人工智能基础编程作业2实验报告

垃圾邮件分类

陈翊辉

PB15111656

邮件预处理

通过观察发现，邮件数据主要包含头部，html标签，base64，图片等需要处理的部分

邮件预处理分为以下几个步骤：

- 去除头部：寻找第一个空行，删除之前内容
- 去除html标签及js代码等：使用正则表达式匹配js代码，使用简单的状态机去除html标签
- base64解码：使用字符串搜索查找base64部分，解码base64
- 将解码的base64重复第2步
- 将非字母删除
- 将字母变为小写

还有部分邮件为中文、朝鲜语、繁体中文、多部分的，手动处理即可

朴素贝叶斯分类器

> 朴素贝叶斯是一种构建分类器的简单方法。该分类器模型会给问题实例分配用特征值表示的类标签，类标签取自有限集合。它不是训练这种分类器的单一算法，而是一系列基于相同原理的算法：所有朴素贝叶斯分类器都假定样本每个特征与其他特征都不相关。举个例子，如果一种水果其具有红，圆，直径大概3英寸等特征，该水果可以被判定为是苹果。尽管这些特征相互依赖或者有些特征由其他特征决定，然而朴素贝叶斯分类器认为这些属性在判定该水果是否为苹果的概率分布上独立的。
>
> 对于某些类型的概率模型，在监督式学习的样本集中能获取得非常好的分类效果。在许多实际应用中，朴素贝叶斯模型参数估计使用最大似然估计方法；换而言之，在不用到贝叶斯概率或者任何贝叶斯模型的情况下，朴素贝叶斯模型也能奏效。

朴素贝叶斯分类器计算比较简单，基本基于贝叶斯定理

$$p(C|F_1,F_2,\ldots,F_n) = \frac{p(C)p(F_1,F_2,\ldots,F_n|C)}{p(F_1,F_2,\ldots,F_n)}$$

即

$$posterior = \frac{prior*likelihood}{evidence}$$

引入规范项的最小二乘分类器

最小二乘法分类器可以直接求解ω

$$min(X\omega - y)^2 + \lambda||\omega||^2$$

易得

$\omega = (X^T X + \lambda * diag(n))^{-1} X^T y$

使用numpy直接计算即可

支持向量机

支持向量机通常可以用凸二次优化求解，这里用SMO算法求解。

1 随机数初始化向量权重α_i并计算偏移b

2 初始化误差项E_i

3 选取两个向量作为需要调整的点 4 令$\alpha_2^{new} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{K}$

5 如果$\alpha_2^{new} > V$

6 令$\alpha_2^{new} = V$

7 如果$\alpha_2^{new} < U$

8 令$\alpha_2^{new} = U$

9 令$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new})$

10 利用更新的

α_1^{new}和α_2^{new}修改E_i 和b的值

11 如果达到终止条件，则停止算法，否则转3

结果分析

朴素贝叶斯

threshold：0.8，

F平均值：0.87，

运行时间：<1s

最小二乘法

lambda：0.01

F平均值：0.67，

运行时间：<1s

支持向量机

- 线性核函数下

F平均值：0.76

运行时间：>10min

- RBF核函数下

F平均值：0.86

运行时间：>20min

交叉验证

交叉验证代码比较容易，只需将数据均有划分为5份，取4份作为训练数据，取1份作为测试数据，每次运行后循环移动即可。

代码详见各方法代码中的交叉验证部分

朴素贝叶斯

threshold	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
F0	0.7904	0.8276	0.8276	0.8208	0.8984
F1	0.8249	0.8556	0.8427	0.8757	0.7545
F2	0.8681	0.8492	0.8314	0.8715	0.8715
F3	0.8877	0.8508	0.8666	0.8649	0.8973
F4	0.8317	0.8827	0.8804	0.8249	0.8229
AVG(F)	0.84056	0.8532	0.8497	0.8517	0.8489

threshold ： 0.6

最小二乘法

λ	**0.01**	**0.1**	**1**	**10**	**100**	**1000**
F0	0.7511	0.6555	0.6471	0.6204	0.6632	0.6801
F1	0.6244	0.7030	0.6893	0.7272	0.6783	0.6175
F2	0.7087	0.6569	0.6305	0.6582	0.7196	0.6154
F3	0.6600	0.7053	0.6701	0.7208	0.6257	0.6114
F4	0.6441	0.6519	0.6941	0.6476	0.6636	0.6111
AVG(F)	0.6777	0.6745	0.6662	0.6749	0.6701	0.6273

λ：0.01

支持向量机

由于svm参数有二维度，只列出平均F

sigma \ C	1	10	100	1000
0.01d	0.7401	0.4904	0.8267	0.6742
0.1d	0.7532	0.5494	0.8642	0.6954
d	0.7427	0.5900	0.8381	0.6771
10d	0.7326	0.5881	0.7910	0.6301
100d	0.7426	0.5707	0.6748	0.6000

从以上数据可以发现，在三种方法中最小二乘法算法最简单，效果也最差；svm算法最复杂，效果也最好；朴素贝叶斯介于两者之间。需要说明的是，朴素贝叶斯与最小二乘法基本都在2s内运行；而svm需要10min以上的时间。

对于同样的数据使用sklearn的svm，F值能达到0.96以上，这说明邮件数据并无问题，自己实现的svm达不到这个正确率，除了算法简陋，调参原始之外，还有一个重要原因是SMO速度太慢，需要迭代多次才能收敛，实际运行中，都因为达到最大次数而提前停止了。

优化

保存预处理的邮件结果

为了避免每次运行都重复对邮件预处理，将预处理后的邮件保存，为了方便打开，将文件名也修改为纯数字：0001，0002.....

处理后的邮件仅包含小写英文字母的单词

如spam的0001

> save up to on life insurance why spend more than you have to life quote savings ensuring your family s financial security is very important life quote savings makes buying life insurance simple and affordable we provide free access to
>
> ......
>
> and or wish to be removed from our list please click here and type remove if you reside in any state which prohibits e mail solicitations for insurance please disregard this email

最小二乘法用梯度下降迭代求解

虽然最小二乘法可以直接求解，但当矩阵维数较大时，求逆矩阵需要时间较长，可以使用梯度下降的方法求解

$\omega_j = \omega_J(1-\alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum(h_\theta(x^i)-y^i)x_j^i$

适当调整学习率α当矩阵维数较大时明显快于直接法求解。

实验心得

本实验应该是这门课程中最难的实验了；本人曾经自学Coursera的machine learning课程，其中也包含了最小二乘法分类器，支持向量机，且支持向量机的实验作业正是垃圾邮件分类，不同的是，ng的课程实验并不需要自行提取邮件，也不强调自己实现核心功能。

如果只是需要自行提取邮件生成数据，或自行实现核心算法，或调参并困难；而这些合并起来就非常困难了，以支持向量机为例：虽然表面上只需要调节sigma和C两个参数，实际上需要调节特征词选取的标准，调节SMO算法的误差接受范围epsilon，最大迭代次数等；其中任何一环不对，都会导致很低的F；还有就是调参需要花费大量时间；使用sklearn的svm能达到0.95的数据，在自行实现的SMO-SVM下就只有0.8甚至更低了。

本实验的最主要的收获是增强了对相关算法原理及应用的理解，比如朴素贝叶斯虽然有公式可以计算概率，但在实际编程中可能出现值太小低于浮点表示范围，而需要做平滑处理，又如svm用直接求解方法几乎不可能，只能使用迭代法；了解了Python非常好用的库如scipy，numpy，cvxopt，sklearn（虽然不让用）。