py文件

__init__

参数

penalty[默认为l2]：正则化范数

gamma[默认为0]：正则化程度

fit_intercept[默认为True]：是否匹配斜率

功能

新建类并记录学习器属性。

返回

无。

sigmoid

参数

x：需要计算Sigmoid函数值的向量

功能

计算学习器中的w[此处可能包含扩展的截距]与x产生的Sigmoid函数结果。

返回

计算出的值。

count_loss

参数

X：训练集中的样本

y：样本对应的结果

功能

按照相应正则化方式计算学习器中的w对训练集产生的损失。

返回

计算出的损失。

count_grad_loss

参数

X：训练集中的样本

y：样本对应的结果

功能

按照相应正则化方式计算训练集上产生的损失对w的梯度。

返回

计算出的损失梯度向量。

fit

参数

X：训练集中的样本

y：样本对应的结果

lr[默认为0.01]：学习率

tol[默认为1e-4]：允许的最大梯度

max_iter[默认为1e3]：最大迭代次数

功能

根据fit_intercept确定是否增广模型，并利用训练集学习。

返回

每次迭代计算出的损失形成的向量。

predict

参数

X: 测试集中的样本。

功能

生成预测结果。

返回

对测试集生成的预测结果。

ipynb文件

Data Cleaning

清除了离散属性缺失的样本，对连续属性以均值替代。

Encode

对离散属性进行最小0最大1的编码。

Data Process

将所有属性归一化并随机打乱，区分出7:3的训练集、测试集。

Train

以给定参数进行训练。

Test

画出训练时的损失曲线，预测并对比正确率。

Update 1 有关错标率的问题

1. 错标率和SVM跑出来的结果不一定有严格的大小关系，因为错标可能会更改实际的超平面信息。
2. 推荐样本数在10000左右，维度在20以上，通过对比来展现模型的效果和能力，可以根据自己cpu的性能调整。
3. 如果实在认为错标率太低，可以将`generate_data`中`if np.abs(pred_n[i]) < 1 and mislabel_value[i] > 0.9 + 0.1 * np.abs(pred_n[i]):`更改为`if np.abs(pred_n[i]) < 1 and mislabel_value[i] > 0.8 + 0.2 * np.abs(pred_n[i]):`或更低，只要在报告中给出你的错标率即可。
4. 如果和sklearn库中svm函数的结果对比相差不大，则你的代码没有大问题。

Machine Learning Lab2

SVM

By Yanwu Gu 2022.10.12

1. Theory of Supported Vector Machine

You can refer to the ppt or the Chap 6 of the textbook.

2. Data

In order to simplify the lab, we give the function `generate_data(dim, num)` for you to freely generate the data. The data was linearly separable, but added some mistakes intentionally. Features, labels and rate of mislabel will be given by the function respectively.

You do not need to modify the function`generate_data(dim, num)`.

3. Tasks, Tips and Requirements

3.1 Tasks

You are required to complete the class `SVM1` and `SVM2` using different methods to find the solution of the supported vector machine. More specificly, since the key of solving SVM is to solve the quadratic programming problem (6.6) in your textbook, you just need to use **two** methods to solve (6.6). The remaining part like predict can be the same.

After finishing the SVM class, you need to test the efficiency of your code. The comparison must include

1. The accuracy,
2. The time of culculation (trainning),

If possible, you can use `sklearn` to compare with your code, feel free to be beaten by it.

3.2 Tips

There are some tips for the lab:

1. We do not recommend you to use existing function to solve the **quadratic programming** problem directly, which will be penalised. Of course, if you cannot complete two methods from scratch, you can use library function.
2. We recommend you to use proper dims to make sure your result reliable, and different dims or numbers of examples will let your report rich in content. But do not let it verbose.

3. Since our data is based on linear kernel, you do not need to try other kernels. But you can try soft margin or regularization to improve the ability of your model. Remember it’s not the key point of this lab.
4. Remember to add your **mislabel rate**, which is generate by the function `generate_data` for us.

3.3 Requirements

- **Do not** use sklearn or other machine learning library, you are only permitted with numpy, pandas, matplotlib, and Standard Library, you are required to **write this project from scratch.**
- You are allowed to discuss with other students, but you are **not allowed to plagiarize the code**, we will use automatic system to determine the similarity of your programs, once detected, both of you will get **zero** mark for this project.

4. Submission

- Report
    - The method you use, and briefly talk about its principle
    - The result of your methods
    - The comparison of your methods
- Submit a .zip file with following contents

  –main.ipynb

  –Report.pdf
- Please name your file as `LAB2_PBXXXXXXXX.zip`, **for wrongly named file, we will not count the mark**
- Sent an email to ml_2022_fall@163.com with your zip file before deadline
- **Deadline: 2022.10.30 23:59:59**
- For late submission, please refer to this

机器学习 Lab2

支持向量机

By Yanwu Gu 2022.10.12

1. 支持向量机的理论

你可以参考演示文档或者书本第六章的相关内容。

2. Data

为了简化实验，我们为你给出了函数 `generate_data(dim, num)` 去自由地生成数据. 这个数据是线性可分的，但是故意在标签值加上了一些错误。特征、标签以及错标率会由函数依次给出。

你不需要去更改`generate_data(dim, num)`.

3. 任务，提示及要求

3.1 任务

你需要去完成类 `SVM1` 和 `SVM2` ，并且使用不同的算法去寻找支持向量机的解。 更具体地说，因为解决支持向量机的关键在于解决书本上的二次规划问题（6.6），你只需要使用两种不同的方法去解决（6.6）。剩下的部分，比如预测，内容可以相同。

在完成了类方法的部分之后，你需要测试你代码的效率。比较应当包含以下内容：

1. 正确率，
2. 计算（训练）的时间消耗。

如果可能的话，你可以使用 `sklearn` 与你的代码比较。如果比不过它，也是没事的。

3.2 提示

这里有一些实验的提示：

1. 我们不推荐你使用已有的库函数去**直接**解决二次规划问题，这是会被扣除一部分分数的。当然，如果你无法使用两种方法去解决，你也可以使用库函数。
2. 我们推荐你使用合适的维度去训练、测试，这会使你的结果更加可靠。同时，不同的维度和样本数也会使你的报告内容更丰富。但是不要让他过于冗杂。
3. 因为我们的数据是基于线性核生成的，你不需要尝试其他的核函数。但是你可以使用软间隔或者正则化等方法来提升你模型的能力。切记，这不是本实验的核心内容。
4. 记得添加你的**错标率**，它会由函数 `generate_data` 生成。

3.3 要求

- 禁止使用 `sklearn` 或者其他的机器学习库，你只被允许使用`numpy`, `pandas`, `matplotlib`, 和 Standard Library, 你需要从头开始编写这个项目。
- 你可以和其他同学讨论，但是你不可以剽窃代码，我们会用自动系统来确定你的程序的相似性，一旦被发现，你们两个都会得到这个项目的零分。

4. 提交

- 报告
    - 你使用的理论，简要讨论它的原理，
    - 你方法的结果，
    - 你方法之间的比较。
- 提交 .zip 文件，包含以下内容

  –main.ipynb

  –Report.pdf
- 请命名你的文件为 `LAB2_PBXXXXXXXX.zip`, **对于错误命名的文件，我们将不会计算分数**
- 请发邮件至 ml_2022_fall@163.com 附带您的文件，在截止日期之前
- **截止日期: 2022.10.30 23:59:59**
- 对于迟交的作业，请点击 [这里](

LAB3

XGBoost

By Yanwu Gu 2022.10.26

1. 实验原理

由于本次实验任务量大，故给大家提供了较为完整的理论推导，仔细阅读本节能够有助于你完成实验。

XGBoost

XGBoost 是由多个基模型组成的一个加法模型，假设第 k 个基本模型是 $f_k(x)$, 那么前 t 个模型组成的模型的输出为

$$y_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = y_i^{(t-1)} + f_t(x_i)$$

其中 x_i 为第表示第 i 个训练样本，y_i 表示第 i 个样本的真实标签; $y_i^{(t)}$ 表示前 t 个模型对第 i 个样本的标签最终预测值。

在学习第 t 个基模型时，XGBoost 要优化的目标函数为:

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^{n} loss(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} penalty(f_k) \\ &= \sum_{i=1}^{n} loss(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{k=1}^{t} penalty(f_k) \\ &= \sum_{i=1}^{n} loss(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + penalty(f_t) + constant \end{aligned}$$

其中 n 表示训练样本的数量, $penalty(f_k)$ 表示对第 k 个模型的复杂度的惩罚项, $loss(y_i, \hat{y}_i^{(t)})$ 表示损失函数,

例如二分类问题的

$$loss(y_i, \hat{y}_i^{(t)}) = -y_i \cdot \log p(\hat{y}_i^{(t)} = 1|x_i) - (1 - y_i) \log(1 - p(\hat{y}_i^{(t)} = 1|x_i))$$

回归问题

$$loss(y_i, \hat{y}_i^{(t)}) = (y_i - \hat{y}_i^{(t)})^2$$

将 $loss(y_i, y_i^{(t-1)} + f_t(x_i))$ 在 $y_i^{(t-1)}$ 处泰勒展开可得

$$loss(y_i, y_i^{(t-1)} + f_t(x_i)) \approx loss(y_i, y_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)$$

其中 $g_i = \frac{\partial\, loss(y_i, y_i^{(t-1)})}{\partial\, y_i^{(t-1)}}$, $h_i = \frac{\partial^2 loss(y_i, y_i^{(t-1)})}{\partial\, (y_i^{(t-1)})^2}$，即 g_i 为一阶导数，h_i 为二阶导数。

此时的优化目标变为

$$Obj^{(t)} = \sum_{i=1}^{n}[loss(y_i, y_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + penalty(f_t) + constant$$

去掉常数项 $loss(y_i, y_i^{(t-1)})$ (学习第 t 个模型时候， $loss(y_i, y_i^{(t-1)})$ 也是一个固定值) 和 constant，可得目标函数为

$$Obj^{(t)} = \sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + penalty(f_t)$$

决策树（回归树）

本实验中，我们以决策树（回归树）为基，因此还需要写出决策树的算法。

假设决策树有 T 个叶子节点，每个叶子节点对应有一个权重。决策树模型就是将输入 x_i 映射到某个叶子节点，决策树模型的输出就是这个叶子节点的权重，即 $f(x_i) = w_{q(x_i)}$ ， w 是一个要学的 T 维的向量其中 $q(x_i)$ 表示把输入 x_i 映射到的叶子节点的索引。例如：$q(x_i) = 3$，那么模型输出第三个叶子节点的权重，即 $f(x_i) = w_3$。

我们对于某一棵决策树，他的惩罚为

$$penalty(f) = \gamma \cdot T + \frac{1}{2} \lambda \cdot \|w\|^2$$

其中 γ, λ 为我们可调整的超参数，T 为叶子数，w 为权重向量. 由于显示问题，$\|w\|$ 实际上为 w 的范数，且 $\|w\|^2 = \sum_{i=1}^{dim} w_i^2$

我们将分配到第 j 个叶子节点的样本用 I_j 表示，即 $I_j = \{i|q(x_i) = j\}(1 \le j \le T)$。

综上，我们在树结构确定（你可以自行确定）时，可以进行如下优化：

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + penalty(f_t) \\ &= \sum_{i=1}^{n}[g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma \cdot T + \frac{1}{2} \lambda \cdot \|w\|^2 \\ &= \sum_{j=1}^{T}[(\sum_{i \in I_j} g_i) \cdot w_j + \frac{1}{2} \cdot (\sum_{i \in I_h} h_i + \lambda) \cdot w_j^2] + \gamma \cdot T \end{aligned}$$

简单起见，我们简记 $G_j = \sum_{i \in I_j} g_i, H_j = \sum_{i \in I_j} h_i$

$$Obj^{(t)} = \sum_{j=1}^{T}[G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2] + \gamma T$$

在上述推导之后，你应该可以推导出最优的权重（对 w_j 优化），请在你的报告中写出这个权重的表达式，同时需要写出这棵决策树的得分 Obj.

构造过程

对于每一棵决策树，即每一个基的训练，我们可以按照以下步骤划分结点

1. 从根节点开始递归划分，初始情况下，所有的训练样本 x_i 都分配给根节点。
2. 根据划分前后的收益划分结点，收益为

$$Gain = Obj_P - Obj_L - Obj_R$$

其中 Obj_P 为父结点的得分，Obj_L, Obj_R 为左右孩子的得分.

3. 选择最大增益进行划分

选择最大增益的过程如下：

1. 选出所有可以用来划分的特征集合 $\mathcal{F}$；
2. For feature in $\mathcal{F}$:
3. 将节点分配到的样本的特征 feature 提取出来并升序排列，记作 sorted_f_value_list；
4. For f_value in sorted_f_value_list ：
5. 在特征 feature 上按照 f_value 为临界点将样本划分为左右两个集合；
6. 计算划分后的增益；
7. 返回最大的增益所对应的 feature 和 f_value。

停止策略

对于如何决定一个节点是否还需要继续划分，我们提供下列策略，你可以选择一个或多个，或自行设定合理的策略

- 划分后增益小于某个阈值则停止划分；
- 划分后树的深度大于某个阈值停止划分；
- 该节点分配到的样本数目小于某个阈值停止分化。

对于整个算法如何终止，我们提供下列策略，你可以选择一个或多个，或自行设定合理的策略

- 学习 M 个颗决策树后停下来；
- 当在验证集上的均方误差小于某个阈值时停下来；
- 当验证集出现过拟合时停下来。

评价指标

你可以在实验中以下列指标来验证你的算法效果和不同参数对于结果的影响

- $RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_{test}^{(i)} - \hat{y}_{test}^{(i)})^2}$，越小越好，
- $R^2 = 1 - \frac{\sum_{i=1}^{m}(y_{test}^{(i)} - \hat{y}_{test}^{(i)})^2}{\sum_{i=1}^{m}(\bar{y}_{test} - \hat{y}_{test}^{(i)})^2} = 1 - \frac{MSE(\hat{y}_{test}, y_{test})}{Var(y_{test})}$，越大越好
- 运行时间

仍然，这些标准不会作为评分的标准

2. 实验数据

在 `train.data` 文件中，有 7154 条 41 维的数据，其中前 40 列为 feature，最后一列为 label.

3. 任务及要求

3.1 任务

1. 完成决策树（回归树）的算法
2. 完成 XGBoost 的算法

3. 书写你的报告

报告应当含有以下内容和其他你觉得必要的内容

1. 有关文档中提出问题的解答
2. 有关于你**两个**停止策略的选择，你关于树的选择，你关于超参数的选择
3. 实验结果的展示（最佳的模型）
4. 不同参数的比较
5. 训练过程 loss 的可视化

一些提示

1. 理论上 XGBoost 应该是一个框架，不应与基的选择有关系，但本实验中不做要求，但应将决策树算法和 XGBoost 算法做明显的区分
2. 本实验可以不做数据预处理
3. 你应当选择合适的数据结构来存储你关于决策树的参数
4. 给出一个回归树的结构示例，仅作参考

```
class RegressionTree(object):
    def __init__(self,): # 初始化回归树
    def _get_best_split(self,): # 获得最佳feature和split
    def _get_split_score(self,): # 获取某一划分的目标函数值
    def _choose_split_point(self,): # 获取最佳的划分点
    def fit(self,):# 训练一棵回归树
    def _predict(self,): # 预测一个样本
    def predict(self,): # 预测多条样本
```

5. 由于 XGBoost 效果很好，对于树的各种属性的选择，在比较经济的情况下也能得到不错的结果，节约时间和内存的开销

3.2 要求

- 禁止使用 `sklearn` 或者其他的机器学习库，你只被允许使用 `numpy`，`pandas`，`matplotlib`，和 Standard Library，你需要从头开始编写这个项目。
- 你可以和其他同学讨论，但是你不可以剽窃代码，我们会用自动系统来确定你的程序的相似性，一旦被发现，你们两个都会得到这个项目的零分。

4. 提交

- 报告推荐格式
  i. 实验目的（可选）
  ii. 实验原理（若不重要可以简要说明）
  iii. 实验步骤（从读取数据、模型训练、使用xx的参数，xx的模型，得到了多少组的结果，总之就是你在每块代码做了什么事情）
  iv. 实验结果（对输出进行总结、比较、可视化）
  v. 实验分析（分析结果出现的原因、分析原因）
- 提交 .zip 文件，包含以下内容（请直接对这两个文件打包）

  –main.ipynb

–Report.pdf

- 请命名你的文件为 `LAB3_PBXXXXXXXX_中文名.zip`，例如 `LAB3_PB19061297_顾言午`，**对于错误命名的文件，我们将不会计算分数，请注意，这次实验开始我们会严格这方面的规定**
- 请发邮件至 ml_2022_fall@163.com 附带您的文件，在截止日期之前
- **截止日期**: 2022.11.20 23:59:59

LAB4

Density Peak Clustering

By HanLei 2022.11.21

1. 实验原理

聚类相关知识详情请回顾课件第九章《聚类》，本次聚类实验主要实现的是《**Clustering by fast search and find of density peaks**》一文中的算法（以下简称**DPC**）

- By Alex Rodriguez and Alessandro Laio
- Published on SCIENCE, 2014
- https://sites.psu.edu/mcnl/files/2017/03/9-2dhti48.pdf
- 每位同学务必仔细阅读原论文

算法思想

集成了 k-means 和 DBSCAN 两种算法的思想

- 聚类中心周围密度较低，中心密度较高
- 聚类中心与其它密度更高的点之间通常都距离较远

算法流程

1. Hyperparameter: a distance threshold d_c
2. For each data point i, compute two quantities:
   - Local density: $\rho_i = \sum_j \chi(d_{ij} - d_c)$,where $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise
   - Distance from points of higher density: $\delta_i = \min_{j:\rho_j > \rho_i} d_{ij}$
     - For the point with highest density, take $\delta_i = \max_j d_{ij}$
3. Identify the cluster centers and out-of-distribution (OOD) points

- Cluster centers: with both high ρ_i and δ_i
- OOD points: with high δ_i but low ρ_i
- Draw a decision graph, and make decisions manually

2. 实验数据

本次实验采用 3 个 2D 数据集（方便可视化）

- Datasets/D31.txt
- Datasets/R15.txt
- Datasets/Aggregation.txt

数据格式

- 每个文件都是普通的 txt 文件，包含一个数据集
- 每个文件中，每一行表示一条数据样例，以空格分隔

注意事项

- 允许对不同的数据集设置不同的超参数

3. 任务及要求

3.1 任务

3.1.1 实验简介

本次实验的总体流程是完成 DPC 算法的代码实现，并在给定数据集上进行可视化实验。具体来说，同学们需要实现以下步骤

1. 读取数据集，（如有必要）对数据进行预处理
2. 实现 DPC 算法，计算数据点的 δ_i 和 ρ_i
3. 画出决策图，选择样本中心和异常点
4. 确定分簇结果，计算评价指标，画出可视化图

助教除大致浏览代码外，以以下输出为评价标准：

- 可视化的决策图
- 可视化的聚类结果图
- 计算出的评价指标值 （DBI）
- 输出只要在**合理范围**内即可，不作严格要求

实验结果需要算法代码和实验报告

- 助教将通过可视化结果和代码来确定算法实现的正确性
- 助教将阅读实验报告来检验同学对实验和算法的理解

3.1.2 评价指标

- 本次实验采用 Davis-Bouldin Index (DBI) 作为评价指标
- 建议统一调用 sklearn.metrics. davies_bouldin_score 进行计算

3.1.3 数据可视化

- 本次实验需要画两个二维散点图：决策图和聚类结果图
- 可视化库推荐 pyplot (也可自行选择别的工具，此处只做教程)
- 代码片段演示

```
# 产生测试数据
import matplotlib.pyplot as plt
import numpy as np

x1 = np.arange(1,10)
x2 = x1**2
fig = plt.figure()
ax1 = fig.add_subplot(111)
# 设置每个样本点的颜色（用于聚类结果展示）
colors = ['r','y','g','b','r','y','g','b','r']
# 设置标题
ax1.set_title('Scatter Plot')
# 设置X轴标签
plt.xlabel('X')
# 设置Y轴标签
plt.ylabel('Y')
# 画散点图
ax1.scatter(x1, x2, c=colors, marker='o')
# 显示所画的图
plt.show()
```

3.2 要求

- 禁止使用 `sklearn` 或者其他的机器学习库，你只被允许使用 `numpy`， `pandas`， `matplotlib`，和 Standard Library，你需要从头开始编写这个项目。
- 你可以和其他同学讨论，但是你不可以剽窃代码，我们会用自动系统来确定你的程序的相似性，一旦被发现，你们两个都会得到这个项目的零分。

4. 提交

- 实验报告可参考关于LAB2的一些反馈
- 报告推荐格式
  i. 实验目的（可选）
  ii. 实验原理（若不重要可以简要说明）
  iii. 实验步骤（从读取数据、模型训练、使用xx的参数，xx的模型，得到了多少组的结果，总之就是你在每块代码做了什么事情）
  iv. 实验结果（对输出进行总结、比较、可视化）
  v. 实验分析（分析结果出现的原因、分析原因）
- 提交 .zip 文件，包含以下内容（请直接对这两个文件打包）

  –main.ipynb

  –Report.pdf
- 请命名你的文件为 `LAB3_PBXXXXXXXX_中文名.zip`，**对于错误命名的文件，我们将不会计算分数**
- 请发邮件至 ml_2022_fall@163.com 附带您的文件，在截止日期之前
- **截止日期**: 2022.12.11 23:59:59
- 对于迟交的作业，请参考 这里

机器学习 LAB5

Comprehensive Experiment

By Benwei Wu 2022.12.24

1. 实验原理

可参考本课程涉及的所有分类模型以及原理

2. Data

在 `train_feature.csv`文件中，有`10000`条`120`维特征数据 ，

在 `train_label.csv`文件中 ，有 `10000`条`1`维标签数据

在 `test_feature.csv` 文件中，有 `3000` 条 `120` 维特征数据

3. 任务、提示以及要求

3.1 任务

- **数据预处理**。需要注意，提供数据包含大量冗余随机特征、outlier数据以及Null数据，你需要综合运用所学的知识进行数据降维、降噪、补缺、特征提取、编码以及必要的其他数据预处理工作。
- **数据划分**。你需要将所提供的 `train` 数据集按照所学的方法拆分成训练集以及测试集。
- **模型训练**。你需要分别使用本课程所学习的线性回归模型、决策树模型、神经网络模型、支持向量机以及XGBoost等分类模型来完成标签预测任务。
- **模型验证**。你需要将 `test_feature.csv` 的数据输入到一个你认为性能最佳的模型中，然后仿照 `train_label.csv` 的文件格式生成对应标签数据文件，命名为 `test_label.csv` ，并将它包含在你所提交的压缩包中。
- **实验分析**。你需要仔细撰写实验报告以及相关分析。

3.2 评价指标

- 你可以在实验中使用下列指标来验证你的算法效果以及不同参数对于结果的影响
  - $E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(x_i)\neq y_i)$，分类错误率，越小越好
  - $Acc(f;D)=1-E(f;D)$，分类精度，越大越好
  - 运行时间
- 你也可以选取你认为合理的评价指标，评价指标的选取不会影响最终的实验分数。
- 但需要注意，在**模型验证**环节，我们将使用$Acc(f;D)$指标对你的生成结果进行判分。当然，这个分数只是最终实验分数的组成，并不唯一，因此我们鼓励你使用更多指标来综合评价模型性能。

3.2 提示

- 原则上你可以直接使用之前在 `lab1-lab4` 中你自己独立编写的程序，但如果部分算法（如神经网络模型）之前你未写过，你可以调用一些已有的算法库。算法复现并不是我们的考察重点，我们更加希望看到的是你整体实验的完整性以及严谨性，是否包括特征预处理、模型评估、调参、模型选择、假设检验等一整套完备流程，我们也将主要基于此对你的实验报告进行判分。

- 你可以调用sklearn库中的MLPClassifier来实现神经网络模型，例如

```
from sklearn.neural_network import MLPClassifier
```

  但我们希望你在调库前能认真阅读官方文档（https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier），理解不同参数的实际含义将有利于你的模型调参。

- 除了本课程所包含的基础模型外，我们鼓励你增加课程外所学习的最新模型。当然你仅需完成基本任务，即可拿到相应的实验分数。
- 不要简单地堆砌实验数据以及程序代码，我们希望看到的是严谨的实验分析。

3.3 要求

- 你可以和其他同学讨论，但是你不可以剽窃代码，我们会用自动系统来确定你的程序的相似性，一旦被发现，你们两个都会得到这个项目的零分。
- 请务必确保 `test_label.csv` 文件数据格式与 `train_label.csv` 相同，否则可能无法判分。

4. 提交

- 报告推荐格式

  i. 实验目的（可选）
  ii. 实验原理（若不重要可以简要说明）
  iii. 实验步骤（从读取数据、模型训练、使用xx的参数，xx的模型，得到了多少组的结果，总之就是你在每块代码做了什么事情）
  iv. 实验结果（对输出进行总结、比较、可视化）
  v. 实验分析（分析结果出现的原因、分析原因）

- 提交 .zip 文件，包含以下内容（请直接对这三个文件打包）

  –main.ipynb

  –test_label.csv

  –Report.pdf

- 请命名你的文件为 `LAB5_PBXXXXXXXX_中文名.zip`，例如 `LAB5_PB19061297_XXX`，**对于错误命名的文件，我们将不会计算分数，请注意，我们会严格执行这方面的规定**
- 请发邮件至 ml_2022_fall@163.com 附带您的文件，在截止日期之前
- **截止日期**: 2023.01.20 23:59:59

LAB5 讲解

实验目的

考察完整进行实验的能力。此处“实验”的含义是指整个任务的过程，不单单指模型的构建。

一般来说，我们整个分类任务可以分为以下部分，供你参考：

1. 获取数据集，对数据进行分析

> 数据集我们已经给出，对于数据的部分特征我们已经给出，你也可以针对你发现的其他特点进行说明和处理。

2. 对数据进行处理，形成测试集和测试集

> 针对你发现的问题，选择合适的处理方式，推荐使用搜索引擎和课程内容 Chap. 2 & 11。

3. 对于任务，选择合适的模型

> 每个模型实际上包含多种对于不同任务的处理方式，关键的是算法的核心。
>
> 例如，在决策树中，实际上存在分类树和回归树两种，你应该选择更合适的方法

4. 利用训练集训练模型，调整超参数以达到在测试集上达到更好的效果，保存模型

> 请注意各个模型有哪些超参数是可以调整的，体现你是通过对比后选出更好的模型。
>
> 注意，调整参数不应该是一个做样子的过程，需要达到“充分”调参。
>
> 如果数量太多可以以图的形式表示。
>
> 尽管我们不需要同学提交模型，但是希望大家学会、养成保存模型的良好习惯。

5. 注意你的评价指标是否合适，同时进行假设检验。

> 参考 Chap 2.

6. 可视化你的结果
7. 挑选一个最好的模型，用其对 test_label 进行预测，提交你的 pred.

> 只需要提交你认为表现最好的模型的预测结果。

1. 对于数据的处理、数据集的划分、模型的验证等内容可以使用已有的库函数，调用的函数一定是要有具体的、有针对性的功能，而不是一个“万能函数”
2. 实验结果正确命名后，与实验报告、代码压缩后提交
3. 调参可以手动调，也可以自动调参数
4. 由于 lab3 我们写的是回归树和回归树为基分类器的 xgboost，我们对于决策树和 xgbclassifier 也可以调包。神经网络也可以调包。具体reference 为：
    i. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier
    ii. https://xgboost.readthedocs.io/en/stable/python/python_api.html#xgboost.XGBClassifier
    iii. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier

主题: 顾言午(YanwuGu)的快速会议日期: 2022-12-24 19:49:37
录制文件：https://meeting.tencent.com/v2/cloud-record/share?id=7fca8550-295d-446a-bf1c-0efc3e52c00a&from=3