# 融合 Jensen-Shannon 散度的推荐算法

王 永[1] 王永东[1] 邓江洲[1] 张 璞[2]
(重庆邮电大学经济管理学院 重庆 400065)[1] (重庆邮电大学计算机科学与技术学院 重庆 400065)[2]

**摘 要** 为充分利用所有评分,缓解数据稀疏性问题,将概率统计领域的 Jensen-Shannon(JS)散度引入相似性度量中,提出了一种新的项目相似性度量算法。该算法将项目的评分信息转化为评分值密度,并依据评分值的密度分布来计算项目相似性。同时,引入评分数量因子,进一步提升了基于JS的相似性度量方法的性能。最后,以基于JS的相似性度量方法为基础,设计了相应的协同过滤算法。在 MovieLens 数据集上的实验结果表明,所提算法在预测误差和推荐准确性方面均有良好的表现。因此,该算法在推荐系统中具有很好的应用潜力。

**关键词** Jensen-Shannon 散度,评分值密度,相似性度量,协同过滤,数据稀疏性

**中图法分类号** TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.02.032

## Recommendation Algorithm Based on Jensen-Shannon Divergence

WANG Yong[1] WANG Yong-dong[1] DENG Jiang-zhou[1] ZHANG Pu[2]
(School of Economics and Managements, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)[1]
(School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)[2]

**Abstract** To fully utilize all the ratings and weaken the problem of data sparsity, the Jensen-Shannon divergence in statistics field was used to design a new similarity measure for items. In this similarity measure, the ratings for items are converted to the density of rating values. Then, the item similarity is calculated according to the density of rating values. Meanwhile, the factor for the number of ratings is also considered to further enhance the performance of the proposed similarity measure based on JS divergence. Finally, a collaborative filtering recommendation algorithm is presented according to the JS-divergence-based item similarity. The test results on MovieLens dataset show that the proposed algorithm has good performance in prediction error and recommendation precision. Therefore, it has high potential to be applied in recommendation system.

**Keywords** Jensen-Shannon divergence, Density of ratings, Similarity measure, Collaborative filtering, Data sparsity

## 1 引言

互联网在带来便捷服务的同时,也产生了信息过载的问题。如何精准地为用户推荐个性化内容成为目前研究的热点之一。协同过滤推荐算法是一种行之有效的解决上述问题的办法。协同过滤推荐算法主要依据历史数据,找到相似用户或者项目,再基于这些相似用户的偏好或项目的特征预测出目标用户的偏好并进行推荐。因此,如何准确地度量用户或者项目间的相似性,一直都是协同过滤算法研究中的一个关键问题。

目前,一些典型的基于项目共同评分的算法,如修正余弦相似性(ACOS)[1]、皮尔逊相关系数(PCC)[2]等,有着广泛的应用。基于评分绝对数量的算法如 JACCARD 算法[3]被提出,该算法从整体统计的角度给出了一种新的度量相似性的方案。然而,这些典型的相似性度量方法难以应付推荐系统中的数据稀疏问题和用户(或项目)冷启动问题[4-5]。为解决数据稀疏问题和用户冷启动问题,国内外学者展开了一系列的研究工作。文献[6-8]提出了基于结构化描述信息的协同过滤推荐算法,免去了信息抽取和过滤过程,通过引入其他数据源(如社交网络信息)来弥补评分矩阵的稀疏性。文献[9-10]在结合外部信息源的基础上,利用矩阵分解的算法框架对高维矩阵进行降维,减少了噪音干扰,有效解决了数据稀疏性和用户冷启动的推荐问题。文献[10]提出了基于最大最小聚类算法的启发式聚类模型,通过该模型对用户进行划分,并引入了项目类别相似度,从用户和项目两个方面实现综合推荐,减小了稀疏性的影响。上述方法一定程度上缓解了数据的稀疏性,但终未脱离共同评分的限制。为充分使用所有的评分信息,基于 Kullback-Leibler(KL)散度[12]和基于巴氏系数

到稿日期:2017-12-06 返修日期:2018-02-17 本文受国家社会科学基金项目(15XGL024),重庆市前沿与应用基础研究计划项目(cstc2015jcyjA40025)资助。

王 永(1977-),男,博士,教授,CCF会员,主要研究方向为数据挖掘、信息系统和加密算法等,E-mail:wangyong1@cqupt.edu.cn(通信作者);王永东(1994-),男,硕士生,主要研究方向为推荐算法;邓江洲(1993-),男,硕士生,主要研究方向为数据挖掘和文本处理;张 璞(1976-),男,博士,副教授,主要研究方向为自然语言和数据挖掘等。

(Bhattacharyya Coefficient，BC)[13]的相似性方法被提出，这类方法利用评分值密度来计算相似度，无须使用共同评分，为解决数据稀疏性问题提供了新的思路，但往往忽略了评分绝对数量的影响。

本文将概率理论领域的 Jensen-Shannon 散度引入到相似性度量方法的设计中，利用评分值的密度计算相似性，有效摆脱了对共同评分的依赖，并从评分数量的角度对方法进行了改进，体现了评分数量的影响；基于此，提出了基于 JS 散度的协同过滤算法，以提高推荐系统的推荐质量。

## 2　基于 JS 散度的相似性度量方法

在推荐系统中，用户-项目评分矩阵用 $\boldsymbol{R}(m,n)$ 表示：

$$\boldsymbol{R}(m,n)=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1i} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2i} & \cdots & r_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ r_{u1} & r_{u2} & \cdots & r_{ui} & \cdots & r_{un} \\ \vdots & \vdots & & \vdots & & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mi} & \cdots & r_{mn} \end{bmatrix}$$

其中，$r_{ui}$ 表示用户 $u$ 对项目 $i$ 的评分，若用户未对项目进行过评分，则记为 $r_{ui}=0$ 。由于存在数据稀疏性，因此实际的评分矩阵中含有大量的 0 元素。基于评分矩阵 $\boldsymbol{R}(m,n)$，本文给出如下定义。

**定义 1**(项目 $i$ 的非零评分集合)　矩阵 $\boldsymbol{R}(m,n)$ 中所有用户对项目 $i$ 的非零评分，表示为 $RI=\{r_{ui}\mid u=1,2,\cdots,m$，且 $r_{ui}\neq 0\}$。

**定义 2**(评分值密度)　在集合 $RI$ 中评分值 $h$ 的密度被被定义为 $\rho_I(h)$，其表达式为：

$$\rho_I(h)=\frac{\# hI}{\# RI} \tag{1}$$

其中，$\# RI$ 是集合 $RI$ 中评分的总数量，$\# hI$ 是集合 $RI$ 中分值为 $h$ 的评分个数。同样地，对于项目 $j$，集合 $RJ$ 中分值为 $h$ 的评分密度记为 $\rho_J(h)$。

### 2.1　JS 散度及其改进

JS 散度(Jensen-Shannon Divergence)也称为 JS 距离或信息半径[14-15]。在概率理论和统计学中，JS 散度是一种通过信息值的密度分布测量其相似度的方法。本文引入 JS 散度来度量两个项目之间的相似性，下面首先给出评分值 $h$ 的平均密度。

**定义 3**(评分值 $h$ 的平均密度)　对于项目 $i$ 和项目 $j$ 的评分集合 $RI$ 和 $RJ$，评分值 $h$ 的平均密度为：

$$M(h)=\frac{1}{2}(\rho_I(h)+\rho_J(h)) \tag{2}$$

基于平均密度，借鉴 JS 散度在概率和统计学中的定义，根据评分矩阵的特点定义项目 $i$ 和 $j$ 的 JS 散度为：

$$D_{\mathrm{JS}}(i,j)=\frac{1}{2}\Big(\sum_{h\in H_I}\rho_I(h)\ln\frac{\rho_I(h)}{M(h)}+\sum_{h\in H_J}\rho_J(h)\ln\frac{\rho_J(h)}{M(h)}\Big) \tag{3}$$

$D_{\mathrm{JS}}$ 的值越小，表示两个评分集合的概率分布越相近，即项目 $i$ 和项目 $j$ 越相似。

JS 散度是根据评分值的密度来测量两个项目之间的差异程度。由于评分值的密度是约分化简后的数值，因此在式(3)所示的 JS 散度计算中未充分考虑评分数量的影响。现基于表 1 所列的示例做进一步的说明。

表 1　计算示例

Table 1　Numerical example

| 项目 | 评分值为 $h_1$ 的数量 | 评分值为 $h_2$ 的数量 | 评分总数 | 评分值为 $h_1$ 的密度 | 评分值为 $h_2$ 的密度 |
|---|---|---|---|---|---|
| $i$ | 1 | 9 | 10 | 1/10 | 9/10 |
| $j$ | 10 | 90 | 100 | 1/10 | 9/10 |
| $k$ | 50 | 50 | 100 | 1/2 | 1/2 |

在表 1 中，项目 $i$ 和项目 $j$ 的评分数量不同，但各评分值的密度相同。根据式(3)计算得到项目 $i$ 与项目 $k$、项目 $j$ 与项目 $k$ 的 JS 散度：$D_{\mathrm{JS}}(i,k)=D_{\mathrm{JS}}(j,k)=0.102$。在此例中，项目 $i$ 与项目 $j$ 的评分数量之间的差异完全没有体现。

为了在 JS 散度中体现评分值数量的影响，本文提出了评分值整体密度的概念，并对 JS 散度进行了改进。

**定义 4**(评分值 $h$ 的整体密度)　将项目 $i$ 和项目 $j$ 的评分集合 $RI$ 和 $RJ$ 合并为一个新的集合；在该合并后的集合中，评分值 $h$ 的密度被称为评分值 $h$ 的整体密度，其定义式为：

$$\hat{M}(h)=\frac{\# hI+\# hJ}{\# RI+\# RJ} \tag{4}$$

为了反映评分数量的影响，利用评分值的整体密度改进 JS 距离的计算。改进后的 JS 距离被定义为：

$$D_{\mathrm{AJS}}(i,j)=\frac{1}{2}\Big(\sum_{h\in H_I}\rho_I(h)\ln\frac{\rho_I(h)}{\hat{M}(h)}+\sum_{h\in H_J}\rho_J(h)\ln\frac{\rho_J(h)}{\hat{M}(h)}\Big) \tag{5}$$

**定理 1**　若 $\# RI\leqslant\# RJ$，则整体密度 $\hat{M}(h)$ 比平均密度 $M(h)$ 更接近于 $\rho_J(h)$。

证明：根据平均密度和整体密度的定义可知 $M(h)=\frac{1}{2}(\rho_I(h)+\rho_J(h))=\frac{1}{2}\Big(\frac{\# hI}{\# RI}+\frac{\# hJ}{\# RJ}\Big)$，$\hat{M}(h)=\frac{\# hI+\# hJ}{\# RI+\# RJ}$，则：

$$\begin{aligned} M(h)-\hat{M}(h) &=\frac{1}{2}\Big(\frac{\# hI}{\# RI}+\frac{\# hJ}{\# RJ}\Big)-\frac{\# hI+\# hJ}{\# RI+\# RJ} \\ &=\frac{(\# hI\cdot\# RJ+\# hJ\cdot\# RI)(\# RI+\# RJ)-(\# hI+\# hJ)\cdot(2\cdot\# RI\cdot\# RJ)}{2\cdot\# RI\cdot\# RJ(\# RI+\# RJ)} \\ &=\frac{\# hI\cdot\# RJ\cdot(\# RJ-\# RI)-\# hJ\cdot\# RI\cdot(\# RJ-\# RI)}{2\cdot\# RI\cdot\# RJ(\# RI+\# RJ)} \\ &=\frac{(\# RJ-\# RI)}{2\cdot(\# RJ+\# RI)}\cdot\Big(\frac{\# hI}{\# RI}-\frac{\# hJ}{\# RJ}\Big) \end{aligned}$$

由于 $\# RI\leqslant\# RJ$，因此可以得到 $\frac{(\# RJ-\# RI)}{2\cdot(\# RJ+\# RI)}\geqslant 0$；

当 $\Big(\frac{\# hI}{\# RI}-\frac{\# hJ}{\# RJ}\Big)\geqslant 0$ 时，有 $M(h)\geqslant\hat{M}(h)$，由定义可知：

$\rho_I(h) \geqslant M(h) \geqslant \rho_J(h)$，$\rho_I(h) \geqslant \hat{M}(h) \geqslant \rho_J(h)$。因此，$\rho_I(h) \geqslant M(h) \geqslant \hat{M}(h) \geqslant \rho_J(h)$。

当$(\frac{\#hI}{\#RI}-\frac{\#hJ}{\#RJ})<0$时，有$M(h) \leqslant \hat{M}(h)$，由定义可知：$\rho_I(h) < M(h) < \rho_J(h)$，$\rho_I(h) < \hat{M}(h) < \rho_J(h)$。因此，$\rho_I(h) < M(h) \leqslant \hat{M}(h) < \rho_J(h)$，等号仅在$\#RI=\#RJ$时成立。综合可知，$\hat{M}(h)$更接近于$\rho_J(h)$。

**推论 1** 整体密度$\hat{M}(h)$总是比平均密度$M(h)$更接近于集合$RI$和$RJ$中评分数量更多的那个集合中的$h$的密度。

证明：由定理1可知，当$\#RI \leqslant \#RJ$时，整体密度$\hat{M}(h)$比平均密度$M(h)$更接近于$\rho_J(h)$。按照同样的方法可证当$\#RJ \leqslant \#RI$时，整体密度$\hat{M}(h)$比平均密度$M(h)$更接近于$\rho_I(h)$。

依据推论1，由于$\hat{M}(h)$总是趋近于评分数量更多的集合中的评分值密度，因此用$\hat{M}(h)$代替$M(h)$计算JS散度，能够更好地反映出评分数量带来的影响。仍然依据示例1，根据式(5)，可得$D_{AJS}(i,k)=0.158$，$D_{AJS}(j,k)=0.102$。项目$i$和项目$j$的各分值密度是相同的，但项目$i$和项目$k$的评分绝对数量悬殊很大，而项目$j$和项目$k$的评分绝对数量差距较小。由式(3)计算出的项目$i$与项目$k$、项目$j$与项目$k$的JS散度也是相同的，可见原始的定义式(3)无法体现评分数量的作用。因此，整体概率密度$\hat{M}(h)$在JS散度的计算中更好地反映了评分数量的影响。

### 2.2 相似性度量方法

依据改进后的JS散度，给出项目相似性的计算公式：

$$Sim_{AJS}(i,j)=\frac{2}{1+e^{D_{AJS}}} \tag{6}$$

由式(6)可知，两个项目的JS距离越小，两者的相似度越高。

## 3 推荐算法

### 3.1 评分数量因子的确定

改进后的JS散度在一定程度上考虑了评分数量的影响，但当两个评分值的密度相近或相同时，该方法的区分度不够显著。例如，项目$i$、项目$j$和项目$k$的评分分别为{5,0,0,0,0}，{5,5,5,5,0}和{5,5,5,5,5}，这3个项目的分值为5分的密度都为1，因此可以得到它们之间的JS相似度也都为1。在该例中，考虑评分个数的影响时，项目$j$比项目$i$应更相似于项目$k$。

为了进一步强调评分数量的影响，本文给出评分数量因子$\omega$：

$$\omega=\begin{cases}1, & \#RI \geqslant \bar{n}, \#RJ \geqslant \bar{n} \\ \sqrt{\frac{2\times\min\{\#RI,\#RJ\}}{\#RI+\#RJ}}, & 其他\end{cases} \tag{7}$$

其中，$\bar{n}$是评分个数的大样本阈值。在概率统计理论中，大样本数据分析的结果相对于小样本数据有更高的可信度。因此，当两个项目评分个数足够多，即评分数据可视为大样本时，弱化数量因子的作用，充分发挥修正后的JS散度的作用。评分数量因子仅在两个项目的评分数量不全为大样本且差异大时才发挥作用。

引入评分数量因子后，得到修正的JS相似度公式：

$$Sim_{WAJS}(i,j)=\omega \cdot Sim_{AJS}(i,j) \tag{8}$$

### 3.2 算法描述

推荐算法的具体步骤如算法1所示。

**算法 1** 推荐算法

输入：用户-项目评分矩阵$\mathbf{R}(m,n)$，目标用户u

输出：用户u的项目推荐列表

1. 依据项目的非零评分集合，计算评分值的密度。
2. 首先根据式(5)计算项目的JS距离；然后根据式(7)计算评分数量因子；最后根据式(8)计算得到最终的项目相似度。
3. 选取最相似的N个项目作为最近邻居集。
4. 根据最近邻居集，预测用户对未评分项目的评分，具体计算公式为：

$$p_{ui}=\frac{\sum_{j\in N_i} Sim_{WAJS}(i,j)\times r_{uj}}{\sum_{j\in N_i}|Sim_{WAJS}(i,j)|} \tag{9}$$

其中，$p_{ui}$是用户u对项目i的预测评分，$N_i$是项目i的最近邻居集，$r_{uj}$是用户u对项目j的评分值。

5. 选取预测值超过阈值的项目作为推荐项目。

## 4 实验结果与分析

### 4.1 数据集

以MovieLens[1)]的最新子集ml-latest-small作为实验数据集，该数据子集包含671个用户对9066部电影的100004条评分记录。评分范围为[0.5,5]，评分间隔为0.5。本文从该数据集中筛选出评分超过20条的电影评分记录作为最终的实验数据集，其中包括671个用户对1303部电影的评分记录，共计69104条，数据稀疏度为92.1%。将数据子集划分为训练集和测试集，训练集占整个子集的80%，剩余的20%为测试集。训练集用于计算最近邻居集，然后根据最近邻居预测出测试集中的项目评分，从而产生推荐结果。

### 4.2 评价指标

评价推荐算法的标准主要包括统计精度和决策支持度两个方面。常用的统计精度指标有平均绝对误差(MAE)和根均方误差(RMSE)，其定义如下[16]：

$$MAE=\frac{\sum_{i=1}^{N}|r_{ui}-p_{ui}|}{N} \tag{10}$$

$$RMSE=\sqrt{\frac{\sum_{i=1}^{N}(r_{ui}-p_{ui})^2}{N}} \tag{11}$$

其中，$N$是有效预测项目的个数。$MAE$和$RMSE$的值越小，表示预测精度越高。

常用的决策支持度指标有查准率(Precision)、查全率(Recall)和综合评价指标(F1-Measure)，其定义如下：

$$Precision=\frac{|I_{pred}\cap I_{real}|}{|I_{pred}|} \tag{12}$$

1) http://www.grouplens.org

$$Recall = \frac{|I_{\mathrm{pred}} \cap I_{\mathrm{real}}|}{|I_{\mathrm{real}}|} \tag{13}$$

$$F1\text{-}Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

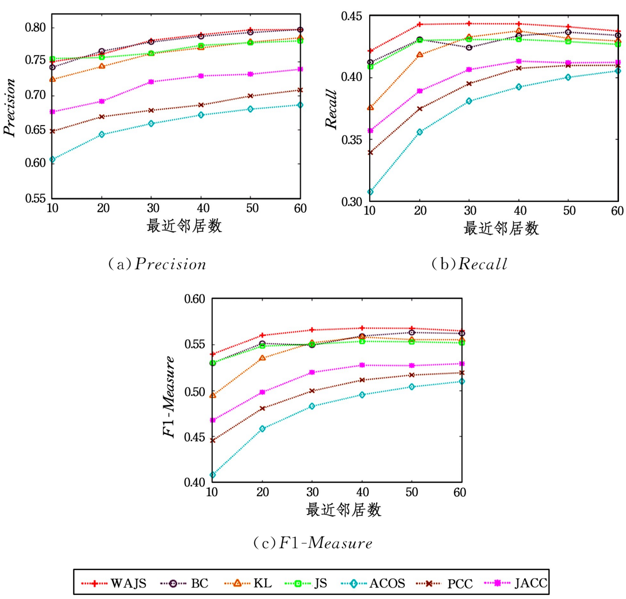
其中，$I_{\mathrm{pred}}$ 是预测评分超过阈值的项目集合，$I_{\mathrm{real}}$ 是用户实际评分超过阈值的项目集合。*Precision* 和 *Recall* 的值越大，表示决策支持的精度越高，但有时 Precision 和 Recall 在性能要求上是相互矛盾的。F1-Measure 是两者的加权调和，综合反映了推荐质量的高低，其值越大，表示推荐结果越好。

### 4.3　实验结果分析

利用 MovieLens 的数据集对本文算法（WAJS）进行性能测试的同时，为了证明其性能优势，还选取了其他一些算法进行对比测试，具体包括经典的相似性度量方法（ACOS[1] 和 PCC[2]）、基于评分个数的方法（JACCARD[3]）和基于评分值密度的方法（KL[12] 和 BC[13]）；另外，还将本文算法与基于原始的 JS 散度的方法做对比，以验证其有效性。在实验中，取高分值 4 分作为推荐阈值。

#### 4.3.1　Precision，Recall 和 F1-Measure

实验测试本文算法及其他算法的 Precision，Recall 和 F1-Measure，如图 1 所示。从中可以看出，基于评分值密度的方法 WAJS，KL，BC 和 JS 明显优于其他方法。主要原因是典型的方法 ACOS 和 PCC 的相似性计算受制于共同评分，可用的数据很有限；而基于评分值密度的算法可以利用项目的所有评分，对数据的利用率远高于 ACOS 和 PCC。基于评分数量的 Jaccard 方法仅仅是利用了评分的总数量，对评分信息的利用过于粗略，因此制约了其评测性能的提升。基于评分值密度的算法中，WAJS 还充分考虑了评分数量的影响，因此在总体性能上略优于 KL，BC 和 JS 方法。




（a）*Precision*　　（b）*Recall*　　（c）*F1-Measure*

图 1　*Precision*，*Recall* 和 *F1-Measure* 的结果比较

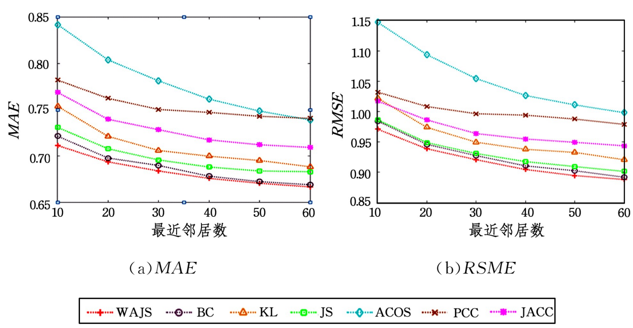
Fig. 1　Result comparison of *Precision*，*Recall* and *F1-measure*

具体而言，在 Precision 的评测中，本文方法 BC 方法要优于 KL 方法，当最近邻居数少于 30 时，本文方法与 BC 方法在 Precision 的评测中各有优劣，但是当最近邻居数大于 30 之后，本文算法的准确率最高。在查全率方面，无论最近邻居数如何设置，本文算法均具有更好的性能。本文算法的 *F1-Measure* 在 0.535 到 0.562 的区间范围内，均高于其他方法，当近邻数为 40 时取得最大值。原理上，JS 散度是在 KL 散度的基础上优化所得，JS 散度较 KL 散度而言具有对称性、有界性等特点，本文又在 JS 散度中加入了评分绝对数量的改良，进一步提升了性能。在 Precision，Recall 与 F1-Measure 的对比中，与 KL 算法相比，基于 JS 的算法的表现更为稳定，波动较小；本文所提 WAJS 方法比原始的 JS 算法有更高的指标值，反映了数量因素在提升算法性能上是有效的。因此，在决策支持精度的性能评测中，本文算法的评测结果优于其他算法。

#### 4.3.2　MAE 和 RMSE

MAE 和 RMSE 主要反映的是预测评分值与实际评分值之间的偏差。在图 2（a）中，基于评分值密度的方法 WAJS，KL 和 BC 明显优于其他两类方法的趋势仍然存在。同时，WAJS 的 *MAE* 总低于 KL 和 BC 的 *MAE* 值，说明 WAJS 方法的误差更小；随着最近邻居数量的增加，WAJS 的 *MAE* 呈现平缓减小的趋势，其总体范围为：$0.667 \leqslant MAE \leqslant 0.711$。在图 2（b）中，随着邻居数的增加，各方法的 *RMSE* 值曲线都有缓慢下降的趋势。其中，基于评分值密度的方法 WAJS，KL 和 BC 依然表现出了更好的性能；且 WAJS 的 *RMSE* 总低于其他方法，其范围为 $0.888 \leqslant RMSE \leqslant 0.971$。

WAJS 在预测准确性方面优于其他方法，其原因主要有两个：1）通过基于评分值的密度分布来度量相似性，摆脱了共同评分的约束，能更充分地利用已有的评分信息；2）在度量相似性的过程中，特意考虑了评分值数量的影响，弥补了基于评分值密度的算法中往往忽略评分值数量的不足。




（a）*MAE*　　（b）*RSME*

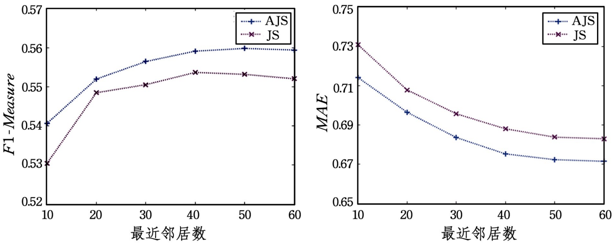
图 2　*MAE* 和 *RSME* 的结果比较

Fig. 2　Result comparison of *MAE* and *RMSE*

#### 4.3.3　整体密度的作用分析

评分值密度是约分化简后的数值，隐藏了原始的分母和分子的实际数值。与平均密度（见式（2））相比，整体密度（见式（4））体现了绝对数量的作用；用整体密度代替原 JS 散度定义式（见式（3））中的平均密度，使得调整后的 JS 散度（见式（5））在计算密度分布相似度时一定程度上反映出了数量的影响。为验证整体密度修正的有效性，分别以式（3）和式（5）作为相似度计算的基础，再依此进行预测和推荐，实验结果的支持度指标 *F1-Measure* 值、精度指标 *MAE* 如图 3 所示。与原始 JS 相比，经整体密度修正所得的 AJS 拥有更高的 *F1-Measure* 和更低的 *MAE*；性能在不同最近邻居维度上均有提升，其中，当最近邻居数为 60 时，AJS 的 *F1-Measure* 值为

0.559,较JS提高了1.33%;*MAE*为0.671,比JS降低了1.68%。因此,整体密度对提高相似性度量的效果是可见的。




(a)*F1-Measure* (b)*MAE*

图3 AJS与JS的性能比较

Fig.3 Performance comparison between AJS and JS

### 4.4 算法分析与总结

1)数据利用率。典型的算法通常使用共同评分进行计算,忽略了非共同评分的价值,并且在本就稀疏的数据环境中,项目间的共同评分更为稀少,这势必造成信息的利用率大幅下降;本文方法则从密度分布入手,充分发挥出所有评分信息的作用,达到了较高的数据利用率。

2)度量的合理性。典型的算法(如皮尔逊相关系数)通常潜在假定项目间存在一定的线性关系,而项目的评分数据是离散的、相对独立的,项目间的关系是多维的、复杂的,假定为线性关系存在局限性;而本文方法则使用改进后的JS散度从评分值密度方面计算项目间的相似性,既可度量线性数据,也可计算非线性数据,无需对评分数据做出任何潜在假定,对项目间的相似度度量更具合理性。

综合各项实验结果和算法分析可知,本文方法拥有更好的性能和效率。

**结束语** 针对协同过滤算法中存在的数据稀疏性问题,引入概率与统计理论领域的JS散度来度量项目之间的相似性;给出了项目评分值的整体密度的定义,并将其应用于对JS散度的改进,改进后的JS散度在计算相似性时更好地体现了评分数量的影响,提高了相似性测量的准确性;最后基于JS散度的相似性设计了相应的推荐方法,该方法以评分值的密度为基础展开计算,不仅充分利用了所有的评分信息,还摆脱了必须依靠共同评分才能完成相似性计算的限制。性能测试结果表明,所提方法有效提高了推荐的质量,较好地解决了数据稀疏性问题,具有很好的应用潜力。

本文方法仍有需要改进的地方:仅从评分这一维度进行相似度的度量忽略了项目自身的属性和特征等因素的衡量,也忽略了推荐结果多样性可能存在不足的问题。今后将围绕这些问题继续展开研究。

## 参考文献

[1] CHOU A Y. The analysis of online social networking:How technology is changing e-commerce purchasing decision[J]. International Journal of Information Systems & Change Management,2010,4(4):353-365.

[2] YANG C C. Correlation coefficient evaluation for the fuzzy interval data[J]. Journal of Business Research,2016,69(6):2138-2144.

[3] GUAN H,GUAN S,ZHAO A. Forecasting Model Based on Neutrosophic Logical Relationship and Jaccard Similarity[J]. Symmetry,2017,9(9):191.

[4] TAKACS G,PILASZY I,NEMETH B,et al. Scalable Collaborative Filtering Approaches for Large Recommender System[J]. Journal of Machine Learning Research,2009,10:623-656.

[5] KIM H N,JI A T,HA I,et al. Collaborative Filtering Based on Collaborative Tagging for Enhancing the Quality of Recommendation [J]. Electronic Commerce Research and Applications,2010,9(1):73-83.

[6] KHROUF H. Hybrid event recommendation using linked data and user diversity[C]// ACM Conference on Recommender Systems. ACM,2013:185-192.

[7] MEYMANDPOUR R,DAVIS J G. Recommendations using linked data[C]// Proceedings of the 5th Ph. d. Workshop on Information and Knowledge. ACM,2012:75-82.

[8] OSTUNI V C,NOIA T D,SCIASCIO E D,et al. Top-N recommendations from implicit feedback leveraging linked open data [C]// ACM Conference on Recommender Systems. ACM,2013:85-92.

[9] BARJASTEH I,FORSATI R,MASROUR F,et al. Cold-Start Item and User Recommendation with Decoupled Completion and Transduction[C]// ACM Conference on Recommender Systems. ACM,2015:91-98.

[10] BINESH N,REZGHI M. A new similarity measure for extraction information from social networks and improve the community detection and recommendation results[C]// Information and Knowledge Technology. IEEE,2015:146-151.

[11] WANG X M,ZHANG X M,WU Y T,et al. A Collaborative Recommendation Algorithm Based on Heuristic Clustering Model and Category Similarity[J]. Acta Electronica Sinica,2016,44(7):1708-1713.(in Chinese)
王兴茂,张兴明,吴毅涛,等.基于启发式聚类模型和类别相似度的协同过滤推荐算法[J].电子学报,2016,44(7):1708-1713.

[12] WANG Y,DENG J Z,DENG Y H,et al. A Collaborative Filtering Recommendation Algorithm Based on Item Probability Distribution[J]. New Technology of Library and Information Service,2016,32(6):73-79.(in Chinese)
王永,邓江洲,邓永恒,等.基于项目概率分布的协同过滤推荐算法[J].现代图书情报技术,2016,32(6):73-79.

[13] PATRA B K,LAUNONEN R,OLLIKAINEN V,et al. A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data[J]. Knowledge-Based Systems,2015,82(3):163-177.

[14] MANNING C D. Foundations of statistical natural language processing[M]. Massachusetts:MIT Press,1999.

[15] MAJTEY A P,LAMBERTI P W,PRATO D P. Jensen-Shannon divergence as a measure of distinguishability between mixed quantum states[J]. Physical Review A,2005,72(5):762-776.

[16] WILLMOTT C J,MATSUURA K. Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance[J]. Climate Research,2005,30(1):79-82.