# 推荐多样性实验

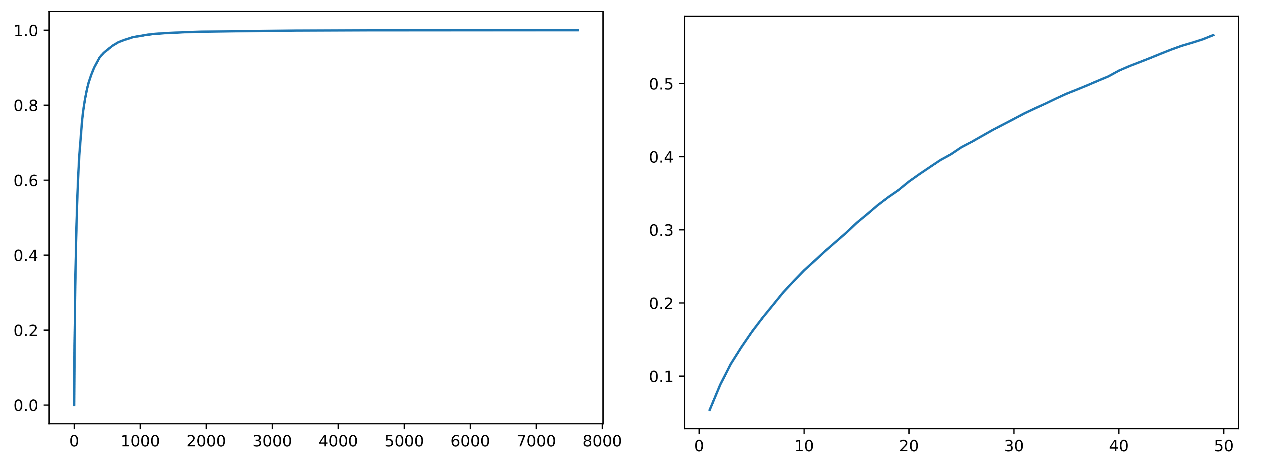
1. **概述**

在之前的survey中论述了提升推荐的多样度必然会降低推荐的准确度，需要在其中找到一个平衡点。为了找到这个平衡点，本实验使用了基于ReRank方法的几种提升多样性的排序方法，包括Reverse Rating、Popularity、Average Rating、Absolute Like、Relative Like等方法，测量了在不同阈值下，准确度和多样度的变化；并对比了这几种方法的效果。实验证明，Absolute Like 方法能够在降低少数准确度的情况下，大幅提升多样性。

1. **实验设置**

实验数据使用济宁市2018年1月1日到1月27日女性用户的滑动数据(like/dislike)，由于实验需要预知推荐系统的Rating值，而实际的Rating值很难获取，因此本实验为每个推荐的Rating进行了估算。

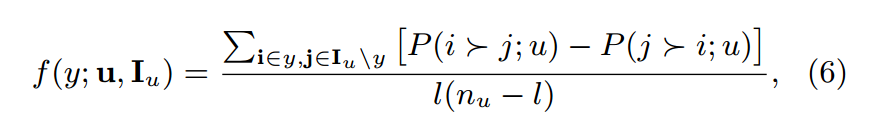
这些滑动数据中，每个用户每天滑动的次数各不相同，最少只滑动1次，最多高达7630次。图2-1(a)为每人每天滑动数据的CDF图，横轴为用户滑动次数，纵轴为少于横轴滑动次数的用户百分比。图2-1(b)将横轴的取值范围限定在50以内。其中滑动次数为10次以上的用户占比75%，20次以上的用户占比68%，因此实验取前20个作为推荐备选()，并假定推荐系统从中取出10个作为推荐()，这些值可以随着当前推荐系统的真实情况进行修改。



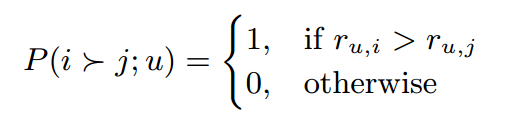
|  |  |
| --- | --- |
| (a) | (b) |
| 图2-1 每人每天滑动次数CDF图 | |

1. **测量方法**

关于准确度的测量，由于缺乏推荐系统完整Rating数据（只有滑动数据），在实验中假设原始数据的准确度为1，实验中测量的准确度**均为相对值**。准确度的测量如下：

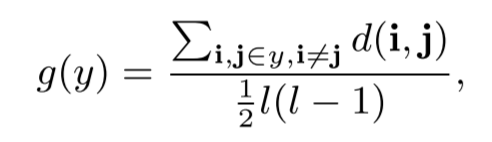


其中表示给用户U推荐的所有备选项目，。并从中抽选N个作为推荐，y为推荐项目集，满足。定义如下：

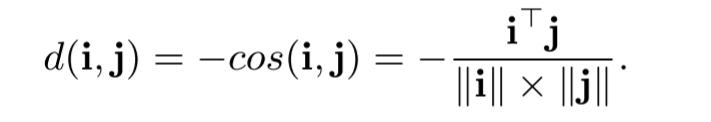


函数的取值范围为[-1, 1]。

多样性的测量使用的是任意两个用户推荐项目的不相似度的平均值，即



为用户i的推荐向量和用户j的推荐向量的不相似度，定义如下：



1. **Re-Rank方法实验**

Re-Rank方法是对于推荐的结果进行重新排序，以提高推荐的多样性。

1. 按照传统推荐方法，计算Rating并排序

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 推荐前10个给用户 | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| Rating | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Popularity | 3 | 9 | 19 | 10 | 8 | 0 | 17 | 5 | 4 | 13 | 2 | 16 | 14 | 7 | 1 | 12 | 6 | 18 | 11 | 15 |
|  |  | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |

1. 按照能提升多样性的方法排序，如这里对Popularity从小到大

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 推荐前10个给用户 | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
| Rating | 14 | 5 | 9 | 19 | 11 | 12 | 3 | 6 | 15 | 18 | 16 | 1 | 4 | 10 | 7 | 0 | 8 | 13 | 2 | 17 |
| Popularity | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|  |  | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |

1. 调整阈值以改善准确度，如，即

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 推荐前10个给用户 | | | | | | | | | |  |  | | | | | | | | |
| Rating | 14 | 9 | 19 | 11 | 12 | 15 | 18 | 16 | 10 | 13 | 17 | 5 | 3 | 6 | 1 | 4 | 7 | 0 | 8 | 2 |
| Popularity | 0 | 2 | 3 | 4 | 5 | 8 | 9 | 10 | 13 | 17 | 19 | 1 | 6 | 7 | 11 | 12 | 14 | 15 | 16 | 18 |
|  |  | | | | | | | | | |  |  | | | | | | | | |

图4-1 Re-Rank方法改进多样性的过程

如图4-1所示为该Re-Rank方法改进多样性的过程，分为三步：第一步使用传统的排序方法，计算物品的rating，并对所有物品的rating值进行排序，并、取前N个作为推荐项目，这是传统推荐系统的推荐步骤。第二步使用能够提高多样化的排序方法进行排序，该排序方法有以下几种：

**Reverse Rating方法**，传统的推荐系统首先计算每个项目i对用户u的Rating值，然后计算每个项目的rank值：，将每个项目按照Rank从小到大排序，取出前N个作为推荐。而Reverse Rating方法的排序结果与传统方法刚好相反：，即优先考虑备选中Rating值低的项目。

**Popularity方法**，该方法首先算出每个项目的被推荐给用户的次数，依照这种原则从小到大排序，即优先考虑备选中不热门的用户，以增加多样度。

**Average Rating方法**，计算每个项目被推荐的平均Rating值，并从小到大排序。由于数据中没有给出具体的Rating值，在实验中假定原先系统推荐给用户第i个物品的Rating值为 ，作为近似的Rating值。

**Absolute Likeability方法**，统计每个项目被like的数量，并从小到大排序。

**Relative Likeability方法**，统计每个项目被like的比率，从小到大排序。

第三步，调整的值，以改善准确率，限制推荐的项目的Rating都满足 。如图中的Rating=3.65比=3.8小，所以被抛弃，其后一位的项目数顺势补充为第五个推荐项目。

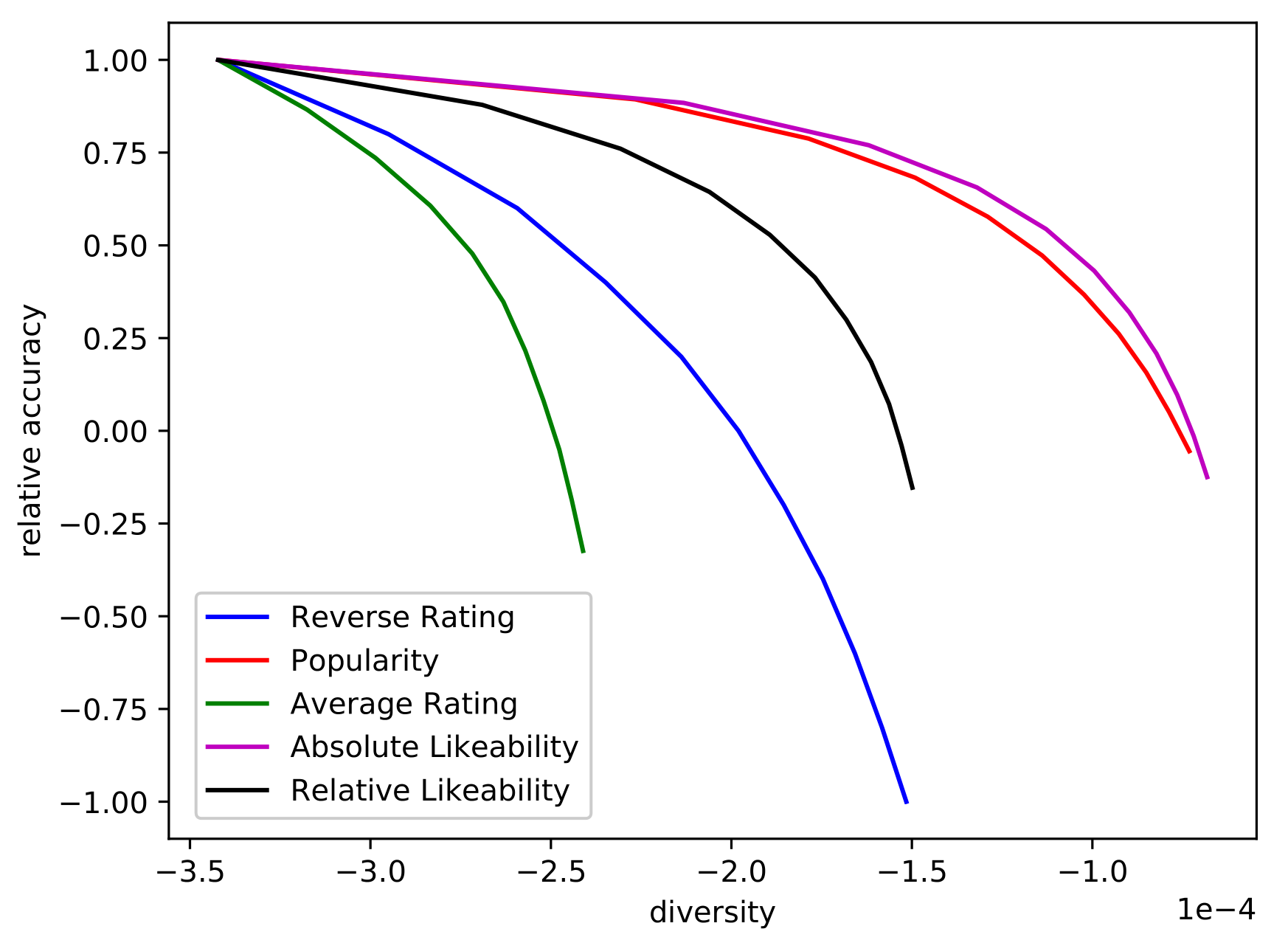


图4-2 不同排序方法的准确度-多样性曲线图

本实验分别针对上述5种排序方法，测量了在不同取值情况下，准确度与多样度的变化曲线，如图4-2所示：当相对准确度为1时为未加入任何优化的系统，其多样度最低，随着阈值的降低，其准确度出现损失，而多样度增高。表中可以看出，Absolute Likeability方法能够在损失少量的准确度情况下，实现多样性的大幅度提升，在因此该方法推荐在推荐系统中使用以提升多样度。

如图4-3所示为使用基于Absolute Likeability的Re-Rank方法后，对各个项目的推荐次数统计的结果，为了更好地展示结果，图中只取了推荐次数最高的200人进行统计。当设定的阈值越低，系统的推荐多样性越明显，可以直观地看到长尾效应的改善。

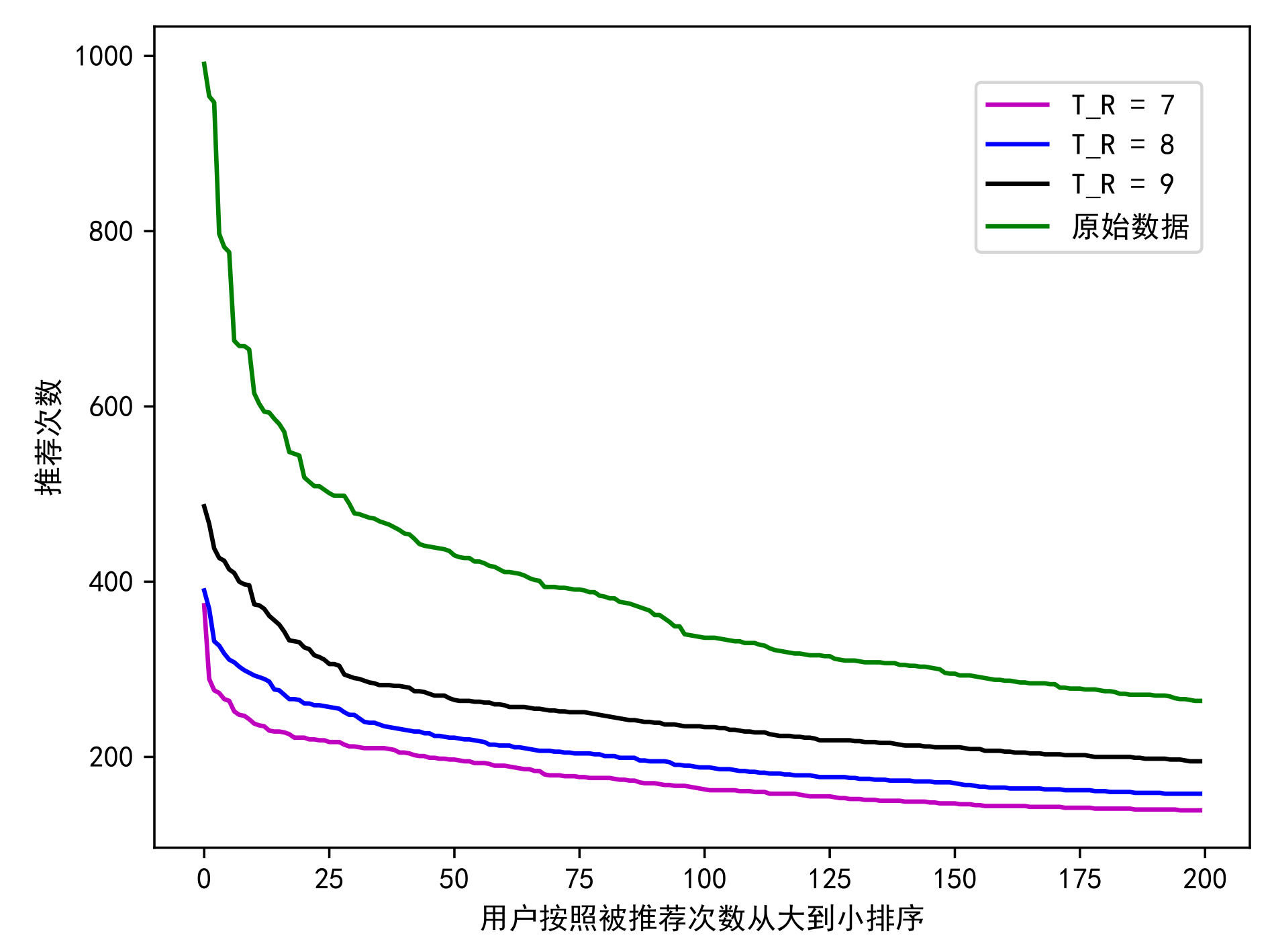


图4-3 使用基于Absolute Likeability的Re-Rank方法

后，对各个项目的推荐次数统计的结果。