南京理工大学紫金学院

计算机 学院

《机器学习》实验指导书

内部教材仅供紫金学院使用

主编：谢玲

南京理工大学紫金学院教务处制

前　言

《机器学习》为计算机科学与技术、人工智能、软件工程专业的专业课程。通过本门课程的学习，使学生了解机器学习的基本概念、应用背景；掌握各类机器学习算法的基本原理；培养学生学会分析研究计算机处理数据对象的特性，以便选择适当的机器学习算法和模型解决实际问题。本课程的学习过程也是进行数据分析和复杂程序设计的训练过程。通过对具体机器学习算法和应用案例的研究，进一步锻炼学生的动手能力，培养学生解决实际问题的能力。

南京理工大学紫金学院

计算机学院实验指导书

课程编号：03323011

课程名称：机器学习

课程学分：2

实验学时：8

适用专业：计算机科学与技术、人工智能、软件工程

学生实验守则

第一条　学生进入实验室从事实验、科研活动，必须严格遵守实验室的各项规章制度。

第二条　实验前必须做好预习，明确实验的目的、内容、方法和步骤，未经预习或无故迟到 15 分钟以上者，指导人员有权取消其实验资格。

第三条　保持实验室的严肃、安静，不得在实验室内大声喧哗、嬉闹，不准在实验室内进食、吸烟和乱吐乱丢杂物。

第四条　学生必须在实验指导老师规定实验开始时，方可动手实验，严禁未经许可就开始实验。学生必须以实事求是的科学态度进行实验，严格遵守操作规程，服从实验教师或实验技术人员的指导(对有特殊要求的实验，必须按要求穿戴安全防护用具后方可进行实验),如违反操作规程或不听从指导而造成仪器设备损坏等事故者，按学校有关规定进行处理。

第五条　学生应备有专用实验记录本，实验记录是原始性记录，是撰写实验报告的主要依据，内容要求真实、客观地反映实际情况，实验结果须经实验教师或实验技术人员认可。

第六条　严防事故，确保实验室的安全。发现异常情况，及时报告实验教师或实验技术人员，并采取相应的措施，减少事故造成的损失。

第七条　实验完成后，应将仪器、工具及实验场地等进行清理、归还，经实验教师或实验技术人员同意后，方可离开实验室。

第八条　独立完成实验报告，按时交给实验教师，不得抄袭或臆造。

实验报告是实验完成后的全面总结，它主要包括：实验名称、实验目的、实验原理、实验仪器设备、实验条件、实验数据、结果分析和问题讨论。实验报告一律用钢笔或圆珠笔书写，统一采用国家标准所规定的单位与符号，要求文字书写工整，不得潦草；作图规范，不得随手勾画。

目　录

实验一 线性回归

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 $(x_i,y_i)(i=1,2,3...,n)$，拟合出函数 $h(x)$有误差，即残差：$r_i=h(x_i)-y_i$，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：$H(x)=w_0+w_1x+w_2x^2+...w_nx^n$，其中 w($w_0$,$w_1$,$w_2$,...,$w_n$)为参数，最小二乘法就是要找到一组 w($w_0$,$w_1$,$w_2$,...,$w_n$)，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，

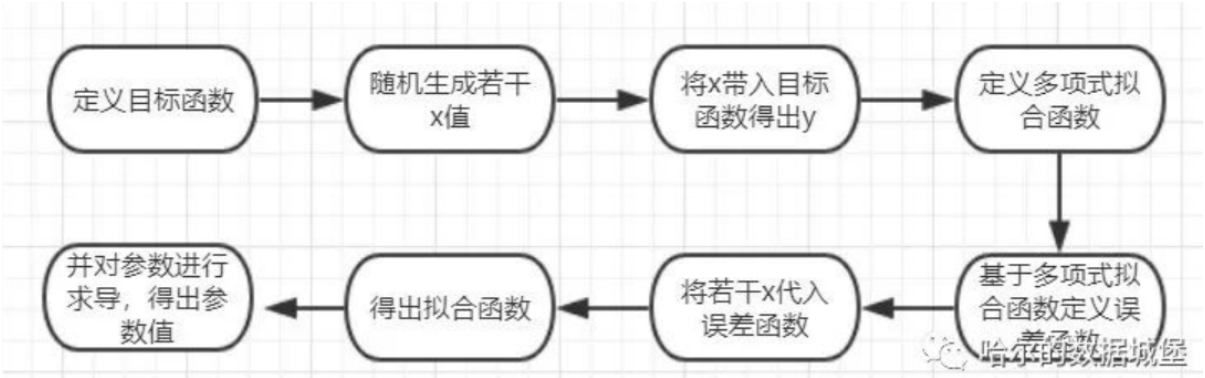
也可以使用“activate root”切回 root 环境。

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。

（3） 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$y=sin2\pi x$, 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。

（2）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xx 对其进行二类分类的线性分类模型：

$$f(x)=\text{sign}(w\cdot x+b)$$

感知机模型对应于输入空间（特征空间）中的分离超平面$w\cdot x+b=0$

感知机学习的策略是极小化损失函数：

$\min L(w,b)=-\sum y_i(w\cdot x_i+b)$，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。

$$w=w+\eta y_i x_i$$

$$b=b+\eta y_i$$

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。

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：

$$P(X=x|Y=c_k)=P\left(X^{(1)}=x^{(1)},\cdots,X^{(n)}=x^{(n)}|Y=c_k\right)$$
$$=\prod_{j=1}^{n}P\left(X^{(j)}=x^{(j)}|Y=c_k\right)$$

这是一个较强的假设。由于这一假设，模型包含的条件概率的数量大为减少，朴素贝叶斯法的学习与预测大为简化。因而朴素贝叶斯法高效，且易于实现。其缺点是分类的性能不一定很高。

朴素贝叶斯法利用贝叶斯定理与学到的联合概率模型进行分类预测：

$$P(Y|X)=\frac{P(X,Y)}{P(X)}=\frac{P(Y)P(X|Y)}{\sum_Y P(Y)P(X|Y)}$$

将输入 x 分到后验概率最大的类 y：

$$y=\arg\max_{ck}P(Y=c_k)\prod_{j=1}^{n}P\left(X_j=x^{(j)}|Y=c_k\right)$$

后验概率最大等价于 0-1 损失函数时的期望风险最小化。

三、实验内容

1．利用鸢尾花数据集构建感知机模型。

（1）IRIS 数据集也称作鸢尾花数据集，整个数据集共有 150 条数据，分为三类，每类 50 条数据，每一条数据都有四个属性：花萼长度，花萼宽度，花瓣长度，花瓣宽度，标签数据共有三种，分别是 Setosa，Versicolour，Virginica。学会导入数据集，并作数据的预处理。

（2）用 python 构建感知机模型，并进行可视化。

（3）利用 sklearn 库进行感知机模型的构建和可视化。

2．利用 K 近邻法进行鸢尾花类别的预测。

（1）K 近邻的 python 实现：建立一个类 KNN，进行模型的构建和可视化。

（2）利用 sklearn 库实现 KNN 模型的构建和可视化。

在 sklearn 库中，KNeighborsClassifier 是实现 K 近邻算法的一个类，一般都使用欧式距离进行测量。这个类的结构如下： class sklearn.neighbors. KNeighborsClassifier (n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=1, **kwargs) n_neighbors：就是选取最近的点的个数：k leaf_size：这个是构造树的大小，值一般选取默认值即可，太大会影响速度。 n_jobs ：默认值 1，选取-1 占据 CPU 比重会减小，但运行速度也会变慢，所有的 core 都会运行。 algorithm: 近邻算法，可选{'auto', 'ball_tree', 'kd_tree', 'brute'}。

（3）构造平衡 Kd 树，并实现书中例题 3.2。

3. 在鸢尾花数据集上用高斯朴素贝叶斯实现分类，并进行预测。直接使用 sklearn 方法。

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

实验三 决策树与随机森林

一、实验目的

1. 掌握决策树原理及实现。

2. 掌握 ID3 算法实现（重点）。

3. 掌握 sklearn 库中决策树的实现方法（CART 树算法）。

4. 了解随机森林原理及实现（加分项）。

二、实验原理及说明

1. 决策树

决策树（decision tree）：是一种基本的分类与回归方法，此处主要讨论分类的决策树。在分类问题中，表示基于特征对实例进行分类的过程，可以认为是 if-then 的集合，也可以认为是定义在特征空间与类空间上的条件概率分布。

决策树通常有三个步骤：特征选择、决策树的生成、决策树的修剪。

用决策树分类：从根节点开始，对实例的某一特征进行测试，根据测试结果将实例分配到其子节点，此时每个子节点对应着该特征的一个取值，如此递归的对实例进行测试并分配，直到到达叶节点，最后将实例分到叶节点的类中。

决策树的构造

决策树学习的算法通常是一个递归地选择最优特征，并根据该特征对训练数据进行分割，使得各个子数据集有一个最好的分类的过程。这一过程对应着对特征空间的划分，也对应着决策树的构建。

1） 开始：构建根节点，将所有训练数据都放在根节点，选择一个最优特征，按着这一特征将训练数据集分割成子集，使得各个子集有一个在当前条件下最好的分类。

2） 如果这些子集已经能够被基本正确分类，那么构建叶节点，并将这些子集分到所对应的叶节点去。

3）如果还有子集不能够被正确的分类，那么就对这些子集选择新的最优特征，继续对其进行分割，构建相应的节点，如果递归进行，直至所有训练数据子集被基本正确的分类，或者没有合适的特征为止。

4）每个子集都被分到叶节点上，即都有了明确的类，这样就生成了一颗决策树。

2. 随机森林

尽管有剪枝等方法，一棵树的生成肯定还是不如多棵树，因此就有了随机森林，解决决策树泛化能力弱的缺点。

而同一批数据，用同样的算法只能产生一棵树，这时 Bagging 策略可以帮助我们产生不同的数据集。Bagging 策略来源于 bootstrap aggregation：从样本集（假设样本集 N 个数据点）中重采样选出 Nb 个样本（有放回的采样，样本数据点个数仍然不变为 N），在所有样本上，对这 n 个样本建立分类器（ID3\C4.5\CART\SVM\LOGISTIC），重复以上两步 m 次，获得 m 个分类器，最后根据这 m 个分类器的投票结果，决定数据属于哪一类。

随机森林在 bagging 的基础上更进一步：

1. 样本的随机：从样本集中用 Bootstrap 随机选取 n 个样本

2. 特征的随机：从所有属性中随机选取 K 个属性，选择最佳分割属性作为节点建立 CART 决策树（泛化的理解，这里面也可以是其他类型的分类器，比如 SVM、Logistics）

3. 重复以上两步 m 次，即建立了 m 棵 CART 决策树

4. 这 m 个 CART 形成随机森林，通过投票表决结果，决定数据属于哪一类（投票机制有一票否决制、少数服从多数、加权多数）

三、实验内容

1．ID3 算法实现。

（1）编写代码计算信息增益，数据集为教材 71 页表 5.1。

（2）用 python 编写 ID3 算法。

2．使用 sklearn 库，对鸢尾花数据建立决策树，并进行可视化。

（1）scikit-learn 决策树算法类库内部实现是使用了调优过的 CART 树算法，既可以做分类，又可以做回归。分类决策树的类对应的是 DecisionTreeClassifier，而回归决策树的类对应的是 DecisionTreeRegressor。两者的参数定义几乎完全相同，但是意义不全相同。在 https://graphviz.org/download/ 上下载 graphviz,安装时注意勾选加环境变量，

在 anaconda prompt 中 pip install graphviz ，再 pip install pydotplus。

（2）使用 DecisionTreeClassifier()构建决策树并进行可视化。

3. 在鸢尾花数据集上使用随机森林进行分类，有条件的可进行可视化。

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

实验四 支持向量机

一、实验目的

1. 掌握线性可分支持向量机的原理及实现。

2. 掌握 sklearn 库中支持向量机的实现方法。

二、实验原理及说明

支持向量机最简单的情况是线性可分支持向量机，或硬间隔支持向量机。构建它的条件是训练数据线性可分。其学习策略是最大间隔法。可以表示为凸二次规划问题，其原始最优化问题为

$$\min_{w,b} \frac{1}{2}\|w\|^2$$
$$s.t. \quad y_i(w \cdot x_i + b) - 1 \geqslant 0, \quad i = 1, 2, \cdots, N$$

求得最优化问题的解为 w*，b^*，得到线性可分支持向量机，分离超平面是

$$w^* \cdot x + b^* = 0$$

分类决策函数是

$$f(x) = \text{sign}\left(w^* \cdot x + b^*\right)$$

最大间隔法中，函数间隔与几何间隔是重要的概念。

线性可分支持向量机的最优解存在且唯一。位于间隔边界上的实例点为支持向量。最优分离超平面由支持向量完全决定。 二次规划问题的对偶问题是

$$\min \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(x_i \cdot x_j\right) - \sum_{i=1}^{N}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0$$
$$\alpha_i \geqslant 0, \quad i = 1, 2, \cdots, N$$

通常，通过求解对偶问题学习线性可分支持向量机，即首先求解对偶问题的最优值 a^*，然后求最优值w^*和b^*，得出分离超平面和分类决策函数。

三、实验内容

1．使用 sklearn 库，使用鸢尾花数据集，对鸢尾花进行分类。

（1）数据准备，载入数据集，并进行数据集分割；

（2）模型搭建；

（3）模型训练；

（4）模型评估；

（5）可视化。

2. 完成习题 7.2，编程实现。

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