统计信号处理大作业

最小二乘法

姓名：王道烩

学号：2015011006

班级：无52

1. 实验目的

使用最小二乘法解决协同滤波问题。

1. 实验原理

要填充的矩阵可以表示为：,其中，表示用户i对电影j的打分是用户i的隐变量与电影j的隐变量的内积。考虑到用户类型和电影类型有限，因此有理由相信评分矩阵M是低秩的。由于矩阵的秩是矩阵奇异值向量的0范数，

可以将其放缩到1范数，即矩阵的核范数。由于举证的核范数满足：

因此上述问题可以通过优化下述目标函数进行求解：



其中为控制矩阵低秩程度的超参数。W为标志矩阵，=1代表用户i对电影j已经打过分了，=0表示未打分，\*表示矩阵对应元素相乘。

对于上述优化问题一般有两种思路，本人采用的方法为交替最小二乘法，及固定U以V

作为优化变量，固定V以U作为优化变量，交替地进行此过程的求解。

公式推导如下：（其中\*代表数量乘 .\*代表矩阵相乘 U(i,:) 为U的第i行 V(:,i)为V的第i列）

（1）固定V以U作为变量时：







又因为



所以上式可写为：



这样就得到了每次U一行的更新公式

1. 固定U以V为变量时：







又因为



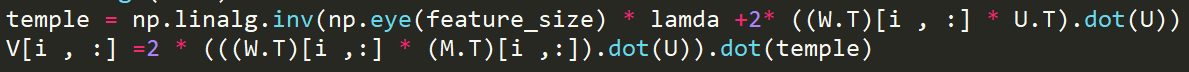
所以上式可以化简如下：



上式就时V每一每一行的更新公式。

将上面的更新公式用代码表示如下：





1. 实验内容
2. 实验环境

操作系统：Ubuntu16.04LTS

工具：python3.5 + numpy + scipy + matplotlib

硬件：CPU：8\*i7 7700HQ 内存：8G

1. 实验步骤

本次实验一共主要有有三个文件，其中：

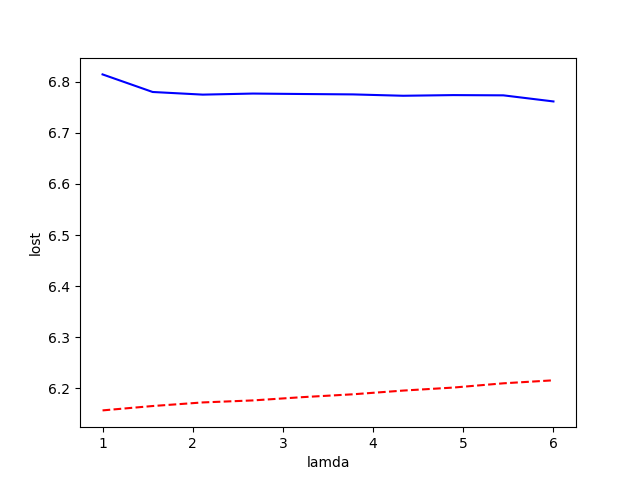
select\_data.py文件用于挑选训练集和测试集。一共实现了三种挑选方法：随机法，均匀挑选测试集和均匀挑选训练集。通过适当的添加删除注释可以选择某种方法来生成文件data\_set\_my.mat文件。内含训练集和测试集数据。

ALS.py文件是算法主题，实现了交替最小二乘算法，通过这行次文件可以读取data\_set\_my.mat文件，并将得到的U V result=保存在resulr.mat文件中。

Plot.py文件主要用于画图来分析损失随着参数以及特征维度feature\_size的变化趋势，从而确定最终选择的参数。

1. 实验结果分析
2. 参数的影响

首先，我选择一个固定的训练集和测试集，来通过选择不同的来观察在训练集上的损失和在测试集上的损失，得到的图象如下：

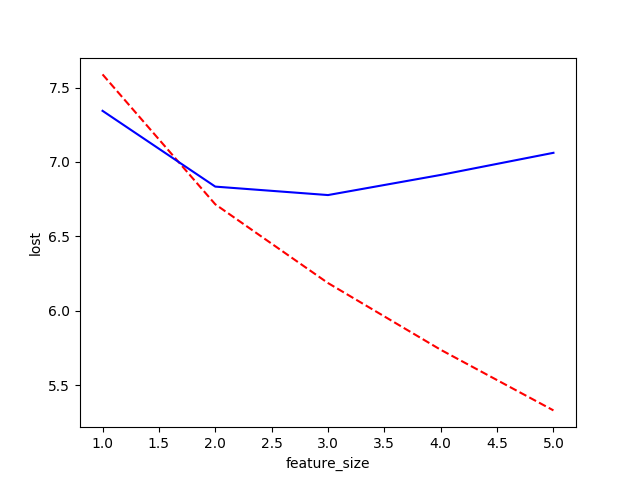


其中红色虚线为在训练集上的损失，蓝色实现为在测试集上的损失。

可以观察到，随着的增大，在训练集上的损失会增加而在测试集上的损失会减小，这是因为的作用可以看做一种避免过拟合的效果。通过增加的值，能够使我们得到的模型更加平滑，这会增加在训练集上的损失，但是增加了模型的泛化能力，所以其在训练集上的损失会降低。通过不断的实验，最终将参数选为10。

1. 特征维度feature\_size的影响

通过固定，改变feature\_size，可以得到不同的特征维度，在训练集和测试集上的损失变化如下：



其中红色虚线为在训练集上的损失，蓝色实线为在测试集上的损失。可以看到，随着特征维度的增加，在训练集上的损失是逐渐降低的，但是在测试集上的损失是现降低再上升的。这是因为，特征维度越大，模型的代表能力就越强，就越能够精确地拟合训练数据，使得在训练集上的损失降低，但是，当特征维度超过某一个值之后，就会出现过拟合现象，这使得模型的泛化能力下降，从而使得其在测试集上的损失上升。

1. 训练集以及测试集选取方法的对比

本人在选择训练集和测试集方面一共使用了三种方法，下对这三种方法进行分析：

随机法：这种方法是最简单的方法，在提供的数据集上随机选取10000个作为测试集，剩下的作为训练集，这种方法得到的最终在测试集上的损失大致为7.8左右。效果不是非常理想。

均匀选取测试集：通过计数，本人发现每行的数据个数最小为15，我在每行上面交替选取11个和10个数据作为测试集，剩下的作为训练集，这样得到的在测试集上面的损失大致为8.1左右，效果更差。

均匀选取训练集：通过两次循环，保证选取的数据集在行和列上分布比较均匀。这样最终得到的在测试集上的损失大致在7.0左右，效果比较好。

通过以上对比，本人最终选择了第三种方法。

最终结果为：=15 特征维度为3 测试集损失为6.90 收敛时间为14秒，迭代次数为75.

1. 总结

通过本次实验，我对最小二乘法在实际问题中的应用有了更加深刻的理解。同时也让自己进一步熟练掌握了Python的使用。