# 融合上下文信息与核密度估计的协同过滤推荐

马瑞新,郭芳清,刘振娇,陈志奎,赵 亮
(大连理工大学 软件学院,辽宁 大连 116620)

**摘 要**:随着互联网信息技术的迅速发展,网络数据量快速增长,如何在海量数据中找到用户感兴趣的信息并实现个性化推荐是目前重要的研究方向。协同过滤算法作为推荐系统中的经典方法被广泛应用于不同场景,但是仍然存在数据稀疏,以及在计算相似度时不能考虑到所有数据的问题,只能够利用具有共同评分的数据,严重影响了推荐的精确度。针对上述存在的问题,提出了一种融合上下文信息与核密度估计的协同过滤个性化推荐算法。该算法通过对用户和项目各自的上下文信息和已经存在的用户评分数据进行处理,通过核密度估计构建用户和项目的兴趣模型,充分挖掘了用户和项目的兴趣分布,以获得更准确的用户和项目兴趣相似度,降低预测评分误差。在公开的数据集上验证表明,将该算法对比传统的协同过滤算法,有效提高了推荐的精确度。

**关键词**:协同过滤算法;核密度估计;上下文信息;兴趣估计模型;推荐系统
**中图分类号**:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2021)04-0034-06
doi:10.3969/j.issn.1673-629X.2021.04.006

# Collaborative Filtering Recommendation Algorithm for Fusion Context Information and Kernel Density Estimation

MA Rui-xin,GUO Fang-qing,LIU Zhen-jiao,CHEN Zhi-kui,ZHAO Liang
(School of Software Technology,Dalian University of Technology,Dalian 116620,China)

**Abstract**:With the development of Internet information technology and the growth of network data,how to find the information that users are interested in from the massive data and realize personalized recommendation is an important research direction at present. As a classic method in the recommendation system,collaborative filtering algorithm is widely used in different scenes,but it still cannot solve the problem of data sparsity,and in the calculation of similarity,it cannot take all the data into account and can only use the common data,which seriously affects the accuracy of recommendation. Aiming at the problems above,we propose a collaborative filtering recommendation fusing context information and kernel density estimation. The algorithm is based on the user and project their own context information and existing user rating data for processing,based on kernel method respectively to build user and project estimation model,fully taping the interest distribution of user and project,so as obtain more accurate similarity of user and project and reduce the prediction error. The validation on the open data set shows that compared with the traditional collaborative filtering algorithm,the proposed algorithm can effectively improve the accuracy of recommendation.

**Key words**:collaborative filtering algorithm;kernel density estimation;context information;interest estimation model;recommendation system

## 0 引 言

不断发展的互联网技术加快了互联网信息的增长速度,造成了严重的信息过载,用户在巨量信息中快速找到有意义的信息越来越困难。当用户面对大量信息时,如何花费较少的时间获取对自己有价值的信息成为一个关键性的难题,个性化推荐[1-2]能够有效地解决此类问题。传统的个性化推荐方法主要包括基于内容的推荐算法、协同过滤推荐算法和混合推荐算法。其中协同过滤推荐算法[3]作为一类经典的推荐算法,在实际中获得了广泛的应用,同时也得到了很多研究者的关注。协同过滤推荐主要包括基于邻居集(neighborhood-based)[4-5]和基于模型(model-based)[6]两种,其中基于邻居集的协同过滤推荐又分为基于用户(user-based)[4]和基于项目(item-based)[5]两种方法,

**收稿日期**:2020-05-28 **修回日期**:2020-09-30
**基金项目**:国家自然科学基金(61906030);中央高校基本科研业务费专项资金(DUT20RC(4)009)
**作者简介**:马瑞新(1975-),男,博士,副教授,CCF会员(C1428M),研究方向为自然语言处理、知识追踪学习;郭芳清(1997-),女,硕士研究生,研究方向为推荐系统。

但是在计算项目或者用户的相似性过程中只利用具有共同评分的数据,不能全面地估计项目(用户)相似度,而且当用户与项目的交互信息较少时,即用户已经评过分的项目远少于全部项目个数时,还会出现数据稀疏等问题。近年来,国内外科研人员提出了不同的思路用于改善传统的协同过滤方法存在的不足。邓秀勤等[7]提出了一种通过构建融合全加权矩阵分解的协同过滤模型来提高准确率的用户协同过滤推荐算法。郭彩云等[8]在用户对标签权重的计算时融入了用户的评分,这种改进的基于标签的协同过滤算法能够考虑用户有不同兴趣程度的项目对推荐结果的影响,进而充分挖掘用户真实的兴趣以提升推荐性能。Liu 等[9]采用关联挖掘技术,提出了被应用于引文推荐中的基于上下文的协同过滤方法。Ushiama 等[10]提出的一种基于推特用户相似性的项目个性化排序方法,不要求用户手动指定自己的兴趣和偏好,着重于利用用户发布的内容提取用户的特征。Huynh 等[11]提出了一种基于上下文属性的协同过滤模型,该模型的计算结果基于评价值的相似因素和上下文属性的相似因素。矩阵分解和聚类也经常被应用于协同过滤推荐中,用于解决数据稀疏性和扩展性等问题,如基于矩阵奇异值分解的 SVD 算法[12]、非负矩阵分解[13]、K-means 聚类[14]、c 均值模糊聚类方法[15-16]等。龚敏等[14]在进行用户聚类时使用 K-means 方法,使用 Slope One 算法生成评分矩阵,在一定程度上缓解了系统的可扩展性问题。目前,一些研究通过改进项目间和用户间的相似度提高推荐精度,但仍然存在着由于仅使用拥有共同评分的数据,而无法充分挖掘用户间和项目间兴趣分布和不能有效缓解数据稀疏等问题。

针对上述问题,该文提出了融合上下文信息与核密度估计的协同过滤个性化推荐方法。该方法融合了用户和项目的上下文信息,通过核密度估计方法构建用户和项目的兴趣模型,利用用户和项目的上下文信息充分挖掘了用户和项目的兴趣分布,更好地估计用户间和项目间的兴趣相似度,以获得更精确的预测评分,最后完成用户和项目间的推荐。在公开的数据集上验证表明,该算法有效地提高了推荐系统的精确度。

## 1　预备知识

传统的协同过滤算法是通过用户的历史行为获得用户的偏好,进而为用户推荐更有可能喜欢的项目。本节将详细介绍推荐过程中所需要的数据的表示、常用的相似性度量方法和分数预测规则。

### 1.1　数据表示

在利用协同过滤方法进行推荐时,所使用的数据包括用户信息、项目信息以及用户对项目的评分信息,通过用户对项目的评分信息,即用户的历史行为的分析可以获得用户对此项目的态度。因此,定义用户集合 $U=\{\text{user}_1,\text{user}_2,\cdots,\text{user}_m\}$,项目集合 $I=\{\text{item}_1,\text{item}_2,\cdots,\text{item}_n\}$,$m$ 表示推荐系统中用户的个数,$n$ 表示其中项目的个数,$r_{u,i}$ 表示用户 $u$ 对项目 $i$ 的评分。根据所处理的数据构建一个包含所有用户与项目的评分矩阵,矩阵表示如表 1 所示。

在该矩阵中,由于有些项目并没有被用户观看或使用,因此没有相对应的评分,此时该用户对该项目无历史行为,$r_{u,i}$ 不存在。如表 1 所示,行向量为每个用户对不同项目的喜好情况,用于表示用户的兴趣,列向量为每个项目对应不同用户的评分结果,表示项目的兴趣。对于评分为空的部分数据,可以通过已经存在的评分数据获得兴趣模型进行预测。但是由于矩阵中缺失部分数据,虽然该直接获得的兴趣模型能够简单方便的进行计算,但得到的兴趣分布较为粗糙,无法充分利用所获得的数据,存在着不足。

表 1　用户 -项目评分矩阵

| | $\text{item}_1$ | $\text{item}_2$ | $\text{item}_3$ | … | $\text{item}_n$ |
|---|---|---|---|---|---|
| $\text{user}_1$ | $r_{1,1}$ | $r_{1,2}$ | $r_{1,3}$ | … | $r_{1,n}$ |
| $\text{user}_2$ | $r_{2,1}$ | $r_{2,2}$ | $r_{2,3}$ | … | $r_{2,n}$ |
| $\text{user}_3$ | $r_{3,1}$ | $r_{3,2}$ | $r_{3,3}$ | … | $r_{3,n}$ |
| … | … | … | … | … | … |
| $\text{user}_m$ | $r_{m,1}$ | $r_{m,2}$ | $r_{m,3}$ | … | $r_{m,n}$ |

### 1.2　相似性度量

在协同过滤推荐算法中,需要获得用户相似度及项目相似度,相似度计算的准确性很大程度影响到推荐效果。一般情况下,有三种具有代表性的相似性度量方法,分别为余弦相似性度量、修正的余弦相似性度量以及 Pearson 相关系数[17]。总体来说,普通的余弦相似性度量计算方法效果较差,为了提高系统推荐性能,Pearson 相关系数和修正的余弦相似性两种方法能够获得更精确的结果,下文将详细介绍这两种方法。

(1)Pearson 相关系数。

设 $I_u=\{i:i\in I,r_{u,i}\neq\varphi\}$ 作为已被用户 $u$ 评分的项目集合,$\bar{r}_{u,*}$ 表示用户 $u$ 所有评分的均值。则用户的相似性计算如式(1)所示:

$$\text{corr}_{u,v}=\frac{\sum_{i\in I_u\cap I_v}(r_{u,i}-\bar{r}_{u,*})(r_{v,i}-\bar{r}_{v,*})}{\sqrt{\sum_{i\in I_u\cap I_v}(r_{u,i}-\bar{r}_{u,*})^2}\sqrt{\sum_{i\in I_u\cap I_v}(r_{v,i}-\bar{r}_{v,*})^2}} \tag{1}$$

设 $u_i=\{u:u\in U,r_{i,u}\neq\varphi\}$ 为对项目 $i$ 评分过的用户集合,$\bar{r}_{i,*}$ 为对项目 $i$ 产生的评分的均值。则项目间相似性计算如式(2)所示:

$$\mathrm{corr}_{i,j}=\frac{\sum_{u\in U_i\cap U_j}(r_{i,u}-\bar{r}_{i,*})(r_{j,u}-\bar{r}_{j,*})}{\sqrt{\sum_{u\in U_i\cap U_j}(r_{i,u}-\bar{r}_{i,*})^2}\sqrt{\sum_{u\in U_i\cap U_j}(r_{j,u}-\bar{r}_{j,*})^2}} \tag{2}$$

(2)修正的余弦相似性。

余弦相似性度量方法是根据向量的坐标值计算两个向量的夹角余弦值,值越大,相似性越大。在协同过滤推荐过程中计算用户间和项目间相似度时,将用户向量看作映射到项目空间的 $n$ 维向量,将项目向量看作时映射到用户空间的 $m$ 维向量。在真实推荐系统中,每个用户评分尺度不同,具有同样喜好程度的用户可能会对相同的项目打不同的分数。直接使用余弦相似性方法,由于忽略评分标准的不同,不能获得准确的度量结果。修正的余弦相似性计算方法可用来弥补以上不足,在计算用户相似度过程中将评分矩阵中用户的真实评分数据减去用户对所有项目产生的评分的平均值来减少不同用户评价标准不同所带来的消极影响,用户相似度计算具体方法如式(3):

$$\mathrm{corr}_{u,v}=\frac{\sum_{i\in I_u\cap I_v}(r_{u,i}-\bar{r}_{u,*})(r_{v,i}-\bar{r}_{v,*})}{\sqrt{\sum_{i\in I_u}(r_{u,i}-\bar{r}_{u,*})^2}\sqrt{\sum_{i\in I_v}(r_{v,i}-\bar{r}_{v,*})^2}} \tag{3}$$

在计算项目间相似度时,为了规避由于评分尺度不同产生的计算问题,将真实的项目评分减去该项目所对应的所有评分的平均值,详细的计算方法如式(4)所示:

$$\mathrm{corr}_{i,j}=\frac{\sum_{u\in U_i\cap U_j}(r_{i,u}-\bar{r}_{i,*})(r_{j,u}-\bar{r}_{j,*})}{\sqrt{\sum_{u\in U_i}(r_{i,u}-\bar{r}_{i,*})^2}\sqrt{\sum_{u\in U_j}(r_{j,u}-\bar{r}_{j,*})^2}} \tag{4}$$

### 1.3 评分预测

在分别获得目标用户与其他用户之间相似度和目标项目与其他项目之间的相似度后,分别将相似度的值按照从小到大的顺序排列,选取相似度值较小的前几个用户和项目作为目标用户和目标项目的邻居集。

根据求得的目标用户邻居集,相似性作为权重,将用户对目标项目的评分的加权平均作为预测评分。由于不同的用户评分尺度不同,修改评分预测策略,预测规则如式(5)所示:

$$p_{u,i}=\bar{R}_u+\frac{\sum_{v\in N_u}\mathrm{corr}_{u,v}\times(r_{v,i}-\bar{R}_v)}{\sum_{v\in N_u}|\mathrm{corr}_{u,v}|} \tag{5}$$

根据求得的目标项目邻居集,相似性作为权重,将项目对目标用户的评分的加权平均作为预测评分,由于不同的用户评分尺度不同,修改评分预测策略,预测规则如式(6)所示:

$$p_{u,i}=\bar{R}_i+\frac{\sum_{j\in N_u}\mathrm{corr}_{i,j}\times(r_{u,j}-\bar{R}_j)}{\sum_{j\in N_u}|\mathrm{corr}_{i,j}|} \tag{6}$$

## 2 融合上下文信息与核密度估计的协同过滤推荐

该文提出的融合上下文信息与核密度估计的协同过滤推荐方法首先挖掘用户和项目在已知数据上的兴趣分布,然后估计用户和项目的兴趣分布,通过计算兴趣度在缺失数据上的扩散,更加准确地构建用户兴趣模型。下面具体介绍该算法的 3 个主要步骤。

### 2.1 基于上下文的分类相似度

在推荐系统中,用户和项目具有相对应的上下文信息,不同的上下文信息被不同的类别包含,通过对上下文信息的类别进行区分和计算,可以分别获得用户和项目的上下文分类相似度。

#### 2.1.1 基于上下文的项目分类相似度

定义集合 $C=\{C_1,C_2,\cdots,C_k\}$ 表示项目上下文信息类别集合,其中 $C_k$ 为项目的一种上下文类别,$C_k=\{C_{k1},C_{k2},\cdots,C_{kk}\}$,$C_{kk}$ 为某种上下文类别中的具体分类信息。例如,在电影推荐系统中,电影可以被具体分为主演、导演、电影类型、拍摄地、上映日期等不同类别的上下文信息,每一类上下文信息中又具有详细的划分,如在类型信息中可以将电影分为喜剧片,爱情片、战争片、动作片等,一个项目可能同属于战争片和动作片。计算项目的上下文信息相似度,需要考虑到项目之间共有的类别比例和共有的类别占整个上下文信息类别集合的比例。$C_i\in C$,定义项目间的上下文分类相似度计算如式(7)所示:

$$\mathrm{sim}_c(i,j)=\frac{|C_i\cap C_j|^2}{|C|\times|C_i\cup C_j|} \tag{7}$$

考虑到在使用项目间距离度量时,需随着项目间相似性的增大,在项目空间上两个项目的距离减小,因此项目 $i$ 与项目 $j$ 的距离计算如式(8)所示:

$$d_{ij}=1-\mathrm{sim}_c(i,j) \tag{8}$$

#### 2.1.2 基于上下文的用户分类相似度

定义集合 $D=\{D_1,D_2,\cdots,D_k\}$ 表示用户上下文信息类别集合,其中 $D_k$ 为用户的一个上下文类别,$D_k=\{D_{k1},D_{k2},\cdots,D_{kk}\}$,$D_{kk}$ 为上下文类别中的具体分类信息。例如用户的上下文信息包括用户的年龄、职业、性别等类别。每一类用户上下文信息也可以被划分为更详细的类别,如用户的年龄可以按照年龄段进行进一步划分。计算用户的上下文信息相似度,需要考虑到用户之间共有的类别比例和共有的类别占整个上下文

信息类别集合的比例。定义用户间的上下文分类相似度如式(9)所示:

$$\mathrm{sim}_d(u,v)=\frac{|D_i\cap D_j|^2}{|D|\times|D_i\cup D_j|} \tag{9}$$

在实际计算用户间距离时,需随着用户间相似性变大,在用户空间上两个用户的距离减小,因此用户 $u$ 与用户 $v$ 的距离计算如式(10)所示:

$$d_{uv}=1-\mathrm{sim}_c(u,v) \tag{10}$$

### 2.2　兴趣估计模型的构建

核密度估计方法利用已测得的样本估计未被测到的样本分布,能够估计未知的密度函数,是一种非参数估计方法[18]。设 $X_1,X_2,\cdots,X_n$ 为总体分布 $X$ 的独立同分布样本,具体的核密度估计定义如式(11)所示:

$$f(X)=\frac{1}{nh}\sum_{i=1}^{n}K\left(\frac{\|X-X_i\|}{h}\right) \tag{11}$$

其中,$K\left(\frac{\|X-X_i\|}{h}\right)$ 为核函数,$K$ 通常是固定值,被称为核函数的窗宽,$h$ 被称作带宽,$h$ 越大密度估计越平滑,$h$ 越小密度估计越易出现多个峰值并且峰顶较尖锐的情况。在文献[18]中,核函数宽窗较核函数的类别对实验结果的影响更大。

在为用户推荐项目时,用户对于没有被评过分的项目也存在兴趣偏好,传统的相似性计算方法,往往仅使用了拥有共同评分的项目,降低了推荐系统性能,在数据稀疏时,可用数据更少难以做出准确的判断。针对这个问题,首先对用户在整个项目空间和项目在整个用户空间上的兴趣密度分布进行估计,然后再分别计算用户间和项目间的兴趣密度分布的相似性,以改善推荐性能。在实际情况中,兴趣密度分布往往会出现多个局部最大值,参数估计方法的参数模型的基本假定与实际情况差异较大,因此非参数估计方法中的核密度估计方法更适用于推荐。核函数有多项式核函数、高斯核函数、线性核函数、三角核函数等种类。高斯核函数(径向基核函数),将有限维数据映射到无穷维,是复杂总和的有限机率分布。兴趣分布可以看作是含有不同种类的不确定因素的一个有限机率分布,因此,首先考虑使用高斯核函数,其定义如式(12):

$$K_g(Z)=\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{z^2}{2h^2}\right\} \tag{12}$$

式(13)和式(14)分别表示用高斯核密度估计方法估计的用户兴趣分布 $P_u$ 与项目兴趣分布 $P_i$。

$$f_{P_u}(j)=\frac{1}{|I_u|\times\sqrt{2\pi}h}\sum_{i\in I_u}r_{u,i}\times\exp\left\{\frac{d_{i,j}^2}{2h^2}\right\} \tag{13}$$

$$f_{P_i}(v)=\frac{1}{|U_i|\times\sqrt{2\pi}h}\sum_{u\in U_i}r_{u,i}\times\exp\left\{\frac{d_{u,v}^2}{2h^2}\right\} \tag{14}$$

### 2.3　相似度计算

本节主要说明如何计算用户间相似度和项目间相似度。在信息论中,相对熵,又称 KL 散度,是两个概率分布之间差别的非对称性度量,类似文献[19]策略,文中通过相对熵分别计算用户相似性和项目相似度,具体定义如式(15)和式(16)所示:

$$D_{\mathrm{KL}}(P_u\parallel P_v)=\sum_{i=1}^{k}P_u(i)\log\frac{P_u(i)}{P_v(i)} \tag{15}$$

$$D_{\mathrm{KL}}(P_i\parallel P_j)=\sum_{u=1}^{k}P_i(u)\log\frac{P_i(u)}{P_j(u)} \tag{16}$$

因为 KL 散度的非对称性,不能将其直接作为度量相似度的标准,该文采用式(17)和式(18)分别获得用户和项目的相似性:

$$\mathrm{corr}_{u,v}=\frac{1}{2}[D_{\mathrm{KL}}(P_u\parallel P_v)+D_{\mathrm{KL}}(P_v\parallel P_u)] \tag{17}$$

$$\mathrm{corr}_{i,j}=\frac{1}{2}[D_{\mathrm{KL}}(P_i\parallel P_j)+D_{\mathrm{KL}}(P_j\parallel P_i)] \tag{18}$$

### 2.4　融合上下文信息和核密度估计的推荐算法描述

在进行 2.1,2.2,2.3 节的分类相似度计算、兴趣模型构建、相似度计算步骤后,综合根据 1.3 节预测评分计算方法得到的预测值获得用户项目评分矩阵中为空的评分预测值。具体的融合上下文信息和核密度估计的推荐算法过程如下所述:

融合上下文信息和核密度估计的推荐算法:

输入:用户信息、项目信息、用户对项目的历史评分数据

输出:预测评分

1. 通过式(8)计算用户间距离,式(10)计算项目间的距离;

2. 计算兴趣分布,通过式(13)计算用户的兴趣分布,式(14)计算在用户空间上项目的兴趣分布;

3. 不断重复步骤 1 和步骤 2,直至获得全部用户和项目的兴趣分布;

4. 计算相似性,通过式(17)计算用户间相似性,式(18)计算项目间的相似性;

5. 通过式(5)和式(6)分别计算预测评分;

6. 综合第 5 步中的结果得到预测评分,根据预测分数完成推荐。

## 3　实验结果及分析

### 3.1　数据集

文中的数据集采用明尼苏达大学提供的 MovieLens[20] 数据集,具体使用的是 MovieLens 中的 ML-100K 数据集。ML-100K 数据集中含有 943 个用户和 1 682部电影,共包括产生的十万条评分记录。数据集中的分数为 1-5 的整数,每名用户对超过 20 部电影拥有评分记录,分数越高则用户对电影的评价越好。为获取完善的上下文信息,从网络上爬取了 MovieLens 数据集中电影名称所对应的上下文属性信息。将数据集按 8∶2的比例随机分配为训练集与测试集。

### 3.2 评价标准

采用精确度(Precision)和平均绝对误差(mean absolute error,MAE)作为评分标准,精确度和平均绝对误差是推荐算法中常用的评估度量。精确度表示预测评分与真实评分相同的项目在测试集中所有项目的占比,精确度越高推荐质量越高。由于预测评分值为浮点型,实际评分值为 1-5 的整数,预测评分与实际评分很难完全相等,因此设置阈值 Threshold,预测评分与实际评分值小于此阈值,则认为实际评分与预测评分相等。精确度计算如式(19)所示:

$$\text{Precision} = \frac{|R|}{m} \tag{19}$$

平均绝对误差表示预测值和观测值之间绝对误差的平均值,MAE 值越小表示推荐越准确,MAE 的计算如式(20)所示:

$$\text{MAE} = \frac{\sum_{i=1}^{m} (P_{u,i} - R_{u,i})}{m} \tag{20}$$

其中,$P_{u,i}$ 表示用户 $u$ 对项目 $i$ 的预测分数值,$R_{u,i}$ 表示用户 $u$ 对项目 $i$ 的实际分数值,$m$ 为测试集中所有的项的个数,$|R|$ 为预测评分与实际评分之差小于阈值的项目个数。

### 3.3 实验结果分析

#### 3.3.1 参数对实验结果的影响

(1)核函数带宽的影响。

该实验分析了高斯核函数带宽对实验结果的影响,设置带宽 $h$ 的范围为 0.5 ~ 1.3,图 1、图 2 分别为在不同带宽情况下实验的 Precision 值和 MAE 值。由图 1 可知,在带宽较小时,精确度的波动较大,随着核函数带宽的增加,精确度变化逐渐趋于平稳,当 $h$ = 0.8 时精确度值最高。由图 2 可知,当 $h$ = 0.8 时,MAE 值达到最低点,在核函数带宽超过 1 后,MAE 值逐渐升高。因此,当 $h$ = 0.8 时,实验效果达到最优状态。

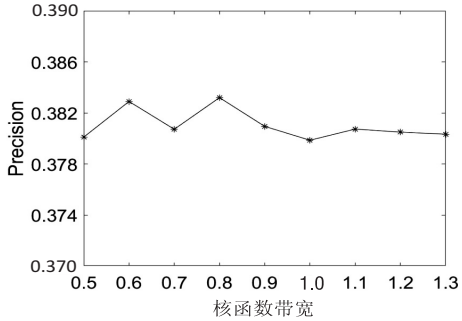


图 1 核函数带宽与 precision 值的关系

(2)基于用户间相似度的评分与基于项目间相似度的评分的权重对实验结果的影响。

最终评分由基于用户间相似度的评分与基于电影项目间相似度的评分两部分组成,两部分的权重不同会对最终预测结果产生影响。表 2 为不同权重下实验的 Precision 值和 MAE 值,基于用户间相似度的评分,简称为(byUser),基于项目间相似度的评分,简称为(byMovie)。从表 2 中可以看出,随着 byUser 在总评分中权重的增加,实验的 Precision 逐渐提高至最高点后下降,MAE 逐渐降低至最低点后升高。当 byUser 与 byMovie 占比分别为 0.45 和 0.55 时,实验的 Precision 最高,当 byUser 与 byMovie 占比分别为 0.55 和 0.45 时,实验的 MAE 达到最低。

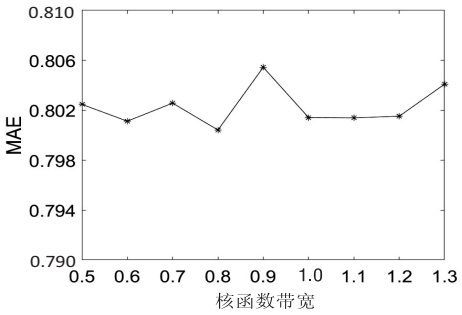


图 2 核函数带宽与 MAE 值的关系

表 2 byUser 与 byMovie 权重对实验结果的影响

| 权重 | Precision | MAE |
|---|---|---|
| byUser=0.3 byMovie=0.7 | 0.376 6 | 0.810 3 |
| byUser=0.35byMovie=0.65 | 0.378 6 | 0.807 0 |
| byUser=0.4 byMovie=0.6 | 0.379 4 | 0.804 5 |
| byUser=0.45byMovie=0.55 | 0.381 1 | 0.802 8 |
| byUser=0.5 byMovie=0.5 | 0.380 6 | 0.801 8 |
| byUser=0.55byMovie=0.45 | 0.379 6 | 0.801 6 |
| byUser=0.6 byMovie=0.4 | 0.377 9 | 0.802 2 |
| byUser=0.65byMovie=0.35 | 0.375 3 | 0.803 8 |
| byUser=0.7 byMovie=0.3 | 0.373 6 | 0.806 4 |

#### 3.3.2 实验对比分析

在 ML-100K 数据集下,将文中的推荐方法分别与基于 Pearson 相关系数的协同过滤算法(P-CF)、基于修正余弦相似度的协同过滤算法(Cos-CF)、基于核方法的协同过滤算法(K-CF)进行对比。

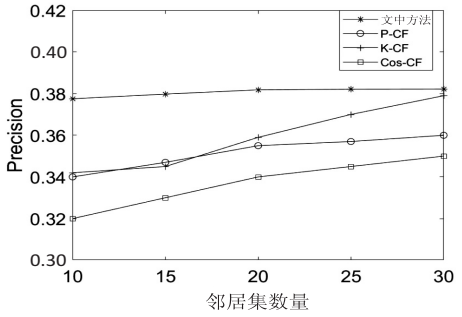


图 3 不同邻居集数量下各算法 Precision 值对比

图 3、图 4 分别为在不同邻居集数量的情况下,四种算法的精确度(Precision)和平均绝对误差(MAE)的值,实验设置邻居集大小分别为 10、15、20、25、30。由

图 3、图 4 可知,文中提出的算法在邻居集大小不同的情况下,均较其他三种算法具有较高的精确度和较低的 MAE 值。其他三种算法在邻居集较大的情况下,精确度才逐渐提高,MAE 值逐渐下降。而邻居集越大,在一定程度上算法的运行时间也会增加。文中提出的方法在邻居集数量较少时,就可以得到远优于其他算法的精确度和 MAE 值,不会受到邻居集大小的限制。

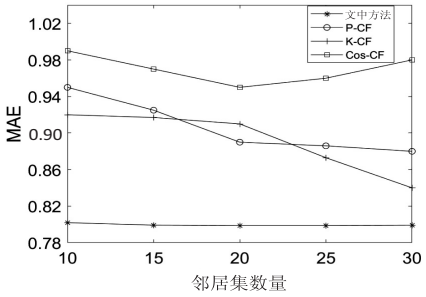


图 4　不同邻居集数量下各算法 MAE 值对比

通过将文中提出的方法与其他三种算法进行对比,提出的融合上下文信息与核密度估计的个性化协同过滤推荐算法在 Precision 值上均高于其他三种算法,MAE 值上均低于其他三种算法,该算法受邻居集大小影响较小,证明了提出的方法的有效性和稳定性。

## 4　结束语

针对传统的协同过滤算法在稀疏性和无法充分利用数据信息等问题,提出了融合上下文信息与核密度估计的协同过滤推荐方法。该方法融合了用户和项目的上下文信息,分别计算用户和项目的上下文分类相似度,利用核密度估计方法对用户和项目的兴趣分布建模,然后用 KL 散度计算兴趣相似度,将相似性度量值进行排序,获取相似性高的近邻集合,最后通过评分预测规则得到预测评分,将合适的项目推荐给用户。通过在公开的电影数据集上验证,证明了该方法在提高推荐系统性能的有效性,并且效果稳定。

**参考文献:**

[1] LI M, WANG L. A survey on personalized news recommendation technology[J]. IEEE Access, 2019, 7: 145861-145879.

[2] LI Z, ZOU X. A review on personalized academic paper recommendation[J]. Computer and Information Science, 2019, 12(1): 33-43.

[3] AMBULGEKAR H P, PATHAK M K, KOKARE M B. A survey on collaborative filtering: tasks, approaches and applications[C]//Proceedings of international ethical hacking conference. Kolkata, India: Springer, 2019: 289-300.

[4] LU Q, GUO F, ZHANG R, et al. User-based collaborative filtering recommendation method combining with privacy concerns intensity in mobile commerce[J]. International Journal of Wireless and Mobile Computing, 2019, 17(1): 63-70.

[5] GUO T, LUO J, DONG K, et al. Locally differentially private item-based collaborative filtering[J]. Information Sciences, 2019, 502: 229-246.

[6] LOEPP B, ZIEGLER J. Towards interactive recommending in model-based collaborative filtering systems[C]//Proceedings of the 13th ACM conference on recommender systems. Copenhagen, Denmark: Association for Computing Machinery, 2019: 546-547.

[7] 邓秀勤,刘太亨,刘富春,等. 基于全加权矩阵分解的用户协同过滤推荐算法[J]. 计算机科学,2019,46(11A):199-203.

[8] 郭彩云,王会进. 改进的基于标签的协同过滤算法[J]. 计算机工程与应用,2016,52(8):56-61.

[9] LIU H, KONG X, BAI X, et al. Context-based collaborative filtering for citation recommendation[J]. IEEE Access, 2015, 3: 1695-1703.

[10] USHIAMA T, TOMINAGA K. A method for personalized ranking of items based on similarity between Twitter users[C]// Proceedings of the 8th international conference on ubiquitous information management and communication. Siem Reap, Cambodia: Association for Computing Machinery, 2014: 1-4.

[11] HUYNH H X, PHAN N Q, PHAM N M, et al. Context-similarity collaborative filtering recommendation[J]. IEEE Access, 2020, 8: 33342-33351.

[12] GUAN X, LI C, GUAN Y, et al. Matrix factorization with rating completion: an enhanced SVD model for collaborative filtering recommender systems[J]. IEEE Access, 2017, 5: 27668-27678.

[13] 康林瑶,唐　兵,夏艳敏,等. 基于 GPU 加速和非负矩阵分解的并行协同过滤推荐算法[J]. 计算机科学,2019,46(8):106-110.

[14] 龚　敏,邓珍荣,黄文明. 基于用户聚类与 Slope One 填充的协同推荐算法[J]. 计算机工程与应用,2018,54(22):139-143.

[15] KOOHI H, KIANI K. User based collaborative filtering using fuzzy C-means[J]. Measurement, 2016, 91: 134-139.

[16] KATARYA R, VERMA O P. Recommender system with grey wolf optimizer and FCM[J]. Neural Computing and Applications, 2018, 30(5): 1679-1687.

[17] Last. fm. Music recommendation service[EB/OL]. (2011-10-01)[2020-05-20]. http:// www. last. fm.

[18] GIVENS G H, HOETING J A. Computational statistics[M]. New York: Wiley-Interscience, 2013.

[19] JIANG K, WANG P, YU N, et al. ContextRank: personalized tourism recommendation by exploiting context information of geotagged web photos[C]//Proceedings of the 2011 sixth international conference on image and graphics. Hefei: IEEE, 2011: 931-937.

[20] GroupLens. MovieLens DataSets[EB/OL]. (1998-04-01)[2020-05-20]. http:// www. grouplens. org.