HW2 矩阵分解

1. 问题描述

个性化推荐是大数据的一个典型应用，其利用已知的用户浏览历史推荐新的信息：给定用户行为矩阵 $\mathbf{X}_{m*n}$，其中 m 是用户数，n 是需要推荐的内容数量，$\mathbf{X}$ 中的元素 $\mathbf{X}_{ij}$ 表示用户 i 对某个内容 j 的推荐/喜好程度或打分。所谓的推荐任务就转化成，当已知 $\mathbf{X}$ 中的一部分值的时候，如何对未知值进行预测。

本次作业的任务是利用视频推荐网站 Netflix 的数据集完成推荐任务。主要用到矩阵分解的相关知识；作为对比，同时了解协同过滤的相关内容。

2. 数据集

作业所使用的是 Netflix 推荐竞赛的一个子集，包含 10000 个用户和 10000 个电影。用户行为数据包含用户对电影的打分，分数的取值范围是 1-5。选取行为数据的 80%作为训练集，其余的 20%作为测试集。

具体的文件格式如下：

(1) 用户列表 users.txt

文件有 10000 行，每行一个整数，表示用户的 id，文件对应本次作业的所有用户。

(2) 训练集 netflix_train.txt

文件包含 689 万条用户打分，每行为一条打分，对应的格式为：

用户 id 电影 id 分数 打分日期

其中用户 id 均出现在 users.txt 中，电影 id 为 1 到 10000 的整数。各项间用空格分开。

(3) 测试集 netflix_test.txt

文件包含约 172 万条用户打分，格式与训练集相同。

(4) 电影名称 movie_titles.txt

文件对应每部电影的年份和名称，格式为：

电影 id，年份，名称

各项之间用逗号分隔。

3. 作业内容

（1）数据预处理

大家首先要体会在大数据处理过程中不得不经历的一个步骤：数据清洗及格式化。

对应到本次作业的问题，就是需要将输入文件整理成维度为“用户*电影”的矩阵 $\mathbf{X}$，其中 $\mathbf{X}_{ij}$ 对应用户 i 对电影 j 的打分。对于分数未知的项，可以采取一些特殊的处理方法，如全定为 0 或另建一个矩阵进行记录哪些已知哪些未知。

这一步的输出为两个矩阵，$\mathbf{X}_{\text{train}}$ 和 $\mathbf{X}_{\text{test}}$，分别对应训练集与测试集。

鉴于大家计算机的计算能力有区别，如果发现处理 10000*10000 的矩阵有困难，可以选取部分用户和部分电影构成的子集，并在作业报告中说明选取规则（不得少于 2000*2000）。

（2）协同过滤

协同过滤（Collaborative Filtering）是最经典的推荐算法之一，包含基于 user 的协同过滤和基于 item 的协同过滤两种策略。

本次作业需要实现基于用户的协同过滤算法。算法的思路非常简单，当需要判断用户 i 是否喜欢电影 j，只要看与 i 相似的用户，看他们是否喜欢电影 j，并根据相似度对他们的打分进行加权平均。用公式表达，就是：

$$score(i,j)=\frac{\sum_k\{sim[\mathbf{X}(i),\mathbf{X}(k)]\cdot score(k,j)\}}{\sum_k sim[\mathbf{X}(i),\mathbf{X}(k)]}$$

其中，$\mathbf{X}(i)$表示用户 i 对所有电影的打分，对应到本次作业的问题中，就是 $\mathbf{X}$ 矩阵中第 i 行对应的 10000 维的向量（未知记为 0）。

$sim[\mathbf{X}(i),\mathbf{X}(k)]$表示用户 i 和用户 k 对于电影打分的相似度，可以采用两个向量的 cos 相似度来表示，即：$\cos(\mathbf{x},\mathbf{y})=\frac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}|\cdot|\mathbf{y}|}$。

通过上面的公式，就可以对测试集中的每一条记录，计算用户可能的打分。

（3）评价指标

采用 $RMSE$（Root Mean Square Error，均方根误差）作为评价指标，计算公式为：

$$RMSE=\sqrt{\frac{1}{n}\left(\sum_{<i,j>\in Test}(\mathbf{X}_{ij}-\tilde{\mathbf{X}}_{ij})^2\right)}$$

其中 $Test$ 为所有测试样本组成的集合，$\mathbf{X}_{i,j}$为预测值，$\tilde{\mathbf{X}}_{i,j}$为实际值，n 为测试样本数。

（4）基于梯度下降的矩阵分解算法

课程业已介绍了矩阵分解的相关知识。对于给定的矩阵 $\mathbf{X}$，可以将其分解为 $\mathbf{U}$、$\mathbf{V}$ 两个矩阵的乘积，使 $\mathbf{UV}$ 的乘积在已知值部分逼近 $\mathbf{X}$，即：$\mathbf{X}_{m*n} \approx \mathbf{U}_{m*k}\mathbf{V}_{n*k}^{\mathrm{T}}$，其中 k 为隐空间的维度，是算法的参数。

基于行为矩阵的低秩假设，可以认为 $\mathbf{U}$ 和 $\mathbf{V}$ 是用户和电影在隐空间的特征表达，它们的乘积矩阵可以用来预测 $\mathbf{X}$ 的未知部分。

本作业可以使用梯度下降法优化求解上述问题。目标函数是：

$$J = \frac{1}{2}\|\mathbf{A}^{\circ}(\mathbf{X} - \mathbf{U}\mathbf{V}^{T})\|_F^{\,2} + \lambda\|\mathbf{U}\|_F^{\,2} + \lambda\|\mathbf{V}\|_F^{\,2}$$

其中，$\mathbf{A}$ 是指示矩阵，$\mathbf{A}_{ij}$=1 意味着 $\mathbf{X}_{ij}$ 的值为已知，反之亦然。°是阿达马积（即矩阵逐元素相乘）。$\|\cdot\|_F$表示矩阵的 Frobenius 范数，计算公式为$\|\mathbf{A}\|_F = \sqrt{\sum_i \sum_j \mathbf{A}_{ij}^2}$。在目标函数 J 中，第一项为已知值部分，为 $\mathbf{UV}$ 的乘积逼近 $\mathbf{X}$ 的误差；后面的两项是为防止过拟合加入的正则项，λ 为控制正则项大小的参数。

当目标函数取得最小值时，算法得到最优解。可分别对 $\mathbf{U}$ 和 $\mathbf{V}$ 求偏导，结果如下：

$$\frac{\partial J}{\partial \mathbf{U}} = [\mathbf{A}^{\circ}(\mathbf{U}\mathbf{V}^{T} - \mathbf{X})]\mathbf{V} + 2\lambda\mathbf{U}$$

$$\frac{\partial J}{\partial \mathbf{V}} = [\mathbf{A}^{\circ}(\mathbf{U}\mathbf{V}^{T} - \mathbf{X})]^{T}\mathbf{U} + 2\lambda\mathbf{V}$$

之后，可迭代对 $\mathbf{U}$ 和 $\mathbf{V}$ 进行梯度下降更新，具体算法如下：

Initialize U and V (very small random value);
Loop until converge:

$\mathbf{U} = \mathbf{U} - \alpha\frac{\partial J}{\partial \mathbf{U}}$;

$\mathbf{V} = \mathbf{V} - \alpha\frac{\partial J}{\partial \mathbf{V}}$;

End loops

算法中 α 为学习率，通常根据具体情况选择 0.0001 到 0.1 之前的实数值。算法的收敛条件，可以选择目标函数 J 的变化量小于某个阈值。

至此，就完成了在给定的 k 和 λ 下，对目标函数 J 进行优化求解的全过程。作业时，可以将每次迭代的目标函数值和测试集上的 $RMSE$ 值 plot 出来，观察算法收敛过程中这两个指标的变化。此外，作业还需要尝试不同的 k 和 λ 值，找到算法的最佳参数。

4. 作业报告提交要求

本次作业总分 100 分，需提交作业报告和代码。报告需要包含的内容如下：

（1）数据整理阶段的简单说明，以及最后是否选用的全量数据。（15 分，未能使用全量数据的酌情扣 5-10 分）

（2）基于用户的协同过滤实现。包含代码的简单介绍，最终的 $RMSE$ 结果，算法的时间消耗。（30 分）

注意，由于本作业中 $\mathbf{X}$ 矩阵非常稀疏，因此使用 Matlab 等对稀疏矩阵有特别优化的工具时，请尽可能使用矩阵运算，避免使用 for 循环。公式如何转化为矩阵形式的运算，请自行推导。

（3）矩阵分解算法结果。（40 分）报告需要包含：

a）对于给定 k=50, λ=0.01 的情况，画出迭代过程中目标函数值和测试集上 $RMSE$ 的变化，给出最终的 $RMSE$，并对结果进行简单分析。

b）调整 k 的值（如 20,50）和 λ 的值（如 0.001,0.1），比较最终 $RMSE$ 的效果，对结果进行简单分析，选取最优的参数组合。

（4）将（2）（3）的结果进行对比，讨论两种方法的优缺点。（15 分）

注意事项：

（1）请将作业报告和代码打包成一个 zip 文件上传到网络学堂。

（2）只需要**提交作业报告和代码**，请***不要提交***中间结果和数据文件。不满足要求的酌情扣分（见 5）。

（3）代码中请给出“详细”的中文注释，解释功能、基本方法等。

（4）评分时会综合作业报告的撰写情况、代码实现的情况、代码注释、代码规范性等。

（5）未按规定提交文件、提交的作业报告或代码不符合要求，将酌情扣分。

5. 数据下载

相关数据集提供清华云盘下载地址：（访问密码：bigdatathu）

https://cloud.tsinghua.edu.cn/d/f5a2f06db98046acb2f9/

本次作业所用的数据在“MatrixFactorization”子目录下。