基于协同过滤的三支粒推荐算法研究

叶晓庆[1] 刘 盾[1] 梁德翠[2]
(西南交通大学经济管理学院 成都 610031)[1] (电子科技大学经济与管理学院 成都 610054)[2]

摘 要 为了降低传统协同过滤算法的推荐成本,并解决该算法评分信息单一的问题,提出了一种基于协同过滤的三支粒推荐算法。该算法在传统协同过滤的基础上,考虑项目特征对用户评分的影响,根据项目特征、粒化用户项目评分矩阵,形成用户对项目粒度的评分矩阵,并以此作为用户偏好的测度依据。同时,该算法在推荐过程中引入三支决策,考虑了推荐过程中产生的误分类成本和学习成本,并基于用户真实的评分偏好构建三支推荐。实验结果显示,基于协同过滤的三支粒推荐算法与传统协同过滤算法相比,不但提高了算法的推荐质量,而且降低了推荐成本。

关键词 协同过滤,三支决策,粒计算,三支粒推荐
中图法分类号 TP18 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.01.014

Three-way Granular Recommendation Algorithm Based on Collaborative Filtering

YE Xiao-qing[1] LIU Dun[1] LIANG De-cui[2]
(School of Economics and Management, Southwest Jiaotong University, Changdu 610031, China)[1]
(School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 610054, China)[2]

Abstract To decrease the recommendation cost and solve the problem of single rating of traditional collaborative filtering algorithm, this paper proposed a three-way granular recommendation algorithm based on collaborative filtering. On the basis of collaborative filtering, this algorithm considers the influence of items' characteristics on users' ratings, and constructs user-item's granulation rating matrix through granulating user-item rating matrix by the characteristics of items, which is used to measure users' preferences. At the same time, this algorithm considers both misclassification cost and teacher cost during the process of recommendation, and constructs three-way recommendation based on users' real preferences on rating. Experimental results show that compared with traditional collaborative filtering algorithm, three-way granular recommendation algorithm based on collaborative filtering not only improves the quality of the recommendation, but also decreases the recommendation cost.

Keywords Collaborative filtering, Three-way decision, Granular computing, three-way granular recommendation

1 引言

近年来,随着互联网的飞速发展,信息膨胀和信息过载已经成为各电商平台和用户面临的难题之一,主要表现在:1)用户很难在海量的信息中筛选出自己感兴趣的信息;2)电商平台在信息过载的境况下很难掌握用户的具体需求,无法做出有效的经营决策,因此各电商平台相继引入推荐系统来解决信息过载问题。推荐系统主要通过建立用户和项目之间的二元关系,根据已有的相似性关系来挖掘每个用户的潜在消费倾向,并针对每一个用户进行个性化推荐,其本质是在大量的信息基础上进行信息过滤。可以看到,推荐系统的运用给电商平台和社交站点带来了巨大的商业利益。例如,Amazon公司的个性化推荐系统为其创造了近35%的销售额。

目前,常见的推荐算法根据其运用环境和推荐过程的不同可分为以下4类:基于知识的推荐、基于内容的推荐、协同过滤推荐以及混合推荐[1-5]。其中,运用最为广泛的是协同过滤推荐,虽然协同过滤算法在各个领域的应用都取得了很好的经济效益,但是该算法依然存在数据稀疏、冷启动等问题[6]。此外,传统的协同过滤算法主要根据用户对项目的单一评分来测度用户的偏好,并没有考虑项目特征对用户评分的影响,导致用户偏好测度不准确;同时,传统协同过滤算法主要根据用户对项目的预测评分来进行推荐,推荐决策只有

到稿日期:2017-03-03 返修日期:2017-06-09 本文受国家自然科学基金项目(71571148,71401026,71201133),四川省科技厅应用基础面上项目(2017JY0220),四川省电子商务与现代物流研究中心项目(DSWL16-2),四川省留学回国人员科技活动择优资助项目(2017-27),川菜发展研究中心项目(CC14SJ12)资助。

叶晓庆(1994—),女,硕士,CCF学生会员,主要研究方向为三支决策与机器学习,E-mail:448013658@qq.com;刘 盾(1983—),男,教授,CCF高级会员,主要研究方向为粗糙集与粒计算,E-mail:newton83@163.com(通信作者);梁德翠(1986—),男,副教授,主要研究方向为三支决策与粒计算,E-mail:decuiliang@126.com。

推荐或不推荐，增大了推荐过程中产生的误分类成本[7]。

为了解决传统协同过滤算法评分信息单一的问题，本文将粒计算思想引入协同过滤算法中。粒计算[8-12]是信息处理中的计算范式和概念，其目的是通过合适的粒度划分和粒层选择来对问题进行求解。粒计算凭借着自身的优势在数据挖掘中得到了广泛的应用[13-15]。粒化是粒计算基础单元的构建，是问题求解空间的一个构造性过程，粒化方法有自顶向下通过分解粗粒子得到细粒子的方法和自底向上将细粒子合并为粗粒子的方法[16]。合理的粒度划分是进行问题求解的关键。

同时，为了降低推荐成本，本文在推荐过程中还引入了三支决策思想。作为决策粗糙集理论[17-18]的一种延伸，三支决策[19-20]主要被用于处理考虑延迟决策的不确定性问题。目前，三支决策已在很多领域都得到成功应用，如垃圾邮件的过滤[21]、政策制定[22]、聚类算法[23]等。同时，三支决策在推荐系统中也得到了很好的应用，ZHANG 等[7]根据推荐过程中产生的误分类成本以及学习成本，提出了三支推荐模型，实验结果证明，三支推荐产生的推荐成本优于二支推荐。之后他们在该模型的基础上提出了基于回归的三支推荐[24]，并通过调整评分阈值使得推荐成本最小。

本文在已有研究的基础上，首先将协同过滤与粒计算相结合，在预测评分过程中考虑项目特征对用户评分的影响，根据项目的特征，采用自顶向下的方法粒化用户对项目的评分信息，将其细分为用户对项目各粒度的评分信息，并以此作为用户偏好的测度依据，提高用户偏好测度的准确性；其次，将三支决策引入粒推荐系统，提出一种基于协同过滤的三支粒推荐算法；最后，利用大众点评成都市的美食数据对模型进行验证。

2　相关概念

本节首先对协同过滤算法、粒度划分和三支决策等基本概念作简要的回顾。

2.1　协同过滤算法

协同过滤算法是目前使用最为广泛的推荐算法，主要有两种形式：基于用户的协同过滤算法[25]和基于项目的协同过滤算法[26]。本文主要讨论基于用户的协同过滤算法，该算法主要根据用户群过去的评分信息，筛选出与当前用户具有相同喜好特征的最近邻用户，并通过最近邻之前的评分记录来预测目标用户对未评分对象的可能评分。

2.2　粒度划分

传统协同过滤算法的数据依据是用户-项目评分矩阵（见表 1）。其中 $I=(i_1,i_2,\cdots,i_n)$ 代表项目集合，$U=(u_1,u_2,\cdots,u_m)$ 代表用户集合，$R=U\times I$ 代表用户 u 对项目 i 的评分 $r_{u,i}$ 所构成的评分集合。协同过滤算法主要通过用户 u 对项目 i 的评分 $r_{u,i}$ 来测度用户的偏好，计算用户之间的相似性。但是，用户购买某个商品或者对某商品给出较高的评分，通常是因为该项目的某些特征满足了用户的偏好需求。例如，用户给予某家餐厅较高的评分的原因可能是这家餐厅的服务好，也可能是这家餐厅的口味佳，用户对餐厅的评分取决于餐厅的口味、环境、服务、位置等多项因素。如果甲和乙都给予某家餐厅好评，而甲是因为喜欢该餐厅的用餐环境，乙是因为喜欢这家餐厅的菜品口味，虽然两人的评分一致，但是偏好并不一致，因此仅根据用户对项目的单一评分并不能准确地测度用户的兴趣偏好，这会降低算法预测评分的准确性。

表 1　用户-项目评分矩阵

Table 1　User-Item rating matrix

用户 \ 项目	i_1	…	i_j	…	i_n
u_1	$r_{1,1}$	…	$r_{1,j}$	…	$r_{1,n}$
…	…	…	…	…	…
u_k	$r_{k,1}$	…	$r_{k,j}$	…	$r_{k,n}$
…	…	…	…	…	…
u_m	$r_{m,1}$	…	$r_{m,j}$	…	$r_{m,n}$

为了准确地测度用户的偏好特征，提高预测评分的准确性，本文在用户偏好测度过程中考虑项目特征对用户评分的影响，将用户对项目的评分粒化为用户对项目各粒度的评分，以此作为测度用户偏好的依据。其中，项目粒度的选取主要取决于项目特征中影响用户偏好的特征。例如，对于某家餐厅，影响用户满意度的因素主要包括餐厅的环境、菜品的口味、服务态度、平均消费以及地理位置等。因此，可以将一家餐厅划分为环境、口味、均价、服务以及位置等粒度，将用户项目评分矩阵转化为用户项目粒度评分矩阵。

2.3　三支决策

传统的协同过滤算法主要根据预测评分的高低进行推荐，推荐状态一般考虑“推荐”或“不推荐”的二支决策模型。本文在二支推荐的基础上引入三支决策思想，由于推荐过程中的误推荐和延迟推荐会产生误分类成本和学习成本，因此以总成本最小为目的进行推荐。在表 2 中，$P(X|[x])$ 表示某一对象 x 的等价类 $[x]$ 属于某一概念 X 的先验概率。

表 2　二支推荐系统

Table 2　Two-way recommendation system

数学条件	决策区域	用户偏好	决策规则
$P(X\|[x])\geqslant\gamma$	正域	喜欢	推荐
$P(X\|[x])<\gamma$	负域	不喜欢	不推荐

三支决策是决策粗糙集理论的一个延伸，主要用于处理边界域的不确定性决策问题。在决策粗糙集理论中，决策者将概念近似地分成 3 个域：正域、负域和边界域。不同域对应不同的决策规则，从而得到三支决策的一种语义：接受正域产生的规则，拒绝负域产生的规则，对边界域产生的规则做延迟决策。在传统决策中，人们往往采用二支决策，即只考虑接受或拒绝两项选择；然而在信息不足或者证据不充分的情况下，无法实时做出接受或拒绝的判定。换言之，此时无论做出上述任何一种决策，造成的决策代价可能都比不做决策高。因此，三支决策在二支决策的基础上提出了延迟决策的选项，以此来规避错误接受或错误拒绝所带来的决策成本。

所谓的三支推荐（见表 3），即在二支推荐的基础上，引入延迟推荐策略，并根据误分类成本和学习成本决定各决策区域的划分阈值 α 和 β。在三支推荐过程中，为了降低误分类

成本，对处于边界域的预测评分采取延迟推荐策略。与此同时，系统需要付出一定的学习成本，对该决策区域的内容进行再学习。

表 3 三支推荐系统

Table 3 Three-way recommendation system

数学条件	决策区域	用户偏好	决策规则
$P(X\|[x])\geqslant\alpha$	正域	喜欢	推荐
$\beta<P(X\|[x])<\alpha$	边界域	不确定	延迟推荐
$P(X\|[x])\leqslant\beta$	负域	不喜欢	不推荐

3 基于协同过滤的三支粒推荐算法

为了提高传统协同过滤算法的推荐质量，降低其推荐成本，本文提出了三支粒推荐算法。该算法主要包括以下两步：1)在预测评分过程中，通过基于协同过滤的粒推荐算法计算预测评分；2)在推荐过程中，根据预测评分、推荐成本以及用户评分偏好实现三支推荐。本节将从粒推荐算法和三支粒推荐模型两个方面对三支粒推荐算法进行详细的阐述。

3.1 粒推荐算法

首先，为了提高用户偏好测度的准确性，解决传统协同过滤算法评分信息单一的问题，三支粒推荐算法将通过基于协同过滤的粒推荐算法来计算预测评分。粒推荐算法的核心是根据项目特征，将表 1 中的用户-项目评分矩阵转化为用户-项目粒度评分矩阵。该算法主要通过用户对项目粒度的偏好程度来判断用户间的相似性，具体步骤如下。

步骤 1 计算用户间的相似性。根据用户-项目粒度评分矩阵来计算用户之间的 Pearson 系数。Pearson 系数在[-1，1]内取值，值越趋近于 1，说明用户间的偏好越一致。具体公式如下：

$$sim(u,v)=\frac{\sum\limits_{k_{s,l}\in K(u)\cap K(v)}(b_{u,s,l}-\overline{b_u})(b_{u,s,l}-\overline{b_u})}{\sqrt{\sum\limits_{k_{s,l}\in K(u)\cap K(v)}(b_{u,s,l}-\overline{b_u})^2}\sqrt{\sum\limits_{k_{s,l}\in K(u)\cap K(v)}(b_{v,s,l}-\overline{b_v})^2}} \tag{1}$$

其中，$K(u)$ 和 $K(v)$ 分别代表 u 和 v 各自的粒度评分集合；$k_{s,l}\in K(u)\cap K(v)$ 代表 u 和 v 共同的粒度评分项；$b_{u,s,l}$ 代表用户 u 对第 s 个项目的第 l 个粒度的评分；$b_{v,s,l}$ 代表用户 v 对第 s 个项目的第 l 个粒度的评分；$\overline{b_u}$ 代表用户 u 对所有项目粒度的平均评分；$\overline{b_v}$ 代表用户 v 对所有项目粒度的平均评分；$sim(u,v)$ 表示用户 u 和用户 v 之间的 Pearson 系数，用来测度 u 和 v 之间的相似程度。

步骤 2 选取最近邻。本文采取固定数量的邻居原则来选取最近邻。根据用户之间的 Pearson 系数，选取与当前用户 Pearson 系数最高的 K 个用户作为最近邻用户。

步骤 3 预测用户对未评分对象的评分。根据近邻用户和当前用户之间的 Pearson 系数以及近邻用户之前的评分记录来计算当前用户对其未评分对象的可能评分。以用户的平均评分作为基准，相似度作为权重来计算目标用户的预测评分。具体公式如下：

$$R_{u,i}^{pre}=\overline{r_u}+\frac{\sum\limits_{v\in k}sim(u,v)*(r_{v,i}-\overline{r_v})}{\sum\limits_{v\in k}(r_{v,i}-\overline{r_v})} \tag{2}$$

其中，$R_{u,i}^{pre}$ 代表用户 u 对项目 i 的预测评分，v 表示用户 u 的最近邻中对 i 进行过评分的前 K 个最近邻，$\overline{r_u}$ 和 $\overline{r_v}$ 分别代表用户 u 和 v 对项目的评分均值。

与传统的协同过滤算法相比，粒推荐算法最大的特点在于它考虑了项目特征对用户评分的影响，通过细分评分粒度形成了用户-项目粒度评分矩阵，并以此作为用户偏好的测度依据，计算用户之间的相似性。

3.2 三支粒推荐模型

同时，为了降低推荐过程中的推荐成本，本文在粒推荐的基础上引入了三支决策，并构建了三支粒推荐模型。三支粒推荐的核心思想是在推荐过程中根据预测评分、推荐成本以及用户的评分偏好决定推荐、不推荐还是延迟推荐。其中推荐成本主要包括错误推荐所产生的误分类成本，以及延迟推荐所产生的学习成本。具体推荐成本如表 4 所列。

表 4 推荐成本矩阵

Table 4 Recommendation cost matrix

决策规则	用户的偏好	
	喜欢(L)	不喜欢(D)
推荐(P)	λ_{PL}	λ_{PD}
延迟推荐(B)	λ_{BL}	λ_{BD}
不推荐(N)	λ_{NL}	λ_{ND}

其中，λ_{PL}，λ_{BL} 和 λ_{NL} 分别代表将用户喜欢(L:$Like$)的项目推荐、延迟推荐以及不推荐给用户时所产生的成本。λ_{PD}，λ_{BD} 和 λ_{ND} 分别代表将用户不喜欢(D:$Dislike$)的项目推荐、延迟推荐以及不推荐给用户时所产生的成本。λ_{PD} 和 λ_{NL} 为误分类成本，λ_{BL} 和 λ_{BD} 为学习成本。

当给定推荐成本矩阵时，需要通过式(3)来计算最小化决策成本，以确定三支决策各决策域的划分阈值，即 α 和 β。

$$TC(\alpha,\beta)=C_P(\alpha,\beta)+C_B(\alpha,\beta)+C_N(\alpha,\beta) \tag{3}$$

其中，$TC(\alpha,\beta)$ 代表阈值为 (α,β) 时所产生的总成本，$C_P(\alpha,\beta)$ 代表推荐产生的成本，$C_B(\alpha,\beta)$ 代表延迟推荐产生的成本，$C_N(\alpha,\beta)$ 代表不推荐产生的成本，其计算方式如下：

$$\begin{cases}C_P(\alpha,\beta)=\lambda_{PL}N_{PL}+\lambda_{PD}N_{PD}\\ C_B(\alpha,\beta)=\lambda_{BL}N_{BL}+\lambda_{BD}N_{BD}\\ C_N(\alpha,\beta)=\lambda_{NL}N_{NL}+\lambda_{ND}N_{ND}\end{cases} \tag{4}$$

其中，N_{PL}，N_{BL}，N_{NL} 分别代表将用户喜欢的项目推荐、延迟推荐、不推荐给用户的数量，N_{PD}，N_{BD}，N_{ND} 分别代表将用户不喜欢的项目推荐、延迟推荐、不推荐给用户的数量，即在不同条件下组合 $\langle u,i\rangle$ 的数量，具体计算公式如下：

$$\begin{cases}N_{PL}=|\{\langle u,i\rangle|R_{u,i}^{real}\geqslant R_u^l,P_{u,i}(X|[x])\geqslant\alpha\}|\\ N_{PD}=|\{\langle u,i\rangle|0<R_{u,i}^{real}<R_u^l,P_{u,i}(X|[x])\geqslant\alpha\}|\\ N_{BL}=|\{\langle u,i\rangle|R_{u,i}^{real}\geqslant R_u^l,\beta<P_{u,i}(X|[x])<\alpha\}|\\ N_{BD}=|\{\langle u,i\rangle|0<R_{u,i}^{real}<R_u^l,\beta<P_{u,i}(X|[x])<\alpha\}|\\ N_{NL}=|\{\langle u,i\rangle|R_{u,i}^{real}\geqslant R_u^l,0<P_{u,i}(X|[x])\leqslant\beta\}|\\ N_{ND}=|\{\langle u,i\rangle|0<R_{u,i}^{real}<R_u^l,0<P_{u,i}(X|[x])\leqslant\beta\}|\end{cases} \tag{5}$$

其中，$|\{\langle u,i\rangle|*\}|$ 代表条件“$*$”下组合 $\langle u,i\rangle$ 的数量；$P_{u,i}(X|[x])$ 代表用户 u 对项目 i 的喜好程度，即用户 u 喜欢

项目 i 的概率；$R_{u,i}^{real}$ 代表用户 u 对项目 i 的真实评分；R_u^l 代表用户喜欢某项目的评分阈值，若 $R_{u,i}^{real} \geqslant R_u^l$ 则代表用户喜欢项目 i，反之则不喜欢。

由此可见，在三支粒推荐过程中，推荐决策取决于阈值 α、阈值 β 以及用户对项目的喜好程度 $P_{u,i}(X|[x])$。

在推荐过程中，误分类成本通常高于延迟推荐所产生的学习成本，学习成本通常高于正确推荐所产生的成本，基于以上情况提出以下假设：

$$\lambda_{PL} \leqslant \lambda_{BL} < \lambda_{NL}; \lambda_{ND} \leqslant \lambda_{BD} < \lambda_{PD} \tag{6}$$

为了使总体推荐成本 $C(\alpha,\beta)$ 最小，根据贝叶斯决策过程，寻找最优阈值 α 和 β，基于式(6)中的假设，可计算最优阈值 (α,β) 为：

$$\begin{cases} \alpha = \dfrac{\lambda_{PD} - \lambda_{BD}}{(\lambda_{PD} - \lambda_{BD}) + (\lambda_{BL} - \lambda_{PL})} \\ \beta = \dfrac{\lambda_{BD} - \lambda_{ND}}{(\lambda_{BD} - \lambda_{ND}) + (\lambda_{NL} - \lambda_{BL})} \end{cases} \tag{7}$$

同时，考虑一种特殊的情况：

$$(\lambda_{NL} - \lambda_{BL})(\lambda_{PD} - \lambda_{BD}) > (\lambda_{BD} - \lambda_{ND})(\lambda_{BL} - \lambda_{PL}) \tag{8}$$

可得，$0 \leqslant \beta < \alpha \leqslant 1$。

特别地，二支推荐过程只考虑了错误推荐所产生的误分类成本，因此其最优阈值 γ 为：

$$\gamma = \frac{\lambda_{PD} - \lambda_{ND}}{(\lambda_{PD} - \lambda_{ND}) + (\lambda_{NL} - \lambda_{PL})} \tag{9}$$

同时，为了测度用户偏好的所属论域，还需要确定用户对项目的喜好程度，即 $P_{u,i}(X|[x])$。预测评分的高低是反映用户对项目喜好程度的依据，一般而言，预测评分越高说明用户喜欢该项目的概率就越高。但是用户对项目的评分不但取决于用户对项目的喜好程度，还受用户评分偏好的影响。由于用户的评分基准不同，为了避免用户评分偏好对用户偏好测度产生影响，本文采用最小最大值法对预测评分进行标准化处理，处理过程如式(10)所示：

$$P_{u,i}(X|[x]) = \begin{cases} 1, & R_{u,i}^{pre} > \mathrm{Max}R_u \\ \dfrac{R_{u,i}^{pre} - \mathrm{Min}R_u}{\mathrm{Max}R_u - \mathrm{Min}R_u}, & \mathrm{Min}R_u \leqslant R_{u,i}^{pre} \leqslant \mathrm{Max}R_u \\ 0, & R_{u,i}^{pre} < \mathrm{Min}R_u \end{cases} \tag{10}$$

其中，$R_{u,i}^{pre}$ 代表用户 u 对项目 i 的预测评分，$\mathrm{Max}R_u$ 代表用户 u 的最大评分，$\mathrm{Min}R_u$ 代表用户 u 的最小评分。

因此，在三支粒推荐算法中，可以根据用户对项目的喜爱程度(即 $P_{u,i}(X|[x])$ 和阈值 (α,β))将用户划分在不同的决策区域，即正域、边界域和负域，并对不同的域采用不同的决策规则，即推荐、延迟推荐和不推荐。

4　实验结果及分析

为了验证三支粒推荐算法相比传统协同过滤算法在推荐质量和推荐成本上得到改善，本节将对在大众点评网站上采集的真实评分数据进行实验分析。

4.1　实验测度标准

4.1.1　推荐质量测度

推荐算法的推荐质量主要通过评分预测准确度来衡量。评分预测准确度通过算法产生的预测评分和用户真实评分之间的误差来反映算法的推荐质量。预测准确度主要包括 4 个指标：平均绝对误差(*MAE*)、平均平方误差(*MSE*)、均方根误差(*RMSE*)以及标准平均误差(*NMAE*)。本文主要采用平均绝对误差(*MAE*)和均方根误差(*RMSE*)来衡量算法的预测准确度，具体计算方法如式(11)和式(12)所示。*MAE* 和 *RMSE* 越小说明评分预测越准确，算法的推荐质量越高。

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |R_{u,i}^{real} - R_{u,i}^{pre}| \tag{11}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} |R_{u,i}^{real} - R_{u,i}^{pre}|^2} \tag{12}$$

4.1.2　评分阈值的测度

为了计算算法的推荐成本，首先需要测度用户对项目真实的喜好，并通过评分阈值 R_u^l 来判断用户是否喜欢某项目，若 $R_{u,i}^{real} \geqslant R_u^l$，则代表用户喜欢项目 i，反之则不喜欢。由于每个用户的评分标准不同，某些用户偏好给出较高的评分，某些用户偏好给出较低的评分，因此需要根据用户的评分标准设置评分阈值 R_u^l。为了消除用户评分标准对用户真实偏好测度的影响，本文在三支推荐过程中通过用户历史评分偏好对预测评分进行归一化处理，为了保证偏好测度的一致性，我们在用户真实评分偏好测度中也需要消除用户评分偏好对用户真实偏好测度的影响，因此本文根据用户的历史评分偏好和评分规则来设置用户的评分阈值 R_u^l，具体如式(13)所示：

$$R_u^l = \mathrm{Max}R_u - \xi * Step \tag{13}$$

其中，$\mathrm{Max}R_u$ 代表用户 u 的历史最高评分。$Step$ 代表评分规则中的评分步长。5 分制评分规则中，评分可以是(1,2,3,4,5)，评分步长为 1；10 分制评分规则中，评分可以是(2,4,6,8,10)，评分步长为 2。ξ 为评分步长的参数值。

4.1.3　推荐成本测度

在计算推荐成本时，主要考虑错误推荐所产生的误分类成本以及延迟推荐所产生的学习成本，因此认为正确的推荐决策不产生推荐成本，即 $\lambda_{PL} = \lambda_{ND} = 0$。根据以上假设，可以得到总推荐成本和平均推荐成本分别为：

$$\begin{aligned} TC(\alpha,\beta) &= C_P(\alpha,\beta) + C_B(\alpha,\beta) + C_N(\alpha,\beta) \\ &= \lambda_{PD}N_{PD} + \lambda_{BL}N_{BL} + \lambda_{BD}N_{BD} + \lambda_{NL}N_{NL} \end{aligned} \tag{14}$$

$$AC = \frac{TC(\alpha,\beta)}{N_{PL} + N_{PD} + N_{BL} + N_{BD} + N_{NL} + N_{ND}} \tag{15}$$

4.2　数据集

本文从大众点评网上采集了四川省成都市 2004 年 12 月到 2014 年 1 月期间 2205 个用户对 2052 家餐厅的 79274 条评分数据作为实验数据[1)]。其中每个用户至少评论过 15 家餐厅，并且每个餐厅至少被 15 个用户评论过，评分数据主要包括用户对餐厅的总体评分以及对每家餐厅的环境、服务以及口味这 3 个粒度的详细评分，将其作为项目评分依据和粒度评分依据。本文保留了 69364 条评分数据作为训练集，将其余 9910 条数据作为测试集。

4.3　实验结果

为了验证三支粒推荐算法在推荐质量和推荐成本上得到了改善，本文根据大众点评网获取的评分数据构建三支粒推

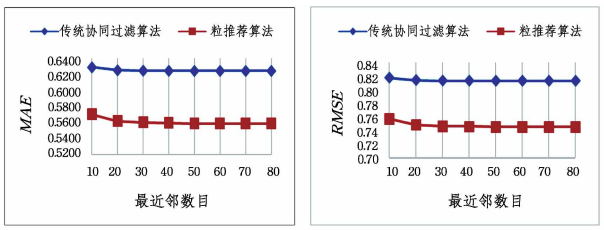
1) http://www.dianping.com/chengdu

荐模型。我们可以获取该网站上用户对餐厅的直接评分，也可以根据标签评分以及文本挖掘来获取用户对餐厅各特征粒度的偏好。由于数据获取的限制，本文主要采集了用户对餐厅的直接评分以及用户对餐厅的环境、口味、服务 3 个标签的评分，以此作为三支粒推荐算法的数据依据。

根据以上数据，本文将用户对餐厅的直接评分作为用户-项目评分矩阵的数据依据，并根据采集的标签评分，将用户对餐厅的评分粒化为用户对餐厅环境、口味以及服务 3 个粒度的评分，以此作为用户-项目粒度评分矩阵的数据依据；同时根据以上两个数据集构建三支粒推荐模型。

4.3.1 推荐质量

首先，为了测度三支粒推荐算法的推荐质量，本文对比了粒推荐算法和传统协同过滤算法在最近邻分别为 10，20，…，80 时，基于训练集产生的预测评分和测试集真实评分之间的平均绝对误差(*MAE*)和均方根误差(*RMSE*)，实验结果如图 1 所示。



(a)平均绝对误差比较 (b)均方根误差比较

图 1 预测准确性的比较

Fig. 1 Comparisons of prediction accuracy

实验结果表明，在保持最近邻个数取值不变的情况下，粒推荐算法的平均绝对误差和均方根误差均低于传统协同过滤算法。由此可见，与传统协同过滤算法相比，粒推荐算法的预测准确度更高，推荐质量更好，该算法通过细分项目的评分粒度可以更准确地测度用户的偏好特征，提高算法的推荐质量。

4.3.2 推荐成本

为了测度各算法的推荐成本，本文首先根据训练集产生的预测评分进行推荐决策，然后根据测试集中的真实评分和评分阈值来判断用户对项目的真实喜好，计算推荐成本。由于本文从大众点评网上获取的评分是 5 分制的，因此令 $R_u^l=\mathrm{Max}\,R_u-1$，即用户喜欢某项目的评分阈值为该用户历史最高评分减 1。

为了验证基于协同过滤的粒推荐算法相比传统协同过滤算法的推荐成本得到改善，本文对比了两种算法在不同成本条件下基于三支推荐产生的平均推荐成本。其中 CF 代表传统协同过滤算法，G_CF 代表基于协同过滤的粒推荐算法。表 5 列出了当 $\lambda_{PD}=80$ 时，两种算法在不同的 λ_{NL}，λ_{BL} 和 λ_{BD} 下产生的平均推荐成本；表 6 列出了当 $\lambda_{NL}=80$ 时，两种算法在不同的 λ_{PD}，λ_{BL} 和 λ_{BD} 下产生的平均推荐成本。

根据表 5 和表 6 可知，在三支推荐环境下，当成本条件相同时，粒推荐算法产生的平均推荐成本均低于传统协同过滤算法。另一方面，与传统协同过滤算法相比，粒推荐算法的预测准确度更高，减少了系统的误分类数量，降低了推荐过程中产生的推荐成本。同时，根据表 5 和表 6 可知：当 λ_{PD} 和 λ_{NL} 保持不变时，λ_{BL} 越小，平均推荐成本越小；当 λ_{PD} 和 λ_{BL} 保持不变时，λ_{NL} 越小，平均推荐成本越小；当 λ_{NL} 和 λ_{BL} 保持不变时，λ_{PD} 越小，平均推荐成本越小。

此外，在表 5 和表 6 的实验基础上，进一步考虑了不同成本环境下三支粒推荐算法基于 8724 条预测评分产生的误推荐数和延迟推荐数，如表 7 和表 8 所列。其中，N_{PD} 代表将不喜欢的项目推荐给用户的个数，N_{NL} 代表将喜欢的项目不推荐给用户的个数，N_B 则代表延迟推荐的个数。

表 5 固定 λ_{PD} 时平均推荐成本的比较

Table 5 Comparisons of average recommendation cost when λ_{PD} is fixed

$(\lambda_{PD},\lambda_{NL})$	$\lambda_{BL}=\lambda_{BD}=35$		$\lambda_{BL}=\lambda_{BD}=30$		$\lambda_{BL}=\lambda_{BD}=25$		$\lambda_{BL}=\lambda_{BD}=20$		$\lambda_{BL}=\lambda_{BD}=15$	
	CF	G_CF	CF	G_CF	CF	G_CF	CF	G_CF	CF	G_CF
(80,120)	29.83	28.61	27.73	26.52	24.83	23.76	21.41	20.25	17.13	16.21
(80,110)	29.56	28.37	27.53	26.38	24.55	23.62	21.10	20.14	16.91	16.09
(80,100)	29.19	28.09	27.23	26.27	24.40	23.52	20.87	20.04	16.71	15.93
(80,90)	29.02	27.73	26.98	26.02	24.19	23.41	20.54	19.88	16.55	15.73
(80,80)	28.34	27.25	26.81	25.65	23.95	23.22	20.37	19.80	16.25	15.64
(80,70)	27.38	26.51	26.18	25.21	23.61	22.93	20.16	19.66	15.97	15.51
(80,60)	—	—	25.26	24.51	23.26	22.47	19.87	19.46	15.76	15.40
(80,50)	—	—	23.43	22.96	22.34	21.84	19.56	19.07	15.52	15.27
(80,40)	—	—	—	—	20.47	20.16	18.73	18.45	15.30	14.96

注：“—”表示不满足式(6)和式(7)的成本条件，即二支决策的情况，下同

表 6 固定 λ_{NL} 时平均推荐成本的比较

Table 6 Comparisons of average recommendation cost when λ_{NL} is fixed

$(\lambda_{PD},\lambda_{NL})$	$\lambda_{BL}=\lambda_{BD}=35$		$\lambda_{BL}=\lambda_{BD}=30$		$\lambda_{BL}=\lambda_{BD}=25$		$\lambda_{BL}=\lambda_{BD}=20$		$\lambda_{BL}=\lambda_{BD}=15$	
	CF	G_CF	CF	G_CF	CF	G_CF	CF	G_CF	CF	G_CF
(120,80)	32.73	31.90	29.63	28.83	25.61	25.09	21.20	20.66	16.65	16.00
(110,80)	32.03	31.16	29.24	28.43	25.29	24.80	21.05	20.58	16.56	15.92
(100,80)	31.23	30.23	28.54	27.68	25.02	24.45	20.90	20.41	16.48	15.85
(90,80)	29.96	28.90	27.88	26.78	24.61	23.98	20.65	20.12	16.39	15.74
(80,80)	28.34	27.25	26.81	25.65	23.95	23.22	20.37	19.80	16.25	15.64
(70,80)	26.31	25.13	25.20	24.06	23.08	22.31	19.93	19.28	16.05	15.43
(60,80)	23.46	22.03	23.28	22.05	21.53	20.79	19.20	18.44	15.77	15.09
(50,80)	—	—	20.50	19.18	19.72	18.80	17.93	17.22	15.22	14.51
(40,80)	—	—	—	—	16.88	15.89	16.14	15.28	14.35	13.50

表 7　固定 λ_{PD} 时不同成本条件下三支粒推荐产生的误推荐数和延迟推荐数

Table 7　Numbers of misclassification and delay recommendation generated by three-way granular recommendation under condition of different cost when λ_{PD} is fixed

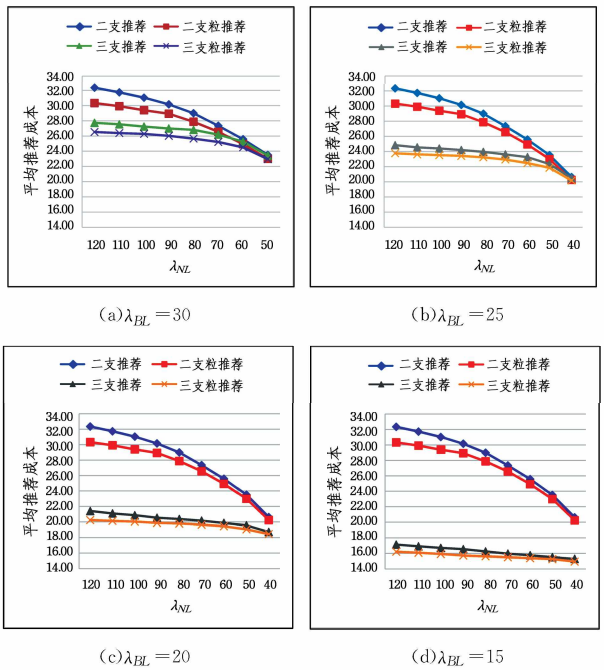
$(\lambda_{PD},\lambda_{NL})$	$\lambda_{BL}=\lambda_{BD}=30$			$\lambda_{BL}=\lambda_{BD}=25$			$\lambda_{BL}=\lambda_{BD}=20$			$\lambda_{BL}=\lambda_{BD}=15$		
	N_{PD}	N_B	N_{NL}	N_{PD}	N_B	N_{NL}	N_{PD}	N_B	N_{NL}	N_{PD}	N_B	N_{NL}
(80,120)	665	4568	343	323	5914	280	137	6933	225	58	7593	191
(80,110)	665	4476	388	323	5870	304	137	6899	243	58	7566	202
(80,100)	665	4364	451	323	5803	343	137	6849	269	58	7528	214
(80,90)	665	4197	532	323	5689	402	137	6796	295	58	7491	225
(80,80)	665	3956	649	323	5525	483	137	6716	343	58	7442	252
(80,70)	665	3505	880	323	5294	598	137	6578	414	58	7371	287
(80,60)	665	2511	1421	323	4872	806	137	6345	532	58	7274	343
(80,50)	665	540	2618	323	3746	1421	137	5931	735	58	7070	451
(80,40)	—	—	—	323	1235	2979	137	4659	1421	58	6662	649

表 8　固定 λ_{NL} 时不同成本条件下三支粒推荐产生的误推荐数和延迟推荐数

Table 8　Numbers of misclassification and delay recommendation generated by three-way granular recommendation under condition of different cost when λ_{NL} is fixed

$(\lambda_{PD},\lambda_{NL})$	$\lambda_{BL}=\lambda_{BD}=30$			$\lambda_{BL}=\lambda_{BD}=25$			$\lambda_{BL}=\lambda_{BD}=20$			$\lambda_{BL}=\lambda_{BD}=15$		
	N_{PD}	N_B	N_{NL}	N_{PD}	N_B	N_{NL}	N_{PD}	N_B	N_{NL}	N_{PD}	N_B	N_{NL}
(120,80)	137	6104	649	73	6858	483	42	7389	343	25	7763	252
(110,80)	199	5808	649	91	6708	483	56	7296	343	26	7726	252
(100,80)	271	5414	649	137	6438	483	67	7197	343	33	7656	252
(90,80)	423	4789	649	208	6858	483	85	7023	343	42	7557	252
(80,80)	665	3956	649	323	6708	483	137	6716	343	58	7442	252
(70,80)	1048	2821	649	565	6438	483	230	6234	343	79	7262	252
(60,80)	1618	1445	649	961	6858	483	423	5401	343	137	6884	252
(50,80)	2204	173	649	168	6708	483	844	4028	343	271	6194	252
(40,80)	—	—	—	2291	6438	483	1618	2057	343	665	4736	252

由表 7 和表 8 可知，在 λ_{PD} 和 λ_{NL} 分别保持不变的情况下，λ_{BL} 越小，N_{PD} 和 N_{NL} 越小，N_B 越大。由此可见，随着学习成本的减小，三支粒推荐算法产生的两种误推荐数量均减小，延迟推荐数量增加。同时，在 λ_{PD} 和 λ_{BL} 保持不变的情况下，λ_{NL} 越小，N_{NL} 越大；在 λ_{NL} 和 λ_{BL} 分别保持不变的情况下，λ_{PD} 越小，N_{PD} 越大，即随着误分类成本的减少，误推荐数量增加，延迟推荐数量减小。根据以上实验结果可知，三支推荐过程中某一决策区域的决策成本越小，该区域的决策数量就越大。由此可见，三支推荐结果偏向推荐成本较小的决策区域。



(a)$\lambda_{BL}=30$　(b)$\lambda_{BL}=25$

(c)$\lambda_{BL}=20$　(d)$\lambda_{BL}=15$

图 2　各算法在不同成本条件下的平均推荐成本

Fig. 2　Average recommendation cost of algorithms under condition of different cost

综合表 5—表 8 的实验结果可知，三支推荐产生的推荐结果偏向推荐成本较小的决策区域，降低了系统的平均推荐成本。

为了进一步验证三支粒推荐算法在推荐成本上的改善情况，本文对比了粒推荐算法和传统协同过滤算法在不同决策环境(即二支决策和三支决策)和成本条件下产生的平均推荐成本(见图 2)，其中 $\lambda_{PD}=80$。

由图 2 可得，在相同的成本条件和算法下，三支推荐产生的推荐成本均低于二支推荐；在相同的成本条件和决策环境下，粒推荐产生的推荐成本均低于传统协同过滤算法；同时，系统的学习成本越低，即 λ_{BL} 越低，三支推荐相比二支推荐的成本改善越明显。因此，与二支推荐相比，三支推荐通过延迟决策减少了系统的误分类数量，降低了系统的平均推荐成本；同时，与传统协同过滤算法相比，粒推荐算法的预测准确度更高，减少了系统的误分类数量，降低了系统的推荐成本。

4.3.3　实验结果分析

通过上述实验分析，可以得到以下结果：

(1)基于协同过滤的粒推荐算法与传统协同过滤算法相比，能够更好地测度用户的偏好特征，在提高算法推荐质量的同时，减少了算法的误分类成本，从而降低了系统的推荐成本。

(2)相比于二支推荐，三支推荐通过延迟推荐减少了误推荐数量，降低了系统的推荐成本。

(3)由表 7 和表 8 可知，三支推荐结果偏向决策成本较小的决策区域(λ_{NL} 和 λ_{PD} 区域)，降低了系统的推荐成本。

综上所述，相比于传统协同过滤算法，三支粒推荐算法解决了评分信息单一的问题，提高了算法的推荐质量。与此同时，在推荐过程中，三支粒推荐算法通过三支决策降低了系统的推荐成本。

结束语　考虑到传统协同过滤推荐算法中存在评分信息单一以及推荐成本较高的不足，本文提出了基于协同过滤的

三支粒推荐算法。为了提高算法的推荐质量，三支粒推荐算法在预测评分过程中，考虑了项目特征对用户偏好的影响，并根据项目特征细分评分粒度，解决了传统协同过滤算法评分信息单一的问题。同时，为了降低系统的推荐成本，该算法在推荐过程中引入了三支决策思想，并根据预测评分、推荐成本以及用户真实的评分偏好实现了三支推荐。实验结果显示，相比于传统协同过滤算法，三支粒推荐算法在推荐质量和推荐成本上都得到了改善。

由于数据获取的限制，本文在三支粒推荐模型构造中只考虑了餐厅的环境、口味以及服务 3 个粒度对用户偏好的影响，在实际运用过程中，可以考虑更多的粒度和粒层，从而可以更准确地测度用户的偏好，提高算法的推荐质量。在未来的工作中，一方面将优化粒度和粒层选取的过程，进一步提高用户偏好测度的准确性；另一方面，将通过更完备的数据集来测试三支粒推荐算法的有效性。

参 考 文 献

[1] ZHAO L, HU N J, ZHANG S Z. Algorithm design for personalization recommendation systems[J]. Journal of Computer Research & Development, 2002, 39(8): 986-991.

[2] DENG A L, ZHU Y Y, SHI B L. A collaborative filtering recommendation algorithm based on Item rating prediction[J]. Journal of Software, 2003, 14(9): 54-65.

[3] DESHPANDE M, KARYPIS G. Item-based top-N recommendation algorithms[J]. ACM Transactions on Information Systems, 2003, 22(1): 143-177.

[4] FENG Z J, XIAN T, FENG G J. An optimized collaborative filtering recommendation algorithm[J]. Journal of Computer Research & Development, 2004, 41(10): 1842-1847.

[5] WANG G X, LIU H P. Survey of personalized recommendation system [J]. Computer Engineering and Application, 2012, 48(7): 66-76. (in Chinese)
王国霞，刘贺平. 个性化推荐系统综述[J]. 计算机工程与应用，2012，48(7)：66-76.

[6] SHU X H. Research of sparsity and cold start problem in collaborative filtering[D]. Hangzhou: Zhejiang University, 2005. (in Chinese)
孙小华. 协同过滤系统的稀疏性与冷启动问题研究[D]. 杭州：浙江大学，2005.

[7] ZHANG H R, MIN F. Three-way recommender systems based on random forests[J]. Knowledge-Based Systems, 2015, 91: 275-286.

[8] PAWLAK Z. Rough sets[J]. International Journal of Computer and Information Science, 1982, 11(5): 341-356.

[9] ZHANG B, ZHANG L. Theory and applications of problem solving [M]. North-Holland: Elsevier Science Publishers, 1992.

[10] ZADEH L A. Fuzzy Logic=Computing with words [M]// Computing with words in Information/Intelligent Systems 1. Physica-Verlag HD, 1999: 103-111.

[11] YAO Y Y. Granular computing: past, present, and future[C]// International Conference on Rough Sets and Knowledge Technology. Springer-Verlag, 2008: 27-28.

[12] YAO Y Y. Granular Computing and sequential three-way decisions[C]// International Conference on Rough Sets and Knowledge Technology. Springer Berlin Heidelberg, 2013: 16-27.

[13] BU D B, BAI S, LI G J. Principle of Granualarity in Clustering and Classification [J]. Chinese Journal of Computers, 2002, 25(8): 810-816. (in Chinese)
卜东波，白硕，李国杰. 聚类/分类中的粒度原理[J]. 计算机学报，2002，25(8)：810-816.

[14] WEI L, MIAO D. Application of granular computing in knowledge reduction[C]// International Conference on Rough Sets and Knowledge Technology. Springer-Verlag, 2006: 357-362.

[15] WU W Z, LEUNG Y, MI J S. Granular computing and knowledge reduction in formal contexts[J]. IEEE Transactions on Knowledge & Data Engineering, 2008, 21(10): 1461-1474.

[16] WANG G Y, ZHANG Q H, HU J. An overview of granular computing[J]. CAAI Transactions on Intelligent Systems, 2007, 2(6): 8-26. (in Chinese)
王国胤，张清华，胡军. 粒计算研究综述[J]. 智能系统学报，2007，2(6)：8-26.

[17] YAO Y Y. Probabilistic rough set approximations [J]. International Journal of Approximate Reasoning, 2008, 49(2): 255-271.

[18] LIU D, LI T R, DA R. Probabilistic model criteria with decision-theoretic rough sets [J]. Information Sciences, 2011, 181(17): 3709-3722.

[19] YAO Y Y. Three-way decisions with probabilistic rough sets [J]. Information Sciences, 2010, 180(3): 341-353.

[20] LIU D, YAO Y Y, LI T R. Three-way Decision-theoretic Rough Sets[J]. Computer Science, 2011, 38(1): 246-250. (in Chinese)
刘盾，姚一豫，李天瑞. 三枝决策粗糙集[J]. 计算机科学，2011，38(1)：246-250.

[21] ZHOU B, YAO Y Y, LUO J G. Cost-sensitive three-way email spam filtering[J]. Journal of Intelligent Information Systems, 2014, 42: 19-45.

[22] LIU D, LI T R, LIANG D C. Three-way government decision analysis with decision-theoretic rough sets[J]. International Journal of Uncertainty Fuzziness and Knowledge-Based Systems, 2012, 20: 119-132.

[23] YU H, ZHANG C, WANG G. A tree-based incremental overlapping clustering method using the three-way decision theory[J]. Knowledge-Based Systems, 2015, 91: 189-203.

[24] ZHANG H R, MIN F, SHI B. Regression-based three-way recommendation [M]. Elsevier Science Inc, 2017.

[25] SARWAR B, KARYPIS G, et al. Analysis of recommendation algorithms for e-commerce [C]// ACM Conference on Electronic Commerce. ACM, 2000: 158-167.

[26] SARWAR B, KARYPIS G, KONSTON J, et al. Item-based collaborative filtering recommendation algorithms [C] // International Conference on World Wide Web. ACM, 2001: 285-295.

[27] LIU J G, ZHOU T, GUO Q, et al. Overview of the Evaluated Algorithms for the Personal Recommendation Systems[J]. Complex Systems and Complexity Sciences, 2009, 6(3): 1-10. (in Chinese)
刘建国，周涛，郭强，等. 个性化推荐系统评价方法综述[J]. 复杂系统与复杂性科学，2009，6(3)：1-10.