

作业答案

1.完成线性回归部分的思维导图

略

2.描述梯度下降算法思想，自己推导银行信贷的案例

$$\theta_{i+1} = \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta)$$

- 沿着梯度下降的方向求解极小值
- 优化过程
 1. 给定初始位置、步长（学习率）
 2. 计算该点当前的梯度的负方向
 3. 向该负方向移动步长
 4. 重复 2-3 步 直至收敛
 - 两次差距小于指定的阈值
 - 达到指定的迭代次数
- 算法分类
 - 1.全梯度下降(FGD): 使用全部样本的梯度值
 - 2.随机梯度下降(SGD): 随机选择一个样本的梯度值更新权重
 - 3.小批量梯度下降(Mini-Batch): 随机选择小部分样本的梯度值更新权重
 - 4.随机平均梯度下降(SAG): 使用以前产生的梯度值来指导当前的梯度计算

3.说明欠拟合和过拟合的相关内容

欠拟合（**对数据迟钝**）是表现在训练集和测试集都不能很好的拟合数据。表现为预测的准确率不足。原因是因为模型过于简单，更深层的原因通常是因为训练量不足或特征值过少。一般欠拟合可以通过增加特征值（组合、泛化、相关性）或增加训练量来矫正。

过拟合（**对数据敏感**）是表现在训练集拟合度高，而测试集拟合度不足。原因是因为模型过于复杂，更深层原因是由于训练中特征值过多，且受异常值影响过大导致的模型过于复杂。一般过拟合需要通过筛选特征值或者清洗异常数据，或者正则化来矫正。

4.使用L1和L2正则化方法实现波士顿房价预测

```
# 1. 导入依赖包
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge, Lasso
from sklearn.metrics import mean_squared_error, mean_absolute_error

# lasso回归
def boston_lasso():
```

```

# 2.读取数据并分割数据集
# 2.1 读取数据
data = pd.read_csv("波士顿房价xy.csv")
# 2.2 数据集分割
x_train, x_test, y_train, y_test = train_test_split(data.iloc[:, :-1],
data.iloc[:, -1], test_size=0.3, random_state=22)

# 3.特征工程-数据预处理-标准化
transfer = StandardScaler()
x_train = transfer.fit_transform(x_train)
x_test = transfer.transform(x_test)

# 4.Lasso模型实例化及模型训练
estimator = Lasso(alpha=0.005)
estimator.fit(x_train, y_train)

# 5.模型训练及评估
y_predict = estimator.predict(x_test)
error_mse = mean_squared_error(y_test, y_predict)
print("L1正则化均方误差：", error_mse)

# 岭回归
def boston_ridge():
    # 2.读取数据并分割数据集
    # 2.1 读取数据
    data = pd.read_csv("波士顿房价xy.csv")
    # 2.2 数据集分割
    x_train, x_test, y_train, y_test = train_test_split(data.iloc[:, :-1],
data.iloc[:, -1], test_size=0.3, random_state=22)

    # 3.特征工程-数据预处理-标准化
    transfer = StandardScaler()
    x_train = transfer.fit_transform(x_train)
    x_test = transfer.transform(x_test)

    # 4.Ridge模型实例化及模型训练
    estimator = Ridge(alpha=0.005)
    estimator.fit(x_train, y_train)
    print("L2正则化梯度下降权重系数：", estimator.coef_)
    print("L2正则化梯度下降偏置：", estimator.intercept_)

    # 5.模型训练及评估
    y_predict = estimator.predict(x_test)
    error_mse = mean_squared_error(y_test, y_predict)
    print("L2正则化均方误差：", error_mse)

if __name__ == '__main__':
    # L1正则化
    boston_lasso()
    # L2正则化
    boston_ridge()

```

