### 4.5.4 冷启动问题

为了解决新项目的冷启动问题，在此我们引入了用户时间权重的概念，通过对用户评价的时间间隔判断用户活跃度并根据用户对新项目偏爱程度等因素创建构建个性化的预测评分公式。由于新项目不存在任何一个用户对它的评价信息，故无法使用协同过滤算法来实现推荐，因此，本章节提出解决该类冷启动问题推荐算法，该算法不仅仅使用用户对物品的评分信息对来进行推荐，而是要综合考虑项目本身的诸多属性信息、标签对项目的标注信息等较为全面的考虑影响推荐系统的诸多因素。实验证明该方式，在解决推荐系统在新项目的冷启动的问题上是较为有效的。

冷启动问题[31]有很多解决方法，本文采用的方法主要是将时间信息加入到推荐算法中。我们都知道，现实世界中，用户群体一般分为两类：一类用户即所谓的活跃用户，这类用户的特点是喜欢追逐并评价新鲜事物；另一类用户即所谓的消极用户，这类用户更多的是追随角色。在这种情况下，积极用户对项目的评价时间往往在项目发布之后很短的一段时间内，而消极用户的反应则要相对滞后。由于这两类用户的普遍存在，推荐系统应用的时候会考虑把新项目优先推荐给积极用户，以此来带动评价。我们同时把时间信息[33]加入到系统中后，冷启动问题也就得到了解决。

由于时间信息对推荐算法有重要的影响，因此我们对用户的时间权重作了如下定义:

时间权重=

其中sum表示用户评价过的项目总数；timeui表示用户评价的第i个项目的时间，datai表示项目i发布的时间。则用户u的时间权重Wu计算公式如下：

(公式4-1)

由此可知，timeui – datai越小，说明项目评价时间与项目发布时间的距离越近，用户越积极。反之亦然。

与实际项目为例，用户评价时间表如下：

**表4.8用户评价时间表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Item | U1 | | U2 | | U3 | |
| Time | Date | Time | Date | Time | Date |
| i1 | 2016 | 2015 | 2015 | 2014 | 2017 | 2017 |
| i2 | 2016 | 2014 | 2016 | 2013 | 2017 | 2014 |
| i3 | 2015 | 2012 | 2017 | 2014 | 2017 | 2016 |

以表一为例，采用公式1计算得到每个用户的时间权重wui即：

根据如上计算可知，用户积极程度由大到小依次是Wu1>Wu3>Wu2，即用户Wu1更偏爱于新发布的产品。

此外我们还根据项目标签信息预测评分值[34]。

1) 基于项目评分信息预测评分值

这里，我们假定有一个用户u，分析用户-项目评分矩阵可以发现用户与项目之间存在评价与未评价，以及评价次数的关系。对于评价与未评价的关系，我们设置一个变量ruj，当ruj的值为1时表示用户评价过该项目，0则表示未做评价。对于评价次数，本文设定以ICk来表示用户评价的个数，UCj表示某一项目被共同评价的次数。在这个关系中，用户评价项目的个数和项目被评价次数可以从用户-项目评分矩阵中得到。根据项目的已评价总数、评分具体信息和用户对项目的评价次数，我们可以定义基于项目评分信息的预测评分值，从而观察和研究他们间的关系。具体公式如下:

(公式4-2)

2) 基于项目属性信息预测评分值

如公式所示表示基于用户u对项目j的预测评分值。如果用户u 对项目 j 做了评价ruj =1, 反之为0。如果项目j具有属性l，则hlj=1，反之为0。

(公式4-3)

3) 基于项目标签信息预测评分值

基于项目标签信息的预测评分计算公式如下：

(公式4-4)

公式(4-4) 中，表示基于标签的用户u对项目j的预测评分值。如果标签g标注的项目j，则bgj=1分之为0。

下面我们基于具体的数据矩阵和公式来围绕研究对象u1计算预测评分值。

**表4.9用户项目评分矩阵**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| rij | i1 | i2 | i3 | i4 |
| u1 | 2 | 0 | 4 | 0 |
| u2 | 0 | 3 | 1 | 0 |
| u3 | 5 | 0 | 2 | 0 |

表4-5 标签-项目标注矩阵

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| bij | i1 | i2 | i3 | i4 |
| t1 | 2 | 3 | 0 | 0 |
| t2 | 0 | 3 | 1 | 0 |
| t3 | 1 | 0 | 0 | 0 |
| t4 | 0 | 2 | 1 | 0 |
| t5 | 0 | 0 | 2 | 0 |

**表4.10属性-项目矩阵** **表4.11 标签-项目标注矩阵**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| hij | i1 | i2 | i3 | i4 |  |  | bij | i1 | i2 | i3 | i4 |
| a1 | 1 | 0 | 1 | 0 |  |  | t1 | 2 | 3 | 0 | 0 |
| a2 | 1 | 1 | 1 | 0 |  |  | t2 | 0 | 3 | 1 | 0 |
| a3 | 1 | 0 | 1 | 1 |  |  | t3 | 1 | 0 | 0 | 0 |
| a4 | 0 | 1 | 0 | 1 |  |  | t4 | 0 | 2 | 1 | 0 |
| a5 | 0 | 1 | 0 | 1 |  |  | t5 | 0 | 0 | 2 | 0 |

其中{i1,i2,…in}为n个项目的集合，{u1,u2…um}表示m个用户，rij表示用户ui对项目j的评分。{a1,a2…,an}表示p个属性的集合，hij表示项目j是否具有an属性。{t1,t2…,tq}表示q个标签的集合,bij表示项目j被标注的次数。为了更好的理解公式与表之间的关系，使用两个三分图的形式来表示。

因此我们可以根据上述理论基础，分别根据用户-项目-标签信息和用户-项目属性信息来挖掘他们之间的潜在关系。从而通过预测评分值来解决新项目的冷启动问题。



**图4-7用户-项目-标签信息 图4-8用户-项目-属性信息**

根据之前的分析发现，结合用户、标签、项目属性提出的个性化推荐算法从理论上可以提高推荐系统的准确率。在此，为了更好的观察用户评分信息、项目属性信息，项目标签信息以及用户评分时效性信息在推荐算法中所产生的影响，我们定义权重, 其中得到预测评分公式如下：

(公式4-5)

由公式4-15可知，该个性化推荐充分利用了用户、标签以及项目属性的相关信息。当, 会综合考虑用户、标签与项目属性信息进行预测评分计算，即用户、标签与项目属性的个性化推荐算法(consider user,tag,and item-attribute personalized recommendation algorithm,简称CUTA)。同理，当时为用户项目属性推荐算法（简称CUA），当时，为用户标签信息推荐算法（简称CUT）。

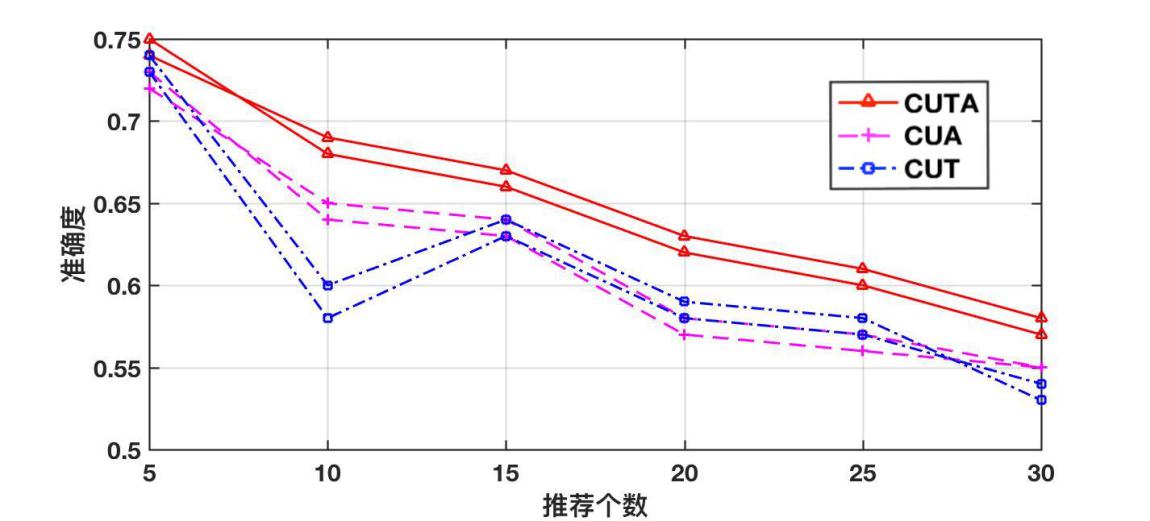
为了解决推荐系统冷启动问题，我们在上述基础上加入了用户时间权重信息，得到CUTATime算法，该算法会在获取基于项目评分(CUTA)、基于标签评分(CUT)、基于项目属性信息(CUA)得到的预测评分值的基础上通过分析用户评价项目的时间信息，获取用户时间权重，并结合之前得到的预测评分实现推荐。

### 6.4.3 冷启动仿真测试

(1) 不考虑新项目的不同推荐算法的准确度对比

为了验证我们提出的冷启动算法的性能，我们对4.5节提出的的三类推荐算法进行了相关的实验对比，我们使用MovieLens数据集，该数据集主要包括用户对电影的评分信息，电影的属性信息以及电影的标签信息等。我们从该数据集中选取至少被30个用户同时评论的电影，以及至少评论过10部电影的用户.经过筛选我们得到2035个用户以及3200部电影并将其50%划分为训练集，另外50%为测试集。

通过调整不同的的取值，并针对近邻个数为5，10，15，20，25，30的情况下进行了相关实验，实验结果如图6-11所示：



**图6-11不同推荐算法的准确度对比**

通过观察我们可以发现，CUTA算法的准确度明显高于CUA和CUT算法，而当邻域个数大于15时，调整的取值对CUA和CUT推荐的准确率并无明显影响。因此我们可以发现，在不考虑新项目的前提下，CUTA的准确率明显优于CUA和CUT算法。

(2) 改善冷启动问题的相关实验对比

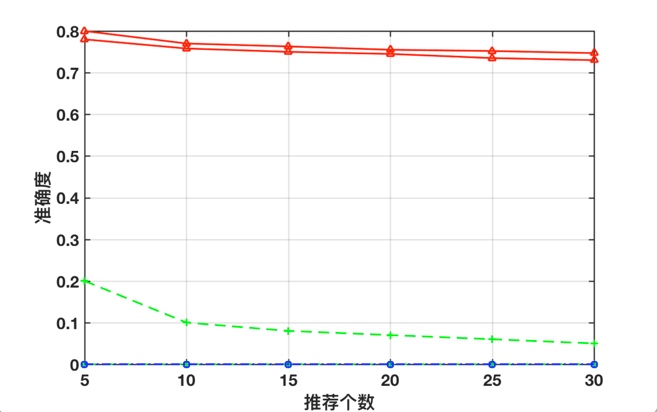
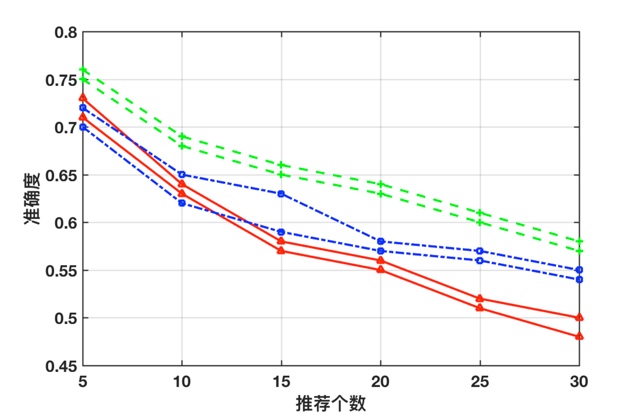
对于冷启动的问题，即通常用户并未曾对项目做出过任何评价，所以在数据集中并不会存在用户对项目的任何评分信息。因此当面临冷启动问题，我们主要会通过项目的基本属性信息和用户的时间权重两项综合来为用户推荐项目。

这里我们借鉴了一个新的概念：新颖度[35]，这里所说的新颖度是指在给目标用户推荐的列表中，新项目的个数占推荐项目的总数的度，即一个比值：

novelty = Recommended\_SUM /SUM (公式6.1)

其中novelty表示新颖度，Recommended\_SUM表示推荐列表中包含的新项目的个数，SUM表示项目总数。

首先我们将实验一中的测试集分成两部分，其中一部分是随机抽取150个项目当作新的项目，同时将训练集中这些对应项目的评分信息和标签信息全都置为0，其余未被抽取的部分为另一部分。的取值和实验一取值相同，并在不同邻域的情况下进行相关实验。



**图6-12解决冷启动问题算法的准确度对比** **图6-13解决冷启动问题算法的新颖度对比**

从实验结果(图6-12，图6-13)可以看出，本文提出的CUTA算法的准确率略高于CUT算法，当近邻值取5左右时，CUTATime的准确度略高于CUT，但是当近邻值大于5时，CUT算法的准确率依然可观。而对于新项目的推荐情况如图6-12，可知CUT算法所推荐的项目的新颖度为0，不能解决冷启动问题，综合考虑，CUTATime算法的新颖度最高，因此在处理新项目冷启动问题时其效果最佳。