计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
50(10): 2185-2194, 2013

基于模糊 C 均值聚类的环境感知推荐算法

张付志　常俊风　周全强
(燕山大学信息科学与工程学院　河北秦皇岛　066004)
(河北省计算机虚拟技术与系统集成重点实验室(燕山大学)　河北秦皇岛　066004)
(xjzfz@ysu.edu.cn)

Context-Aware Recommendation Algorithm Based on Fuzzy *C*-Means Clustering

Zhang Fuzhi, Chang Junfeng, and Zhou Quanqiang
(*School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei* 066004)
(*Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province (Yanshan University), Qinhuangdao, Hebei* 066004)

Abstract Aiming at the deficiencies of existing context-aware recommendation algorithms, this paper proposes a context-aware recommendation algorithm based on fuzzy *C*-means clustering. Firstly, the fuzzy *C*-means clustering algorithm is used to perform clustering of historical contextual information and produce clusters and membership matrix. Then contextual information for the active user is matched with the cluster of historical contextual information, and non-membership data, which accord with the condition, are mapped into membership data by using membership degree of clustering as a mapping coefficient. Finally, we choose ratings of membership users, which conform to the active contextual information, to generate recommendation for the user. Compared with the existing algorithms, the proposed algorithm can not only solve the problem of inaccuracy recommendation due to the change of user context, but also overcome traditional hard clustering by using fuzzy *C*-means clustering algorithm. Moreover, the problem of data sparseness caused by clustering is solved by using membership mapping function. The experiments are conducted on the MovieLens dataset and the effectiveness of the proposed algorithm is verified.

Key words context-aware; fuzzy *C*-means clustering; membership matrix; membership mapping; recommenda-tion algorithm

摘　要　针对现有环境感知推荐算法存在的不足,提出一种基于模糊 C 均值聚类的环境感知推荐算法.首先采用模糊 C 均值聚类算法对历史环境信息进行聚类,产生聚类及隶属矩阵;然后匹配活动用户环境信息与历史环境信息聚类,采用聚类隶属度作为映射系数将符合条件的非隶属数据映射为隶属数据,最终选择与活动环境匹配的隶属用户评分数据为用户产生推荐.同现有算法相比,该算法不仅解决了因用户环境改变不能准确推荐项目的问题,而且通过采用模糊聚类算法克服了传统硬聚类问题,并且借助于隶属映射函数解决了聚类产生的数据稀疏性问题.在 MovieLens 数据集上比较了新算法和其他算法的性能,验证了所提算法的有效性.

关键词　环境感知;模糊 C 均值聚类;隶属矩阵;隶属映射;推荐算法

中图法分类号　TP391

收稿日期:2011-12-05;**修回日期**:2012-08-24

基金项目:国家自然科学基金项目(61379116);河北省自然科学基金项目(F2011203219,F2013203124);教育部高等学校博士学科点专项科研基金项目(20101333110013);河北省高等学校科学技术研究重点项目(ZH2012028)

传统推荐算法的不足之处在于无法根据不同的环境向用户推荐不同的项目.例如,在一些场合某种产品对一个用户的实用性很大程度上依赖于时间(如季节或者月份、早上或者下午),或者依赖于该用户在哪些情况下与谁共享该产品.在这些情况下,推荐算法不仅需要采用历史数据挖掘用户的偏好,更需要根据目标用户的当前环境(如时间、位置或同行者)发现用户的偏好.在一些研究领域中[1]已经证明用户的购买行为会根据环境不同而改变.Palmisano等人[2]的研究表明,将环境信息加入到用户的建模过程中不仅可以提高对用户行为的预测精度,而且可以识别更多用户同源的购买方式.

在推荐系统领域,环境感知推荐(context-aware recommendation)方法可分为3类[3]:环境前过滤(pre-filtering)、环境后过滤(post-filtering)和环境建模(contextual modeling).在环境前过滤方法中,首先根据用户的当前环境信息对数据集进行过滤,然后在过滤后的数据集上利用传统的推荐算法进行推荐.在环境后过滤方法中,首先在整个数据集上进行top-N推荐,然后根据用户的当前环境信息对推荐列表进行调整.在环境建模方法中,环境信息直接用于评分预测的建模过程中.一些学术研究表明,将环境信息引入到推荐算法中有助于提高推荐系统的性能.Adomavicius等人[4]提出了一种基于降维的环境感知推荐方法,通过将基于环境信息的多维推荐模型转化为二维推荐模型以提高推荐精度.但是,这种方法容易产生数据稀疏性问题.Hussein等人[5-6]提出了一种基于规则的服务选择方法,利用环境信息对原始数据进行过滤,并采用基于项目的协同过滤推荐算法完成预测过程.由于该方法在推荐前对原始数据进行过滤,也容易产生数据稀疏性问题.Panniello等人[7]提出了一种环境前过滤推荐算法和两种环境后过滤推荐算法,并在两种不同的数据集上对3种算法进行了实验对比.但是,这些过滤算法对于环境硬分解导致偏离用户偏好的问题无能为力.Liu等人[8]提出了一种环境感知服务推荐方法,通过对环境信息进行分类,分别采用距离相似度和语义相似度计算方法对环境相似性进行度量.但是,这种环境相似度计算方法没有考虑环境粒度和稀疏度对推荐产生的负面影响.Hussein[9]提出了一种基于移动环境的推荐系统分析与设计方法,但是所采用的k-Means聚类方法和传统的协同推荐算法在推荐精度方面还有待于提高.文献[10]通过项目的星级可视化标签(visualized tags)显示用户对项目的评分,并且通过用户选择所需环境进行推荐.这种方法虽然提供了可视化效果,但是通过选择环境类型导致的环境硬分解及数据稀疏性问题仍有待于解决.文献[11]将原始数据根据评分时的环境信息进行分类,通过选取与活动环境相同的数据为用户进行推荐,但是对于硬分解所产生的数据稀疏性问题仍然没有很好解决.

针对现有环境感知推荐算法存在的不足,本文提出了一种基于模糊C均值聚类(fuzzy C-means clustering, FCMC)的环境感知推荐算法.具体贡献如下:1)提出了一种基于模糊C均值聚类的环境感知推荐模型;2)提出了一种历史环境发现算法,通过对历史环境信息进行聚类,克服了环境信息硬分解导致的忽略用户偏好问题;3)提出了一种环境匹配及隶属映射算法,有效解决了评分数据稀疏性问题;4)设计了相应的环境感知推荐算法,并在MovieLens数据集上进行了模拟实验,验证了该算法的有效性.

1 基础定义及相关知识介绍

1.1 基础定义

为了在动态环境中提高推荐系统的推荐质量,我们不仅要获得用户对项目的偏好评分,而且需要捕获用户评分时的环境信息,以便根据用户所处环境推荐不同的项目.

定义1. 环境(context).推荐系统中用户与系统交互时周围环境的特征信息,表示为$Context=\{\boldsymbol{E}_1,\boldsymbol{E}_2,\cdots,\boldsymbol{E}_n\}$,其中$\boldsymbol{E}_i(i=1,2,\cdots,n)$表示环境信息类型向量.

形式上讲,传统的推荐效用函数R可表示为

$$R:User\times Item\rightarrow Rating,$$

其中,$User$表示用户,$Item$表示项目,$Rating$表示用户对项目的评分.引入环境信息后,环境感知推荐效用函数R可表示为

$$R:User\times Item\times Context\rightarrow Rating,$$

其中,$Context$表示环境信息.

为了更好地理解环境感知推荐,我们以表1中用户在不同环境下的项目评分为例来说明环境信息对推荐产生的影响,"√"对应的选项表示用户所处的环境,"?'表示待预测评分.

表1中的环境Contexts包括Place, Time和Fellow 3类,而每一类又分别拥有自己的子类.显然,环境感知推荐算法需要根据不同的环境层次为用户推荐不同的项目.例如,$User1$和$User2$在Indoor,

Table 1 Rating of Two Users in Different Contexts

表 1 两个用户在不同环境下的项目评分

Item ID	Contexts							Rating of *User*1	Rating of *User*2
	Place		Time			Fellow			
	Indoor	Outdoor	Morning	Noon	Afternoon	Friends	Family		
*Item*1	√				√		√	2	2
		√	√			√		5	5
*Item*2	√				√		√	4	?
		√	√			√		2	?
*Item*3	√				√		√	1	?
		√	√			√		5	?
*Item*4	√				√		√	3	3
		√	√			√		4	4

Afternoon 和 Family 环境下和在 Outdoor,Morning 和 Friends 环境下对 *Item*1 和 *Item*4 的偏好相同,因此可将 *User*1 和 *User*2 看作相似用户,那么可以根据 *User*1 在不同环境下对项目的偏好为 *User*2 作推荐. 从表 1 可以看出,在 Indoor, Afternoon 和 Family 环境下,*User*1 对 *Item*2 的评分为 4,而对 *Item*3 的评分为 1,因此在该环境下为 *User*2 推荐 *Item*2;同样,在 Outdoor, Morning 和 Friends 环境下,*User*1 对 *Item*2 的评分为 2,而对 *Item*3 的评分为 5,因此为 *User*2 推荐 *Item*3.

定义 2. 活动用户(active user). 推荐系统中需要得到推荐服务的目标用户,记为 u_a.

定义 3. 活动环境(active context). 推荐系统中活动用户 u_a 当前所处的环境,表示为 $E_a=(e_{a1},e_{a2},\cdots,e_{ak})$, $k=1,2,\cdots,n$.

1.2 模糊 C 均值聚类(FCMC)[12-13]理论简介

模糊聚类算法是一种基于函数最优方法的聚类算法,使用微积分计算技术求最优代价函数. 在基于概率算法的聚类方法中将使用概率密度函数,为此要假定合适的模型. 模糊聚类算法中向量可以同时属于多个聚类,从而解决假定合适模型困难的问题.

隶属度函数是指对论域 D 中的任一元素 X,都有一个数 $\mu_A(X)\in[0,1]$ 与之对应,$\mu_A(X)$ 称为 X 对 A 的隶属度;而当 X 在 D 中变动时,$\mu_A(X)$ 就是一个函数,称为 A 的隶属度函数. 其中,A 称为 D 上的模糊集. 隶属度 $\mu_A(X)$ 越接近于 1,说明 X 属于 A 的程度越高;$\mu_A(X)$ 越接近于 0,说明 X 属于 A 的程度越低. 对于论域中的有限个对象 $\{X_1,X_2,\cdots,X_n\}$,模糊集 A 形式化为

$$A=\{(\mu_A(\boldsymbol{X}_i),\boldsymbol{X}_i)\}. \tag{1}$$

模糊 C 均值聚类是用隶属度确定每个数据点属于某个聚类程度的一种聚类方法. FCMC 把 n 个向量 $\boldsymbol{X}_i(i=1,2,\cdots,n)$ 分成 c 个模糊组,并求每组的聚类中心,使得非相似性指标的价值函数达到最小. FCMC 使用模糊划分,使得每个给定数据点用值在 0～1 之间的隶属度来确定属于各个组的程度. 与引入模糊划分相适应,隶属矩阵 $\boldsymbol{U}$ 允许有取值在 0～1 之间的元素. 通过数据归一化,一个数据集的隶属度的总和等于 1,表示为式(2):

$$\sum_{i=1}^{c}u_{ij}=1,\forall j=1,2,\cdots,n. \tag{2}$$

那么,FCMC 的价值函数一般化形式如下:

$$J(U,H_1,\cdots,H_c)=\sum_{i=1}^{c}J_i=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^m d_{ij}^2, \tag{3}$$

其中,$u_{ij}\in[0,1]$, H_i 为模糊聚类 i 的聚类中心,$d_{ij}=\|H_i-X_j\|$ 为第 i 个聚类中心与第 j 个数据点之间的欧氏距离;$m\in[1,\infty)$ 是一个加权指数,这里设 $m=2$. 我们采用拉格朗日最值法可以求得式(3)达到最小值的必要条件,构造函数如下:

$$\begin{aligned}\bar{J}(U,H_1,\cdots,H_c,\lambda_1,\cdots,\lambda_n)=&\\ J(U,H_1,\cdots,H_c)+\sum_{j=1}^{n}\lambda_j\Big(\sum_{i=1}^{c}u_{ij}-1\Big)=&\\ \sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^m d_{ij}^2+\sum_{j=1}^{n}\lambda_j\Big(\sum_{i=1}^{c}u_{ij}-1\Big),&\end{aligned} \tag{4}$$

其中,$\lambda_j(j=1,2,\cdots,n)$ 是 n 个拉格朗日乘子. 对所有输入参数求导,用 h_i 表示 H_i 的第 i 个属性,得到式(3)达到最小时的必要条件为

$$h_i=\frac{\sum_{j=1}^{n}u_{ij}^m x_j}{\sum_{j=1}^{n}u_{ij}^m}, \tag{5}$$

$$u_{ij}=\frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}}. \tag{6}$$

基于上述两个条件,模糊 C 均值聚类算法通过迭代得到聚类中心和隶属矩阵.

2 基于模糊 C 均值聚类的环境感知推荐

2.1 环境感知推荐模型

环境感知推荐模型通过引入用户所处环境信息向用户提供适合当前环境的推荐结果,该模型主要包括用户视图层、环境提供器层、环境感知层和推荐服务层,其结构如图 1 所示:

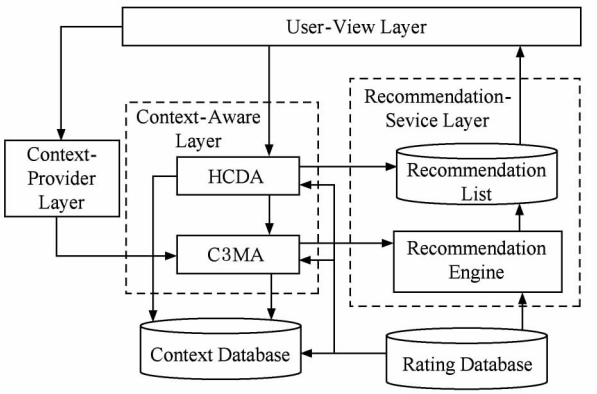

Fig. 1 Model of context-aware recommendation.

图 1 环境感知推荐模型

在图 1 中,各层之间通过相互协作实现环境感知推荐. 首先,通过用户视图层和环境提供器层分别得到用户信息和用户活动环境信息,并将其发送到环境感知层;然后,环境感知层通过历史环境发现算法(historical context discovery algorithm, HCDA)发现历史环境信息的相关性,通过环境匹配及隶属映射算法(context matching and membership mapping algorithm, C3MA)将用户活动环境信息同历史环境信息进行匹配,并且对匹配出的评分数据进行隶属映射,以降低评分数据的稀疏性;最后,推荐服务层从环境信息库和用户评分库中分别选择匹配出的环境信息和历史用户评分,通过推荐引擎计算得到推荐列表,并将推荐结果发送到用户视图层.

在上述模型中,环境提供器层可以直接采用现有的环境提供技术实现,在此不作详细阐述. 下面重点介绍该模型中的 3 个核心模块所使用的算法,即历史环境发现算法、环境匹配及隶属映射算法和环境感知推荐算法.

2.2 历史环境发现算法

现有的一些研究采取环境信息硬分解或者硬聚类的方法,使得环境信息只能属于其中的一个聚类,并且聚类之间无相关性. 因此,只能从环境信息聚类中取得极少部分用户评分,难以准确定位用户的偏好. 为了解决这一问题,我们采用模糊 C 均值聚类方法,使环境信息同时属于多个聚类,从而解决环境信息硬分解和硬聚类带来的问题. 利用模糊 C 均值方法对历史环境信息进行聚类,得到的不是环境信息属于或者不属于某个聚类的关系,而是属于某个聚类的程度. 为了更好地发现用户行为和环境信息的相关性,我们提出一种历史环境发现算法. 利用该算法可以得到多个环境信息聚类以及环境信息隶属于聚类程度的隶属矩阵 $\boldsymbol{U}$. 目前确定聚类数据的普遍规则是 $c_{\max}\leqslant\sqrt{n}$, $c_{\max}$ 为最大聚类数, n 为数据集的样本数[14]. 为了降低该算法时间复杂度,本文聚类数确定为 $c=\frac{c_{\max}}{2}$.

为了向活动用户提供更加精确的推荐结果,首先需要发现用户历史评分及环境信息的相关和相似程度,并根据相关和相似程度将环境信息归属到不同的相似聚类中.

定义 4. 环境数据库(contextual database). 存储用户评分时所处环境信息的数据库,记为 CDB.

历史环境发现算法 HCDA 的基本思想是:首先提取 CDB 中用户的历史评分环境,然后采用模糊 C 均值聚类方法将提取的历史评分环境信息进行聚类,得到各个聚类及隶属矩阵 $\boldsymbol{U}$. 具体步骤如下:

1) 从环境数据库 CDB 中提取用户的历史评分环境信息;

2) 填充随机数 $u_{ij}\in[0,1]$ 初始化隶属矩阵 $\boldsymbol{U}$, 要求满足 $\sum_{i=1}^{c}u_{ij}=1, j=1,2,\cdots,n$;

3) 利用式(7)计算初始聚类中心 $H_i(i=1,2,\cdots,c)$,利用式(8)计算数据点和聚类中心的距离:

$$h_i=\frac{\sum_{j=1}^{n}(u_{ij}^{(l-1)})^m p_j^{(l)}}{\sum_{j=1}^{n}(u_{ij}^{(l-1)})^m},1\leqslant i\leqslant c, l=1,2,3,\cdots, \tag{7}$$

$$d_{ij}^2=(h_i-p_j^{(l)})^T A(h_i-p_j),\quad 1\leqslant i\leqslant c,1\leqslant j\leqslant n; \tag{8}$$

4) 利用式(6)重新计算隶属矩阵 $\boldsymbol{U}$,然后转步骤 3),直到隶属矩阵 $\boldsymbol{U}$ 变化值小于某阈值 ε;最终得到稳定聚类中心 $H_i(i=1,2,\cdots,c)$ 和隶属矩阵 $\boldsymbol{U}$.

根据上述算法思想,给出历史环境发现算法的描述如下:

算法 1. 历史环境发现算法 *HCDA*.

输入:用户历史评分环境数据库 CDB;

输出:历史评分环境聚类中心 $H_i(i=1,2,\cdots,c)$,隶属矩阵 $\boldsymbol{U}$.

① 初始化阈值 ε 和 m;

② $context \leftarrow getContext(\text{CDB})$; /* $\{E_j=(e_{j1},e_{j2},\cdots,e_{jn})\mid j=1,2,\cdots,n\}$ */

③ 初始化隶属矩阵 $\boldsymbol{U}=\{[u_{ij}]_{c\times n}\mid u_{ij}\in[0,1],\forall I,\forall j\sum_{i=1}^{c}u_{ij}=1\}$;

④ repeat

⑤ for $l=1$ to $context.size$ do

⑥ 利用式(7)计算聚类中心;

⑦ 利用式(8)计算距离;

⑧ end for

⑨ for $k=1$ to n do

⑩ for $i=1$ to c do

⑪ if($d_{ik}^2>0$) then

⑫ 利用式(6)计算 u_{ij};

⑬ else

⑭ $u_{ij}\leftarrow 0$;

⑮ end if

⑯ end for

⑰ end for

⑱ until $\|U^{(l)}-U^{(l-1)}\|<\varepsilon$;

⑲ return $H_i(i=1,2,\cdots,c)$ and U.

该算法由两部分组成:第 1 部分为行①~③,用 0~1 的随机数初始化隶属矩阵,并且从 CDB 中提取环境信息;第 2 部分为行④~⑲,可分为两个阶段,第 1 阶段为行④~⑧,分别利用式(7)和式(8)计算聚类原型和距离;第 2 阶段为行⑨~⑲,实现划分矩阵的更新,直到小于某个确定的阈值或者相对于上次的改变量小于某个阈值,最终得到聚类中心和隶属矩阵 $\boldsymbol{U}$.

2.3 环境匹配及隶属映射算法

2.3.1 环境匹配

环境匹配的目的是通过寻找到与活动环境信息相似的环境信息集,选择符合活动环境信息的历史用户评分数据用于用户推荐. 从环境提供器中得到捕获的活动用户环境信息后,需要将活动环境信息与历史环境信息进行匹配,得到与活动环境信息最相似的历史环境信息. 环境匹配的结果不是一条环境记录,而是与活动环境相同或者相近的环境信息聚类. 这样不仅可以改善传统的环境硬匹配问题,而且能够解决环境感知推荐中出现的环境无匹配问题.

环境可以进行细粒度划分,产生不同层次的环境信息. 例如,将年龄信息划分为 3 个年龄段,将位置信息划分为 2 个层次:

$$Age=\{\text{youth},\text{middleageed},\text{old}\}\Rightarrow\{\{10-18\},\{19-50\},\{50-\}\},$$

$$Location\Rightarrow\{\text{Indoor},\text{Outdoor}\}\Rightarrow\{\{\text{mall},\text{cinema},\text{lab},\cdots\},\{\text{square},\text{seaside},\cdots\}\}.$$

通常情况下,环境信息的取值有两种类型:一种是数值型,例如整型或者实数型数据;另一种是非数值型,例如位置信息等字符型数据或者时间类型数据. 我们需要将这些非数值型数据进行量化,转化为数值型数据.

对于数值型环境信息,我们采用 Manhattan 距离公式[15]计算两条环境记录中某一类环境信息之间的距离,并且采用相似度计算式(10)计算两类环境信息之间的相似度.

$$d(E_i,E_i')=\sum_{j=1}^{n}|e_{ij}-e_{ij}'|, \tag{9}$$

$$sim(E_i,E_i')=\frac{1}{1+d(E_i,E_i')}. \tag{10}$$

在历史环境发现过程中得到 c 个聚类中心,表示为 $H_i(i=1,2,\cdots,c)$. 其中,每个中心由一组向量组成 $H_i=(h_{i1},h_{i2},\cdots,h_{ik})$,活动环境为 $E_a=(e_{a1},e_{a2},\cdots,e_{ak})$. 为了找到与活动环境最相似的聚类中心,需要将式(9)和式(10)变形,得到环境信息相似度计算公式:

$$d(H_i,E_a)=\sqrt{\sum_{q=1}^{k}(h_{iq}-e_{aq})^2},i=1,2,\cdots,c, \tag{11}$$

$$sim(H_i,E_a)=\frac{1}{1+d(H_i,E_a)}=\frac{1}{1+\sqrt{\sum_{q=1}^{k}(h_{iq}-e_{aq})^2}},i=1,2,\cdots,c. \tag{12}$$

对于非数值型环境信息,需要进行环境信息量化后才能利用式(11)和式(12)计算相似度. 例如,对于 Boolean 型数据,可将 true 转化为 1,将 false 转化为 0,然后使用式(11)和式(12)计算环境信息相似度. 同样,对于日期时间型环境信息,可将某一时刻(如 2011-01-01 0:00 时)作为原点,并以某一粒度的时间单位(例如秒)为度量手段,将日期时间型数据转化为数值型数据,然后利用式(11)和式(12)计算环境信息相似度.

2.3.2 隶属映射

将环境信息相似度进行排序,选择与活动环境最相似的环境信息聚类,称之为活动环境聚类.根据活动环境聚类中的环境信息可以得到部分用户信息及用户在这些环境下的历史评分数据,利用这些评分数据为活动用户做推荐.然而,由于用户的历史评分不可能属于相同环境,而是分布在不同的环境中,因此根据匹配出的环境信息聚类只能得到用户的极少部分历史评分数据,这无疑使评分数据更加稀疏.为此,我们提出隶属映射函数 $Mapping(y)$.其基本思想是通过将 2 个不同环境的隶属度比值作为系数,使 2 个不同环境中的评分数据可以映射到彼此聚类中,得到隶属于不同聚类的评分数据.

定义 5. 隶属映射函数.对任意需要映射至不同聚类的用户评分数据,通过将该数据当前所在聚类隶属度与需要映射至聚类的隶属度比值作为系数所形成的一元线性函数,其表达式如下:

$$Mapping(y): x_j = y_j(x_i) = \begin{cases} x_i, & E_i = E_j, \\ \dfrac{B_i}{B_j} \times x_i, & E_i \neq E_j, \end{cases}$$

其中,$i, j = 1, \cdots, n$,x_j 表示我们所需要得到的将评分 x_i 映射到 E_j 环境信息聚类后的新评分值;x_i 表示用户在 E_i 环境信息聚类中的评分;B_i 表示 x_i 的环境信息在 E_i 中的隶属度;B_j 表示 x_i 的环境信息在 E_j 中的隶属度.

该函数通过使用隶属矩阵将处于不同环境聚类中的历史用户评分数据进行相互映射,以便降低评分数据的稀疏性,提高推荐的精度.在隶属矩阵中,每一个聚类都存在隶属度值 B_i,并且 $\sum_{i=1}^{n} B_i = 1$.下面以表 2 中的用户评分数据为例,说明隶属映射函数的应用.

Table 2 The User's Rating

表 2 用户评分数据

Item ID	Rating	Context Clustering
*Item*1	2	E_1
*Item*2	3	E_1
*Item*3	4	E_2
*Item*4	5	E_2
*Item*5	4	E_3
*Item*6	3	E_3

假设环境隶属度分别为 $B_1 = 0.5$, $B_2 = 0.3$, $B_3 = 0.2$, *Item*1 和 *Item*2 所属的评分环境聚类为 E_1, *Item*3 和 *Item*4 所属的评分环境聚类为 E_2, *Item*5 和 *Item*6 所属的评分环境聚类为 E_3.假设根据环境匹配得到与活动环境最相似的历史环境聚类为 E_1,则只能得到环境聚类 E_1 中用户对 *Item*1 和 *Item*2 的评分.为了降低评分数据的稀疏性,我们需要将环境聚类 E_2 或者 E_3 中的评分数据映射到 E_1 中.例如,根据上述隶属映射函数,可将环境聚类 E_2 中 *Item*3 的评分映射为环境聚类 E_1 中的评分 $x_1 = \dfrac{B_2}{B_1} \times x_2 = \dfrac{0.3}{0.5} \times 4 = 2.4$,这样就完成了评分数据的隶属映射.

将用户在推荐系统中进行评分活动时的信息记录称为用户活动概貌(user activity profile, UAP),其中包括用户对项目的评分数据,以及在评分时所处的环境信息.根据以上分析,我们提出一种环境匹配及隶属映射算法.其基本思想是:首先将感知的活动环境信息进行量化,然后将量化后的环境信息与环境聚类中的所有聚类中心进行匹配,选择与活动环境最相似的环境聚类,并采用隶属映射函数将非活动环境聚类中的评分数据映射为活动环境聚类中的评分,进而得到项目-评分矩阵.具体步骤如下:

1) 获取当前活动用户的活动环境;

2) 将非数值型环境信息转化为数值型环境信息,并根据式(11)和式(12)计算环境相似度;

3) 计算所有历史环境聚类与当前活动环境的环境相似度,选择相似度最高者为活动环境聚类;

4) 使用隶属映射函数将活动用户在其非活动环境中的项目评分映射为活动环境中的评分;

5) 采用隶属映射函数将过滤出的用户在非活动环境聚类中的项目评分映射为活动环境聚类中的评分;

6) 生成活动用户-项目评分矩阵.

根据上述步骤,给出环境匹配及隶属映射算法的描述如下:

算法 2. 环境匹配及隶属映射算法 *C3MA*.

输入:聚类中心集合 $\{H_1, H_2, \cdots, H_c\}$, UAP;

输出:匹配活动环境的用户-项目评分矩阵.

① 初始化 $r = []$; /* 初始化评分矩阵 */

② $Context_a \leftarrow AwareContext(u_a)$; /* $u_a \in UAP$ */

③ $Context'_a \leftarrow FindMaxMatching(\{H_i \mid i = 1, 2, \cdots, c\}, Context_a)$;

/* 寻找与当前用户活动环境匹配相似度最高的环境聚类 */

④ for each user $u_i \in UAP$ do

⑤ for each rating $r_{ik} \in u_i$ do

⑥ if($Context_{ik} = Context'_a$) then

⑦ $r'_{ik}(r_{ik} \times 1$; /* r'_{ik} 为环境匹配后的映射评分 */

⑧ else

⑨ $w \leftarrow r_{ik}/B_a$; /* 隶属映射系数 */

⑩ $r'_{ik} \leftarrow w \times r_{ik}$;

⑪ end if

⑫ end for

⑬ end for

⑭ $matrix_{context_a}(user, item, rating) \leftarrow r$;

⑮ return $matrix_{context_a}(user, item, rating)$.

/* 返回匹配活动环境的用户-项目评分矩阵 */

该算法主要包括两部分:第 1 部分为行②③,将捕获到的活动环境信息与环境聚类中心匹配,得到符合活动环境信息的历史环境信息聚类;第 2 部分为行④~⑮,选择匹配的历史环境聚类中的用户和评分数据,并采用隶属映射函数将非活动环境聚类中的用户评分映射为活动环境聚类中的评分,以降低评分数据的稀疏性,最终返回活动环境下的评分数据.

2.4 环境感知推荐算法

本节在上述 2 种算法的基础上,提出一种新的环境感知推荐算法(context-aware recommendation algorithm, CARA),以便向用户推荐符合其当前环境的项目. 其基本思想如下:首先,在离线状态下采用历史环境发现算法,得到环境信息聚类及隶属矩阵;然后,根据环境提供器提供的活动环境信息,利用环境匹配及隶属映射算法得到符合活动环境信息的评分数据 $matrix_{context_a}(user, item, rating)$;最后,根据传统的相似度计算公式和评分预测公式为用户进行推荐,产生用户推荐列表.

本文采用相关相似性计算用户之间的相似度,其度量公式如下[16]:

$$sim(i,j) = \frac{\sum_{v \in I_{ij}} (R_{i,v} - \overline{R_i})(R_{j,v} - \overline{R_j})}{\sqrt{\sum_{v \in I_{ij}} (R_{i,v} - \overline{R_i})^2} \sqrt{\sum_{v \in I_{ij}} (R_{j,v} - \overline{R_j})}}, \quad (13)$$

其中,$R_{i,v}$ 表示用户 i 对项目 v 的评分,和 $\overline{R_j}$ 分别表示用户 i 和用户 j 对项目的平均评分,I_{ij} 表示用户 i 和用户 j 共同评分的项目集合.

假设用 N_u 表示用户 u 的最近邻居集合,则用户 u 对项目 v' 的预测评分 $P_{u,v'}$ 计算公式如下[16]:

$$P_{u,v'} = \overline{R_u} + \frac{\sum_{n \in N_u} sim(u,n)(R_{n,v'} - \overline{R_n})}{\sum_{n \in N_u} |sim(u,n)|}, \quad (14)$$

其中,$\overline{R_u}$ 和 $\overline{R_n}$ 分别表示用户 u 和用户 n 的平均评分,$sim(u,n)$ 表示用户 u 和用户 n 之间的相似度,$R_{n,v'}$ 表示用户 n 对项目 v' 的评分.

根据上述算法思想,给出环境感知推荐算法的描述如下:

算法 3. 环境感知推荐算法 *CARA*.

输入:活动用户 u_a,UAP;

输出:项目预测评分(predicted rating).

① $\{H_i, u_{ij} \mid 1 \leqslant i \leqslant c, 1 \leqslant j \leqslant n\} \leftarrow HCDA(UAP)$;

② $ActiveContext \leftarrow getContext(ContextSupplier(u_a))$;

③ $matrix_{context_a}(user, item, rating) \leftarrow C3MA(\{H_i \mid i = 1, 2, \cdots, c\}, UAP)$;

④ for $i = 1$ to c do

⑤ $neighborArray[i] \leftarrow sim(u_i, u_a)$; /* $sim(u_i, u_a)$ 通过式(13)计算得到 */

⑥ end for

⑦ $neighborArray' \leftarrow Ranking(neighborArray)$;

⑧ 利用式(14)计算预测评分;

⑨ return predicted rating.

该算法分为两部分:第 1 部分为行①~③,根据历史环境发现算法得到环境信息聚类和环境隶属矩阵,利用环境匹配及隶属映射算法得到符合活动环境的评分矩阵;第 2 部分为行④~⑨,首先利用式(13)计算活动用户和活动环境聚类中用户的相似度,然后按相似度降序排序得到邻居集列表,最后利用式(14)计算项目预测评分.

3 实验与评价

3.1 数据集

实验数据集取自 MovieLens 站点(http://movielens.umn.edu),该站点是一个基于 Web 的研究型推荐系统,注册用户必须至少对其中的 15 部电影进行评价后才可以使用该系统. 该数据集包括 943 个用户对 1 682 部电影的 10 万条投票记录,用户评分数据集的稀疏度为 $1 - 100\,000/(943 \times 1\,682) = 0.937\,0$. 下载的文件中包括:$u.data$ 为未排序的全部 10 万条投票记录;$u.item$ 为电影信息,包括电影的名称

和分类；*u.user* 为用户信息，包括用户的年龄、性别和职业；*u.genre* 为电影的分类信息. 本文使用 *u.user* 文件中的 *Age*，*Sex* 和 *Occupation* 作为环境信息. 为了方便起见，将 *Age*，*Sex* 和 *Occupation* 分别简写为 A，S 和 O. 将这些环境信息进行组合，可得到 7 种不同的环境信息类型，表示为 $Context=\{A, S, O, AS, AO, SO, ASO\}$. 可以任意选取不同的环境信息类型进行实验.

3.2 评价指标

推荐质量的评价标准主要有统计精度度量和决策支持精度度量两类[17]. 统计精度度量方法中常用的指标是平均绝对偏差(*MAE*)；决策支持精度度量中常用的指标为召回率(*recall*)和准确率(*precision*)[18].

1) *MAE* 指标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. 显然，*MAE* 越小推荐质量越高. 假设预测的用户评分集合表示为 $\{p_1, p_2, \cdots, p_N\}$，对应的实际用户评分集合为 $\{q_1, q_2, \cdots, q_N\}$，则平均绝对偏差定义为[13,16]

$$MAE=\frac{\sum_{i=1}^{N}|p_i-q_i|}{N}. \tag{15}$$

2) 召回率反映待推荐项目被推荐的比率，准确率表示算法推荐成功的比率. 召回率和准确率是一对互斥指标，通常将两者结合，使用 *F-measure* 指标评价推荐的质量. *F-measure* 值越大推荐质量越高. *F-measure* 指标的计算公式如下[19]：

$$F\text{-}measure=\frac{2\times recall\times precison}{recall+precison}. \tag{16}$$

3.3 实验结果及分析

1) 不同用户不同环境信息推荐结果比较

随机选取 7 个用户，计算在 A，AS，SO 和 ASO 这 4 个环境中 7 个用户推荐首项目的情况，具体实验结果如表 3 所示：

Table 3 Recommendation Items of Seven Users in Four Contexts

表 3 在 4 种环境下 7 个用户推荐的项目

User ID	*A*	*AS*	*SO*	*ASO*
User1	61	189	160	171
User2	288	237	255	302
User3	181	331	264	321
User4	358	264	11	327
User5	25	229	342	186
User6	303	268	533	185
User7	176	418	169	307

2) *MAE* 和 *F-measure* 值比较

随机选取 7 个用户作为目标用户，在环境信息类型为 $\{A, S, O\}$ 的条件下，将本文提出的 CARA 算法与 EPF[20-21] 和 Filter PoF[12] 两种环境感知推荐算法进行比较，实验结果如图 2 所示：

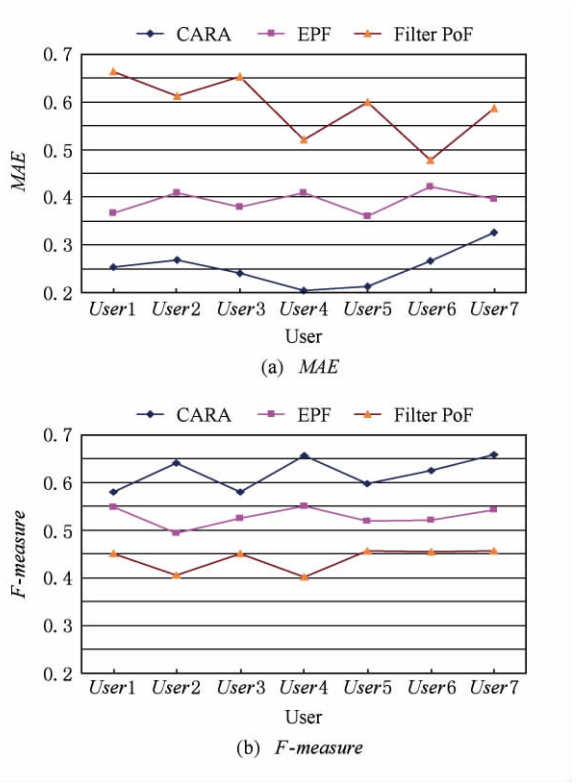

Fig. 2 Comparison of *MAE* and *F-measure* for three algorithms.

图 2 3 种算法的 *MAE* 和 *F-measure* 值比较

从图 2(a)可以看出，在随机选取的 7 个用户中，Filter PoF 算法的 *MAE* 值保持在 0.478～0.664 之间，EPF 算法的 *MAE* 值在 0.365～0.417 之间，CARA 算法的 *MAE* 值在 0.204～0.329 之间. 显然，CARA 算法的 *MAE* 值最小，推荐精度最高. 从图 2(b)可以看出，CARA 算法的 *F-measure* 值在 0.587～0.653 之间，EPF 算法的 *F-measure* 值在 0.496～0.551 之间，Filter PoF 算法的 *F-measure* 值在 0.402～0.457 之间. 显然，CARA 算法的 *F-measure* 值最大，推荐质量较高. 可见，本文提出的 CARA 算法优势比较明显.

为了进一步验证 CARA 算法的优势，我们将环境信息组合为 7 种不同类型，将本文提出的 CARA 算法与 EPF 和 Filter PoF 两种算法进行比较. 图 3 给出了 3 种算法在这 7 类环境中的 *MAE* 和 *F-measure* 值比较.

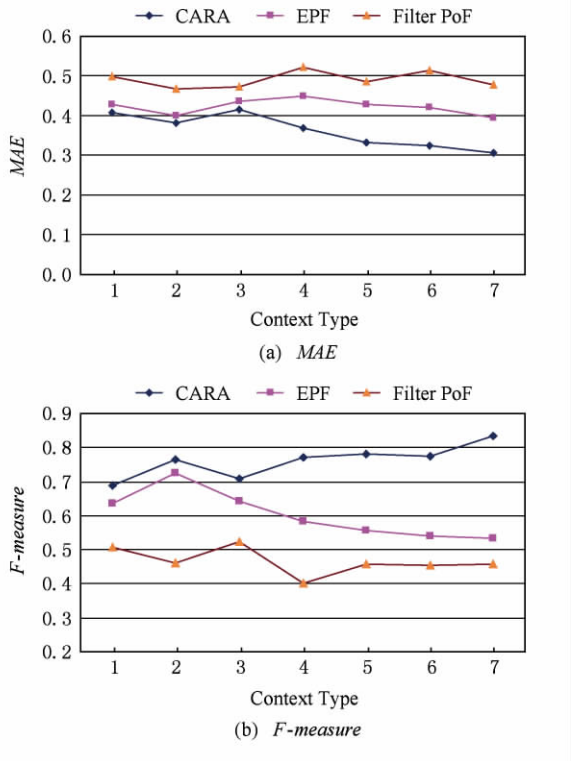

Fig. 3 Comparison of *MAE* and *F-measure* for three algorithms in seven contextual types.

图 3 3 种算法在 7 种环境中的 *MAE* 和 *F-measure* 值比较

从图 3(a)可以看出,本文提出的 CARA 算法的 *MAE* 值要优于 EPF 和 Filter PoF 算法. 在前 3 类环境中,CARA 算法的 *MAE* 值与 EPF 算法的 *MAE* 值比较接近. 但是,随着环境类型的改变,CARA 算法的 *MAE* 值逐渐降低,而 EPF 算法和 Filter PoF 算法的 *MAE* 值变化趋势不太明显. 可见,CARA 算法在环境类型较多的情况下,性能呈较优状态. 从图 3(b)可以看出,在不同的环境中,CARA 算法的 *F-measure* 值也优于 EPF 和 Filter PoF 算法,并且从第 4 类环境信息开始呈明显增长趋势,进一步说明了在环境类型较多的情况下,CARA 算法的性能较优.

为了进一步评价本文提出的 CARA 算法的性能,在环境信息类型为 $\{A,S,O\}$ 的条件下,通过改变邻居用户的个数(依次取 2,4,6,8)来比较 CARA 算法、EPF 算法和 Filter PoF 算法的 *MAE* 值,实验结果如图 4 所示.

从图 4 可以看出,3 种算法的 *MAE* 值随着邻居个数的增加而减小. 与 EPF 算法和 Filter PoF 算法相比,CARA 算法的 *MAE* 值相对最小. 可见,CARA 算法的推荐精度最高.

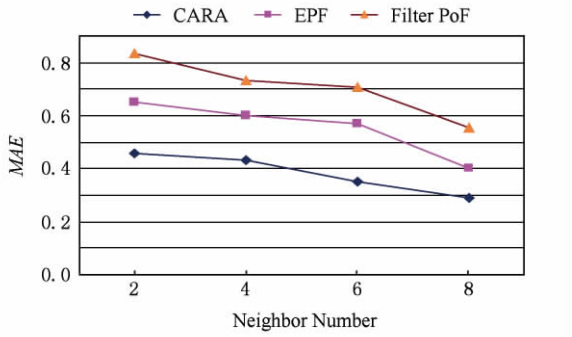

Fig. 4 Comparison of *MAE* with different number of neighbors.

图 4 不同邻居个数的 *MAE* 值比较

综上,本文提出的 CARA 算法在环境信息类型和邻居用户个数较多时,推荐精度较高. 但是,增加邻居个数和环境信息类型必然会影响算法的运算效率. 因此,对于不同的应用场合,应根据具体要求选择环境类型和邻居用户个数.

4 结 论

环境感知推荐算法是推荐系统中的一个新的研究热点,本文在这方面进行了一些有益的探索. 文中提出了一种基于模糊 C 均值聚类的环境感知推荐算法. 通过历史环境发现对历史环境进行模糊聚类,克服了传统环境感知推荐算法中的硬聚类和硬分解问题. 通过将活动环境与聚类进行匹配寻找相似环境聚类,不仅减少了推荐计算量,而且解决了环境硬匹配问题. 采用隶属映射函数解决可能出现的数据稀疏性问题. 同现有的算法相比,本文提出的 CARA 算法不仅能够根据不同环境为用户推荐不同的项目,而且提高了推荐的质量.

参 考 文 献

[1] Bettman J R, Luce M F, Payne J W. Constructive consumer choice processes [J]. Journal of Consumer Research, 1998, 25(3): 187-216

[2] Palmisano C, Tuzhilin A, Gorgoglione M. Using context to improve predictive models of customers in personalization applications [J]. IEEE Trans on Knowledge and Data Engineering, 2008, 20(11): 1535-1549

[3] Ricci F, Rokach L, Shapira B, et al. Recommender Systems Handbook [M]. Berlin: Springer, 2010: 230-238

[4] Adomavicius G, Sankaranarayanan R, Sen S, et al. Incorporating contextual information in recommender systems using a multidimensional approach [J]. ACM Trans on Information Systems, 2005, 23(1): 103-145

[5] Hussein T, Linder T, Gaulke W, et al. Context-aware recommendations on rails [C] //Proc of the 2009 Workshop on Context-aware Recommender Systems. New York: ACM, 2009

[6] Jiang T, Tuzhilin A. Improving personalization solutions through optimal segmentation of customer bases [J]. IEEE Trans on Knowledge and Data Engineering, 2009, 21(3): 305-320

[7] Panniello U, Tuzhilin A, Gorgoglione M, et al. Experimental comparison of pre-vs.-post-filtering approaches in context-aware recommender systems [C] //Proc of the 3rd ACM Conf on Recommender Systems. New York: ACM, 2009: 265-268

[8] Liu L, Lecue F, Mehandjiev N, et al. Using context similarity for service recommendation [C] //Proc of the 2010 IEEE 4th Int Conf on Semantic Computing. Los Alamitos, CA: IEEE Computer Society, 2010: 277-284

[9] Hussein G. Mobile recommender system analysis & design [C] //Proc of the 1st Int Conf on Networked Digital Technologies. Los Alamitos, CA: IEEE Computer Society, 2009: 14-19

[10] Pessemier T D, Deryckere T, Martens L. Context aware recommendations for user-generated content on a social network site [C] //Proc of the 7th European Conf on European Interactive Television Conference. New York: ACM, 2009: 133-136

[11] Baltrunas L, Ricci F. Context-dependent items generation in collaborative filtering [C] //Proc of the 2009 Workshop on Context-aware Recommender Systems. New York: ACM, 2009

[12] Berget I, Mevik B H, Nis T. New modifications and applications of fuzzy *C*-Means methodology [J]. Computational Statistics & Data Analysis, 2008, 52(5): 2403-2418

[13] Li X, Lu X, Tian J, et al. Application of fuzzy *C*-means clustering in data analysis of metabolomics [J]. Analytical Chemistry, 2009, 81(11): 4468-4475

[14] Yu Jian, Cheng Qiansheng. The search range of the optimal number of clusters in fuzzy clustering methods [J]. Science China: Series E, 2002, 32(2): 274-280 (in Chinese)
(于剑, 程乾生. 模糊聚类方法中的最佳聚类数的搜索范围[J]. 中国科学: E辑, 2002, 32(2): 274-280)

[15] Adomavicius G, Kwon Y. New recommendation techniques for multicriteria rating systems [J]. Intelligent Systems, 2007, 22(3): 48-55

[16] Zhang Guangwei, Li Deyi, Li Peng, et al. A collaborative filtering recommendation algorithm based on cloud model [J]. Journal of Software, 2007, 18(10): 2403-2411 (in Chinese)
(张光卫, 李德毅, 李鹏, 等. 基于云模型的协同过滤推荐算法[J]. 软件学报, 2007, 18(10): 2403-2411)

[17] Deng Ailin, Zhu Yangyong, Shi Baile. A collaborative filtering recommendation algorithm based on item rating prediction [J]. Journal of Software, 2003, 14(9): 1621-1628 (in Chinese)
(邓爱林, 朱扬勇, 施伯乐. 基于项目评分预测的协同过滤推荐算法[J]. 软件学报, 2003, 14(9): 1621-1628)

[18] Ma Hongwei, Zhang Guangwei, Li Peng. Survey of collaborative filtering algorithms [J]. Journal of Chinese Computer Systems, 2009, 30(7): 1282-1288 (in Chinese)
(马宏伟, 张光卫, 李鹏. 协同过滤推荐算法综述[J]. 小型微型计算机系统, 2009, 30(7): 1282-1288)

[19] Goldbreg K, Roeder T, Gupta D, et al. Eigentaste: A constant time collaborative filtering algorithm [J]. Information Retrieval, 2001, 4(2): 133-151

[20] Xu Hailing, Wu Xiao, Li Xiaodong, et al. Comparison study of Internet recommendation system [J]. Journal of Software, 2009, 20(2): 350-362 (in Chinese)
(许海玲, 吴潇, 李晓东, 等. 互联网推荐系统比较研究[J]. 软件学报, 2009, 20(2): 350-362)

[21] Dey A K. Understanding and using context [J]. Personal and Ubiquitous Computing, 2001, 5(1): 4-7

Zhang Fuzhi, born in 1964. Professor and PhD supervisor. His main research interests include intelligent information processing, network and information security, and service-oriented computing.

Chang Junfeng, born in 1986. Master. Her main research interests include intelligent information processing and information security.

Zhou Quanqiang, born in 1985. PhD candidate. His main research interests include intelligent information processing and information security.