

# 基于用户的协同过滤实验结果

Author: 李鹏

Date: 2020/05/07

## 实验任务1输出

- 训练集与测试集大小

```
=====
User-Based-Collaborative-Filtering
=====
```

训练集大小: 980204

测试集大小: 20005

训练集例子:

	user_id	movie_id	ratings
741853	4433	695	3
300817	1782	2770	4
69550	466	30	4

测试集例子:

	user_id	movie_id	ratings
478972	2939	2183	4
698420	4175	2375	3
143115	922	516	1

- 训练集和测试集中的“用户到电影”对应的序列

训练集用户到电影索引举例：

索引值： 4433

对应的列表：

[695, 1198, 920, 608, 1358, 3494, 2407, 1676, 2467, 2064, 1090, 25, 1250, 1258, 950, 2248, 2529, 1253, 849, 1201, 1376, 1584, 2797, 1252, 1240, 1617, 2028, 347, 1097, 1805, 3843, 1094, 2100, 2968, 527, 2944, 3365, 3487, 2921, 2951, 1021, 36, 2011, 3836, 750, 1266, 1573, 1210, 908, 3704, 1288, 1320, 3062, 383, 2922, 1294, 198, 1089, 3386, 3476, 3703, 1073, 3730, 293, 541, 1289, 2288, 1215, 2021, 1214, 185, 904, 412, 1086, 6, 1283, 1303, 1219, 2716, 1748, 2640, 3681, 368, 2841, 34

索引值： 1782

对应的列表：

[2770, 785, 3176, 2599, 3543, 968, 2034, 2762, 3928, 2901, 1959, 2369, 2987, 250, 3185, 2028, 201, 2683, 2013, 3081, 2794, 1093, 2070, 2541, 858]

测试集用户到电影索引举例：

索引值： 2939

对应的列表：

[2183, 1394, 2804, 1480]

索引值： 4175

对应的列表：

2375

## 实验任务2输出

- 训练集中“电影到用户”对应的索引

训练集电影到用户索引举例：

索引值： 695

对应的列表：

[4433, 2608, 2559, 5120, 2632, 953, 2768, 2376, 1690, 1680, 309, 1051, 3045, 5675, 3082, 4312, 58, 3618, 818, 2441, 2142, 4879, 3387, 889, 5643, 4335, 4235, 3450, 5837, 352, 5, 3539, 3498, 4123, 2472, 2776, 2581, 3690, 5961, 4375, 718, 5511, 2909, 1958, 4347, 301, 1053, 4447, 2793, 3841, 3416, 4834, 5039, 2162, 3464, 1983, 6036, 5297, 1248, 3603, 42, 5831, 970, 520, 1194, 5249, 4735, 1943, 163, 4327, 5026, 2507, 903, 5333, 2391, 1284,

索引值： 2770

对应的列表：

[1782, 3684, 3550, 1582, 3123, 4343, 1889, 2261, 1544, 2378, 1851, 5444, 1390, 3380, 60, 119, 170, 587, 1130, 5880, 1088, 5099, 2054, 331, 1025, 3192, 3931, 1794, 5205, 4266, 1606, 1861, 2631, 3878, 3755, 5, 1416, 3082, 4725, 2888, 4054, 2188, 877, 4682, 2106, 1005, 2095, 617, 2102, 4085, 5888, 675, 4999, 4072, 2214, 1803, 3726, 1221, 188, 4922, 3526, 4918, 5784, 323, 5030, 1597, 1199, 5459, 39, 777, 3822, 77, 2891, 19, 1738, 5070, 3032, 5636, 4280, 5208, 1055, 1329, 4437, 4186, 3868, 1053, 2780, 3595, 198, 5872, 517, 5850, 3462, 5717, 2086, 4001, 5896, 3973, 423, 5867, 2551, 5246, 4378, 3470, 3267, 518

- 相似度矩阵计算过程重要数据

```
得到最终的相似度矩阵:  
输出样例:  
C[u][v]: 0.8119379983823597  
movies num or users num: 74  
movies num or users num: 111  
sim_matrix: 0.008958710805466192  
输出样例:  
C[u][v]: 4.756805851963565  
movies num or users num: 74  
movies num or users num: 363  
sim_matrix: 0.02902326410020775
```

## 实验任务3输出

- 产生Top-N推荐 (N=10, K=5)

```
推荐结果:  
需要推荐电影的用户id: 695  
推荐的电影:  
电影id: 2918 电影名称: Ferris Bueller's Day Off (1986)  
电影id: 47 电影名称: Seven (Se7en) (1995)  
电影id: 356 电影名称: Forrest Gump (1994)  
电影id: 318 电影名称: Shawshank Redemption, The (1994)  
电影id: 608 电影名称: Fargo (1996)  
电影id: 587 电影名称: Ghost (1990)  
电影id: 1200 电影名称: Aliens (1986)  
电影id: 2804 电影名称: Christmas Story, A (1983)  
电影id: 1198 电影名称: Raiders of the Lost Ark (1981)  
电影id: 2762 电影名称: Sixth Sense, The (1999)
```

## 实验任务4输出

- 评测模型

N= 10, K=5

```
N=10 K= 5  
100%|██████████| 4697/4697 [09:02<00:00, 6.49it/s]  
Metric: {'Precision': 5.04, 'Recall': 11.82, 'Coverage': 44.7, 'Popularity': 6.989832}
```

N=10, K=10

```
N=10  K= 10
100%|██████████| 4697/4697 [14:52<00:00, 3.45it/s]
Metric: {'Precision': 5.83, 'Recall': 13.69, 'Coverage': 36.2, 'Popularity': 7.124474}
```

(准确率和召回率都有提升；覆盖率稍有下降)

N=10, K=20

```
N=10  K= 20
100%|██████████| 4697/4697 [32:39<00:00, 2.32it/s]
Metric: {'Precision': 6.22, 'Recall': 14.61, 'Coverage': 29.67, 'Popularity': 7.226768}
评测模型:
```

(效果提升很多，说明还能继续提高K来增加拟合能力)

N=10, K=50

```
100%|██████████| 4697/4697 [1:06:10<00:00, 1.05s/it]
Metric: {'Precision': 6.3, 'Recall': 14.8, 'Coverage': 22.43, 'Popularity': 7.340144}
```

(效果提升不多，耗时特多，说明系统性能已经基本稳定在这个结果了，K小一点更合适)