DOI:10.7544/issn1000-1239.2018.20160928
55(2):391-399,2018

基于密度峰值聚类的动态群组发现方法

王海艳[1,2,3]　　肖亦康[1]

[1](南京邮电大学计算机学院　南京　210023)

[2](江苏省无线传感网高技术研究重点实验室　南京　210003)

[3](江苏省大数据安全与智能处理重点实验室　南京　210023)

(wanghy@njupt.edu.cn)

Dynamic Group Discovery Based on Density Peaks Clustering

Wang Haiyan[1,2,3] and Xiao Yikang[1]

[1](*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing* 210023)

[2](*Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing* 210003)

[3](*Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing* 210023)

Abstract　Group recommendation has recently received wide attention due to its significance in real applications. As a premier step of group recommendation, group discovery is very important and discovery results will impact a lot on the performance of group recommendation. The higher similarity the groups have, the better effectiveness and stability the recommendation results will possess. However, current group discovery methods seldom consider the dynamicity of users' tendency with variance of time context, nor do they support the existence of groups overlapping. In order to address the problems above, a dynamic group discovery method based on density peaks clustering (DGD-BDPC) is put forward in this paper. In the proposed DGD-BDPC method, quantitative users' dynamic tendency is firstly obtained by dynamic poisson factorization. And secondly, users' tendency under different time nodes for various items will be predicted with the employment of high order singular value decomposition (HOSVD) and user sets with high similarity will then be built according to users' tendency. Finally, user sets will be clustered with a modification of density peaks clustering algorithm and group discovery will be realized successfully. Experimental results show that the proposed dynamic group discovery method based on density peaks clustering has higher accuracy, lower error and better stability compared with some other methods.

Key words　time context; dynamicity; similarity; density peaks clustering; group discovery

摘　要　近年来,群组推荐由于其良好的实用价值得到了广泛关注.群组发现作为群组推荐的前提环节,其发现结果对推荐效果有着至关重要的影响,群组相似度越高,推荐的效果和稳定性越好.针对现有群组发现方法中存在忽略用户倾向具有时间迁移性和群组可重叠性展开研究,提出了一种基于密度峰值聚类的动态群组发现方法.该方法首先通过动态泊松分解得到量化的用户动态倾向,然后通过高阶奇异值分解预测不同的时间节点下用户对不同项目的倾向,并根据计算所得的用户倾向构建高相似度用户集合,最后利用改进的基于密度峰值的聚类算法对用户集合进行划分,实现群组发现.仿真实验对比结果表明:上述基于密度峰值聚类的群组发现方法具有更好的群组推荐效果.

收稿日期:2016-12-07;**修回日期**:2017-09-28

基金项目:国家自然科学基金项目(61772285,61373138)

This work was supported by the National Natural Science Foundation of China (61772285, 61373138).

关键词 时间上下文;动态性;相似度;密度峰值聚类;群组发现

中图法分类号 TP311

随着互联网技术的飞速发展,网络上的服务数量也随之急剧增长.然而,这种增长远远超过个人或系统所能接受、处理和有效利用的范畴.在这种环境下,能够针对不同用户需求的推荐系统应运而生,推荐理论及其相关技术已成为学术界和工业界的一个热门研究课题.

传统的服务推荐系统如协同过滤技术普遍侧重于向单个用户进行推荐,但在现实生活的许多日常活动中,用户是以群组形式出现的,例如出行旅游、网上团购等[1].因此,群组推荐系统需要同时考虑所有用户的倾向来进行推荐.另一方面,针对某单一用户进行推荐容易产生效果不理想的情况,而需要将其放入到群组中,通过群组推荐往往能获得良好效果,并能有效缓解新用户引起的冷启动问题.目前面向群组的推荐系统研究受到越来越多的关注,2011年 ACM 推荐系统大会(RecSys2011)以"为家庭群组推荐电影"为主题,举办了上下文感知电影推荐挑战赛(CAMRa2011),促进了群组推荐在电影、餐饮、旅游等领域的推广与应用[2-3].群组发现作为群组推荐的前提步骤,其群组划分结果对推荐效果起重要作用.群组的内在相似度决定了群组推荐的精确度,高相似度群组的推荐效果能够达到甚至超过单个用户推荐的精度,且在群组规模增大时也具有良好的稳定性[4].

现有群组发现方法往往只考虑用户与项目间的二元关系[5],而较少关注时间因素的影响,这类方法在实际应用中是不合理的,因为用户的选择倾向会随着时间的推移而发生变化,具有时间迁移性.时间上下文对推荐的影响主要有以下 2 点:1)项目的流行程度会随着时间推移而改变;2)用户对项目的倾向可能会随时间变化而改变[6].例如,考虑学术研究这一应用场景,由于每年都会举办大量的国际、国内学术会议,诞生大量的学术论文和在相应研究领域的突破性研究成果,随着科技的发展,每年的研究热点会发生变化,学者们的研究倾向也会随之变化,这是极具时效性的.因此在进行学术热点或相关成果推荐时把时间考虑在内是很有必要的.

通过对现有相关工作进行调研与分析,已有的群组发现方法主要存在以下 2 个问题:

1) 大部分群组发现方法都假设用户倾向是静态的,忽视了时间因素带来的倾向迁移性问题,往往导致实际计算出的用户间相似度值精确度较低.

2) 聚类算法在群组发现中是一个典型应用,但大部分聚类算法将每一处理对象划分到互斥的数据集中,即簇之间无交集,这并不符合实际情况中同一个用户可以分属于不同群组的事实.另外该类分配方法还会导致处在相邻群组边界区的用户只能得到一个群组的推荐结果,影响这类用户的推荐结果精度.

针对以上存在问题,本文提出了一个解决方案,主要工作如下:

1) 提出一种用户动态倾向的计算方法.考虑到用户倾向随时间的迁移性,首先通过动态泊松分解得到已有观察值的用户倾向,再利用张量分解对高维数据的处理能力,预测出不同时间节点不同项目下用户的倾向.

2) 提出一种基于密度峰值聚类算法的群组发现方法.利用用户选择倾向计算的结果构建高相似度用户集合,然后对原有的密度峰值聚类算法进行修改并实现用户群组划分,新的群组划分方法能够满足同一个用户可以隶属于多个群组的需求.

1 相关工作

1.1 用户倾向获取

大多数传统的推荐系统中,系统根据用户已有的评分数据或隐式信息量化用户倾向,但是用户倾向会随着时间的推移而改变,因此在量化用户间倾向时应该考虑已知的评分数据或隐式信息与时间因素的关系.

Gang 等人[6]利用用户隐式反馈信息,结合时间上下文进行推荐,指出时间上下文在项目流行程度、用户偏好和用户打分习惯 3 个方面影响着推荐结果.Victor 等人[7]基于准二元组理论,利用上下文环境中用户行为来表现用户对不同项目的倾向,能够将相同的用户划分进不同的群组,但无法保证用户群组的内在相似度.Yi 等人[8]结合互联网中项目的显式评分和用户点击提取用户偏好,构建了一个个性化推荐系统.但该系统采用的历史记录是静态的,无法准确表现用户的动态倾向.Laurent 等人[9]在泊松分解的基础上提出了动态泊松分解,能够同时考虑时间因素对用户选择倾向和项目流行度的影响,缓解了传统推荐系统处理用户倾向的静态局限性.

1.2 群组发现

由于大多数数据集可能不包含用户的群体信息,人们提出了不同的群组发现方法对群体进行划分,目前群组推荐系统中的群组发现主要是根据用户的倾向和人口统计学信息等特征对用户进行聚类.最基本的想法是将选择倾向相似的用户构成群组,因为群组中用户内在相似度越高越容易生成高质量的群组推荐.

Linas 等人[4]对比分析了群组推荐与个人推荐,实验结果显示相似度高的群组推荐精度能达到甚至会超过个人推荐的精度,且在群组规模增大时依然能保持良好的稳定性.Jing 等人[10]基于用户会受到某些潜在因素的影响的假设,提出了潜在群组模型,该方法将用户的潜在因素偏好聚合为群体倾向从而进行群体推荐.Boratto 等人[11]直接把用户-项目评分矩阵作为 K-means 算法的输入,根据用户倾向将用户聚类形成群组,但此聚类方法导致用户只能属于一个群组.Ntoutsi 等人[2]提出一种聚类算法,初始时每个群组中只有一个用户,然后比较每个用户群组的内在相似度.当偏好最相似的 2 个群组的相似度超过给定阈值时将 2 个群组合并,最终相似度超过阈值的用户都被划为同一个群组.Seko 等人[12]提出基于内容的群体推荐方法,该方法假设群体的选择受物品种类影响的假设,但这种方法只能应用于预定义的群体.

1.3 密度峰值聚类算法

密度峰值聚类算法由 Rodriguez 等人[13]提出,由于其良好的适应能力和极高的运行效率而备受关注.近年来,不少学者也将该算法应用到了多个领域.

冯国香[14]在复杂社区网络的研究中应用了该算法,并对其进行改进,克服了邻近矩阵为整数的缺点且实现了重叠社区网络的划分.Chen 等人[15]利用密度峰值聚类算法提出了一种根据图像估算人年龄的方法,在大规模图像数据集试验中取得了良好的效果.Liu 等人[16]将该算法应用到了城市出租车运营领域,提出一种变体密度峰值聚类算法用于发现城市出租车的需求热点,且时间与内存消耗都很低.Mykola 等人[17]针对模式识别中数据的复杂异构型问题,提出了局部密度以及流式距离测量方法,可以更精确地捕捉局部和全局流式信息.

2 群组发现方法

群组发现的核心思想是让相似度高的用户聚集成群组,本文围绕这一核心思想展开工作.考虑到用户倾向的时间迁移性,本文采用动态泊松分解的方法获取用户的动态倾向量化值.为计算用户间相似系数,其中的缺省值通过张量分解预测得到,最后构造高相似度用户集合,通过对已有的聚类算法进行改进,最终得到用户群组.

2.1 用户动态倾向提取

用户对项目的选择倾向随着时间的变化而变化,正如引言中给出的推荐学术研究热点问题,仅考虑用户的历史记录这一全局静态因素是不可取的.例如,学者 A 与 B 在某一时间段内对某研究领域的论文点击量相近,但学者 A 的点击量呈逐渐增加趋势,B 则相反,也就是说学者 A 与 B 对该研究领域的关注倾向是相反的,具有较低的相似度.然而,如果静态地考虑 A 与 B 在时间段内的历史数据,则会得到两者是具有较高相似度的用户这一截然相反的结果.

定义 1. 泊松分解[18].泊松分解本质上属于产生式概率模型,它假设每个观测元素 f_{nm} 服从期望为 y_{nm} 的泊松分布:$f_{nm}\sim Poisson(y_{nm})$.期望值矩阵 $\boldsymbol{Y}\in\mathbb{R}^{N\times M}$ 被分解为用户隐藏特征矩阵 $\boldsymbol{U}\in\mathbb{R}^{N\times K}$ 和项目隐藏特征矩阵 $\boldsymbol{V}\in\mathbb{R}^{M\times K}$,且隐藏特征矩阵的每个元素 u_{nk} 和 v_{mk} 服从 Gamma 分布,其中 $K\ll\min(M,N)$ 是隐藏特征向量的维数,并使用用户隐藏特征矩阵 $\boldsymbol{U}$ 和项目隐藏特征矩阵 $\boldsymbol{V}$ 的内积来近似期望值矩阵 $\boldsymbol{Y}$:$\boldsymbol{Y}\sim\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}$.

为了获取用户的动态倾向,本文采用文献[9]提出的动态泊松分解方法.该方法是在泊松分解这一静态方法上的改进,优势在于能够利用显式的评分或隐式信息高效地发现用户的倾向序列,并同时考虑用户倾向和项目流行度,有效缓解了时间因素对推荐结果的影响.

首先,用状态空间模型作为泊松分解的动态部分,状态空间模型表示基于前一个时间节点的当前状态的高斯分布情况,$u_{nk,t}$ 表示第 n 个用户在时间节点 t 的第 k 个元素,$v_{mk,t}$ 表示第 m 个项目在时间节点 t 的第 k 个元素,其分布表达式为

$$\begin{cases}u_{nk,t}\mid u_{nk,t-1}\sim N(u_{nk,t-1},\sigma_u^2),\\ v_{mk,t}\mid v_{mk,t-1}\sim N(v_{mk,t-1},\sigma_v^2).\end{cases}\tag{1}$$

然后,通过以下递推表达式获得时间节点 t 下用户和项目的相关系数:

$$\begin{cases}\begin{cases}u_{nk,1}\sim N(\mu_u,\sigma_u^2),\ t=1,\\ u_{nk,t}\mid u_{nk,t-1}\sim N(u_{nk,t-1},\sigma_u^2),\ t>1.\end{cases}\\ \begin{cases}v_{mk,1}\sim N(\mu_v,\sigma_v^2),\ t=1,\\ v_{mk,t}\mid v_{mk,t-1}\sim N(v_{mk,t-1},\sigma_v^2),\ t>1.\end{cases}\end{cases}\tag{2}$$

用户在时间节点 t 的元素表达式为 $\overline{u_{nk}}+u_{nk,t}$,其中 $\overline{u_{nk}}$ 为用户的全局系数:$\overline{u_{nk}}\sim N(\mu_u,\sigma^{\frac{2}{u}})$,项目的表达式与此对应,状态空间模型能够随时间提取用户的倾向,而静态全局因素是不受时间影响的.

最后,时间节点 t 下用户 n 对项目 m 选择倾向的泊松分布如下:

$$f_{nm,t}\sim Poisson(\sum_{k=1}^{K}e^{(u_{nk,t}+u_{nk})}e^{(v_{mk,t}+v_{mk})}). \quad (3)$$

动态泊松分解同时考虑不同时间节点下的用户反馈信息及项目流行度,它将两者有机融合,本文将动态泊松分布加权平均的结果作为该时间节点下用户对项目选择倾向的量化值,计算方法如下:

$$P_{nm,t}=\sum_{l=1}^{K}f_{nm,t,l}l. \quad (4)$$

2.2 用户倾向值分解

在计算用户相似度时,由于本文引入了时间因素,不同时间节点下不同用户间的倾向值会出现缺省,无法计算,因此,在计算相似度之前需要先用张量分解预测补全每个用户的倾向值.

定义 2. 张量分解[19]. 张量分解是对矩阵分解的 N 维拓展,它将 N 阶张量分解为一个核心张量和 N 个因子矩阵的乘积:$\boldsymbol{X}\approx\boldsymbol{C}\times_1\boldsymbol{U}^1\times_2\boldsymbol{U}^2\times_3\cdots\times_N\boldsymbol{U}^N$,其中 $\boldsymbol{U}^i\in\mathbb{R}^{I_i\times T_i}$,$1\leqslant i\leqslant N$,核心张量 $\boldsymbol{S}\in\mathbb{R}^{T_1\times T_2\times\cdots\times T_N}$,因子矩阵是每一维上的主要成分.

张量分解主要以下 3 点优势:1)不需要预过滤和后过滤. 不同于许多现有的依靠拆分、预过滤和后过滤的算法,张量分解利用所有已知的评分将用户和项目建模,而拆分、预过滤以及后过滤上下文会导致不同上下文设定间的信息丢失. 2)简化计算. 许多现有的方法采用一系列代价巨大的技术,而张量分解是一个单一的简化计算的模型. 3)解决 N 维数据的能力. 张量分解方法对于任意数量的上下文变量都具有通用的处理能力.

高阶奇异值分解(high order singular value decomposition, HOSVD)属于张量分解模型,它包含密度矩阵 $\boldsymbol{D}$,能缓解引入时间因素后造成的严重的数据稀疏性问题,因此本文采用该方法进行张量分解,将用户-项目-时间 3 维张量分解为 3 个维度上的因子矩阵以及 1 个核心张量.

本文只考虑了时间这一个上下文因素,因此只用单个上下文变量 C 来描述模型,$\boldsymbol{Y}$ 为包含倾向值的 3 维张量. 对大量上下文变量的 n 维泛化是类似的,不在此赘述. 用 $\boldsymbol{Y}\in y^{n\times m\times c}$ 表示已有的系数稀疏张量,其中 n 为用户数量,m 为项目数量,$c_i\in\{1,2,\cdots,c\}$ 为时间因素变量. 倾向数据为 $\boldsymbol{Y}\in\{0,1,\cdots,s\}^{n\times m\times c}$,0 表示用户对该项目没有点击过该项目. 0 是特殊的,因为它不能表示用户是否对该项目具有倾向. 2 个张量的相乘操作用 $\times_U$ 表示,例如,$\boldsymbol{T}=\boldsymbol{Y}\times_U\boldsymbol{U}$ 表示为 $T_{ijk}=\sum_{i=1}^{n}Y_{ijk}U_{ij}$,用 $\boldsymbol{U}_{i*}$ 表示第 i 行的矩阵 $\boldsymbol{U}$.

3 维张量被分解为 3 个矩阵:$\boldsymbol{U}\in\mathbb{R}^{n\times d_U}$,$\boldsymbol{I}\in\mathbb{R}^{m\times d_I}$,$\boldsymbol{C}\in\mathbb{R}^{c\times d_C}$,以及核心张量 $\boldsymbol{S}\in\mathbb{R}^{d_U\times d_I\times d_C}$. 针对单个用户 u,项目 i 和上下文 c 的决策函数为

$$F_{ijk}=\boldsymbol{S}\times_U\boldsymbol{U}_{i*}\times_I\boldsymbol{I}_{j*}\times_C\boldsymbol{C}_{k*}. \quad (5)$$

这个分解模型能够通过调整参数 d_U,d_I,d_C 分别控制用户、项目和时间的维度. 这个特性对于现实中大规模数据集的处理很有利,因为矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{M}$ 的规模可能会增大带来潜在存储问题.

损失函数定义为

$$L(\boldsymbol{F}',\boldsymbol{Y})=\frac{1}{\|\boldsymbol{S}\|_1}\sum_{i,j,k}D_{ijk}l(F_{ijk},Y_{ijk}), \quad (6)$$

其中 $l:\mathbb{R}\times y\rightarrow\mathbb{R}$ 是一个点态损失函数,惩罚估算值和真实值,F_{ijk} 已由式(5)给出. 需要注意的是,全局损失函数 L 由张量 $\boldsymbol{Y}$ 中的真实值定义.

最终得到的目标函数如下:

$$R[\boldsymbol{U},\boldsymbol{I},\boldsymbol{C},\boldsymbol{S}]=L(\boldsymbol{F}',\boldsymbol{Y})+\Omega[\boldsymbol{Y},\boldsymbol{I},\boldsymbol{C}]+\Omega[\boldsymbol{S}], \quad (7)$$

其中加入了 Frobenius 范数 $\Omega[\boldsymbol{U},\boldsymbol{I},\boldsymbol{C}]$,因为单纯的最小化上述损失函数会导致过拟合现象,$\Omega[\boldsymbol{S}]$是作用于核心张量 $\boldsymbol{S}$ 的范数,其目的是降低核心张量复杂度.

时间上下文张量分解的用户倾向预测算法如下所示:

算法 1. 用户倾向张量分解预测算法.

输入:原用户倾向值张量 $\boldsymbol{Y}$、维度 d;

输出:用户倾向值张量 $\boldsymbol{F}$.

初始化:$\boldsymbol{U}\leftarrow\mathbb{R}^{n\times d_U}$,$\boldsymbol{I}\leftarrow\mathbb{R}^{m\times d_I}$,$\boldsymbol{C}\leftarrow\mathbb{R}^{c\times d_C}$,$\boldsymbol{S}\in\mathbb{R}^{d_U\times d_M\times d_C}$,$t=t_0$;

遍历:for (i,j,k) in $\boldsymbol{Y}$

$\eta=\frac{1}{\sqrt{t}}$, $t=t+1$;

$F_{ijk}=\boldsymbol{S}\times_U\boldsymbol{U}_{i*}\times_I\boldsymbol{I}_{j*}\times_C\boldsymbol{C}_{k*}$;

$\boldsymbol{U}_{i*}=\boldsymbol{U}_{i*}-\eta\lambda_U\boldsymbol{Y}_{i*}-\eta\partial_{\boldsymbol{U}_{i*}}l(F_{ijk},Y_{ijk})$;

$\boldsymbol{I}_{j*}=\boldsymbol{I}_{j*}-\eta\lambda_I\boldsymbol{I}_{j*}-\eta\partial_{\boldsymbol{M}_{j*}}l(F_{ijk},Y_{ijk})$;

$\boldsymbol{C}_{k*}=\boldsymbol{C}_{k*}-\eta\lambda_C\boldsymbol{C}_{k*}-\eta\partial\boldsymbol{C}_{k*}l(F_{ijk},Y_{ijk})$;

$\boldsymbol{S}=\boldsymbol{S}-\eta\lambda_S\boldsymbol{S}-\eta\partial_S l(F_{ijk},Y_{ijk})$;

end for

最小化:$\boldsymbol{F}\leftarrow R[\boldsymbol{U},\boldsymbol{I},\boldsymbol{C},\boldsymbol{S}]=L(\boldsymbol{F}',\boldsymbol{Y})+\Omega[\boldsymbol{U},\boldsymbol{I},\boldsymbol{C}]+\Omega[\boldsymbol{S}]$.

算法 1 输入为包含原用户倾向值的 3 维张量 $\boldsymbol{Y}$,以及维度 d.将原张量初始化分解为 $\boldsymbol{U},\boldsymbol{I},\boldsymbol{C}$ 这 3 个矩阵,通过迭代得到近似张量 $\boldsymbol{F}'$,最小化目标函数得到最终的包含预测倾向值得张量 $\boldsymbol{F}$.其算法复杂度为 $O(Kd_Ud_Id_C)$,与已知的倾向值数量和维度线性相关.

2.3 基于密度峰值聚类算法的群组发现

鉴于密度峰值聚类算法能够适用于各种形状的聚类并能自动发现聚类个数,且适应能力和运行效率极高,有利于解决用户群组划分中用户分布情况和群组个数未知的问题,因此本节借鉴密度峰值聚类算法[13]的思想对用户群组进行划分,提出了一种基于密度峰值聚类的动态群组发现方法(dynamic group discovery method based on density peaks clustering, DGD-BDPC).

2.3.1 密度峰值聚类算法

密度峰值聚类算法[13]的基本思想如下:假设类簇中心周围节点的局部密度一般低于该类簇中心的局部密度,并且与具有更高密度的节点的距离都较大.首先计算每一个数据节点 i 局部密度 ρ_i 和该点到更高局部密度的节点的最短距离 δ_i;然后根据这 2 个值画出决策图,在决策图里面找到类簇中心;最后根据每一个节点的最近更高局部密度节点的类别确定其所属类簇中心.由于该算法对于多种类型的数据显示出了良好的适应能力和极高的运行效率,所以本文将其引入到用户群组的划分中.

该算法需要解决 2 个问题:1)需要计算不同节点之间的距离,本文将用户倾向相似度系数作为节点间的距离;2)聚类算法只能解决标准的群组划分结果,它将剩余的待划分节点都划分到最近更高局部密度节点所在的类簇中,因此无法处理包含重叠节点的群组划分问题.本文通过定义群组贡献的适应度函数方法来判断剩余节点是否应该划分进该群组.

2.3.2 用户节点距离矩阵

密度峰值聚类算法需要用到不同节点之间的距离,即距离矩阵,由第 2.2 节可计算得到包含用户倾向的张量 $\boldsymbol{F}$,然后通过皮尔逊相似度计算公式计算得到用户间相似度并作为节点之间的距离,在此不再赘述.根据 Linas 等人[4]的实验结论,相似度高于 0.27 的用户即为高相似度用户.所以本文将用户间相似度低于 0.27 的系数直接设为 0,表示用户间没有连接,这样保证了在下一步聚类操作中得到的群体必然为具有高相似度的用户群体.与实际距离相反,用户间相似度值越大距离越小,反之越大.设矩阵 $\boldsymbol{A}$ 为已知的连接矩阵,如表 1 所示:

Table 1 Users Distance Matrix

表 1 用户距离矩阵

User	u_1	…	u_i	…	u_k
u_1	A_{11}				
⋮		⋱			
u_i			A_{ii}		
⋮				⋱	
u_k			A_{ki}		A_{kk}

其中 A_{ij} 表示节点 i 与 j 之间的距离,不同于一般的网络结构,用户间不存在间接距离,因此简化了距离计算.

2.3.3 重叠用户群组发现

得到用户距离矩阵后,基于密度峰值聚类算法,本文修改了其处理方法以实现重叠用户群组发现.用户群组发现过程具体步骤如下:

1) 计算所有节点的局部密度和与该点更高局部密度最近的距离.这 2 个值通过 2.3.2 节得到的用户距离矩阵计算得出,修改的局部密度 ρ 计算公式如下:

$$\rho_i = \sum_{j=1}^{k} X(1-d_{ij}-d_c), \tag{8}$$

其中,若 $x<0$ 则 $X(x)=1$,否则 $X(x)=0$,d_c 是一个截断距离,一般来说,可以选择 d_c 使得节点的平均邻居数大概是数据集中节点总数的 1%~2%.基本上,节点 i 的密度 ρ_i 等于该点的距离小于截断距离 d_c 的点的个数.密度峰值聚类算法对于节点密度大小相对比较敏感,而对于大数据集,其对于 d_c 的取值都具有很好的鲁棒性.

用户节点 i 到更高局部密度用户的距离 δ_i 是到任何比其密度大的节点的最小值,计算公式如下:

$$\delta_i = \min_{j:\rho_j>\rho_i}(d_{ij}). \tag{9}$$

2) 画出相应的决策图.将 δ 且 ρ 异常大的用户节点作为群组中心.

图 1 为决策图的示例.图 1 中圆圈代表用户节点,不同颜色表示各个不同的群组.其中节点 1 和 10 同时具有异常大的 δ 和 ρ,因此分别作为群组的中心节点.

3) 划分重叠区.为了得到群组间重叠的用户节点,不能再按原算法中的方法进行划分.找到每个群组的边界区域,这个边界区域的定义为:分配到某个群组中的节点,同时与其他群组的节点距离小于 d_c 的节点集合.然后针对集合中的每个节点,本文采用 LFM 算法中的重叠节点判定方法[20].LFM 算法在

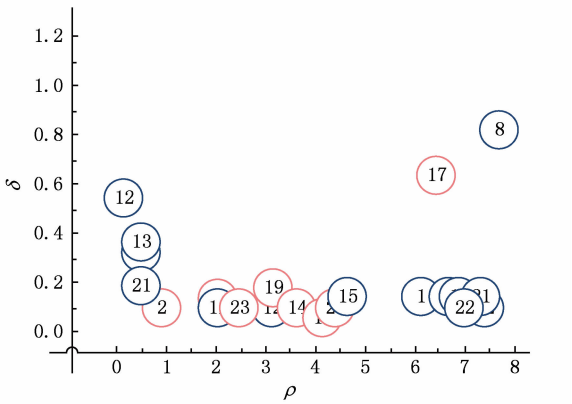

Fig. 1 Distance decision graph

图 1 距离决策图

可发现重叠节点的社区算法中是精度和效率都非常高的算法,但该算法在处理局部密集较高的簇时效率会下降,本文只采用其重叠节点的识别方法因此并不会影响群组划分算法的效率. 定义的适应度函数表示为

$$f_G=\frac{K_{\text{in}}^G}{(K_{\text{out}}^G+K_{\text{out}}^G)^{\alpha}}, \tag{10}$$

其中,K_{in}^G指群组 G 中区内度值,数值上等于 2 端都在群组内边的数目和的 2 倍;K_{out}^G指群组 G 的区外度值,数值上等于只有一端在群组 G 内的边的数目. 很容易看出当 $K_{\text{out}}^G=0$ 时划分出的群组最简单,群组外没有连边,不过这是一种理想状态,一般 $K_{\text{out}}^G>0$. α 用来控制用户集合中群组的规模,它的取值范围只能是正实数,Lancichinetti 等人[20]对实际网络的研究发现 $\alpha<0.5$ 时整个用户集合被划分成一个群组,$\alpha>2$ 时用户集合中每个节点都被划分成一个单独的群组,对于大多数的实际情况取值都非常接近于 1,本文中 $\alpha=0.98$.

节点对社区有没有贡献用节点适应度函数通过下面的式子反映出来:

$$f_G^A=f_{G+\{A\}}-f_{G-\{A\}}, \tag{11}$$

其中,$f_{G+\{A\}}$表示节点 A 从属于群组 G 时群组 G 的适应度,$f_{G-\{A\}}$表示节点 A 不属于群组 G 时群组的适应度. 当 $f_G^A>0$ 时才能说明节点 A 对群组 G 有贡献,可加入到群组 G 中,f_G^A 越大,说明节点 A 越应该加入到群组中.

本文修改的基于密度峰值聚类的可重叠的动态群组发现算法(DGD-BDPC)如下:

算法 2. 基于密度峰值聚类的动态群组发现算法.

输入:用户距离矩阵 $\boldsymbol{U}$;

输出:群组集合 G.

Step1. for each u in $\boldsymbol{U}$

$u_{\rho_i}=\sum_{j=1}^{k}X(1-d_{ij}-d_c)$;

$u_{\delta_i}=\min_{j:\rho_j>\rho_i}(d_{ij})$;

end for

Step2. for each u in $\boldsymbol{U}$

$u_{\delta_{\min}}\leftarrow u$;

end for

for each u_{center}

$G_i\leftarrow u_{\text{center}}$;

end for

Step3. for each u in G_i

if $d_{uG_j}<d_c$ and $j\neq i$

$U_{\text{edge}}\leftarrow u$;

end if

end for

Step4. for each u in U

if $f_G^A>0$

$G\leftarrow u$;

end if

end for

该算法复杂度为 $O(u^2)$,主要的计算耗时集中于每个节点的密度与距离计算,群组的构建操作只需对用户进行一次遍历即可完成复杂度为 $O(u)$.

3 实验仿真及效用评估

3.1 实验场景

为了检测本文提出的基于密度峰值聚类算法的群组发现的效果,采用的测试样本使用了为提供科学研究成果的交流与共享而建立的 arXiv.org 真实论文数据集,它包含 2003—2013 年 75 000 篇学术论文以及 5 000 个用户,每篇学术论文至少有 20 次点击量. 实验硬件环境:CPU 为酷睿 i5 处理器 2.3 GHz,内存 8 GB,系统为 Ubuntu 14.04 LTS.

3.2 实验评估

3.2.1 实验结果评价准则

本文采用 *nDCG* 和 *RMSE* 这 2 个参数来对比衡量不同群组划分方法得到的群组推荐效果.

nDCG (normalized discounted cumulative gain)[21]用来评估生成的 Top-k 推荐列表的准确度,这是一个标准的 IR. $p_1,p_2,\cdots,p_l$ 是项目的评分列表,u 表示用户,r_{up_i}表示用户对项目 p_i 的真实评分,k 表示推荐结果的前 k 个,其计算方法如下:

$$DCG_k^u = r_{up_1} + \sum_{i=2}^{k} \frac{r_{up_i}}{\mathrm{lb}\, i}, \tag{12}$$

$$nDCG_k^u = \frac{DCG_k^u}{IDCG_k^u}. \tag{13}$$

RMSE(root mean square error)是用来衡量预测评分与实际评分的误差指标,向量 $\boldsymbol{r}_i$ 表示群组对项目的实际评分,$\boldsymbol{r}_i'$ 表示群组对项目的预测评分,计算方法如下:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\boldsymbol{r}_i - \boldsymbol{r}_i')^2}{n}}. \tag{14}$$

3.2.2 推荐效果对比实验

本实验以文献[22]提出的采用 *K*-means 群组发现方法的潜在群体模型(latent group model, LGM)、文献[7]提出的基于用户行为的群组发现方法(online role mining, ORM)、以及随机用户群组(Random)作为参照对象与本文所提出的 DGD-BDPC 方法进行对比. 实验选取了 Average, LM (least misery)两种传统的融合策略获得推荐结果.

实验 1. 对比引入时间因素后动态群组的推荐效果,在保持推荐生成策略相同的条件下对比 4 种群组发现方法的推荐效果. 本实验中用 *Value* 来表示推荐精度改善百分比,计算方法如下:

$$Value = \frac{nDCG_{\mathrm{Method}} - nDCG_{\mathrm{Random}}}{nDCG_{\mathrm{Random}}}. \tag{15}$$

对比无任何优化处理的 Random 方法,表 2 给出了在不同群组规模下,LGM, ORM, DGD-BDPC 三种群组发现方法在 Average 和 LM 策略下的推荐精度改善百分比. 由于 Random 对比自身的改善值均为 0,因此不在表 2 中显式列出. 从表 2 中可以看出本文提出的 DGD-BDPC 方法改善效果始终高于另外 2 个静态方法,且在群组规模增大时依然具有较好的改善效果. 对比 Average 和 LM 两种策略可以发现,在本文的数据集应用场景下,Average 策略具有更好的改善效果.

Table 2 Improvement Comparison of Recommendation Effects

表 2 推荐效果改善对比 %

Group Size	Average			LM		
	LGM	ORM	DGD-BDPC	LGM	ORM	DGD-BDPC
4	1.48	1.24	**2.01**	0.84	0.81	**1.82**
8	2.02	2.05	**2.24**	1.56	1.17	**2.22**
16	2.25	1.94	**2.58**	1.60	1.39	**2.24**
32	2.7	1.9	**2.97**	1.82	1.14	**2.28**
64	3.49	2.7	**3.82**	2.86	2.39	**3.32**

在 Average 策略下 4 种群组发现方法精度及误差对比结果如图 2 和图 3 所示:

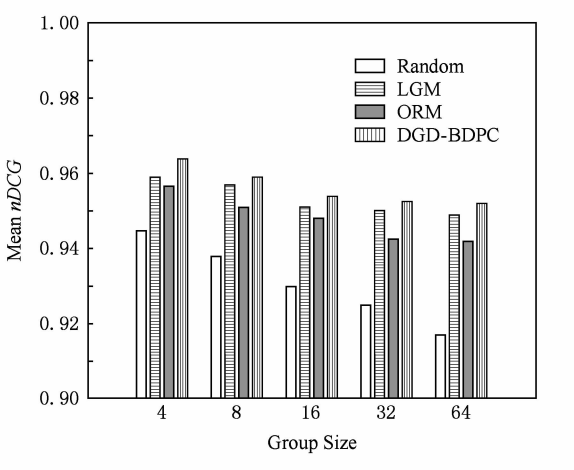

Fig. 2 Mean *nDCG* in Average strategy

图 2 Average 策略的 *nDCG*

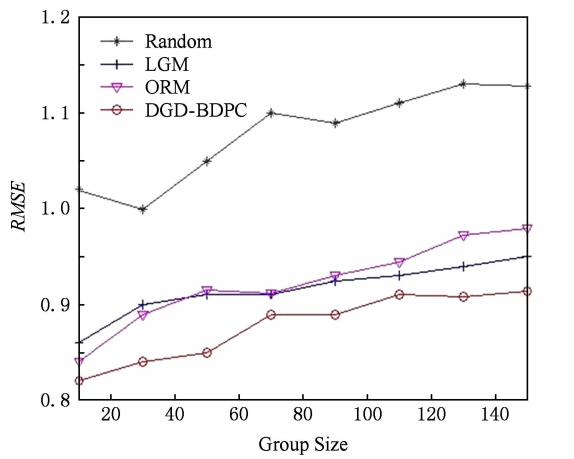

Fig. 3 *RMSE* in Average strategy

图 3 Average 策略的 *RMSE*

在 LM 策略下 4 种群组发现方法精度及误差对比结果如图 4 和图 5 所示:

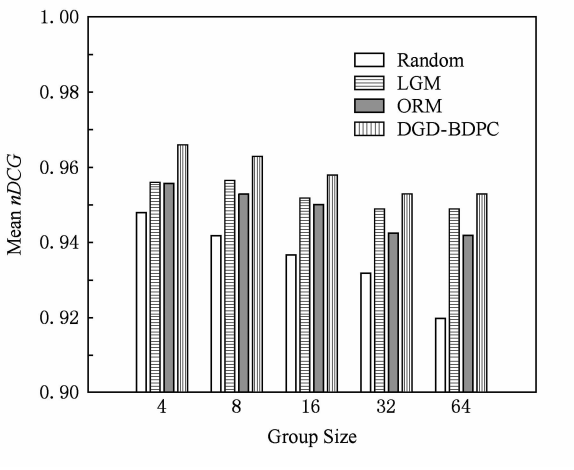

Fig. 4 Mean *nDCG* in LM strategy

图 4 LM 策略的 *nDCG*

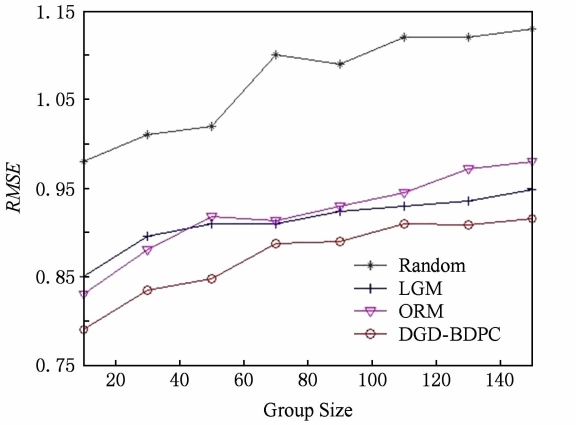

Fig. 5 *RMSE* in LM strategy

图 5 LM 策略的 *RMSE*

对比图 2 和图 4 可以看出,在群组规模较小的情况下 4 种方法差别不大. 随着群组规模的增大,随机群组的精度急剧下降. LGM 和 ORM 方法在群组规模增大时仍然保持良好的推荐精度,但是 ORM 由于不能保证群组相似度的缺点,推荐精度在群组规模相对较大时逐渐劣于 LGM 方法. 本文提出的 DGD-BDPC 群组发现方法较 LGM 和 ORM 方法具有更好的精确度和误差率,且在群组规模增大时也具有良好的稳定性.

对比分析图 3 和图 5,可以看到随机化的群组始终存在较高的误差率. ORM 和 LGM 方法在群组相对较小时误差率都很低且很接近,但同样由于无法保证用户相似度的问题,ORM 方法的误差率逐渐超过了 LGM 方法. 本文提出的群组发现方法始终保持相对较低的误差率,且当群组规模增大时,与其他 3 种方法对比,DGD-BDPC 方法增长趋势逐渐平稳.

3.2.3 群组可重叠问题检测实验

以现实应用为例,某学者主要研究机器学习算法,但近几年对服务推荐领域产生兴趣,由于该学者往年的学术关注记录绝大多与机器学习相关,因此传统的聚类算法在群组划分时很可能会将其划分到关注机器学习的学者群组中,因此也无法得到服务推荐领域的学术推荐,该部分用户即是所谓的群组重叠区域用户.

实验 2. 检测本文提出的群组发现 DGD-BDPC 方法对边界区用户的重叠处理的效果. 将各个群组间重叠区域内的用户作为实验对象,该部分用户通过 3.3.3 节的步骤 3 获得,分别对其采用非重叠处理(non-overlapping)和重叠处理(overlapping)两种方式进行对比最终推荐结果. 对于非重叠处理的用户只采用所在群组的推荐结果直接进行推荐,对于重叠处理的用户则采用多个群组的推荐结果进行推荐. 这 2 种处理方法的推荐精度结果如图 6 所示:

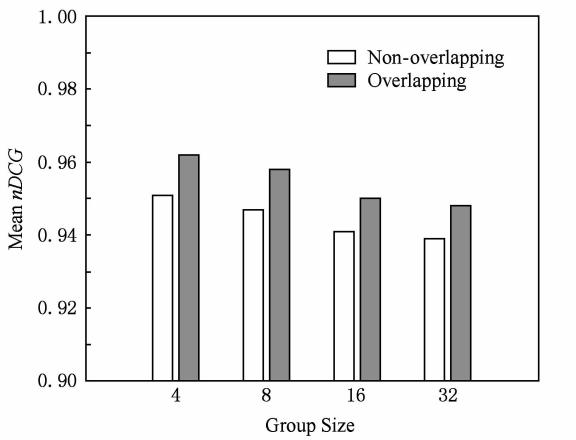

Fig. 6 Comparison between overlapping and non-overlapping

图 6 重叠处理对比图

从图 6 可以看出,无论群组规模大小,处在重叠区的用户接受多个群组的推荐结果其 *nDCG* 明显高于只接受单个群组推荐的用户,充分显现了本文对用户群组重叠性问题的处理方法有效改善了推荐效果,更符合现实中用户可以隶属多个群组的情况,避免了用户只能获得单个群组推荐结果的局限性.

4 总 结

群体推荐在推荐系统中越来越流行,而群组发现作为群组推荐首要的环节起到了至关重要的作用. 本文提出了一种考虑用户倾向动态性的基于密度峰值聚类的群组发现方法,通过引入时间因素根据用户的历史数据精确量化用户的倾向,解决了用户倾向的时间迁移性问题导致的用户相似度计算误差. 进而采用修改的基于密度峰值聚类的算法划分得到可互相重叠群组,保证了重叠区用户不会只接受单一的群组推荐,提高了该部分用户的推荐结果精度.

在后续的研究中,我们将进一步研究群组推荐阶段需要进一步考虑的问题,比如:如何解决个人用户对群组推荐结果的影响、如何缓解群组推荐中的冷启动问题等,不断提高群组推荐的效果.

参 考 文 献

[1] Zhang Yujie, Du Yulu, Meng Xiangwu. Research on group recommendation systems and their applications [J]. Chinese Journal of Computers, 2016, 39(4): 745-764 (in Chinese)
(张玉洁, 杜雨露, 孟祥武. 组推荐系统及其应用与研究[J]. 计算机学报, 2016, 39(4): 745-764)

[2] Ntoutsi E, Stefanidis K. Fast group recommendations by applying user clustering [C] //Proc of the 31st Int Conf ER (ICER 2012). Berlin: Springer, 2012: 126-140

[3] Yu Zhiwen, Zhou Xingshe, Hao Yanbin. TV program recommendation for multiple viewers based on user profile [J]. User Modeling and User-adapted Interaction, 2006, 16 (1): 63-82

[4] Linas B, Francesco R. Group recommendations with rank aggregation and collaborative filtering [C] //Proc of the 8th ACM Conf on Recommender Systems (RecSys 2010). New York: ACM, 2010: 26-30

[5] Ricci F, Rokach L. Recommender Systems Handbook [M]. Berlin: Springer, 2010: 40-62

[6] Gang Tian, Wang Jian. Time-aware Web service recommendations using implicit feedback [C] //Proc of the 21st IEEE Int Conf on Web Services (ICWS 2014). Piscataway, NJ: IEEE, 2014: 273-280

[7] Victor W, Chi Huangchi. Online role mining without over-fitting for service recommendation [C] //Proc of the 20th IEEE Int Conf on Web Services (ICWS 2013). Piscataway, NJ: IEEE, 2013: 58-65

[8] Yi Xing, Hong Liangjie, Zhong Erheng. Beyond clicks: Dwell time for personalization [C] //Proc of the 8th ACM Conf on Recommender Systems (RecSys 2014). New York: ACM, 2014: 113-120

[9] Laurent C. Dynamic poisson factorization [C] //Proc of the 9th ACM Conf on Recommender Systems (RecSys 2015). New York: ACM, 2015: 155-162

[10] Jing Shi, Wu Bin. A latent group model for group recommendation [C] //Proc of the 4th IEEE Int Conf on Mobile Service (MS 2015). Piscataway, NJ: IEEE, 2015: 233-238

[11] Boratto L, Carta S. Group identification and individual recommendations in group recommendation algorithms [C] // Proc of the 4th ACM Conf on Recommender Systems (RecSys 2010). New York: ACM, 2010: 27-34

[12] Seko S, Yagi T. Group recommendation using feature space representing behavioral tendency and power balance among members [C] //Proc of the 5th ACM Conf on Recommender Systems (RecSys 2011). New York: ACM, 2011: 101-108

[13] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. Science, 2014, 344(6191): 1492-1496

[14] Feng Guoxiang. Research on overlapping community detection method based on density peaks [D]. Changchun: Jilin University, 2015 (in Chinese)
(冯国香. 基于密度峰值的重叠社区发现算法研究[D]. 长春: 吉林大学, 2015)

[15] Chen Yewang, Lai Dehe. A new method to estimate ages of facial image for large database [J]. Multimedia Tools & Application, 2015, 75(5): 1-19

[16] Liu Dongchang, Cheng Shifen. Density peaks clustering approach for discovering demand hot spots in city-scale taxi fleet dataset [C] //Proc of the 18th IEEE Int Conf on Intelligent Transportation System (ICITS 2015). Piscataway, NJ: IEEE, 2015: 1831-1836

[17] Mykola P, Jian Z. A robust density-based clustering algorithm for multi-manifold structure [C] //Proc of the 31st Annual ACM Symp on Applied Computing (SAC 2016). New York: ACM, 2016: 832-838

[18] Yu Yonghong, Gao Yang, Wang Hao. A ranking based poisson matrix factorization model for point-of-interest recommendation [J]. Journal of Computer Research and Development, 2016, 53(8): 1651-1663 (in Chinese)
(余永红, 高阳, 王皓. 基于 Ranking 的泊松矩阵分解兴趣点推荐算法[J]. 计算机研究与发展, 2016, 53(8): 1651-1663)

[19] Wang Licai. Understanding and using contextual information in recommender system [C] //Proc of the 34th Annual ACM SIGIR Conf on Information Retrieval (SIGIR 2011). New York: ACM, 2011: 1329-1330

[20] Lancichinetti A, Fortunato S. Detecting the overlapping and hierarchical community structure in complex network [J]. New Journal of Physics, 2009, 11: 2-20

[21] Toon D, Simon D. Comparison of group recommendation algorithms [J]. Multimedia Tools & Application, 2014, 72 (3): 2497-2541

[22] Gopalan P, Hofman J. Scalable recommendation with hierarchical poisson factorization [C] //Proc of the 31st Conf on Uncertainty in Artificial Intelligence (CUAI 2015). Arlington, VA: AUAI, 2015: 235-242

Wang Haiyan, born in 1974. PhD. Professor. Senior member of CCF. Her main research interests include service computing, trusted computing and big data intelligent processing technology.

Xiao Yikang, born in 1992. Master. His main research interests include service computing, big data analysis and processing (cocomboco@gmail.com).