭回归和Lasso回归相关内容，为下节课做准备

## 板书设计

线性回归模型┌─────────────────────────────────────────┐│ 1. 线性回归基础 ││ \* 一元线性回归: y = wx + b ││ \* 多元线性回归: y = w₁x₁ + w₂x₂ + ... + b ││ ││ 2. 最小二乘法 ││ \* 损失函数: J(w) = 1/n∑(yᵢ - ŷᵢ)² ││ \* 目标: min J(w) ││ ││ 3. 求解方法 ││ \* 正规方程: w = (X^T X)⁻¹X^T y ││ \* 梯度下降: w := w - α∇J(w) ││ ││ 4. 方法比较 ││ \* 正规方程: 一步求解,特征少时高效 ││ \* 梯度下降: 迭代求解,大数据时高效 │└─────────────────────────────────────────┘

## 教学反思（课后填写）

### 教学效果

* 学生对线性回归的基本概念理解良好
* 正规方程的矩阵推导部分学生理解有困难
* 梯度下降可视化效果好，学生兴趣高

### 问题与改进

* 时间控制需要改进，讲授部分超时
* 学生动手能力参差不齐，需要准备不同难度的练习
* 可以增加更多实际应用案例，提高学习兴趣

### 后续调整

* 优化PPT内容，减少数学推导细节，增加直观解释
* 提前准备代码模板，降低编程实现门槛
* 增加课前预习材料，提高课堂效率