推荐系统课程实验报告

基于 SVD 的协同过滤算法

霍超凡

1 实验概述

协同过滤算法集群体之智慧，与基于内容的推荐算法不同，协同过滤算法不会分析根据单个物品和单个用户来计算两者的匹配程度，而是采用群体的思想，放眼全局，系统中的任何单个物品和用户都会对最终的决策有影响，多个用户和物品协作完成推荐的任务。SVD 是继基于领域的协同过滤算法的另一种算法，算法的核心思想不变，只是换了一种方法。在前一个实验中，一个用户是由这个用户对物品的评分所组成的向量刻画的，这个向量非常稀疏，SVD 将用户和物品映射到低维空间中，可以解决这种稀疏问题。在一个稀疏度高达 90%矩阵中做矩阵分解是不现实的，而是通过一个优化函数，最小化用户向量和物品向量的点积与用户对物品评分之间的差距。本次实验通过实现 SVD 协同过滤算法，探讨了其中的一些细节，比如模型参数抉择和过拟合问题，加入限制减轻数据量小的问题。

2 算法细节

给定一个评分矩阵R，其中第 u 行、第 i 列为用户 u 对物品 i 的评分，对这个评分矩阵进行 SVD 分解得到

$$\mathrm{R} = \mathrm{U}\Sigma\mathrm{V}$$

经过分解后，R 矩阵可以被分解成若干个矩阵的加权和

$$\mathrm{R} = \sigma_1 R_1 + \sigma_2 R_2 + \cdots + \sigma_k R_k$$

其中$\mathrm{R_i}, i = 1,2, \ldots, k$表示某一个因素，用户对物品的评分是由若干个因素（比如用户的打分偏差）构成的

$$\mathrm{r} = \sigma_1 \mathrm{r}_1 + \sigma_2 r_2 + \cdots + \sigma_k r_k$$

σ反映这个因素对最终评分的影响程度，通过 SVD 分解后，把强因素和弱因素分离，去掉弱因素达到降维的目的。从另一种角度理解，SVD 分解的结果可视为两个矩阵的乘积

$$\mathrm{R} = \mathrm{PQ}$$

评分矩阵中的任意元素可由 P 的行向量与 Q 的列向量点积得到，

$$\mathrm{R_{ui}} = P_{u,:} \cdot Q_{:,i}$$

行向量$\mathrm{P_{u,:}}$和列向量$\mathrm{Q_{:,i}}$是分别对用户 u 和物品 i 的一个刻画，两者的点积（即两者的相似度）等于评分。最终的结果使得相匹配的物品和用户在隐语义空间中相近较近。为了克服稀疏矩阵的分解问题，引入优化问题

$$min \quad \sum_{ui}(p_u \cdot q_i - r_{ui})^2 + \lambda(|p_u|^2 + |q_i|^2)$$

上述优化问题p_u是对用户 u 的一个向量刻画，对应于矩阵分解中 P 的行向量，q_i是对物品i的一个向量化刻画，对应于矩阵 Q 的列向量，第二项是为了防止过拟合，λ是权衡两者的系数，这个优化问题旨在寻找用户和物品的最简单的向量化表示，使两者的点积能够反映两者的相关度。解决上述优化问题的算法使 SGD 算法，反复对参数求梯度、更新参数。

另有算法引入用户和物品的偏差以提高性能，优化函数变成

$$min \quad \sum_{ui}(p'_u \cdot q'_i + b_u + b_i - r_{ui})^2 + \lambda(|p'_u|^2 + |q'_i|^2 + b_u^2 + b_i^2)$$

其中b_u, b_i表示用户 u、物品 i 对分值的偏差，心细的读者可以马上发现，如果$p_u = (1, b_u, p'_u), q_i = (b_i, 1, q'_i)^T$，这两个优化函数完全一致，与之前的优化函数相比只不过对参数做出了一些限制，那么这个限制对最终模型的结果是否有提升呢？这个问题类似于神经网络中是否添加偏置项。在文章[1]还引入了全局因素，对上述优化函数做出改进

$$min \quad \sum_{ui}(p'_u \cdot q'_i + b_u + b_i + \mu - r_{ui})^2 + \lambda(|p'_u|^2 + |q'_i|^2 + b_u^2 + b_i^2)$$

新添加的μ这一项是所有评分的一个平均，对原始优化函数在参数方面进一步作出限制，理论上，第一个优化函数应该不会比后面两个优化函数差，如果后面两个优化函数存在一个最有解，那么第一个优化函数也存在相同的最有解或甚至更好的解。在实验中，分别实现了这三种情形，结果和分析放在了第三节。

3 实验结果

当评分矩阵规模较小时，使用矩阵分解要比优化问题更高效，那么也会带来问题，评分矩阵存在很多缺失值，缺失值如何处理？在实验中，我同一将缺失值添为 0，这种使用矩阵分解的 SVD 协同过滤算法在训练集上具有较小的 RMSR，但是在测试集上具有非常高的 RMSR，这是因为在分解之前我们对那些出现在测试集中的评分同一添为 0。图 1 给出了这种模型的 PR 曲线图，效果不错（注意到，在前面报告中提到计算 precise 等指标时，放眼到整个数据集，不仅有测试集还有训练集）。

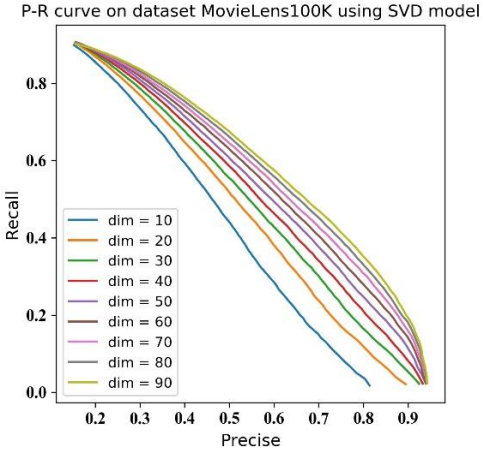

图 1：使用矩阵分解的 SVD 协同过滤算法在数据集 MovieLens 数据集上的 PR 曲线

图 2 给出了一些 SVD 模型在数据集上的 RMSE 指标，左图是在训练集上的 RMSE 指标，右图是在测试集上的 RMSE 指标，横坐标 dim 表示向量$\mathbf{p}_u, q_i$的维度，当向量维度增加后，模型在训练集上的效果变好，但是出现了过拟合，这或许不是λ取值不当的缘故，实验中λ同一设为 0.15。图例中的 raw SVD 是使用在第二节提到的第一种优化函数得到到的模型，with global mean 在此基础上把向量中的一个维度变成平均评分，with bias 表示添加了偏置项（对应于第二个优化函数），with global mean and bias 表示同时添加全局平均评分和偏置，实验中使这几个模型中的参数数目尽可能保持一致。对比这四个模型，可以发现添加了限制的模型在训练集上的 RMSE 不如 raw SVD，但是在测试集上的结果要好于 raw SVD，非常有意思的是，仅仅添加 global mean 会使结果变差，但是结合 bias 模型后性能在 bias 模型的基础上又有了提升。

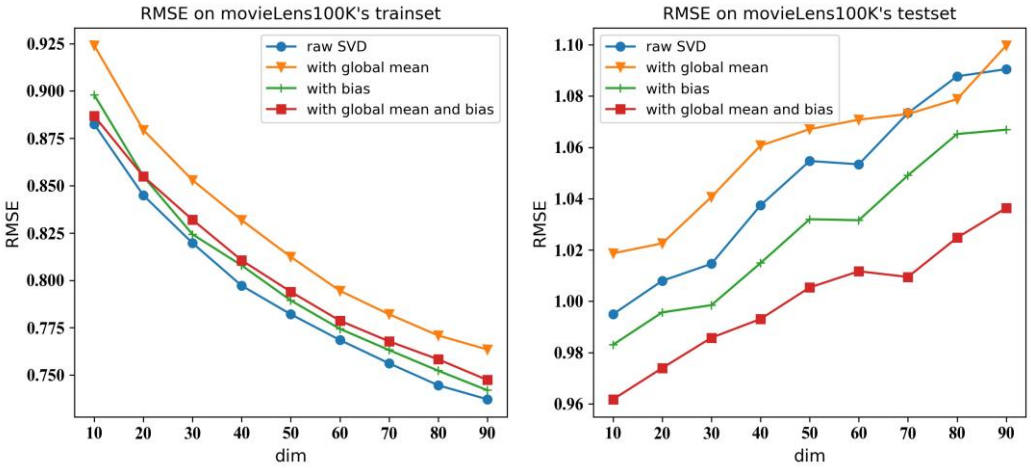

图 2：使用不同优化函数的 SVD 协同过滤算法在 movieLens 训练集和测试集上的 RMSE 指标

基于 SVD 算法更倾向于推荐未知的物品（没有出现在训练集的商品），使用 SVD 算法得到的 precise 往往非常低，top3 范围内仅为 1%左右，但是新鲜度达到 90%以上，如果计算精度时把那些评分为 0 的去掉或许会使结果好一点，也更能反映出模型的效果，进一步的修正目前还没做。

参考文献

[1] Koren, Yehuda. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'08).. 426-434. 10.1145/1401890.1401944.

[2] recommendation-system, site: https://github.com/hopebo/recommendation-system.

[3] Recommender_system_algorithm_comparision, site: https://github.com/tehruhn/Recommender_system_algorithm_comparision