南京理工大学紫金学院

# 《机器学习》实验报告



|  |  |
| --- | --- |
| **课程编号：0332301** |  |
| **课程名称：机器学习** |  |
| **学 院：计算机学院** |  |
| **专 业：** |  |
| **学 号：** |  |
| **姓 名：** |  |
| **任课教师：戴治波** |  |

2022年11月

## 实验一 线性回归（黑体三号）

# 一、实验目的

1. 搭建机器学习开发平台。
2. 掌握线性回归分析的基本思想和基本方法（难点）。
3. 掌握最小二乘法原理及实现（重点）。

# 二、实验原理及说明

线性回归是在已有数据集上通过构建一个线性的模型来拟合该数据集特征向量的各个分量之间的关系，对于需要预测结果的新数据，我们利用已经拟合好的线性模型来预测其结果。最小二乘法是用的比较广泛的一种方法。

高斯于 1823 年在误差独立同分布的假定下，证明了最小二乘方法的一个最优性质:在所有无偏的线性估计类中, 最小二乘方法是其中方差最小的！ 对于数据 (𝑥𝑥𝑖𝑖,𝑦𝑦𝑖𝑖)(𝑖𝑖=1,2,3...,n)，拟合出函数 ℎ(𝑥𝑥)有误差，即残差：𝑟𝑟𝑖𝑖=ℎ(𝑥𝑥𝑖𝑖)−𝑦𝑦𝑖𝑖，此时 L2 范数(残差平方和) 最小时， h(x) 和 y 相似度最高， 更拟合一般的 H(x) 为 n 次的多项式：

𝐻𝐻(𝑥𝑥)=𝑤𝑤0+𝑤𝑤1𝑥𝑥+𝑤𝑤2𝑥𝑥2+...𝑤𝑤𝑛𝑛𝑥𝑥𝑛𝑛，其中 w(w0,w1,w2,...,wn)为参数，最小二乘法就是要找到一组 w(w0,w1,w2,...,wn)，使得残差平方和最小。

# 三、实验内容

1．搭建机器学习开发平台。

1. 安装 Anaconda 开发平台，使用 jupyter notebook 进行编辑。
2. 建立虚拟环境：

创建虚拟环境：使用 conda create -n your\_env\_name python=X.X（3.6、3.8 等）， anaconda 命令创建 python 版本为 X.X、名字为 your\_env\_name 的虚拟环境。 your\_env\_name 文件可以在 Anaconda 安装目录 envs 文件下找到。

激 活 虚 拟 环 境 ： 使 用 如 下 命 令 即 可 激 活 创 建 的 虚 拟 环 境 activate your\_env\_name(虚拟环境名称)，此时使用 python --version 可以检查当前 python 版本是否为想要的（即虚拟环境的 python 版本）。

退出虚拟环境：使用如下命令即可退出创建的虚拟环境 deactivate env\_name，

也可以使用“activate root”切回 root 环境。

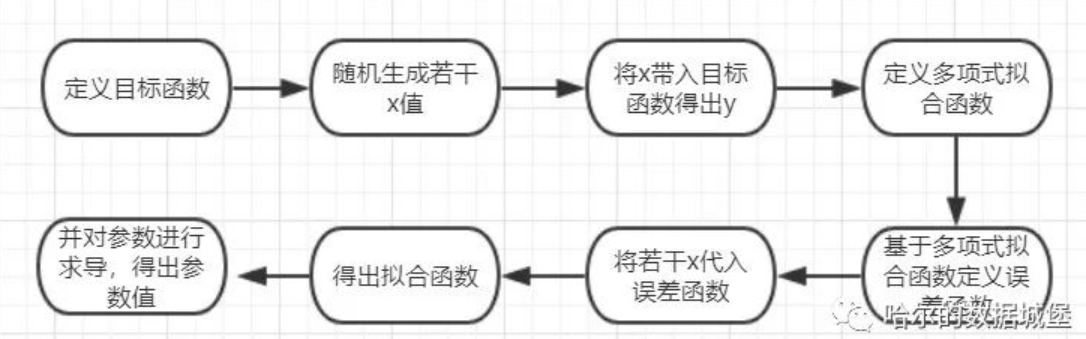
删除虚拟环境：使用命令 conda remove -n your\_env\_name(虚拟环境名称) --al

即可删除。

1. 在虚拟环境中安装相应的包文件，如：pandas、numpy、matplotlib、scipy、sklearn等，具体实验可具体安装。可以使用 conda list 命令查看安装了哪些包。

2．最小二乘法实现。

1. 最小二乘法的 python 实现：我们用目标函数𝑦𝑦=𝑠𝑠𝑖𝑖𝑛𝑛2𝜋𝜋𝑥𝑥, 加上一个正态分布的噪音干扰，再用多项式去拟合（分别取 0 阶、1 阶、3 阶、9 阶进行拟合）。程序流程如下：



写出代码和可视化结果。

目标函数：代入生成的x，生成对应的y

|  |
| --- |
| def func(x):  return np.sin(2\*np.pi\*x) |

随机生成10个x进行实验：

|  |
| --- |
| x = np.linspace(0, 1, 10) |

构造多项式拟合函数：

|  |
| --- |
| #多项式  def fit\_func(p,x):  """  eg:p = np.poly1d([2,3,5,7])  　　　print(p)==>>2x3 + 3x2 + 5x + 7  """  f = np.poly1d(p)  return f(x) |

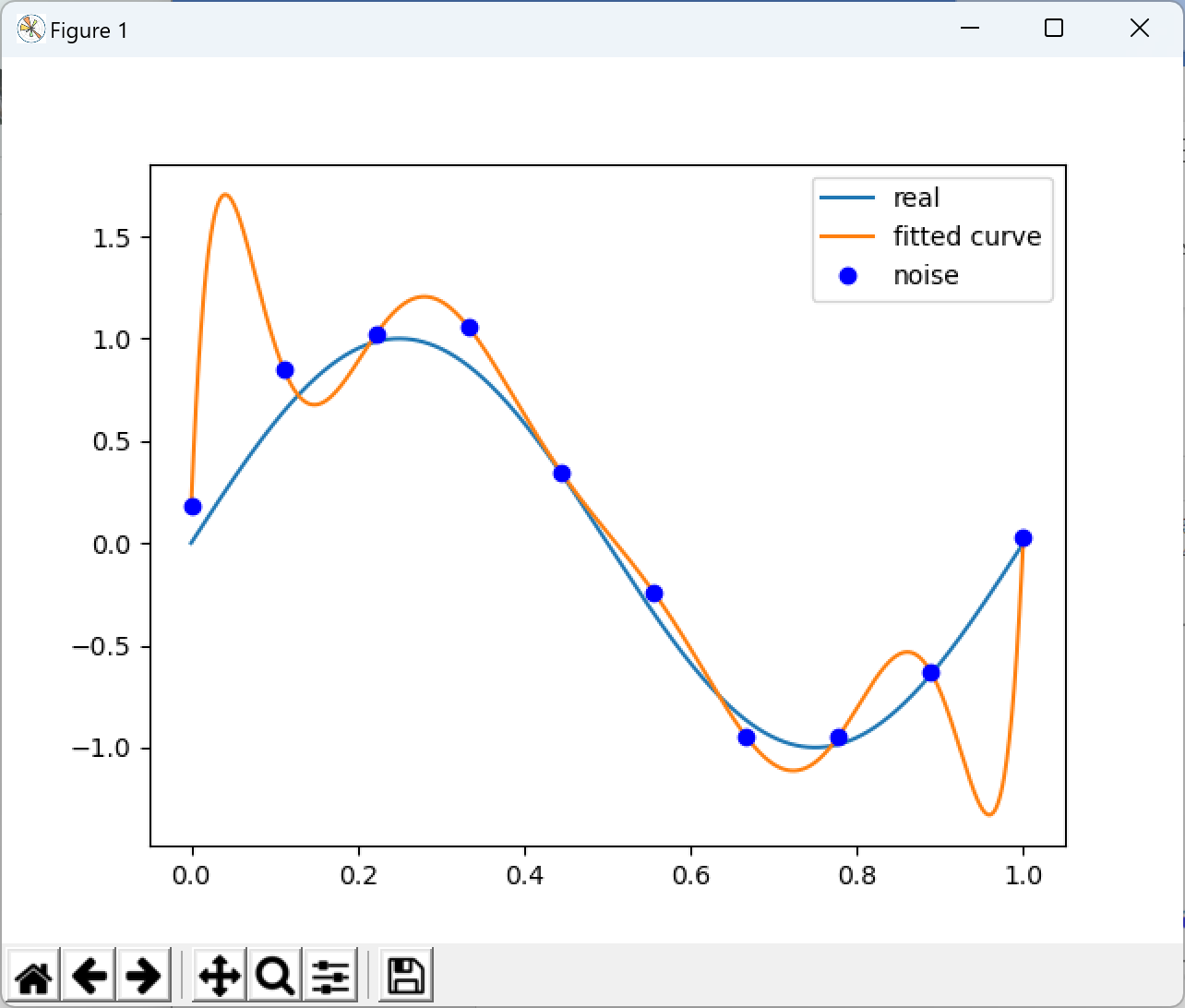
计算误差：

|  |
| --- |
| #残差  def residuals\_func(p, x, y):  ret = fit\_func(p, x) - y  return ret |

leastsq 是 scipy 库 进行最小二乘法计算的函数，也就是通过误差函数以及数据点进行我们前面讲的对参数进行求导操作，最后得出我们拟合出来的函数。

|  |
| --- |
| def fitting(M=0):  """  n 为 多项式的次数  """  # 随机初始化多项式参数  #numpy.random.rand(d0)的随机样本位于[0, 1)之间。d0表示返回多少个  p\_init = np.random.rand(M+1) #生成M+1个随机数的列表  # 最小二乘法  p\_lsq = leastsq(residuals\_func, p\_init, args=(x, y)) # 三个参数：误差函数、函数参数列表、数据点  print('Fitting Parameters:', p\_lsq[0])    # 可视化  plt.plot(x\_points, func(x\_points), label='real')  plt.plot(x\_points, fit\_func(p\_lsq[0], x\_points), label='fitted curve')  plt.plot(x, y, 'bo', label='noise')  plt.legend()  return p\_lsq    # M=0  p\_lsq = fitting(M=0) |

我们从一次函数依次增加项式，找到最合适的拟合曲线。到9次的时候，已经过拟合这些点了 。



1. 过拟合的情况下，引入正则化项进行优化实现，写出代码和可视化结果。

# 四、实验安全事项

实验过程中注意用电安全。

# 五、实验提交方式

√ 实验报告 □ 现场打分 □ 线上平台提交

实验二 线性分类模型

一、实验目的

1. 掌握感知机原理及实现。

2. 掌握 K 近邻基本思想和基本方法（重难点）。

3. 了解朴素贝叶斯原理及实现。

二、实验原理及说明

1. 感知机

感知机是根据输入实例的特征向量𝑥𝑥x 对其进行二类分类的线性分类模型：

𝑓𝑓(𝑥𝑥)=sign(𝑤𝑤⋅𝑥𝑥+𝑏𝑏)

感知机模型对应于输入空间（特征空间）中的分离超平面𝑤𝑤⋅𝑥𝑥+𝑏𝑏=0

感知机学习的策略是极小化损失函数：

min𝐿𝐿(𝑤𝑤,𝑏𝑏)=−∑yi(w⋅xi+b)，其中损失函数对应于误分类点到分离超平面的总距离。感知机学习算法是基于随机梯度下降法的对损失函数的最优化算法，有原始形式和对偶形式。算法简单且易于实现。原始形式中，首先任意选取一个超平面，然后用梯度下降法不断极小化目标函数。在这个过程中一次随机选取一个误分类点使其梯度下降。𝑤𝑤=𝑤𝑤+𝜂𝜂𝜂𝜂𝑖𝑖𝑥𝑥𝑖𝑖

𝑏𝑏=𝑏𝑏+𝜂𝜂𝜂𝜂𝑖𝑖当实例点被误分类，即位于分离超平面的错误侧，则调整 w, b 的值，使分离超平面向该无分类点的一侧移动，直至误分类点被正确分类。

2. K 近邻

k 近邻法是基本且简单的分类与回归方法。k 近邻法的基本做法是：对给定的训练实例点和输入实例点，首先确定输入实例点的 k 个最近邻训练实例点，然后利用这 k个训练实例点的类的多数来预测输入实例点的类。

k 近邻模型对应于基于训练数据集对特征空间的一个划分。k 近邻法中，当训练集、距离度量、k 值及分类决策规则确定后，其结果唯一确定。

4 k 近邻法三要素：

距离度量、k 值的选择和分类决策规则。常用的距离度量是欧氏距离及更一般的 pL 距离。k 值小时，k 近邻模型更复杂；k 值大时，k 近邻模型更简单。k 值的选择反映了对近似误差与估计误差之间的权衡，通常由交叉验证选择最优的 k。常用的分类决策规则是多数表决，对应于经验风险最小化。

k 近邻法的实现需要考虑如何快速搜索 k 个最近邻点。kd 树是一种便于对 k 维空间中的数据进行快速检索的数据结构。kd 树是二叉树，表示对 k 维空间的一个划分，其每个结点对应于 k 维空间划分中的一个超矩形区域。利用 kd 树可以省去对大部分数据点的搜索， 从而减少搜索的计算量。

3. 朴素贝叶斯

朴素贝叶斯法是典型的生成学习方法。生成方法由训练数据学习联合概率分

布 P(X,Y)，然后求得后验概率分布 P(Y|X)。具体来说，利用训练数据学习 P(X|Y)和 P(Y)的估计，得到联合概率分布：P(X,Y)＝P(Y)P(X|Y)其中，概率估计方法可以是极大似然估计或贝叶斯估计。朴素贝叶斯法的基本假设是条件独立性