

《机器学习》编程作业 2——逻辑回归

本次的训练营作业使用 `python+numpy` 实现。对于作业文件，推荐大家安装 `anaconda` 后使用 `jupyter notebook` 打开。

作业说明主要需要参考英文原版 `pdf`（在文件夹里），本文件只是一个作业提示，告诉大家需要完成作业中的哪些地方以及注意事项。对于其他不需要大家完成的程序，我们鼓励大家看懂。

首先需要说明的是，每一个需要大家完成的点都会有如下类似的语句：

your code here (appro ~ 1 lines)

大家需要把自己的代码写在这行注释下面，括号里是提示大家大概需要写几行代码。这里只是老师根据自己的经验对代码的估计，并不是说你一定要写这么多行。假如你写的代码多余注释中的估计行数，不要担心！只要你的代码能够正常运行就可以了。

在有些地方，为了降低编程难度同时提高代码规范性，我们已经为大家写了一部分代码，大家只需要修改即可。例如，在初始化参数 `theta` 中，你会看到如下的代码：

```
theta = None
```

这里，大家只需要删除赋值等式右边的 `None`，修改为自己编写的表达式即可。

本次作业需要完成的地方有：

- 1、Sigmoid 函数
- 2、Costfunction 函数（不含正则化）
- 3、gradient 函数（不含正则化）
- 4、predict 函数
- 5、regularized cost 函数
- 6、regularized gradient 函数
- 7、改变正则化参数 λ 并查看决策边界的变化效果（选做）

需要说明的是，这次作业使用了 `scipy.optimize` 库。`scipy.optimize` 是一个优化算法包。如果你没有听说过这个库，没有关系。正如吴恩达老师在视频中说的

那样，你并不需要去手写自己的优化算法，也不需要看懂源码。

为了让大家更好的看到自己写的算法究竟怎么样，在这里使用了一个机器学习库 `sk-learn`。如果你不熟悉这个库或从未听说过，不用着急。这并不影响你正常完成本次编程作业。

在完成一个地方的代码之后，我们强烈建议同学们运行一下代码看看是否报错，同时通过对比英文原版作业中的期望输出值来检验自己的结果是否正确。

这是训练营的第二个作业了，难度比第一个作业要大一些。相信你已经在第一个作业中学到了一些编程技巧，也获得了一些自己的经验。大家可以把在完成作业当中遇到的问题和解决的办法记录下来，这都将成为自己的宝贵财富。