**与文献[35 用户间多相似度协同过滤推荐算法]的对比实验**

**基于改进Slope One数据填充的用户间多相似度的协同过滤算法**

A 数据填充

为缓解数据稀疏问题，对用户—项目评分矩阵运用Slope One算法进行数据填充。结合文献[35]对项目进行分类的思想，对传统Slope One进行改进。传统Slope One算法根据用户对不同项目之间的评分线差来预测用户对项目的评分，改进后的Slope Ones算法先将项目分类，然后在当前项目所属的类别中，再来计算用户在当前类中物品的评分线差，并预测出用户对该项目在当前类中的预测评分。同一项目可能属于不同类别，此时需要计算出该项目在所属各类别上的预测评分，最后取它们的加权平均值作为最后的预测评分。

将项目的所有类型表示为集合，即为某一类型，每个项目可以属于集合T中的一种或多种类型。

数据填充算法步骤如下：

Step 1 根据式(1)计算项目i与项目j之间的评分差均值，其中i与j都被评分过，且同属于类型，将该评分差均值记为项目间的评分偏差。

(1)

其中，分别表示用户u对项目i和项目j的评分。U(i)是对项目i评分的用户，那么是对项目i和项目j都评分过的用户的集合，即为对项目i和项目j都评分过的用户数。

Step 2 将Step1中计算得出的同类型项目间的评分偏差结合用户的历史评分，通过式(2)预测用户u在类型上对未评分项目的评分。

(2)

其中，表示用户u在类型上评分过的项目的集合。

以此类推，可以得到用户u在目标项目所属m个类型上，对该项目的所有预测评分。

Step 3 取Step2中目标项目所有预测评分的加权平均值作为用户u对该项目最后的预测评分。

(3)

式中，，表示用户u评分过的属于类型的项目的数量，表示用户u评价过的项目所属类型的集合，，表示用户u评分过的属于类型中任意一种类型的项目数量的综合。很显然，有

(4)

权重反映了用户对于类型在目标项目所属类型集合中的关注程度，权重越大，则用户对于类型的项目评分次数相对于用户对于类型集合中的其他类型项目的评分次数的比重就越高，由此可以反映出用户对于某一类型的喜好程度，预测出的填充值才更真实有效。

Step 4 计算出所有用户对未评分项目的预测评分，并作为填充数据填入用户-项目评分矩阵中。

因为本填充算法是基于项目分类的基础上的，所以极有可能对用户u未评分过的项目未能通过上述方式预测出评分，在此统一采取留空的方式，即这种情况下不进行填充，以减少填充数据的噪点。

B 用户间的多相似度计算

由前可知，将项目的所有类型表示为集合，即为某一类型，每个项目可以属于集合T中的一种或多种类型。通过分别计算用户之间对于不同项目类型的各自的评分相似度，来描述用户对于这n种类型项目的不同喜好程度，更加准确地刻画出用户之间对于不同类型项目的兴趣和口味的异同。

对于任意两个用户x和y，他们关于集合T中各个类型的k个相似度表示为，，···，。

对于用户间多相似度的计算，与传统采用相关相似性或余弦相似性计算方式不同的是，在计算过程中筛选出用户对于该类型的项目的评分记录，在当前类型上计算用户间的相似度。

设集合和分别表示用户x和y评分过的属于类型的项目的集合，其中，他们共同评分过属于类型的项目的集合为，和分别为用户x和用户y对项目i的评分，则他们关于类型的余弦相似度为：

(5)

若‾和‾分别表示x和y对于属于类型的项目的平均评分，用户x和用户y关于类型的相似度还可以用皮尔森相关系数[硕18]计算得到，如下公式所示。就是对比论文中的相关相似度

(6)

以上为两种相似度计算的方式，理论上选一个就好，因为对比的论文实验两种相似度都用上了，所以这两个相似度是都要计算的。

通过以上相似度计算公式可以计算出用户x和用户y在项目类型集合T上不同类型上各自独立的评分相似度。

C 基于用户间多相似度的预测评分

由步骤B计算出用户间关于不同类型项目的相似度，计算出各类相似度的加权平均值，作为用户间的相似度。

用户间余弦相似度可表示为：

(7)

用户间皮尔森相关系数可表示为：

(8)

这个与B中两个不一样的相似度计算方式相对应。

式中，，表示用户x和用户y共同评分过的属于类型的项目的数量，表示用户x和用户y评分过的属于所有项目类型集合T中的任意一种项目的数量（就是用户X和Y共同评分过的项目的类型总数量），由此可见，

权重反映了用户x和用户y对某一类型项目的关注程度，值越大，则关注度越高。

计算出用户间的相似度之后，通过k-means方法来对用户进行聚类。聚类是数据挖掘中的经典问题，它的目的是将数据对象分成多个类，使得同一个聚类中对象之间具有较高的相似性，而不同聚类中的对象差别较大[硕47]。K-means聚类算法属于基于划分的聚类算法中最常用的一种。其主要思想是：首先随机选择k个节点作为聚类中，(K值会影响聚类效果)然后按照最近邻原则把剩余待聚类节点分配到各个聚类中，通过计算聚类中各节点间距离与其他节点平均距离最小的节点重新作为聚类中心，一直这样重复执行，直到聚类中心不再变化或聚类次数达到要求为止。

在与目标用户x相似性较高的若干个聚类中选择大于某一阈值的用户作为目标用户x的近似邻居集合N。

根据近似邻居集合N中用户的评分利用下式计算出目标用户u对未评分项目的预测评分值。

(9)

(10)

(9)(10)两式也根据B中选择的相似度计算的式子加以区分。（9）中用了余弦相似度，（10）中用了皮尔逊相关系数，就是对比论文中的相关相似度。

其中，为目标用户的近似邻居集合，和表示用户x和用户y对项目的平均评分。

D 实验对比

针对相关相似性和余弦相似性两种不同的用户相似性度量方法，分别进行了两组实验，每组实验分别从数据集中选择100、500和1000个用户的评分记录作为测试数据集，并进一步在这个测试数据集中随机抽取80%的评分记录作为训练集，另外20%作为测试集。采用平均绝对偏差MAE。统计精度度量方法中的平均绝对偏差MAE（Mean Absolute Error）易于理解，可以直观地对推荐质量进行度量，是最常用的一种推荐质量度量方法。MAE通过计算预测的用户评分与实际的用户评分之间的偏差度量预测的准确性，MAE越小，推荐质量越高。

设预测的用户评分集合表示为，对应的实际用户评分集合为，则平均绝对偏差MAE定义为

**MovieLens数据集**

下载网址

<https://blog.csdn.net/qq_22562651/article/details/80485329>

相关介绍

<https://www.letiantian.me/2014-11-20-introduce-movielens-dataset/>

实验使用的测试数据集是GroupLens研究产品组（http://www.grouplens.org）提供的一个著名电影评分数据MovieLens，它是有10万条记录的数据集，记录了943个用户对1682部电影的评分，每个用户至少对20部电影进行了评分，评分值范围为[1,5]。5表示“perfect”（非常好），而“1”表示“poor”（差），用户通过对不同电影上的不同评分表达了自己的兴趣。用户和产品的评分矩阵密度为

说明此数据集的评分矩阵是相当稀疏的。

关于实验参数的选择见对比论文[用户间多相似度协同过滤算法]的第4节，实验结果与分析。