



机器学习实验指导书

日期：2025 年 3 月 25 日

目 录

实验注意事项:	3
实验一: 贝叶斯决策论	4
实验二: 线性判别函数和参数、非参数估计	7
实验三: 基于决策树算法完成鸢尾花卉品种预测任务	14
实验四: 神经网络学习	16
实验五: 支持向量机	19
实验六: 集成学习	21

实验注意事项

1. 本次实验共有六个题目，包含了本学期的主要内容，目的是促进同学们巩固课堂讲解的内容，并通过实验实现加深对机器学习理论和方法的理解。
2. 实验实现不限编程语言（Python、C++ or Matlab），推荐使用python。
3. 实现过程中不得使用现成工具包（如Sklearn、Pytorch等，对机器学习感兴趣可以自行了解），代码要求同学们亲自书写。【思维发散除外】
4. 除了代码，要求同学们认真撰写实验报告，格式规范、分析透彻、内容丰富。
5. 代码与实验报告不得有任何抄袭内容。

祝大家完美完成全部题目，并取得好成绩！

实验一：贝叶斯决策论

【实验背景】

信用风险是指银行向用户提供金融服务后，用户不还款的概率。信用风险一直是银行贷款决策中广泛研究的领域。信用风险对银行和金融机构，特别是商业银行来说，起着至关重要的作用，但是一直以来都比较难管理。

【实验说明】

本实验以贷款违约为背景，要求使用贝叶斯决策论的相关知识在训练集上构建模型，在测试集上进行贷款违约预测并计算分类准确度。

【实验数据说明】

训练数据集 train.csv 包含 9000 条数据。

测试数据集 test.csv 包含 1000 条数据。

最后一列为样本标签。

注意，训练集和测试集中都有缺失值存在。

【数据字段说明】

表 1-1 贷款违约数据字段说明

字段	描述
loan_id	贷款记录唯一标识
user_id	借款人唯一标识
total_loan	贷款数额
year_of_loan	贷款年份
interest	当前贷款利率
monthly_payment	分期付款金额
grade	贷款级别
employment_type	所在公司类型

industry	工作领域
work_year	工作年限
home_exist	是否有房
censor_status	审核情况
issue_date	贷款发放的月份
use	贷款用途类别
post_code	贷款人申请时邮政编码
region	地区编码
debt_loan_ratio	债务收入比
del_in_18month	借款人过去 18 个月逾期 30 天以上的违约事件数
scoring_low	借款人在贷款评分中所属的下限范围
scoring_high	借款人在贷款评分中所属的上限范围
known_outstanding_loan	借款人档案中未结信用额度的数量
known_dero	贬损公共记录的数量
pub_dero_bankrup	公开记录清除的数量
recircle_bal	信贷周转余额合计
recircle_util	循环额度利用率
initial_list_status	贷款的初始列表状态
app_type	是否个人申请
earlies_credit_mon	借款人最早报告的信用额度开立的月份
title	借款人提供的贷款名称
policy_code	公开可用的策略_代码=1新产品不公开可用的策略_代码=2
f 系列匿名特征	匿名特征 f0-f4，为一些贷款人行为计数特征的处理
early_return	借款人提前还款次数
early_return_amount	贷款人提前还款累积金额
early_return_amount_3mon	近 3 个月内提前还款金额
isDefault	贷款是否违约（预测标签）

【实验注意事项】

1. 实验不限制使用何种高级语言，推荐使用 python 中 pandas 库处理 csv 文件。

(1) 安装: `pip install pandas/conda install pandas` 【建议使用 conda 命令，安装 anaconda 环境】

(2) 导入: `import pandas as pd`

2. 在进行贝叶斯分类之前重点是对数据进行预处理操作，如，缺失值的填充、将文字表述转为数值型、日期处理格式（处理成“年-月-日”三列属性或者以最早时间为基准计算差值）、无关属性的删除、多列数据融合等方面。

3. 数据中存在大量连续值的属性，不能直接计算似然，需要将连续属性离散化。

4. 另外，特别注意零概率问题，贝叶斯算法中如果乘以 0 的话就会失去意义，需要使用平滑技术。【如拉普拉斯平滑等相关知识】

5. 实验目的是使用贝叶斯处理实际问题，不得使用现成工具包直接进行分类。【该点切记！！！这个一定要自己写，才能感受贝叶斯的魅力】

6. 实验代码中需要有必要的注释，具有良好的可读性。

实验二：线性判别函数与参数、非参数估计

一、线性判别函数

【实验目的】

掌握线性判别函数算法的原理

【实验数据格式】

实验数据的格式如表 2-1 所示。

表 2-1 线性判别函数实验数据格式样例

	x1	x2	y
1	1.9643	4.5957	1
2	2.2753	3.8589	1
3	2.9781	4.5651	1
4	2.9320	3.5519	1
5	3.5772	2.8560	1

【实验内容及说明】

采用 exp2_1.mat 中的数据，实现线性判别函数分类算法，其 x1、x2 为二维自变量，y 为样本类别。编程实现线性判别函数分类，并做出分类结果可视化。

xxx.mat 格式的数据文件可以使用 scipy.io 进行读取处理，**算法实现部分不可借助现成库。**

(安装: `pip install scipy` 导入: `import scipy`)

二、 最大似然估计

【实验目的】

掌握用最大似然估计进行参数估计的原理； 当训练样本服从多元正态分布时，计算不同高斯情况下的均值和方差

【实验数据格式】

实验数据的格式如表 2-2 所示。

表 2-2 最大似然估计实验数据格式样例

样 本	类1			类2		
	x1	x2	x3	x1	x2	x3
1	0.42	-0.087	0.58	-0.4	0.58	0.089
2	-0.2	-3.3	-3.4	-0.31	0.27	-0.04
3	1.3	-0.32	1.7	0.38	0.055	-0.035
4	0.39	0.71	0.23	-0.15	0.53	0.011
5	-1.6	-5.3	-0.15	-0.35	0.47	0.034
6	-0.029	0.89	-4.7	0.17	0.69	0.1
7	-0.23	1.9	2.2	-0.011	0.55	-0.18
8	0.27	-0.3	-0.87	-0.27	0.61	0.12
9	-1.9	0.76	-2.1	-0.065	0.49	0.0012
10	0.87	-1	-2.6	-0.12	0.054	-0.063

【实验内容及说明】

使用表 2-2给出的三维数据或者使用 exp2_2.xlsx 中的数据：

(1) 编写程序，对类 1 和类 2 中的三个特征 x_i 分别求解最大似然估计的均值 $\hat{\mu}$ 和方差 $\hat{\sigma}^2$ 。

(2) 编写程序，处理二维数据的情形 $p(x) \sim N(\mu, \Sigma)$ 。对类 1 和类 2 中

任意两个特征的组合分别求解最大似然估计的均值 $\hat{\mu}$ 和方差 $\hat{\Sigma}$ （每个类有3种可能）。

(3) 编写程序，处理三维数据的情形 $p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ 。对类 1 和类 2 中三个特征求解最大似然估计的均值 $\hat{\mu}$ 和方差 $\hat{\Sigma}$ 。

(4) 假设该三维高斯模型是可分离的，即 $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2)$ ，编写程序估计类 1 和类 2 中的均值和协方差矩阵中的参数。

(5) 比较前 4 种方法计算出来的每一个特征的均值 μ_i 的异同，并加以解释。

(6) 比较前 4 种方法计算出来的每一个特征的方差 σ_i 的异同，并加以解释。

数据读取：可以使用python的pandas库读取xxx.xlsx格式的文件

三、非参数估计

【实验目的】

掌握用非参数的方法估计概率密度

了解parzen 窗方法的原理

了解k近邻方法的原理

【实验数据】

实验数据的格式如表 2-3 所示。

表 2-3 非参数估计实验数据格式样例

样 本	类1			类2			类3		
	x1	x2	x3	x1	x2	x3	x1	x2	x3
1	0.28	1.31	-6.2	0.011	1.03	-0.21	1.36	2.17	0.14
2	0.07	0.58	-0.78	1.27	1.28	0.08	1.41	1.45	-0.38
3	1.54	2.01	-1.63	0.13	3.12	0.16	1.22	0.99	0.69
4	-0.44	1.18	-4.32	-0.21	1.23	-0.11	2.46	2.19	1.31
5	-0.81	0.21	5.73	-2.18	1.39	-0.19	0.68	0.79	0.87
6	1.52	3.16	2.77	0.34	1.96	-0.16	2.51	3.22	1.35
7	2.20	2.42	-0.19	-1.38	0.94	0.45	0.60	2.44	0.92
8	0.91	1.94	6.21	-0.12	0.82	0.17	0.64	0.13	0.97
9	0.65	1.93	4.38	-1.44	2.31	0.14	0.85	0.58	0.99
10	-0.26	0.82	-0.96	0.26	1.94	0.08	0.66	0.51	0.88

【实验内容及说明】

Parzen 窗估计：

使用表 2-3中的数据或者使用 exp2_3.xlsx 中的数据进行Parzen 窗估计和设计分类器。窗函数为一个球形的高斯函数如公式 2-1所示：

$$\varphi\left(\frac{(x-x_i)}{h}\right) \propto \exp[-(x-x_i)^T(x-x_i)/(2h^2)] \quad (2-1)$$

编写程序，使用 Parzen 窗估计方法对任意一个的测试样本点 x 进行分类。对分类器的训练则使用表2-2中的三维数据。令 $h = 1$ ，分类样本点为 $(0.5, 1.0, 0.0)^T$ ， $(0.31, 1.51, -0.50)^T$ ， $(-0.3, 0.44, -0.1)^T$ 。

k-近邻概率密度估计：

对表 2-3表格中的数据使用k-近邻方法进行概率密度估计：

- 1) 编写程序，对于一维的情况，当有 n 个数据样本点时，进行k-近邻概率密度估计。对表格中的类1的特征 x_3 ，用程序画出当 $k=1, 3, 5$ 时的概率密度估计结果。
- 2) 编写程序，对于二维的情况，当有 n 个数据样本点时，进行k-近邻概率密度估计。对表格中的类2的特征 $(x_2, x_3)^T$ ，用程序画出当 $k=2, 4, 6$ 时的概率密度估计结果。
- 3) 编写程序，对表格中的3个类别的三维特征，使用k-近邻概率密度估计方法。并且对下列点处的概率密度进行估计：

$(-0.52, 0.92, 0.92)^T$ ， $(0.24, 0.88, 4.2)^T$ ， $(-0.91, 0.72, -0.41)^T$ 。

数据读取：可以使用 python 的 pandas 库读取 xxx.xlsx 格式的文件

四、KNN 实战

【实验目的】

掌握 KNN 算法的使用。

【实验数据】

数据集存放在 exp2_4.txt 中，共有 1000 条数据

exp2_4.txt 中实验数据格式如表 2-4 所示。

表 2-4 KNN 实验数据格式样例

x1	x2	x3	label
40920	8.326976	0.953952	largeDoses
14488	7.153469	1.673904	smallDoses
26052	1.441871	0.805124	didntLike

其中，前三列是样本数据，最后一列是样本标签

用学过的 **KNN 方法** 来构建一个分类器，判断一个样本所属的类别

【具体任务】

一、数据预处理

1. 将 e2.txt 中的数据处理为模型对应数据格式

2. 是否需要对特征值进行归一化处理？目的是？

二、数据可视化分析

将预处理好的数据以散点图的形式进行可视化（使用不同颜色区分类别），通过直观感觉总结规律，感受 KNN 模型思想与人类经验的相似之处。

三、构建 KNN 模型并测试

1. 输出测试集各样本的预测标签和真实标签，并计算模型准确率。

2. 选择哪种距离更好？欧氏还是马氏？

3. 改变数据集的划分（如 70%-30%，80%-20%）以及 k 的值，观察模型准确率随之的变化情况。

注意：选择训练集与测试集的随机性

四、使用模型构建可用系统

利用构建好的 KNN 模型实现系统，输入为新的数据的三个特征，输出为预测的类别。

【实验要求】

1. 编程语言不限，推荐使用 Python 或者 MATLAB
2. KNN 模型需要自己实现，不可使用现成的第三方库
3. 实验报告中提供的代码需要有必要的注释

实验三：基于决策树算法完成鸢尾花品种预测任务

【实验说明】

本实验通过鸢尾花数据集 `iris.csv` 来实现对决策树进一步的了解。其中，Iris 鸢尾花数据集是一个经典数据集，在统计学习和机器学习领域都经常被用作示例。数据集内包含 3 类共 150 条记录，每类各 50 个数据，每条记录都有 4 项特征：花萼长度、花萼宽度、花瓣长度、花瓣宽度，可以通过这 4 个特征预测鸢尾花卉属于（`iris-setosa`，`iris-versicolour`，`iris-virginica`）三个类别中的哪一品种。Iris 数据集样例如下图所示：

表 3-1 决策树实验数据格式样例

SepalLength	SepalWidth	PetalLength	PetalWidth	Species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa

本实验将五分之四的数据集作为训练集对决策树模型进行训练；将剩余五分之一的数据集作为测试集，采用训练好的决策树模型对其进行预测。训练集与测试集的数据随机选取。本实验采用准确率（accuracy）作为模型的评估函数：预测结果正确的数量占样本总数，如公式 3-1 所示

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (3 - 1)$$

【实验要求】

1. 本实验要求输出测试集各样本的预测标签和真实标签，并计算模型准确率。另外，给出 3 个可视化预测结果（如散点图）。
2. 可以分别尝试 ID3, C4.5, cart 等决策树算法，并评判效果。
3. （选做）：对实现的决策树模型进行预剪枝与后剪枝。
4. （选做）：分别对 c4.5 和 cart 决策树进行剪枝并比较不同。

【注意事项】

编程语言不限，推荐使用 python；

决策树模型需要自己实现，**不可使用已有的第三方库**；

实验报告中提供的代码需要有必要的注释。

实验四：神经网络学习

【实验目的】

掌握 BP 神经网络的基本原理和基本的设计步骤 ；

了解 BP 算法中各参数的作用 and 意义 。

【实验数据】

CIFAR-10 数据集，数据集中包含 50000 张训练样本，10000 张测试样本，可将训练样本划分为 49000 张样本的训练集和 1000 张样本的验证集，测试集可只取 1000 张测试样本。其中每个样本都是 32×32 像素的 RGB 彩色图片，具有三个通道，每个像素点包括 RGB 三个数值，数值范围 0 ~ 255，所有照片分属 10 个不同的类别：飞机（ airplane ）、汽车（ automobile ）、鸟类（ bird ）、猫（ cat ）、鹿（ deer ）、狗（ dog ）、蛙类（ frog ）、马（ horse ）、船（ ship ）和卡车（ truck ）。

数据集展示如图 4-1 所示。

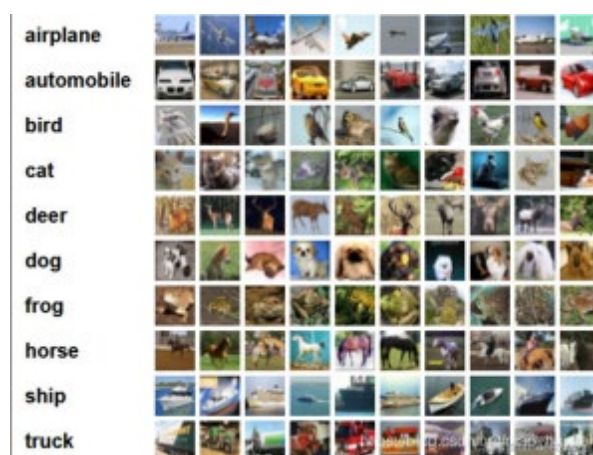


图 4-1 神经网络实验数据格式样例

【实验内容及说明】

用神经网络对给定的数据集进行分类，画出 loss 图，给出在测试集上的精确度；

不能使用 `pytorch` 等框架，也不能使用库函数，所有算法都要自己实现；

神经网络结构图如图 4-2 所示。

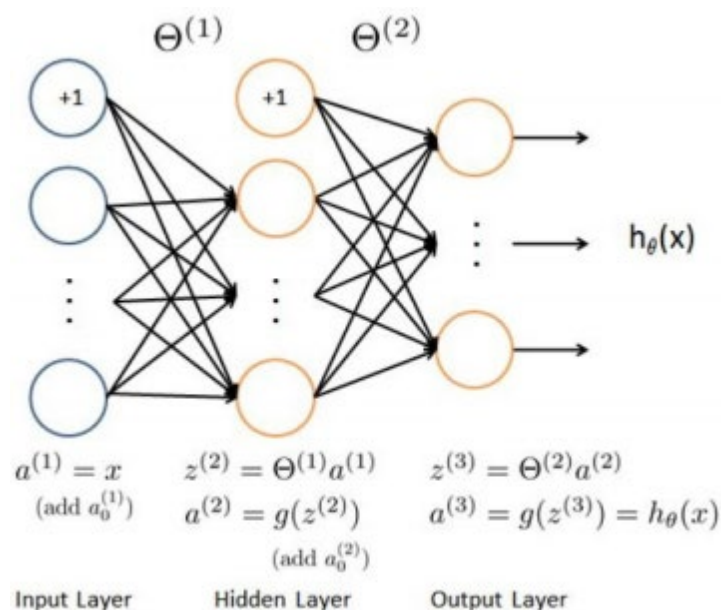


图 4-2 基础神经网络结构图

整个神经网络包括 3 层——输入层，隐藏层，输出层。输入层有 $32*32*3$ 个神经元，隐藏层有 1024 个神经元，输出层有 10 个神经元（对应 10 个类别）。训练 10 个 epoch。注意事项：三层网络模型较为简单，模型准确率不需要很高，保证正确实现神经网络的搭建和训练即可。

其他提示：

1. 建议使用批处理和矩阵运算代替 for 循环，可以提高效率。

2. RGB 图像的维度是：3（通道数） \times 32（长） \times 32（宽），可以根据自己的需求选择平均通道值还是最大最小通道值。

3. 输出层需要加入激活函数

【思维发散】

可以尝试添加卷积层，修改隐藏层神经元数，层数，学习率，正则化权重等参数，探究参数对实验结果的影响。（尝试使用 pytorch 或 tensorflow，将结果对比截图放入实验报告）

实验五：支持向量机

【实验目的】

掌握线性 SVM 的基本原理和基本设计步骤；

掌握非线性 SVM 的基本原理和基本设计步骤。

【实验数据】

Exp5_1.mat 数据格式如表 5-1 所示，其中 x_1 、 x_2 为二维自变量， y 为样本类别：

表 5-1 支持向量机实验数据格式样例

	x_1	x_2	y
1	1.9643	4.5957	1
2	2.2753	3.8589	1
3	2.9781	4.5651	1
4	2.9320	3.5519	1

Exp5_2.mat 数据格式如表 5-2 所示，其中 x_1 、 x_2 为二维自变量， y 为样本类别。

表 5-2 支持向量机实验数据格式样例

	x1	x2	y
1	0.107143	0.603070	1
2	0.093318	0.649854	1
3	0.097926	0.705409	1
4	0.155530	0.784357	1

【实验内容及说明】

使用 exp5_1.mat 数据，构造线性 SVM 对数据进行划分，给出可视化的划分边界结果。

探究不同程度的惩罚因子 c 对样本分类误差的影响。

（选做）构造使用 Gaussian kernels 的 SVM，对 exp5_2.mat 数据进行划分，给出可视化的划分边界的结果。

【注意事项】

编程语言不限，推荐使用 python；

xxx.mat 数据文件可以使用 scipy.io 进行读取处理，模型需要自己实现，不可使用已有的第三方库；实验报告中提供的代码需要有必要的注释。

实验六：集成学习

【实验目的】

用集成方法对数据集进行分类

【实验数据】

Spaceship Titanic 数据集

【实验内容及说明】

利用若干算法，针对同一样本数据训练模型，使用投票机制，少数服从多数，用多数算法给出的结果当作最终的决策依据，对 Spaceship Titanic 数据集进行分类，给出在测试集上的精确度；

除了投票法外，也可以使用其他的集成学习方法。

实验来自kaggle入门赛

<https://www.kaggle.com/competitions/spaceship-titanic>，可以参考原网站代码与预处理部分，但与公开代码不同点在于，集成学习所用的基学习器需要自己实现，不能调用现成的第三方库。

数据集的分析是一个开放性问题，可以参考网站中的预处理方式。所选算法包括但不限于课堂上学习的模型例如决策树、SVM、KNN、神经网络。

需要在网站上提交，不要求精确度很高，但要求模型由自己实现，如果有优化酌情加分。