结合用户聚类和评分偏好的推荐算法*

高茂庭，段元波
(上海海事大学 信息工程学院，上海 201306)

摘 要： 针对推荐算法中用户评分矩阵维度高、计算量大的问题，为更加真实地反映用户本身评分偏好，提出一种结合用户聚类和评分偏好的推荐算法。先利用PCA降维和K-means聚类对用户评分矩阵进行预处理，在最近邻选取方法上，添加用户共同评分数量作为约束，利用用户和相似簇的相似度对相似簇内评分加权求和生成基本预测评分；再综合用户评分偏置和用户项目类型偏好，建立用户评分偏好模型；最后通过多元线性回归确定每部分的权重，生成最终的预测评分。对比实验结果表明，新算法能更真实地反映用户评分，有效减少计算量并提高推荐系统的预测准确率，更好地满足用户对于推荐系统的个性化需求。

关键词： 协同过滤；降维；聚类；用户偏好；推荐系统
中图分类号： TP301.6 **文献标志码：** A **文章编号：** 1001-3695(2018)08-2260-05
doi:10.3969/j.issn.1001-3695.2018.08.005

Recommendation algorithm based on user clustering and rating preference

Gao Maoting, Duan Yuanbo
(*College of Information Engineering, Shanghai Maritime University, Shanghai* 201306, *China*)

Abstract: To solve the problem of high dimensionality and computational complexity of the user scoring matrix in recommendation system and reflected the user's preference more realistically, this paper proposed a recommendation algorithm based on user clustering and user rating preference. Firstly, it used the PCA dimensionality reduction and K-means clustering to preprocess user rating matrix, used the number of user common rating items as the constraint in the nearest neighbor selection and used the similarity between user and similar cluster to sum weighted scores and generated a basic prediction rating. Secondly, it established the user scoring preference model by combining the user rating bias and the user item type preference. Finally, it used multiple linear regression to determine the weight of each component and obtained the final prediction score. The experimental results show that the new algorithm can reflect the user rating more accurately, reduces the computational complexity and improves the prediction accuracy of the recommendation system effectively, and meets the user's personalized requirements for the recommendation system better.

Key words: collaborative filtering; dimensionality reduction; clustering; user preference; recommendation system

0 引言

随着信息技术的迅猛发展，网络数据呈现出爆炸式的增长趋势，人们从信息匮乏时代进入了信息过载时代。作为一种能够有效解决信息过载问题的手段，推荐系统受到了越来越多的关注，各大电子商务和社交网站，如Amazon、淘宝网、微博等都不同程度地使用推荐系统为用户进行推荐。现有流行推荐算法主要包括协同过滤推荐算法、基于内容的推荐算法、混合推荐算法以及基于用户—产品二部图网络结构的推荐算法。其中，协同过滤推荐算法[1]是使用最广泛、最成功的推荐算法。Tapestry[2]邮件过滤系统被认为是最早提出的协同过滤推荐系统。

协同过滤推荐算法的核心思想就是寻找与目标用户有相似兴趣的近邻用户，根据近邻用户的信息产生推荐。但由于用户和项目数量往往比较大，算法存在数据稀疏性和计算量大[3]的问题，为此，很多学者提出了一些新的方法进行改进。Goldberg等人[4]提出利用主成分分析法以及形式化符号的表示方法将用户描述文件映射为阶数是评分等级的方阵，为用户评分矩阵的降维提供了新思路。You等人[5]提出了一种结合项目聚类和slope one方案的推荐算法，使用项目聚类算法将项目分组到几个群集，并在每个群集中应用slope one方案，以预测目标用户的未知项目的评分等级。Frémal等人[6]提出了一种基于项目元数据信息的聚类方法，算法根据项目类型进行聚类，由于项目可以有多种类型可以放置在几个集群中，所以每个集群提供自己的评级预测，然后使用加权策略将这些结果合并成一个最终的预测结果。Guo等人[7]提出了一种基于多视角聚类的推荐算法，用户从评级模式和社会信任关系的观点进行迭代聚类，同时使用支持向量机理论，根据用户、项目和预测相关的特征来确定给定项目的预测评分。邓爱林等人[8]提出了基于项目聚类的协同过滤推荐算法，通过用户对项目评分的相似性对项目进行聚类，通过相似簇搜索目标项目的最近邻，有效提高了推荐系统的实时响应速度。黄创光等人[9]提出了自适应选择近邻对象的推荐群以及信任子群，通过不确定近邻的动态方法对预测评分进行计算，可有效缓解近邻值不确定性问题。

上述对协同过滤推荐算法的改进大多是根据目标用户最近邻信息来产生推荐，却没有充分考虑用户本身的评分偏好，有时难以给出用户真正满意的推荐。因此，本文提出一种结合用户聚类和评分偏好的推荐算法，对最近邻选取和预测评分生成方法进行改进，同时结合用户本身的评分偏好，以更准确地刻画用户对项目的真实喜好，降低推荐算法计算量并提高推荐的质量。

1 传统协同过滤推荐算法

协同过滤推荐算法是以“物以类聚，人以群分”原则为基础，即若当前用户与大多数用户对项目的评分数据相似，那么他对未知的项目也会有类似的项目评分。算法流程如图1所示。

收稿日期：2017-04-10；修回日期：2017-05-15 基金项目：国家自然科学基金资助项目(61202022)
作者简介：高茂庭(1963-)，男，江西九江人，教授，博士，主要研究方向为智能信息处理、数据库与信息系统(mtgao@163.com)；段元波(1993-)，男，山东临沂人，硕士，主要研究方向为数据仓库与数据挖掘、数据库与信息系统.

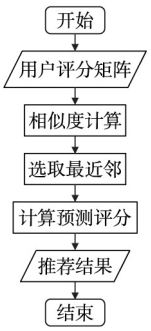

图1　传统的协同过滤推荐算法流程

1.1　用户评分矩阵

在协同过滤推荐系统中，用户对项目的评分数据以用户评分矩阵 $R=(r_{ij})_{m\times n}$ 表示，其中包含 m 个用户的集合 user = $\{u_1,u_2,\cdots,u_m\}$ 和 n 个项目的集合 item = $\{it_1,it_2,\cdots,it_n\}$，$r_{ij}$ 表示用户 i 对项目 j 的评分等级。

1.2　相似度度量方法

用户间相似性计算是协同过滤推荐算法的关键之一，通常相似度度量有以下几种方法: 余弦相似性、修正的余弦相似性和皮尔逊相关相似性。Herlocker 等人[8] 通过比较发现，在基于用户的推荐系统中，皮尔逊相关相似性更关注用户共同评价过的项目，比其他相关性度量方法更胜一筹，皮尔逊相关系数计算如下:

$$\mathrm{sim}(x,y)=\frac{\sum_{i\in I_{xy}}(r_{xi}-\bar{r}_x)(r_{yi}-\bar{r}_y)}{\sqrt{\sum_{i\in I_{xy}}(r_{xi}-\bar{r}_x)^2(r_{yi}-\bar{r}_y)^2}} \tag{1}$$

其中: I_{xy} 表示用户 x 与 y 共同评分过的所有项目的集合; $\bar{r}_x$ 和 $\bar{r}_y$ 表示用户 x 和用户 y 对所有项目评分的平均值; r_{xi} 和 r_{yi} 分别表示用户 x 和用户 y 对项目 it_i 的评分。

1.3　传统预测评分计算方法

预测评分直接反映目标用户对当前项目的喜好程度，其计算建立在与目标用户相似近邻用户对该项目的真实评分基础上。传统协同过滤推荐算法在选择目标用户的最近邻时一般采用 K 近邻算法，即按照相似度由高到低选择 k 个用户作为其近邻。设集合 N_a 表示目标用户 U_a 的最近邻集合，则目标用户对未评分项目 it_i 的预测评分 r_{ai} 采用式(2) 计算。

$$r_{ai}=\bar{r}_a+\frac{\sum_{j\in N_a}\mathrm{sim}(U_a,U_j)(r_{ji}-\bar{r}_j)}{\sum_{j\in N_a}\mathrm{sim}(U_a,U_j)} \tag{2}$$

其中: $\bar{r}_a$ 和 $\bar{r}_j$ 分别为用户 U_a 和 U_j 所有已有评分的平均值; $\mathrm{sim}(U_a,U_j)$ 为用户 U_a 和 U_j 的相似度。

2　结合用户聚类和评分偏好的推荐

传统协同过滤推荐算法在实际应用场景中往往存在一些不足: a) 推荐系统中用户和项目的数量十分庞大，数据维度很大，但单个用户有过评分的项目却较少，在进行相似度计算时需要在整个用户空间上进行，计算量非常大; b) 仅仅依据目标用户最近邻的评分数据产生预测，而没有考虑用户本身评分偏好情况，例如，不同的用户之间存在不同的评分标准，而且同一用户对不同的项目类型也存在不同喜爱情况。对于个性化推荐而言，用户本身偏好是不可忽视的。

针对数据纬度高且稀疏、计算量大的问题，文献[4] 利用 PCA 技术对数据进行降维处理，但在计算相似度时需要在整个用户空间上进行，计算量依然较大; 文献[5,10,11] 利用聚类技术有效地解决了计算量大的问题，但在预测评分时却忽视了用户与相似簇之间相似度的影响，而且在选取最近邻时没有考虑到彼此之间共同评分数量等因素。针对用户本身评分偏好问题，文献[9] 从用户本身评分数据出发进行推荐，但当用户评分记录较少时，该方法会存在很大的偏差。

为此，本文提出结合用户聚类和评分偏好的推荐算法，针对数据维度高、计算量大的问题，利用 PCA 技术和 K-means 聚类技术对原始数据进行预处理，降低数据维度、减少计算次数; 同时在最近邻的选取上添加共同评分项目数量作为约束条件，并考虑用户与相似簇之间相似度的影响，根据最近邻和相似簇生成一个基本预测评分。在用户评分偏好上，用已有评分记录去挖掘用户评分偏置和用户项目类型偏好。算法流程如图 2 所示。

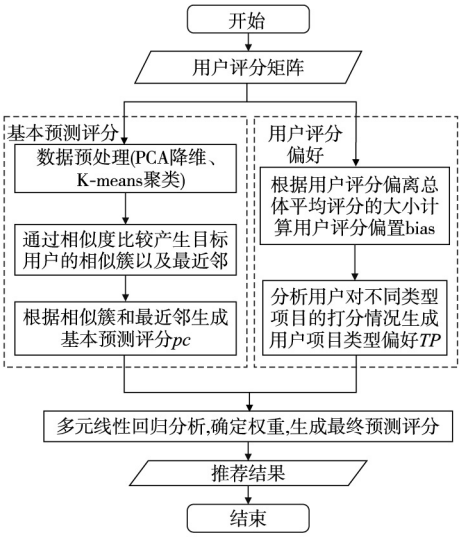

图2　结合用户聚类和评分偏好的推荐算法流程

2.1　基本预测评分

2.1.1　数据预处理

推荐系统中用户和项目数据十分庞大，导致用户—项目评分矩阵维度非常高，降维是解决维度灾难的有效方法，PCA 算法是一种常用的降维技术，借助正交变换实现只需要少数新变量就能解释原始数据中的大部分信息。该算法根据保留原始数据的信息比例来确定主成分的个数，一般而言，选取原始数据 95% 的信息量就可以达到很好的降维效果。因此，为了降低用户评分矩阵的维度，先利用 PCA 技术对原始用户评分矩阵进行降维处理，得到一个维度较低的矩阵，再对降维后的数据矩阵采用 K-means 进行聚类处理，生成 k 个簇 cent = $\{c_1,c_2,\cdots,c_k\}$，其中，每一个簇都用其聚类中心表示。

2.1.2　目标用户的相似簇以及最近邻的选取

经过数据预处理后，评分矩阵的维度得到控制，同时在选取目标用户最近邻时不再需要比较全部用户，只需计算其与每个簇中心间的相似度，将相似度最高的前 α 个簇定义为目标用户的相似簇集 $\mathrm{clust}_a=\{c_1,c_2,\cdots,c_\alpha\}$，由此，目标用户只需与相似簇集中用户进行比较从中选择出最近邻，从而大大降低了计算次数。

为了防止目标用户与选取的最近邻可能仅仅对很少的一部分项目共同有过评分现象发生，在选取目标用户的最近邻时与前面介绍的传统 K 近邻选取方法不同，不再仅仅根据用户相似度，同时还考虑了两者之间共同评分的项目数量。本文对 K 近邻选取方法进行改进，首先从目标用户 U_a 的相似簇集 clust_a 中筛选出相似度大于规定阈值 γ 的用户组成目标用户的候选近邻集。

$$C(U_a)=\{U_x\mid \mathrm{sim}(U_a,U_x)>\gamma, a\neq x, U_x\in \mathrm{clust}_a\}$$

然后对目标用户候选近邻集中的用户按照有共同评分项目的数量进行排序，选择前 k 个用户作为其最近邻。

$$NN(U_a)=\{U_1,U_2,\cdots,U_k,\forall U_i\in \mathrm{clust}_a\}$$

2.1.3　生成基本预测评分

在得到目标用户 U_a 相似簇集 clust_a 和最近邻 $NN(U_a)$ 后，就可以对目标用户未评分的项目生成一个基本预测评分。由于不同的簇和目标用户之间的相似度就代表了簇内的用户和目标用户的整体相似情况，所以相似度高的簇内用户所占的比重就应该较高。因此在生成基本预测评分时，不能仅仅考虑用户之间的相似度，还需要综合考虑与相似簇之间相似度的影响。

已知目标用户的每个相似簇中都包含其若干个最近邻，首先对每一个相似簇 c_i 中的簇内最近邻，利用式(2)来生成一个簇内评分 G_i，然后根据每个相似簇与目标用户的相似度作为权重对这些簇内评分进行加权求和，从而生成基本预测评分 pc_{ai}：

$$pc_{ai} = \frac{\sum_{s=1}^{\alpha} G_s \times \mathrm{sim}(U_a, c_s)}{\sum_{s=1}^{\alpha} \mathrm{sim}(U_a, c_s)} \tag{3}$$

其中：G_s 为目标用户根据第 s 个相似簇中最近邻生成的簇内评分；$\mathrm{sim}(U_a, c_s)$ 为目标用户 U_a 与簇 c_s 的相似度。

2.2 用户评分偏好建模

上述操作可以生成目标用户对未评分项目的一个基本预测评分，但是这个评分仅仅是依靠目标用户的最近邻信息来生成的，而没有考虑到用户本身的一些影响，因此本文从用户本身角度出发，通过建立用户偏好模型来修正基本预测评分的结果，以达到提高推荐准确性的目的。本文对用户偏好模型从用户评分偏置和用户项目类型偏好程度两方面进行考虑。

2.2.1 用户评分偏置

每个用户都有自己的评分准则，有些用户评分比较严格，往往给项目评较低的分，另一些用户却相反，对项目评分总是较高。例如，图 3 所示是 MovieLens [13] 数据集中的两名用户实际打分情况分布图。

图 3 中横轴表示不同的评分等级，纵轴表示该评分等级数量占总体评分的比重。可以看出用户 1 的评分主要分布在 3 分以下，占据了 75% 左右，而用户 2 的评分却和用户 1 截然相反。说明用户 1 的评分准则比较宽松，而用户 2 的评分则比较严格，因此，在对目标用户进行评分预测时，需要考虑到各个用户本身的评分准则，本文定义其为用户评分偏置。现在需要对用户评分偏置进行准确的建模，从而可以更好地模拟用户真实的打分标准。总体用户的评分平均值代表了用户评分的集中趋势，反映了总体用户的评分准则，可以以此作为评判用户偏置的标准，因此本文中将用户评分偏置定义为目标用户已有打分偏离总体用户平均打分的距离和，计算步骤如下：

a) 计算全部用户已有评分的平均值 μ：

$$\mu = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}}{|N|} \tag{4}$$

其中：N 为有用户评分项目数量总和。

b) 通过累加目标用户 U_a 已有评分与 μ 的差值来计算目标用户的评分偏置：

$$\mathrm{bias}_a = \frac{\sum_{i=1}^{t}(r_{aj} - \mu)}{|t|} \tag{5}$$

其中：t 为目标用户已有评分项目数量。

为了证明用户评分偏置值的有效性，本文在 MovieLens 数据集上进行了统计分析，依次计算了每个用户的评分偏置，绘制了全部用户评分偏置分布图，如图 4 所示。

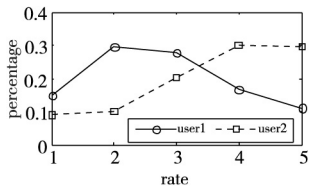

图3 用户评分分布对比图

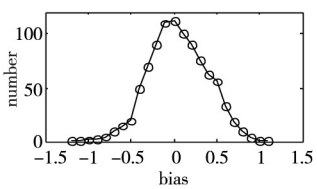

图4 用户评分偏置结果分布图

从图 4 中可以看出用户的评分偏置是符合正态分布的，说明具有有效性，用户评分偏置的均值和方差计算证明如下：

$$E(\mathrm{bias}) = \frac{1}{N}\sum_{a=1}^{N}\mathrm{bias}_a = \frac{1}{N}\sum_{a=1}^{N}\left(\frac{\sum_{i=1}^{t}(r_{aj} - \mu)}{|t|}\right) = \frac{1}{N}\left(\sum_{a=1}^{N}\sum_{i=1}^{t} r_{aj} - N\mu\right) = 0$$

$$\mathrm{var}(\mathrm{bias}) = \frac{1}{N-1}\sum_{a=1}^{N}(\mathrm{bias}_a - \mu)^2$$

2.2.2 用户项目类型偏好

每个项目均有各自的类型属性，比如一部电影的类型可分为剧情、爱情、动画等，项目的类型属性往往由领域专家确定。而不同的用户对于不同的项目类型的喜爱程度是不同的，例如图 5 是随机挑选了 MovieLens 数据集中的一个用户，对该用户评分过的项目按照类型进行统计分析。

图 5 中横轴表示不同的类型，纵轴表示评分的数量，实线表示该类型对应评分大于等于 3 分的数量，虚线表示该类型对应评分小于 3 分的数量。可以发现该用户对于类型 1、5、8、15、17 有观看数量较多比较明显的喜爱倾向，而且普遍打分较高，但是对于类型 9、10、12、13 观看的数量很少，说明该用户在观看时倾向于选择自己喜欢的类型去观看，因此在进行推荐时应多推荐用户感兴趣的类型项目。考虑到用户对于自己喜欢的类型应该有相对较多的评分记录，可以从这些评分中挖掘其偏好情况。因此本文从用户已评分项目和项目类型进行分析建模，根据每个用户已有评分的项目建立项目所属类型矩阵，通过计算用户对不同类型的评分数量以及评分等级确定出该用户对每种项目类型的偏好程度，这样做可以通过概率大小直观地计算出用户对于不同项目类型的实际偏好，具体步骤如下。

设项目集中的项目共分为 s 个类型，用集合 $T = \{t_1, t_2, \cdots, t_s\}$ 表示，一个项目可以同时属于多个类型，对于一个项目 it_i，其类型向量表示为 $V_{it_i} = \{h_{i1}, h_{i2}, \cdots, h_{is}\}$，其中，当项目属于类型 t_j 时，$h_{ij} = 1$，否则，$h_{ij} = 0$。若用户 U_a 评分过的项目总数为 w，则用户 U_a 已经评分的项目类型集合可以用一个 $w \times s$ 矩阵 H_a 表示：

$$H_a = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1s} \\ h_{21} & h_{22} & \cdots & h_{2s} \\ \vdots & \vdots & & \vdots \\ h_{w1} & h_{w2} & \cdots & h_{ws} \end{bmatrix}$$

由于每一列都对应同一种类型的项目，对某一列进行求和结果即为用户评价过该类型的数量，所以，通过上面矩阵可以计算出用户 U_a 所有评分过的项目中任意类型属性 t_j 所占的比例 $P(t_j)$：

$$P(t_j) = \frac{\sum_{i=1}^{w} h_{ij}}{|w|} \tag{6}$$

于是，可以确定用户 U_a 对属于类型 t_j 的项目评分为 r 等级的比例 $P_a(r|t_j)$ 为

$$P_a(r|t_j) = \frac{\sum_{i=1}^{w}(h_{ij} \cap \mathrm{rating}(it_i) == r)}{|w|} \tag{7}$$

其中：$\mathrm{rating}(it_i)$ 为用户对项目 it_i 的实际评分。根据式(7)可以依次计算出用户 U_a 对每种项目类型属性打分值的比重，从中选取比重最高的评分作为用户 U_a 对项目类型 t_j 的倾向评分 $M_a(t_j)$。对于待评分项目 it_i，计算到其类型向量 V_{it_i}，则用户 U_a 对该项目的类型偏好 TP_i 计算如下：

$$TP_i = \sum_{j=1}^{s} h_{ij} \times p(t_j) \times M_a(t_j) \tag{8}$$

2.3 多元线性回归确定权重系数产生推荐

经过式(3)(5)(8)的计算，分别生成了基本预测评分、用户评分偏置、用户项目类型偏置，之后就可以结合用户偏好模型来修正基本预测评分从而产生最终的预测评分。因此将式(3)(5)(8)结合起来，在计算目标用户 U_a 对未评分项目 it_i 预测评分时使用式(9)进行计算。

$$p_{ai} = pc_{ai} + \lambda_1 \mathrm{bias}_a + \lambda_2 TP_i \tag{9}$$

其中：pc_{ai} 表示根据目标用户的最近邻对目标项目的基本预测评分；bias_a 表示用户评分偏置；TP_i 表示用户项目类型偏好；λ_1、λ_2 为权重系数。

因此最后需要确定式(9)中的权重系数 λ_1、λ_2 来生成最终的预测评分。已知现有的用户评分矩阵中已经存在了部分用户评分记录，从已有评分中选择一部分作为训练集。分别计算训练集中每个数据的 pc、bias、TP 三个分量，然后利用多元线性回归模型，确定权重系数 λ_1、λ_2，从而计算出预测评分，产生推荐结果。

本文所提出的算法，首先利用 PCA 和 K-means 技术对数据进行了预处理，其中 PCA 进行降维时间复杂度较高，为

$O(n^3)$，聚类操作的时间复杂度为 $O(nkt)$，其中 n 为对象总数，k 为簇数，t 为迭代次数。式(3)主要根据目标用户的相似簇集和簇内评分进行加权求和，时间复杂度为 $O(\alpha n)$，其中 α 表示相似簇的个数。式(4)(5)主要通过计算目标用户的评分与整体评价评分之间的偏离程度来计算用户的评分偏置。式(8)通过遍历用户已有评分数据，找出每个类别的比例从而计算出项目类型偏好，这部分的时间复杂度为 $O(n^2)$。但是一个用户所属的簇和兴趣爱好一般是相对稳定的，因此降维、聚类和兴趣模型部分可以定期离线完成。

3　实验结果及分析

3.1　数据集和实验环境

为了评价所提出算法的性能，采用在推荐系统领域著名的 MovieLens 100K[12] 数据集，该数据集是由美国明尼苏达大学的 GroupLens 小组提供并维护的，它包含 943 名用户对 1 683 部电影的 10 000 条评分记录。

本文的实验环境为 Windows 7 操作系统，Intel Core i5 处理器和 4 GB 内存，代码使用 Python 语言实现。

3.2　实验的度量标准与设计

本文使用的度量标准是平均绝对误差(mean absolute error，MAE)，它是一种常见的推荐准确性度量方法，可以对推荐质量作直观度量，MAE 越小，表示推荐的质量就越高，准确性越好。MAE 评价准则如式(10)所示。

$$\mathrm{MAE}=\frac{\sum_{i=1}^{N}|p_i-r_i|}{|N|} \tag{10}$$

其中：p_i 表示对项目 i 的预测评分；r_i 表示该项目的实际评分；N 表示进行预测的项目数量。

本文实验设计目的是确定算法中的参数以及和其他推荐算法进行对比来检验本文算法的推荐质量，因此本文分别进行了最佳聚类数量的实验来确定 K-means 算法中合适的 k 值；最佳近邻数量实验来确定最近邻选取的合适数量以及与其他推荐算法进行对比实验。

3.3　最佳聚类数目的确定实验

本实验的目的是为了寻找算法中进行 K-means 聚类时合适的 k 值，依次选择 $k=5$，8，10，13，15 进行实验，实验结果如表 1 所示。

表 1　不同聚类数目下的 MAE 值

聚类数	5	8	10	13	15
MAE	0.812	0.805	0.781	0.796	0.804

通过表 1 发现，当 $k=10$ 时 MAE 值最小，此时效果比较好，考虑到该数据集有 19 个类别，所以建议聚类数 k 值为类别的一半向上取整比较合适，因此在下面实验中会选择 $k=10$ 进行实验。

3.4　最佳近邻数确定实验

本实验的目的是测试在不同的最近邻数量下，算法的 MAE 值变化情况。实验中固定聚类数量 $k=10$，依次选取最近邻数量为 5、10、15、20、25、30。实验结果如图 6 所示，它描述了在不同的近邻数量下的 MAE 值。

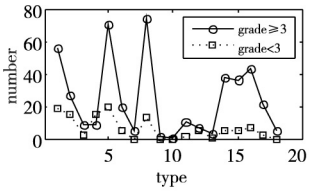

图5　用户对不同项目类型评分分布图

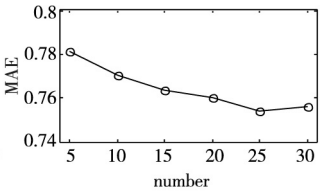

图6　不同近邻数下的MAE值变化图

由图 6 可以看出，随着选择的最近邻数量的增多，MAE 值在逐渐降低，当最近邻数量为 25 时，MAE 值表现最佳。考虑到最近邻数量过多可能会加入一些相似度较低的用户，因此最近邻数量在 25 ~ 30 比较合适。

3.5　结合评分偏好推荐与基本预测评分进行推荐对比实验

本实验主要结合用户评分偏好的推荐结果与 2.1 节提出的基本预测评分结果进行比较，目的是为了验证结合用户偏好模型对推荐的作用，实验在生成基本预测评分时相似度阈值为 0.4，选取聚类数量和最近邻数量为 10，分别从实验数据集中选取了 50、100、150、200 名用户进行实验。结果如图 7 所示。

图 7 中实线表示结合用户评分偏好的推荐结果，虚线为 2.1 节生成的基本预测评分结果。可以看出，在不同的用户数量下结合了用户评分偏好的算法的 MAE 值均低于相同实验条件下基本预测评分的结果。实验表明通过结合用户评分偏好可以获得更高的预测准确性，起到了积极的促进作用。

3.6　与其他协同过滤推荐算法对比实验

3.6.1　MAE 对比实验

下面将本文提出的结合用户评分偏好的推荐算法与传统基于项目的协同过滤算法(item-based collaborative filtering，IB)，同时选取了文献[6，7，13]所提的算法(分别简写为 WS、ICCS、PT)与本文算法(UCSP)进行对比实验，本文算法在生成基本预测评分时相似度阈值为 0.4，选取聚类数量 $k=10$，分别选取近邻数为 5、10、15、20、25、30 进行对比，实验结果如图 8 所示。

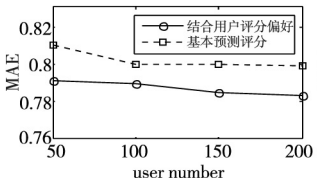

图7　结合用户评分偏好推荐与基本预测评分结果对比结果

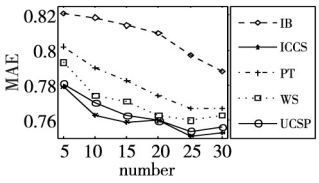

图8　本文算法与其他推荐算法对比实验结果

图 8 中横轴表示最近邻的数量，纵轴表示对应的 MAE 值。可以看出，随着最近邻数量的增加，MAE 的值都是在减少的。但本文算法相较于传统的和基于项目(IB)的推荐算法而言，结合用户评分偏好的算法都可以获得更低的 MAE 值，推荐的效果更好。与其他改进的算法相比，本文算法推荐质量虽然略低于文献[7]中的提出的改进算法，相比于其他的改进算法在推荐质量上可以达到更加准确的效果，说明结合用户偏好对推荐质量的提升起到了积极效果。表 2 是在不同的近邻 k 下本文算法对传统基于用户(UB)和基于项目(IB)的推荐算法的 MAE 提升值。

表 2　不同 k 值下本文算法提升值

算法	10	15	20	25	30	ave
UB	0.078	0.081	0.084	0.085	0.081	0.08
IB	0.05	0.053	0.059	0.061	0.053	0.05

3.6.2　算法时间效率对比实验

本实验主要考察所提出的结合用户评分的推荐算法的时间效率，随机选取 100 名、200 名、500 名用户的未评分项目进行预测实验，在相同的实验环境下与基于用户的协同过滤推荐进行运行时间对比，实验结果如表 3 所示。

表 3　算法运行时间对比

用户数量	100	200	500
基于用户协同过滤运行时间/s	63.01	137.74	374.84
本文所提算法运行时间/s	21.41	57.21	217.65

从表 3 实验结果可以很明显地看出，本文所提的算法在运行时间效率上比传统基于用户的协同过滤推荐算法在时间上要快很多，这也说明本文算法在计算复杂度上明显好于传统的协同过滤推荐。

4　结束语

推荐系统作为一个热门研究领域，既能帮助用户更好地使用互联网信息，又能提高用户忠诚度和推广产品。本文从降低数据维度和减少计算量，提高推荐质量出发，提出了结合用户聚类和评分偏好的个性化推荐算法。先利用降维和聚类技术

对数据进行预处理，改进最近邻选取方法，根据相似簇和最近邻生成基本预测评分，从而降低了数据的维度，并且有效地减少了计算量；同时建立了用户兴趣模型，从用户评分偏置和用户项目类型偏好两方面出发进行建模；最后通过多元线性回归确定每部分的权重，生成最终推荐结果。算法对不同用户进行针对性分析，充分结合了用户本身的评分偏好，实验结果表明，新算法对推荐系统的准确性起了积极的作用。

协同过滤推荐算法是推荐系统研究的核心内容，随着机器学习和数据挖掘技术的发展，在不同的应用场景下涌现出越来越多的新思路。例如，信任网络和结合时间属性的模型的应用。同时，也有很多研究者通过情感分析和领域知识对推荐算法进行改进，使得推荐算法不断向前发展。

参考文献：

[1] Herlocker L, Konstan A, Borchers S A, *et al*. An algorithmic framework for performing collaborative filtering [C] // Proc of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 1999: 230–237.

[2] Goldberg D, Nichols D, Oki B M, *et al*. Using collaborative filtering to weave an information tapestry [J]. Communications of the ACM, 1992, 35(12): 61–70.

[3] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions [J]. IEEE Trans on Knowledge & Data Engineering, 2005, 17(6): 734–749.

[4] Goldberg K, Roeder T, Gupta D, *et al*. Eigentaste: a constant time collaborative filtering algorithm [J]. Information Retrieval Journal, 2001, 4(2): 133–151.

[5] You Haipeng, Li Hui, Wang Yunmin, *et al*. An improved collaborative filtering recommendation algorithm combining item clustering and slope one scheme [C] //Lecture Notes in Engineering & Computer Science, vol 2215. 2015: 313–316.

[6] Frémal S, Lecron F. Weighting strategies for a recommender system using item clustering based on genres [J]. Expert Systems with Applications, 2017, 77(7): 105–113.

[7] Guo Guibing, Zhang Jie, Yorke-Smith N. Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems [J]. Knowledge-Based Systems, 2015, 74(1): 14–27.

[8] 邓爱林，左子叶，朱扬勇. 基于项目聚类的协同过滤推荐算法 [J]. 小型微型计算机系统，2004，25(9)：1665–1670.

[9] 黄创光，印鉴，汪静，等. 不确定近邻的协同过滤推荐算法 [J]. 计算机学报，2010，33(8)：1369–1377.

[10] Herlocker J. Clustering items for collaborative filtering [C] // Proc of ACM SIGIR Workshop on Recommender Systems. New York: ACM Press, 1999.

[11] Beutel A, Beutel A, Ahmed A, *et al*. ACCAMS: additive co-clustering to approximate matrices succinctly [C] //Proc of the 24th International Conference on World Wide Web. Switzerland: International World Wide Web Conferences Steering Committee, 2014: 119–129.

[12] MovieLens_100K [DB/OL]. https://grouplens.org/datasets/movielens/.

[13] Forsati R, Barjasteh I, Masrour F, *et al*. PushTrust: an efficient recommendation algorithm by leveraging trust and distrust relations [C] //Proc of the 9th ACM Conference on Recommender Systems. New York: ACM Press, 2015: 51–58.

[14] Sarwar B, Karypis G, Konstan J, *et al*. Item-based collaborative filtering recommendation algorithms [C] //Proc of International Conference on World Wide Web. New York: ACM Press, 2001: 285–295.

[15] Altingovde I S, Subakan Ö N, Ulusoy Ö. Cluster searching strategies for collaborative recommendation systems [J]. Information Processing & Management, 2013, 49(3): 688–697.

[16] 王茜，杨莉云，杨德礼. 面向用户偏好的属性值评分分布协同过滤算法 [J]. 系统工程学报，2010，25(4)：131–138.

[17] 郭均鹏，赵梦楠. 面向在线社区用户的群体推荐算法研究 [J]. 计算机应用研究，2014，31(3)：696–699.

[18] 崔春生. 推荐系统中显式评分输入的用户聚类方法研究 [J]. 计算机应用研究，2011，28(8)：2856–2858.

[19] 范波，程久军. 用户间多相似度协同过滤推荐算法 [J]. 计算机科学，2012，39(1)：23–26.

[20] 黄震华，张佳雯，田春岐，等. 基于排序学习的推荐算法研究综述 [J]. 软件学报，2016，27(3)：691–713.

(上接第 2244 页)

[8] 冯剑红，李国良，冯建华. 众包技术研究综述 [J]. 计算机学报，2015，38(9)：1713–1726.

[9] Chen Zhao, Fu Rui, Zhao Ziyuan, *et al*. gMission: a general spatial crowdsourcing platform [J]. Proceedings of the VLDB Endowment, 2014, 7(13): 1629–1632.

[10] DiDi [EB/OL]. [2017-06-05]. http://www.xiaojukeji.com/website/about.html.

[11] BaiDuWaiMai [EB/OL]. [2017-06-05]. http://waimai.baidu.com/waimai?qt=about.

[12] 张琳，刘彦，王汝传. 位置大数据服务中基于差分隐私的数据发布技术 [J]. 通信学报，2016，37(9)：46–54.

[13] Hassan U U, Curry E. Multi-armed bandit approach to online spatial task assignment [C] //Proc of the 11th International Conference on Ubiquitous Intelligence and Computing. Washington DC: IEEE Computer Society, 2014: 212–219.

[14] McSherry F, Talwar K. Mechanism design via differential privacy [C] //Proc of the 48th Annual IEEE Symposium on Foundations of Computer Science. Washington DC: IEEE Computer Society, 2007: 94–103.

[15] Dwork C, Roth A. The algorithmic foundations of differential privacy [J]. Foundations & Trends® in Theoretical Computer Science, 2014, 9(3–4): 211–407.

[16] To H, Ghinita G, Shahabi C. Framework for protecting worker location privacy in spatial crowdsourcing [J]. Proceedings of the VLDB Endowment, 2014, 7(10): 919–930.

[17] Xiong Ping, Zhang Lefeng, Zhu Tianqing. Reward-based spatial crowdsourcing with differential privacy preservation [J]. Enterprise Information Systems, 2016, 11(10): 1–18.

[18] Qardaji W, Yang Weining, Li Ninghui. Differentially private grids for geospatial data [C] //Proc of the 29th International Conference on Data Engineering. Washington DC: IEEE Computer Society, 2013: 757–768.

[19] Sweeney L. *k*-anonymity: a model for protecting privacy [J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(5): 557–570.

[20] 吴英杰，唐庆明，倪巍伟，等. 基于取整划分函数的 k 匿名算法 [J]. 软件学报，2012，23(8)：2138–2148.

[21] Hu Jie, Huang Liusheng, Li Lu, *et al*. Protecting location privacy in spatial crowdsourcing [M] //Web Technologies and Applications. Cham: Springer International Publishing, 2015: 113–124.

[22] Chow C Y, Mokbel M F, Liu Xuan. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments [J]. GeoInformatica, 2011, 15(2): 351–380.

[23] Kazemi L, Shahabi C. A privacy-aware framework for participatory sensing [J]. ACM SIQKDD Explorations Newsletter, 2011, 13(1): 43–51.

[24] Kleinberg J, Tardos É. Algorithm design [M]. Boston: Addison Wesley, 2005.

[25] Vu K, Zheng Rong, Gao Jie. Efficient algorithms for *k*-anonymous location privacy in participatory sensing [C] //Proc of IEEE INFOCOM. Piscataway, NJ: IEEE Press, 2012: 2399–2407.

[26] Datar M, Immorlica N, Indyk D, *et al*. Locality-sensitive hashing scheme based on p-stable distributions [C] //Proc of the 20th Annual Symposium on Computational Geometry. New York: ACM Press, 2004: 253–262.

[27] Shen Yao, Huang Liusheng, Li Lu, *et al*. Towards preserving worker location privacy in spatial crowdsourcing [C] // Proc of IEEE Global Communications Conference. Piscataway, NJ: IEEE Press, 2015: 1–6.

[28] Yao A C. How to generate and exchange secrets [C] // Proc of the 27th Annual Symposium on Foundations of Computer Science. Washington DC: IEEE Computer Society, 1986: 162–167.

[29] Xu Jian, Wang Wei, Pei Jian, *et al*. Utility-based anonymization using local recoding [C] // Proc of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2006: 785–790.

[30] 童咏昕，袁野，成雨蓉，等. 时空众包数据管理技术研究综述 [J]. 软件学报，2017，28(1)：35–58.