项目简介与要求

陈欣鸿

2017.12.7

Rank

- 标签是1/0，提交上来是1.0/0.0，1.00000…/0.00000…
- 测试集是 N 行，提交上来的行数不对
- 文件名不是下划线而是中划线
- 文件名学号位数不对，无法确认是谁的rank
- 压缩包格式不对，无法解压
- 内容包含有非结果字符，比如多了一行“正确率”字样，多了一列文本序号等等

Project

- 三个任务
  - 二元分类，以 **F1** 作为评测指标
  - 多元分类，以 **Average Accuracy** 作为评测指标
  - 回归，以 **RMSE（MSE开方）** 作为评测指标
- 竞赛制
  - 组队后，每个队伍每天可提交自己的结果到 ftp，TA 会跑 rank，然后把 rank 的情况发给大家，如果提交的结果是空的则分数为 0，排名越高的分数越高
- 占实验期末总评 **50%**

数据集介绍

- 每个任务都会提供一个数据集
- **友情提示，每个数据集都经过了随机处理以及数据处理，不用费心在网上找原数据集，就算找到了也不一样。**
- 二元分类：神秘数据集

训练集**有效**行数	测试集行数	**有效**属性个数	输出
48000	**12000**	**13**	**1 或 0**

数据集介绍

- 多元分类：
- 从某网站上收集的文本数据集，分了三个类别：LOW,MID,HIG
- 每行一开始是标签，用 \t\t 跟后面隔开
- 后面跟着文本，一行中的文本可能有多个句子，用〈sssss〉隔开

训练集**有效**行数	测试集行数	输出
62522	**8671**	**类别标签**

- 提示：回忆之前的实验中关于文本处理的方法

数据集介绍

- 回归：与 NN 提供的数据集属于同一个数据集，关于自行车数量预测的一个任务。

训练集**有效**行数	测试集行数	**有效**属性个数	输出
16637	**742**	**7**	**自行车数量**

- 提示：预处理数据，是否可以抛弃一些属性

算法

- 没有规定使用某种算法
- 学过的算法：KNN, NB, PLA, DT, LR, NN
- 全新的算法：SVM, SVR, …
- 鼓励大家尝试新算法
- **所有算法，都必须是自己实现的，不可以调用现成库**
- 比如想在 NN 里面用 PLA，PLA部分也要自己实现
- 选择你认为效果最好的方法，在 pre 的时候展示
- 在 Project 报告中，将你使用的方法展示出来

Presentation

- 17-19周进行，展示顺序如何确定的问题等下讲
- pre 的时候**不要求已经完成了所有的算法的设计，也不是要求展示最终版本**，只是每个任务都要有开始尝试，并且有自己的规划
- 内容：团队如何分工，自行测试的方法，每个任务使用的方法，结果，改进思路。
- 时间：**每组展示 5 分钟，提问时间 1 分钟**
- 要求：每个人都要到场，每个组的所有成员都要发言

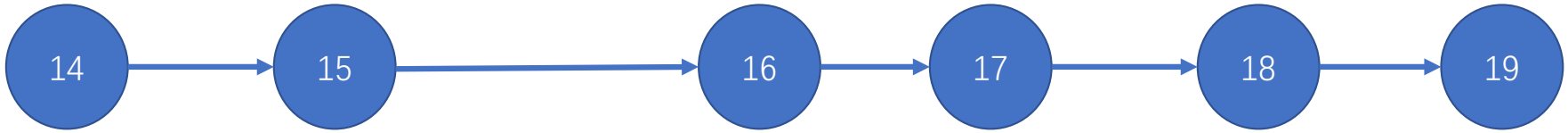
评分标准

- Project 占期末实验总评 50%
- Project 100%中：
  - 排名 40%（小组算分）
  - Pre 20% （单独算分）
  - 报告 40% （单独算分）
  - 加分（5~20%）（小组算分）
    - 有足够的理论依据证明自己的优化正确且有效
    - 尝试新算法

最终提交

- DDL：19周周五（1月11日）23:59:59
- 提交内容：
  - 实验报告，每个人一份，命名为：组号_学号_姓名拼音_report.pdf，如“2_15350000_xiaoming_report.pdf”
  - 展示PPT，每个组一份，命名为：组号_presentation.pptx，如“2_presentation.pptx”
  - 结果是三个csv文件，交到 ftp 三个对应文件夹。每个csv命名为：组号.csv，如“2.csv”，无论之前是否已经有成绩，都要提交一份最后的结果跑最终 rank。
  - 源码zip，包含多个文件，命名为：组号_code.zip，如“2_code.zip”，里面包含一个 **readme** 文件，阐述各文件用途

时间轴



1. 12月10日晚23:59:59前确认分组
2. **可跨时段组队，每组1-3人**
3. 12月11日公布pre顺序安排，随机安排

1. 设计自己的算法，三个任务最好同步进行
2. 基础算法实现成功后，尝试提交结果查看算法效果
3. 尝试对算法进行优化，思考可能存在的问题，不要盲目地乱优化，就算是调参数也要知道怎么调，调什么，调了之后会有怎样的理论效果，实际效果又是什么

1. 按照随机安排的 pre 顺序来实验课教室 pre，没有安排到 pre 的组别就不用过来了。
2. pre 完**不可以离开**，最后会签到。
3. 展示的时候着重展示目前自己尝试过的算法，有什么效果，自己做过什么优化，除了提交rank之外，自己如何做测试等。
4. 每个组是**一个整体**，但是评分分开评分，意思就是，每个组的同学对自己组目前项目的情况不能只知道自己的那部分，不是自己负责的部分，要知道大概的情况，比如用的什么算法，有没有特殊的处理，什么效果，下一步打算怎么做；这些是**提问重点**。

组队名单上报

- 12月10日晚23:59:59前确认分组
- 可以跨上课时段组队
- 一组1~3人，评分标准**没有区别**，推荐组队完成
- 确认分组后，上交一份 txt 文件到 ftp “**组队信息**”文件夹，到时候没有组队的同学会直接强制组队。
- txt命名为：组长学号.txt
- txt中包含以下内容：
  - 组内各成员学号，姓名，所属上课时段（包括组长）
  - 队伍名字（自己定一个，会在 rank 的时候出现）
  - 周四7-8节，周四9-10节，周五5-6节三个时段中，选择一个**组内所有成员都有空的时间段，至少选择一个上报**。

提交 Rank 重点注意事项

- 只需要提交**结果！结果！**
- 不要多一列文章序号，不要多一行文字介绍
- Test 几行有效数据，答案就提交几行，请提交前自行确认。

- 每天都可以提交！记得自己存好最佳 rank 的结果文件，最后上交

- 为了避免有个别组别很晚才开始做 project，**每周六一定要提交有数据的一个结果**！少提交一次，pro的分数扣 5 分，自己衡量。

提交 Rank 要求

- 每天 ftp 的结果文件夹会每天清空，提交的时候**将对应 csv 交到对应任务的文件夹**，命名要求前面已经讲过，**请严格按照要求，否则无 rank，浪费一天的等待**。

- 每天可以提交十个版本，**多于十个版本的不会处理**，用**v1~v10**区分，就算如果只有一个版本，也要加“v1”。

补充内容

逻辑回归

- 之前的课件中：

> - 不同的参数设置代表着不同的模型，**在某种模型下利用给定数据 *x* 得到给定标签 *y* 的概率，是这个问题中的似然（*likelihood*）**

- 这个说法是有问题的，**基于 *x* 得到 y 的概率，是后验**
- 似然指的是**基于 *w* 得到 *y* 的概率，*w* 是我们的模型**

梯度下降

- 这是一种在求解机器学习模型的模型参数的时候常用的优化方法之一， 不是特定只能用于 LR 的方法
- 使用的时候需要：
  - 正确标签 y
  - 假说模型 h，比如我们之前使用的 logistic function
  - 利用 y 和 h 就可以得到损失函数，至于具体是什么函数又有不同的形式，比较常用的是负对数似然，这时候一般就是利用的最大似然估计法；还可以是均方误差（MSE），均方根误差（RMSE）
- 当损失函数是凸函数的时候，只有一个最优值。
- 当损失函数不是凸函数的时候，可能会存在局部最优解。

机器学习技巧

- 单一弱模型的效果可能不是很好
- 如何使用一定的技巧，训练多个模型，将这些模型联合起来？
- 比如随机森林
- 但是要记住，**技巧是通用的**，不是针对任何模型的

- **Bagging**
- **Adaboost**

Bagging

- 也叫 Bootstrap aggregation
- 原始数据集为 $X = \{x_1, x_2, \ldots, x_N\}$
- **有放回地**抽取 N 个样本，构成新的 bootstrap 数据集 X_B
- 该数据集是**有可能出现重复的样本**的
- 这样构成的数据集，理论上是与原数据集同分布的，但是实际肯定会有区别
- 生成 M 个这样的 bootstrap 数据集，用这些数据集分别训练对应的模型，**对于不同的任务进行不同的融合**，即可得到比单一模型表现要好的融合模型

Bagging

- 分类：
- 多数投票，权重投票，…

- 回归：
- 取均值，权重均值，…

Adaboost

- Adaptive Boosting，初始是为了解决分类问题
- “Boosting can give good result even if the base classifiers have a performance that is only slightly better than random”
- 引自“Pattern Recognition and Machine Learning”
- Bagging 得到的是用多个与原数据集同分布的重新采样的数据集训练出来的模型，**这多个模型之间本身是没有联系的**，把这多个模型的结果用一定的方式综合起来作为最终的预测结果

Adaboost

- AdaBoost 对这多个分类器训练是**顺序进行**的，某个分类器在训练的时候会利用到上一个分类器的预测结果
- 直接对原数据集进行训练，训练 M 次就能得到 M 个不同的模型。
- **步骤：**
  - 对 N 个数据点的权重初始化为 1/N
  - **每个数据点的权重**可以理解成表示这个数据点出现了几次
  - 利用一定的假说模型（比如 PLA，LR 等等分类模型），得出预测值 y
  - 对二元分类可以使用的损失函数（也可以用别的）：
  - $Loss_m = \sum_{n=1}^{N} w_n^m I(y_m(x_n) \neq t_n)$
  - *I*(*argument*)函数当 argument 为 true 的时候为 1，argument 为 false 的时候为 0；
  - t_n 是正确标签值

Adaboost

- **步骤：**
  - 对 N 个数据点的权重初始化为 1/N
  - **每个数据点的权重**可以理解成表示这个数据点出现了几次
  - 利用一定的假说模型（比如 PLA，LR 等等分类模型），得出预测值 y
  - 对二元分类可以使用的损失函数（也可以用别的）：
  - $Loss_m = \sum_{n=1}^{N} w_n^m \, I(y_m(x_n) \neq t_n)$
  - 如果这个权重 w_n^m 可以理解成在第 m 个模型下，某数据点 x_n 现在出现了w_n^m 次，那么如果这个数据点是当前模型分类错误的，也就相当于错了 w_n^m 次

Adaboost

- **步骤：**
  - 通过最小化损失函数，可以得到最佳的 w_n^m
  - 计算错误率 $\epsilon_m = \frac{\sum_{n=1}^{N} w_n^m I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^m}$
  - 分子是当前模型的误差，分母是一共有多少个数据点
  - ϵ_m 越大，代表着模型越差
  - 这个数值用在哪里呢？

Adaboost

- **步骤：**
  - 在损失函数里面，一个数据点的权重越大，那么它一旦被分错，造成的损失就会越大
  - 在机器学习领域，自然是要找到使损失函数最小的模型参数
  - 那么如果我们可以使得下一次训练的时候，这些当前**分类错误**的点的**权重增大**，**分类正确**的点的**权重减小**，那么下一次训练的时候是不是就可以**更加关注当前分类错误的这些点**了？

Adaboost

- **步骤：**
  - ϵ_m 这个数值用在哪里呢？
  - ϵ_m 要越小越好，且这个值是一个(0,1)的数，那么我们设计这样一个数值：（**不唯一**）
  - $u_m = \sqrt{\frac{1-\epsilon_m}{\epsilon_m}}$
  - 当 ϵ_m 小于 1/2 的时候，这个值大于1
  - 对于一个弱分类器，也得比随机好一点点，也就是错误率应该小于 1/2
  - 那么对于分类正确的点：$w_n^{m+1} = w_n^m / u_m$
  - 对于分类错误的点：$w_n^{m+1} = w_n^m * u_m$

Adaboost

- **步骤：**
  - 根据这样的思路，训练了 M 个分类器之后，这 M 个分类器也不是简单的多数投票。既然都算了错误率了，当然要用上
  - $\ln(u_m) = \ln(\sqrt{\frac{1-\epsilon_m}{\epsilon_m}})$
  - 这个数值**可以充当**每个模型的权重（不唯一），在决策的时候：
  - $y(x_n) = sign(\sum_{m=1}^{M} \ln(u_m)\, h_m(x_n))$

总结

- Adaboost 一定要在 Project 中任何一个任务中使用

- Project 对任务有任何问题的，**先看 PPT，再有问题的再问我**
- 要是想知道在当前数据集下，大概是个怎样的准确率，可以划分验证集之后，调用现成库函数跑一下训练集，然后在验证集上看看准确率
- 抄袭问题不再多说，**如果 Project 发现抄袭，实验总评直接就是不及格的了**。