DOI: 10.11992/tis.201806002
网络出版地址: http://kns.cnki.net/kcms/detail/23.1538.TP.20180629.1153.004.html

基于图游走的并行协同过滤推荐算法

顾军华[1,2], 谢志坚[1,2], 武君艳[1,2], 许馨匀[1,2], 张素琪[3]
（1. 河北工业大学 人工智能与数据科学学院, 天津 300401; 2. 河北工业大学 河北省大数据计算重点实验室, 天津 300401; 3. 天津商业大学 信息工程学院, 天津 300134）

摘　要: 针对目前协同过滤推荐算法存在的数据稀疏性问题和可扩展性问题, 本文进行了相关研究。针对稀疏性问题, 在传统的皮尔逊相关相似度中引入交占比系数计算用户间直接相似度, 该方法缓解了用户间共同评分项的占比问题; 提出一种基于图游走的间接相似度计算方法, 该方法根据用户间的直接相似度建立用户网络图, 在用户网络图上通过游走计算用户间的间接相似度, 并进行推荐。在 Spark 平台上实现本文方法的并行化, 缓解了数据规模增加带来的可扩展性问题。实验结果表明: 本文提出的算法在不同数据集上均取得了良好效果, 有效地提高了推荐准确度, 并且在分布式环境下具有良好的可扩展性。

关键词: 协同过滤; 推荐; 用户网络图; 游走; 相似度; 间接相似度; 并行; Spark 平台
中图分类号: TP391　　**文献标志码**: A　　**文章编号**: 1673−4785(2019)04−0743−09

中文引用格式: 顾军华, 谢志坚, 武君艳, 等. 基于图游走的并行协同过滤推荐算法 [J]. 智能系统学报, 2019, 14(4): 743–751.
英文引用格式: GU Junhua, XIE Zhijian, WU Junyan, et al. Parallel collaborative filtering recommendation algorithm based on graph walk[J]. CAAI transactions on intelligent systems, 2019, 14(4): 743–751.

Parallel collaborative filtering recommendation algorithm based on graph walk

GU Junhua[1,2], XIE Zhijian[1,2], WU Junyan[1,2], XU Xinyun[1,2], ZHANG Suqi[3]
(1. School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China; 2. Hebei Province Key Laboratory of Big Data Computing, Tianjin 300401, China; 3. School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China)

Abstract: This study aims to solve the problem of data sparsity and scalability of collaborative filtering recommendation algorithms. For the sparseness problem, the traditional Pearson correlation similarity is introduced to calculate the direct similarity between the users using the cross-ratio coefficients. This method alleviates the proportion of common scoring items among users. An indirect similarity calculation method based on graph walk is proposed in the paper. This method builds a user network map based on the direct similarity between users, calculates the indirect similarity between users by walking on the user network map, and makes recommendations. The parallelization of this method on the Spark platform mitigates the scalability problem caused by increase of the data size. Experimental results on Movielens dataset and IPTV dataset show that the proposed algorithm achieves good results on different datasets, effectively improves the recommendation accuracy rate, and has good scalability in a distributed environment.

Keywords: collaborative filtering; recommendation; user network map; walk; similarity; indirect similarity; parallel; Spark platform

近年来随着互联网科技的发展, 大数据在促进社会进步的同时, 也带来了“信息过载”问题。如何快速从海量数据中获取有价值的信息成为当前大数据发展的关键性问题[1]。为满足人们在大数据中快速获取有价值信息的需求, 推荐系统应运而生。推荐系统的目标是根据用户的个性化需求将最符合用户喜好的信息挑选出来并推荐给用

收稿日期: 2018−06−01.　**网络出版日期**: 2018−07−02.
基金项目: 河北省科技计划项目 (17210305D); 天津市科技计划项目 (16ZXHLSF0023); 天津市自然科学基金项目 (15JCQNJC00600).
通信作者: 张素琪. E-mail: zhangsuqie@163.com.

户，以减轻用户的选择负担。协同过滤推荐算法是一种目前应用最广泛的推荐算法[2]，可以在用户没有明确提出自己需求的情况下，根据用户的行为对用户进行推荐。但由于大数据环境下用户和项目的数量不断增长，协同过滤推荐算法面临着严重的数据稀疏性和可扩展性问题[3]。

针对稀疏性问题，许多学者从不同角度进行了相关研究。SUN 等[4]采用聚类和时间影响因子矩阵来监测用户兴趣漂移程度，更准确的预测项目的评分。彭宏伟等[5]提出一种基于矩阵分解的上下文感知 POI推荐模型，有效地缓解稀疏性问题。WU 等[6]将异构信息网络建模为张量，并提出两种随机梯度下降方法同时进行分解。MA 等[7]提出了一种局部概率矩阵分解的方法，降低稀疏性的同时有效地缓解了每个局部模型的过拟合问题。以上的方法均通过缓解数据稀疏性问题来提高推荐的准确度。

针对协同过滤推荐算法在处理大规模数据所遇到的可扩展性问题，许多学者在并行方法上进行了相关研究。杨志文[8]、LU F[9]、KUPISZ[10] 等将协同过滤推荐算法部署在 Hadoop 和 Spark 并行平台上，取得了良好的执行效率。

本文针对协同过滤推荐算法的数据稀疏性问题和可扩展性问题进行研究。针对稀疏性问题，在皮尔逊相关相似度的基础上引入交占比系数来计算用户的直接相似度，提出了一种基于图游走的协同过滤推荐算法 (GW_CF)，使用图游走的方法计算用户的间接相似度，然后根据直接相似度和间接相似度重建用户的相似度矩阵，最后进行推荐。在 Movielens-100k 数据集和 IPTV 数据集上实验，验证 GW_CF 在提高推荐准确度上的有效性。针对可扩展性问题，在 Spark 平台上实现 GW_CF 算法，并使用 Movielens-1M 和 Movielens-100k 数据集进行实验，验证 GW_CF 算法的可扩展性。

1　相关工作

1.1　问题定义

基于近邻的协同过滤问题可以描述为[11]：已知用户集合表示为 $U=\{u_1,u_2,\cdots,u_a,\cdots u_b,\cdots,u_n\}$，项目集合表示为 $S=\{s_1,s_2,\cdots,s_i,\cdots,s_j,\cdots s_m\}$，用户-项目评分矩阵 $\boldsymbol{R}=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix}$，$r_{ai}$ 表示用户 u_a 对项目 s_i 的评分。基于近邻的协同过滤推荐算法的流程：1) 根据评分矩阵 $\boldsymbol{R}$ 计算用户的相似度；2) 计算目标用户的近邻用户集合；3) 根据近邻用户的评分预测目标用户对未评分项目的评分，从而生成推荐列表。

1.2　用户相似度

用户相似度指用户与用户之间行为中表现出的相似程度，皮尔逊相关相似度是一种常用的计算相似度的方法，反映了两个用户的偏好信息的线性相关程度。用户 u_a 和用户 u_b 的皮尔逊相关相似度计算公式[12-13] 如下：

$$\mathrm{sim}(a,b)=\frac{\sum\limits_{s_i\in s_{ab}}(r_{ai}-\bar{r}_a)(r_{bi}-\bar{r}_b)}{\sqrt{\sum\limits_{s_i\in s_{ab}}(r_{ai}-\bar{r}_a)}\sqrt{\sum\limits_{s_i\in s_{ab}}(r_{bi}-\bar{r}_b)}} \tag{1}$$

式中：s_{ab} 为用户 u_a 和用户 u_b 共同评分项目的集合；r_{ai} 为用户 u_a 对项目 s_i 的评分；$\bar{r}_a$ 为用户 u_a 对集合 s_{ab} 中项目评分的平均值。$\mathrm{sim}(a,b)$ 的值域为 $[-1,1]$，$\mathrm{sim}(a,b)$ 越大，表示两个用户的相似度越高。

1.3　近邻用户

近邻用户表示与目标用户偏好信息最相似的一组用户，可以通过式 (1) 计算用户的相似度，然后计算目标用户的近邻用户。目标用户的多个近邻用户组成目标用户的近邻用户集合，常用的计算近邻用户集合的方法分为两类：基于数量的近邻用户集合和基于阈值的近邻用户集合。

基于阈值的近邻用户集合包含以目标用户为中心，与目标用户的相似度大于 Value 的用户。基于数量的近邻用户集合包含与目标相似度最大的 Top-K 个近邻用户。

1.4　个性化推荐

首先计算目标用户的近邻用户集合，然后对目标用户进行推荐。目标用户 u_a 对未评分项目 s_i 预测评分的计算公式[14] 如式 (2)，最后将预测评分最大的 K 个项目推荐给目标用户。

$$r_{ai}=\bar{r}_a+\frac{\sum\limits_{u_b\in N_{u_a}}\mathrm{sim}(a,b)\times(r_{bi}-\bar{r}_b)}{\sum\limits_{u_b\in N_{u_a}}\mathrm{sim}(a,b)} \tag{2}$$

式中：$\bar{r}_a$ 表示用户 u_a 已评项目的平均评分；$\bar{r}_b$ 表示用户 u_b 已评项目的平均评分；N_{u_a} 表示用户 u_a 的近邻用户集合；$\mathrm{sim}(a,b)$ 表示目标用户 u_a 与近邻用户 u_b 的相似度。

2　改进的皮尔逊相关相似度

皮尔逊相关相似度计算方法如式 (1)，仅仅考虑了用户的共同评分项，而忽视了共同评分项目与每个用户所有评分项的比例关系。这会导致如果两个用户仅有极少数共同评分项目，并且两个

用户对共同评分项目的评分极度相似, 使用皮尔逊相关相似度计算得到的用户的相似度, 远远大于用户的真实相似度, 降低了推荐的准确度。例如, 用户 u_a 曾对 200 个项目进行了评分, 用户 u_b 对 300 个项目进行了评分, 两个用户仅拥有 10 个共同评分项目, 且两个用户对每个共同评分项目的评分均相同。使用传统皮尔逊相关相似度计算两者的相似度为 1(两个用户完全相似)。但实际上, 除了 10 个共同评分项目以外, 用户 u_a 和用户 u_b 还各自拥有大量的非共同评分项目, 两个用户的喜好并不完全相同, 利用皮尔逊相关相似度得到的结果远远大于两个用户的真实相似度。针对这个问题, 本文在皮尔逊相关相似度基础上, 引入交占比系数来缓解共同评分项占比的问题, 交占比反映了两个用户的共同评分项在两个用户评分中的占比, 加入交占比系数的皮尔逊相关相似度计算公式如下:

$$\mathrm{sim}(a,b)=\frac{2\times|s_{ab}|}{|s_a|+|s_b|}\times\frac{\sum\limits_{s_i\in s_{ab}}(r_{ai}-\bar{r}_a)(r_{bi}-\bar{r}_b)}{\sqrt{\sum\limits_{s_i\in s_{ab}}(r_{ai}-\bar{r}_a)}\sqrt{\sum\limits_{s_i\in s_{ab}}(r_{bi}-\bar{r}_b)}} \tag{3}$$

式中: $|s_{ab}|$ 表示用户 u_a 和用户 u_b 共同评分项目的个数; $|s_a|$ 表示用户 u_a 的评分项目个数; $|s_b|$ 表示用户 u_b 的评分项个数; 其他变量的含义和式 (1) 相同。

表 1 为用户评分示例, u_1、u_2 和 u_3 表示 3 个用户, $s_1, s_2\cdots, s_8$ 表示 8 个项目, 表中的值表示用户对项目的评分, 表中的空值 (—) 表示该用户未曾对该项目评分。根据式 (1) 计算用户 u_1 和用户 u_2 的相似度, u_1 和 u_2 的共同评分项集合 $s_{12}=\{s_3,s_4,s_5\}$, u_1 和 u_2 对 s_{12} 的评分均为 [2,3,5], 得到 $\mathrm{sim}(1,2)=1$, 显然这并不能准确的反映用户 u_1 和用户 u_2 的相似程度。使用加入交占比的式 (3) 计算用户 u_1 和用户 u_2 的相似度, $|s_{12}|=3$, $|s_1|=6$, $|s_2|=5$, 得到 $\mathrm{sim}(1,2)\approx 0.545$, 显然 0.545 更符合用户 u_1 和用户 u_2 的真实相似度。

3　基于图游走的协同过滤推荐算法 (GW_CF)

相似度计算是协同过滤推荐算法的关键部分, 得到用户相似度之后可以确定用户的近邻用户集合。但以往计算用户的相似度时只考虑用户的直接相似相似度, 这样将会遗失目标用户的间接近邻用户[15-16]。例如图 1 所示, u_1、u_2 和 u_3 表示 3 个用户, $s_1, s_2, \cdots, s_5$ 表示用户 u_1 的评分项目, $s_4, s_5, \cdots s_8$ 表示用户 u_2 的评分项目, $s_6, s_7, \cdots, s_{10}$ 表示用户 u_3 的评分项目。$\mathrm{sim}(1,2)$、$\mathrm{sim}(2,3)$、$\mathrm{sim}(1,3)$ 表示用户 u_1、u_2 和 u_3 的相似度。依据式 (3) 计算用户 u_1 和用户 u_3 的相似度, 由于用户 u_1 和用户 u_3 没有共同评分项, 所以 $\mathrm{sim}(1,3)=0$。但是用户 u_1 和 u_2 拥有共同评分项目 s_4 和 s_5, 那么 $\mathrm{sim}(1,2)>0$, 同理 $\mathrm{sim}(2,3)>0$。由于相似性具有传递性, 因此用户 u_1 和 u_3 可以通过共同的相似用户 u_2 建立间接相似度, 使得 $\mathrm{sim}(1,3)>0$。如果两个用户没有共同评分项目, 但间接相似度大于 0, 称这两个用户为间接近邻用户。在数据稀疏时, 为用户寻找间接近邻用户能够有效地提高推荐的准确度。本文提出了基于图游走的方法, 首先根据用户的直接相似度矩阵建立用户网络图, 其次在用户网络图上进行游走计算间接相似度, 然后根据间接相似度和直接相似度重建用户的相似度矩阵, 最后进行推荐。

表 1　用户评分示例表

Table 1　User rating

用户	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
u_1	4	—	2	3	5	1	—	4
u_2	—	3	2	3	5	—	2	—
u_3	—	3	4	3	—	1	—	4

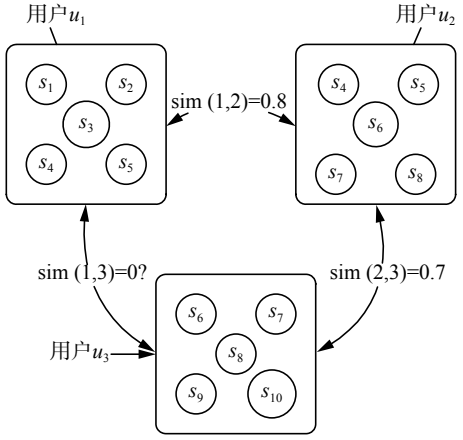

图 1　间接相似度关系图

Fig. 1　Indirect similarity diagram

3.1　构建用户网络图

使用用户网络图来说明用户间的相似关系,从目标用户开始游走后停留在某个用户的概率越高意味着它与目标用户更相似。为了建立用户网络图, 首先使用式 (3) 计算用户间的直接相似度, 然后根据直接相似度建立用户近邻矩阵。为每个用户选择 T 个直接近邻用户, 其他非 T 用户的相似度置 0, 得到的近邻矩阵如式 (4) 所示:

$$\mathbf{SU}=\begin{bmatrix} \mathrm{su}_{11} & \mathrm{su}_{12} & \cdots & \mathrm{su}_{1n} \\ \mathrm{su}_{21} & \mathrm{su}_{22} & \cdots & \mathrm{su}_{2n} \\ \vdots & \vdots & & \vdots \\ \mathrm{su}_{n1} & \mathrm{su}_{n2} & \cdots & \mathrm{su}_{nn} \end{bmatrix} \tag{4}$$

式中: 对每个用户 u_a 建立 T 近邻集合 N_{u_a}; 如果用

户 u_b 不是用户 u_a 的 T 近邻用户，则 $\mathrm{su}_{ab}=0$；若用户 u_b 是用户 u_a 的 T 近邻用户，则 $\mathrm{su}_{ab}=\mathrm{sim}(a,b)$。在游走过程中不考虑用户和自身的相似度，所以令 $\mathrm{su}_{aa}=0$。

然后对矩阵 **SU** 按列进行归一化，得到矩阵 $\mathbf{SU}^*$，以矩阵 $\mathbf{SU}^*$ 作为邻接矩阵建立用户网络图。矩阵 $\mathbf{SU}^*$ 中的 SU^*_{ab} 表示从当前用户节点 u_b 下一步游走到用户节点 u_a 的概率。

3.2 基于用户网络图游走

用向量 $\boldsymbol{r}^k=[r_1^k\ r_2^k\ldots r_b^k\ldots r_n^k]$ 中 r_b^k 表示第 k 次游走之后停留在节点 u_b 的概率，向量 $\mathbf{su}_a=[\mathbf{su}_{a1}\ \cdots\ \mathbf{su}_{ab}\ \cdots\ \mathbf{su}_{an}]$ 中 $\mathbf{su}_{ab}=\mathrm{SU}^*_{ab}$，则向量 $\boldsymbol{r}_a^{k+1}=\mathbf{su}_a\times(\boldsymbol{r}^k)^{\mathrm{T}}$ 为 k+1 次游走后停留在节点 u_a 的概率。整个用户网络图的游走过程公式如下：

$$\left(\boldsymbol{r}^{k+1}\right)^{\mathrm{T}}=\mathbf{SU}^*\times\left(\boldsymbol{r}^k\right)^{\mathrm{T}} \tag{5}$$

式中：$\mathbf{SU}^*$ 为用户网络图的邻接矩阵；$\boldsymbol{r}^k$ 为第 k 次游走后停留在各个节点的概率向量；$r_b^0=\begin{cases}1, b=a\\0, b\neq a\end{cases}$，其中 $b=a$ 表示从用户节点 u_a 开始游走。

在用户网络图中存在着与其他用户的相似度都很低甚至可以忽略不计的特殊用户节点。在用户网络图中此类节点只有入度，没有出度，如图 2 中节点 D，此时由于图中 D 节点只有入度，没有出度，用户网络图演变为非强连通图，以式 (5) 的方法游走到图中节点 D 时将无法跳转到其他节点。整个用户网络图的游走最终停留在类似节点 D 的死节点，无法求得用户的间接相似度，因此对式 (5) 进行变形如下：

$$\left(\boldsymbol{r}^{k+1}\right)^{\mathrm{T}}=p\times\mathbf{SU}^*\times\left(\boldsymbol{r}^k\right)^{\mathrm{T}}+(1-p)\times\boldsymbol{t}^{\mathrm{T}} \tag{6}$$

式中：p 表示 n 次游走后在当前节点继续游走的概率；$(1-p)$ 表示随机远程跳转到目标节点的概率。p 的大小与式 (6) 的收敛速度成反比，p 太大会导致收敛速度太慢从而影响算法的性能，p 如果太小则无法反映游走的效果，因此令 $p=0.85$。向量 $\boldsymbol{t}=[t_1\ t_2\ \cdots t_n]$ 表示远程跳转的目标节点，$t_b=\begin{cases}1, b=a\\0, b\neq a\end{cases}$。

当式 (6) 经过有限次迭代后，向量 $\boldsymbol{r}^k$ 收敛[17-18]。在理想情况下，当 k 趋于无穷大时，$\boldsymbol{r}^{k+1}=\boldsymbol{r}^k=r$，那么式 (6) 可以表示为 $\boldsymbol{r}^{\mathrm{T}}=p\times\mathbf{SU}^*\times\boldsymbol{r}^{\mathrm{T}}+(1-p)\times\boldsymbol{t}^{\mathrm{T}}$。对式 (6) 进一步变形得到式 (7)，在从不同用户顶点开始游走查找它的间接相似用户时，$(1-p)\times(\boldsymbol{I}-p\times\mathbf{SU}^*)^{-1}$ 只需要计算一次。相对于式 (6) 的多次迭代，式 (7) 大大降低了计算的复杂度。

$$\boldsymbol{r}^{\mathrm{T}}=(1-p)\times(\boldsymbol{I}-p\times\boldsymbol{S}^*)^{-1}\times\boldsymbol{t}^{\mathrm{T}} \tag{7}$$

式中：向量 $\boldsymbol{r}=[r_1\ \cdots\ r_b\cdots\ r_n]$ 中 r_b 表示从用户 u_a 开始游走最终停留在用户 u_b 的概率；r_b 被视作用户 u_a 和用户 u_b 的相对相似程度。不考虑用户和它本身的相似度，因此令 $r_{b=a}=0$。r_b 越大，表示用户 u_a 和目标用户 u_b 的间接相似度越高。

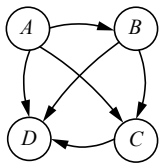

图 2 非强连通用户网络示例图

Fig. 2 Non-strong connected user network

3.3 重建相似度矩阵

向量 $\boldsymbol{r}$ 反映了各个用户与目标用户的相似程度相对大小，游走过程中的多次累加导致 r_b 过大，进行推荐之前需要将向量 $\boldsymbol{r}$ 映射到直接相似度同一个数量级上，因此需要重建相似度矩阵。

集合 $N'_{u_a}=\{u_b|u_b\in N_{u_a}\mathrm{su}_{\mathrm{ab}}\neq 0\}$ 表示直接近邻集合 N_{u_a} 中和目标用户相似度大于 0 的用户集合。利用该集合中用户与目标用户的直接相似度和向量 r 对应元素的映射关系，将向量 $\boldsymbol{r}$ 转化为目标用户和其他用户的间接相似度向量，重建的相似度计算公式为：

$$\mathrm{sim}(a,b)=\begin{cases}\dfrac{\sum\limits_{u_b\in N'_{u_a}}\dfrac{\mathrm{su}_{\mathrm{ak}}}{r_k}}{\left|N'_{u_a}\right|}\times r_a, & u_b\notin N'_{u_a}\\ \mathrm{su}_{ab}\ \ u_b\in N'_{u_a}\end{cases} \tag{8}$$

式中：$\mathrm{sim}(a,b)$ 表示目标用户 u_a 和用户 u_b 的重建相似度；su_{ab} 表示目标用户 u_a 和用户 u_b 的直接相似度；$\left|N'_{u_a}\right|$ 表示集合 N'_{u_a} 中用户个数。

3.4 生成推荐结果

以每个用户顶点为起点进行游走查找其间接相似用户，得到重建的用户相似度矩阵，进一步得到目标用户的近邻用户集合。然后利用式 (2) 对目标用户的未评分项目进行评分预测，并将评分最高的 Top-K 个项目推荐给目标用户。

4 基于图游走的并行协同过滤推荐算法

4.1 Spark 介绍

Spark 是基于内存的分布式并行计算平台[19]，它拥有 Hadoop 平台和 MapReduce 框架的全部优点，并且 Spark 运算的中间结果能存储在内存中，提高了并行计算的速度，因此 Spark 更适合进行数据挖掘与机器学习等需要迭代处理算法的实现[19-21]。Spark 集群启动时包括一个 Master 节点和若干个 Worker 节点，其中 Master 节点主要负责集群资源的管理，Worker 节点主要负责数据的计算。当在 Master 节点使用 spark-submit 命令提交

作业时, 首先在本地客户端启动一个 Driver 进程; Driver 进程会根据设置的参数向 Master 节点申请相应的集群资源, 主要有 Worker 节点个数、每个 Worker 节点上 Executor 的内存和 CPU 数量; Master 节点与 Worker 节点进行通信, 通知 Worker 节点启动 Executor 并向 Driver 进程注册; Driver 进程与 Worker 节点连接起来, 将需要执行的任务分配给集群中的各个 Worker 节点, Worker 节点按照任务分配从 HDFS 上读取数据并缓存到内存中, Driver 进程对各个 Worker 节点处理完的结果进行收集和汇总。在 Spark 平台实现基于图游走的协同过滤算法能够有效地提高算法的时间效率。

4.2　相似性计算的并行化

由于皮尔逊相关相似度计算公式较为复杂, 全局搜索较多, 因此在实现本文方法并行化时引入中间变量 Q, Q_{ai} 反映了用户 u_a 在项目 s_i 上的相似度权重, 计算公式如下:

$$Q_{ai}=\frac{(r_{ai}-\bar{r}_a)}{\sqrt{\sum_a (r_{ai}-\bar{r}_a)}} \tag{9}$$

式中: r_{ai} 表示用户 u_a 对项目 s_i 的评分; $\bar{r}_a$ 表示用户 u_a 的评分均值。皮尔逊相关相似度公式可以变形为

$$\mathrm{sim}(a,b)=\frac{2\times|s_{ab}|}{|s_a|+|s_b|}\times\sum_{s_i\in s_{ab}} Q_{ai}\times Q_{bi} \tag{10}$$

因此, 求用户 u_a 和用户 u_b 的相似度 $\mathrm{sim}(a,b)$ 的过程转化为 5 步: 1)对于用户 u_a 和用户 u_b 的共同评分项 $s_i\in s_{ab}$, 计算中间变量 Q_{ai} 和 Q_{bi}; 2)求用户 u_a 和用户 u_b 的 Q 乘积 $Q_{ai}\times Q_{bi}$; 3)计算 $\sum_{s_i\in s_{ab}} Q_{ai}\times Q_{bi}$ 得到皮尔逊相关相似度; 4)交占比系数得到用户的直接相似度; 5)使用游走的方法求得用户的间接相似度并重建相似度。

4.3　基于图游走的协同过滤算法并行化流程

基于图游走的协同过滤推荐算法在 Spark 平台上的并行化包括 3 部分, 分别是读入数据创建 RDD、计算用户的相似度以及生成推荐列表, 该算法的并行化主要体现在计算用户相似度和生成推荐列表。基于图游走的并行协同过滤推荐算法示意图如图 3 所示。

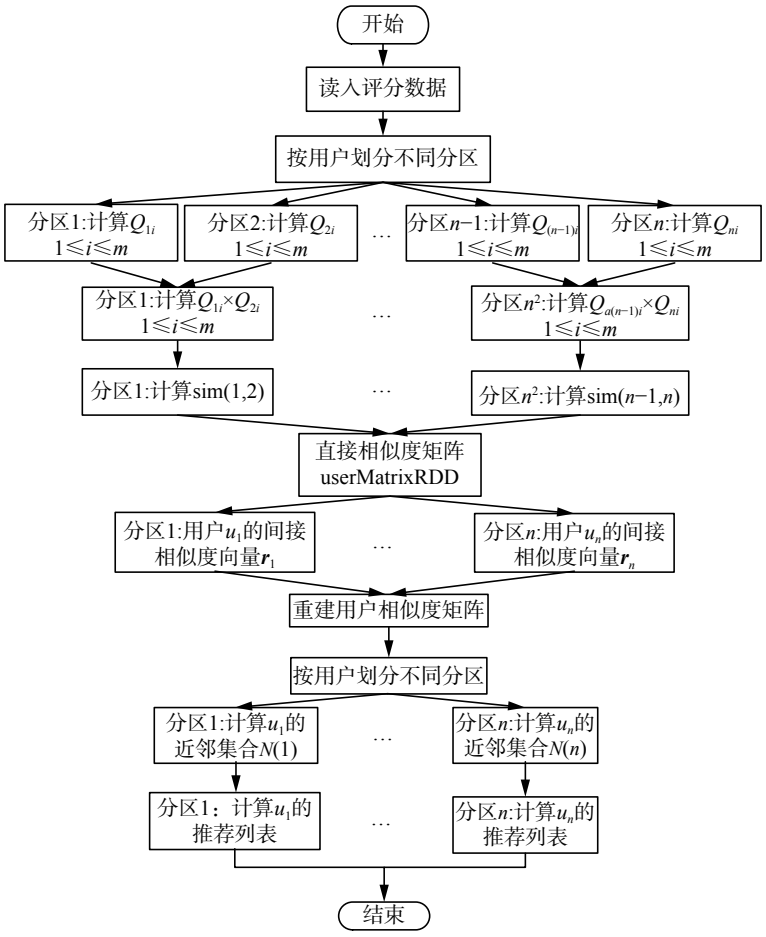

图 3　基于图游走的并行协同过滤示意图

Fig. 3　Parallel collaboration filtering based on graph walk schematic

具体过程如下：

1) 读入用户行为数据，构建 RDD_1；

2) 将 RDD_1 转换成 $(u_a,(s_i,r_{ai}))$ 形式的 RDD_2，$(1\leqslant a\leqslant n,1\leqslant i\leqslant m)$ 按照用户 ID 进行聚集得到 $RDD_3\,(u_a,\text{Iterable}[(s_i,r_{ai})])$，使用 flatMap 算子计算每个用户的中间变量 Q，并按照项目 ID 进行聚集得到 RDD_4；

3) 根据 RDD_4 计算用户 u_a 和用户 u_b 的 $Q_{ai}\times Q_{bi}$，得到形如 $((u_a,u_b),(Q_{ab},1,\text{toNum}))$ 的 RDD_5，其中的 1 和 toNum 是为了便于计算交占比系数而设置的；

4) 将 3) 的 RDD_5 使用 ReduceByKey 算子统计其共同评分项，计算结合交占比系数的相似度，得到形如 $((u_a,u_b),\text{sim}_{ab})$ 的 RDD_6；

5) 利用 4) 的相似度 RDD_6，构造用户相似度矩阵 userMatrixRDD，使用 Spark 中的线性代数库 Breeze，调用其库函数 inv() 计算 userMatrixRDD 的逆矩阵 invMatrixRDD，进一步通过式 (7) 和 (8) 求得得间接相似度，重建相似度矩阵得到 RDD_7；

6) 根据 RDD_7 按用户划分得到 RDD_8，并进一步得到目标用户的近邻用户集合 RDD_9，最后进行推荐。

5　实验与评价

实验使用 Movielens 数据集和 IPTV 数据集[20]进行实验。Movielens 是一个基于 Web 的研究型推荐系统，用于接收用户对电影的评分并提供电影的推荐列表，Movielens 数据集在协同过滤研究领域得到了广泛研究，也是使用最多的数据集之一。IPTV 数据集来源于天津市 IPTV 电视用户的收视日志数据，经过对日志数据进行预处理和隐式评分处理，形成 IPTV 数据集。相比于 Movielens 数据集，IPTV 数据集应用性更高。Movielens-100k 数据集包含 943 用户，1 682 项目，共 10 万条评分记录；Movielens-1M 数据集包含 6 040 个用户和 3 952 个项目，共计 100 万条评分记录。IPTV 数据集选取 193 用户，8 200 项目，共计 43 175 条评分记录。

使用平均绝对误差 MAE 和准确率 precision 作为衡量推荐准确度的指标。MAE 反映了评分预测误差的大小，误差越小表明推荐准确度越高，计算公式 (11) 如下：

$$\text{MAE}=\frac{\sum_{i=1}^{N}\left|r_i-r_i'\right|}{N}\tag{11}$$

式中：N 表示预测评分记录数量，r_i 表示该条记录的预测评分，r_i' 表示该条记录的实际评分。

准确率 precision 反映了推荐的准确度，准确率越高，表明推荐的准确度越高。准确率的计算公式为

$$\text{precision}=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{|R(u)|}\tag{12}$$

式中：$|R(u)|$ 表示推荐给用户的所有项目个数；$|R(u)\cap T(u)|$ 表示推荐给用户的项目中推荐正确的项目个数。

以加速比作为可扩展性的实验指标，加速比为

$$S_p=\frac{T_1}{T_p}\tag{13}$$

式中：T_1 表示使用 1 个节点时任务执行时间；T_p 表示使用 p 个节点时任务执行时间；S_p 表示加速比，反映了并行后运行效率的提升情况。$S_p=p$ 时为线性加速比，加速比越接近线性加速比时，算法的可扩展性越好。

5.1　相似度交占比系数的有效性验证实验

在原始的皮尔逊相关相似度的基础上，为了比较加入交占比 (YPCC) 和未加入交占比 (PCC) 对预测评分误差的影响进行本次实验。此次实验使用 Movielens-100k 数据集，共 943 用户，1 682 个项目，共 10 万条评分记录，稀疏度为 94.12%，训练集和测试集按 8:2 分割。实验结果如图 4。

图 4 中，Top-K 表示近邻用户选取的个数，MAE 表示评分预测的平均绝对误差，PCC 表示未加入交占比系数计算用户相似度进行评分预测的误差曲线，YPCC 表示加入交占比系数计算用户相似度进行推荐的误差曲线。

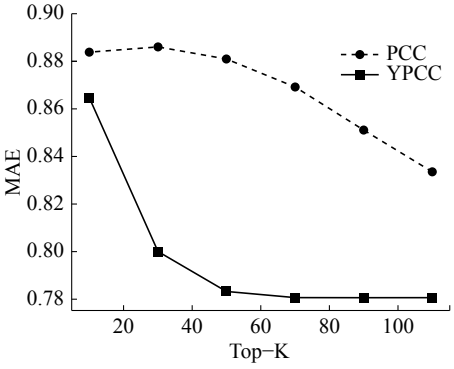

图 4　交占比系数有效性验证实验

Fig. 4　Trial ratio validity validation experiment

从图 4 中可以看出，协同过滤推荐算法的预测评分误差受到近邻用户个数 Top-K 的影响。随着近邻用户个数 Top-K 的增加，PCC 和 YPCC 曲线均呈现下降趋势并最终趋于稳定，但是 YP-

CC 曲线明显低于 PCC 曲线, 尤其 Top-K 在 [40, 60] 时差距最明显。实验结果表明, 无论近邻用户个数如何选取, 在皮尔逊相关相似度上加入交占比系数均可以有效地减小评分预测误差。

5.2　基于图游走方法的有效性验证实验

5.2.1　Movielens 数据集实验

为了验证基于图游走方法在降低评分预测误差和提高推荐准确率上的有效性, 本次实验使用 Movielens-100k 数据集, 训练集和测试集按 8:2 分割。先通过实验确定用户直接近邻个数 T 的最优取值, 然后比较在不同的推荐近邻个数 Top-K 下, 本文方法和基于用户的协同过滤推荐算法 (BSCF) 与基于聚类的协同过滤推荐算法 (k-means_CF) 的 MAE 和 precision。

图 5 为选取不同直接近邻个数时的评分预测误差曲线, T 表示直接近邻用户选取的个数。图 5 表明, 当 $T>60$ 时, MAE 趋于稳定。

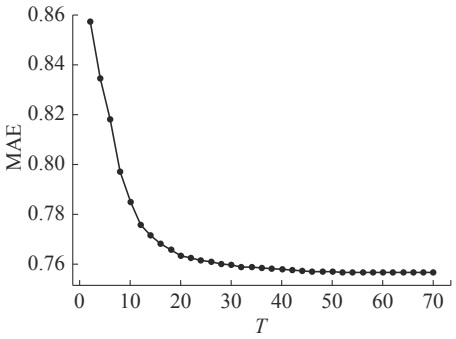

图 5　参数 T 测试图

Fig. 5　Parameter T test chart

图 6 的实验中 T=60。推荐时近邻用户个数 Top-K 作为单一变量, 对基于图游走的协同过滤推荐算法 (GW_CF)、基于用户的协同过滤推荐算法 (BSCF)、基于聚类的协同过滤推荐算法 (k-means_CF) 进行对比实验。图 6Top-K 表示推荐时近邻用户选取的个数, MAE 表示评分预测的平均绝对误差。从图中可以看出, 随着近邻用户个数 Top-K 的增加, 3 条曲线均呈下降趋势, BSCF 曲线和 k-means_CF 曲线比较接近, GW_CF 曲线明显低于另两条曲线, 当 Top-K 大于 80 时更加明显。实验结果表明: GW_CF 算法在降低评分预测误差方面是有效的。

图 7 中虚线反映了使用 GW_CF 推荐的准确度, 实线反映了使用 BSCF 推荐的准确度。生成推荐列表时推荐项目数为 10, 从图中可以看出, 随着近邻用户个数 Top-K 的增加, 两条曲线呈上升趋势, GW_CF 准确率曲线高于 BSCF 曲线。实验结果表明, 基于图游走的协同过滤推荐算法 GW_CF 可以有效地提高推荐准确率。

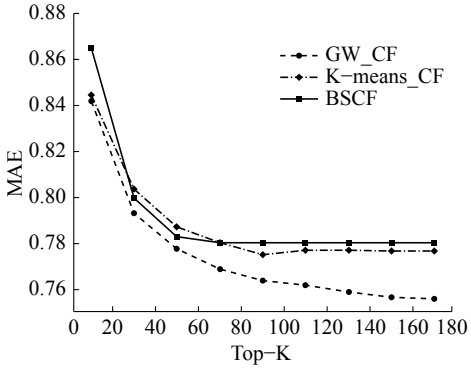

图 6　图游走效果图

Fig. 6　Random walk effect graph

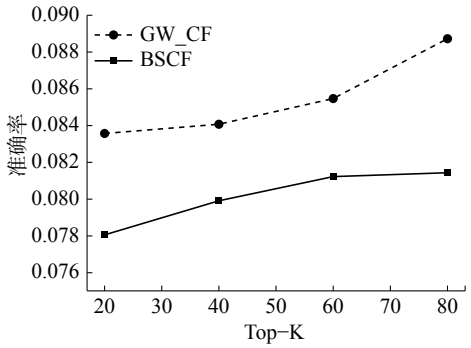

图 7　准确率对比图

Fig. 7　Accuracy comparison chart

5.2.2　IPTV 隐式评分数据集实验

为了验证基于图游走方法在降低评分预测误差和提高推荐准确率上的有效性, 本次实验使用 IPTV 数据集, 训练集和测试集按 8:2 分割。先通过实验确定用户直接近邻个数 T 的最优取值, 然后比较在不同的推荐近邻个数 Top-K 下, 基于图游走的协同过滤推荐算法 (GW_CF) 和基于用户的协同过滤推荐算法 (BSCF) 的 MAE 和 precision。

图 8 为选取不同直接近邻个数时的评分预测误差曲线。图 8 表明, 当 $T>20$ 时, MAE 趋于稳定。

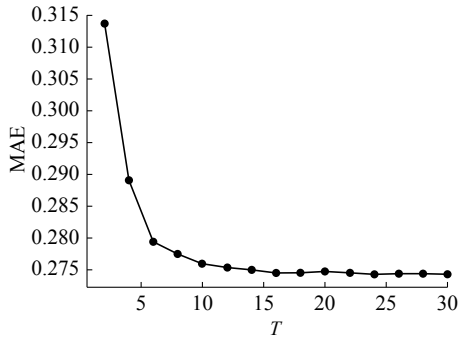

图 8　参数 T 测试图

Fig. 8　Parameter T test chart

图 9 的实验中 T=20, 推荐近邻用户 Top-K 作为单一变量, 对基于图游走的协同过滤推荐算

法 (GW_CF) 和基于用户的协同过滤推荐算法 (BSCF) 进行对比实验。从图 9 中可以看出，随着近邻用户个数 Top-K 的增加，两条曲线均呈下降趋势，GW_CF 曲线明显低于 BSCF 曲线。实验结果表明：GW_CF 算法在降低评分预测误差方面是有效的。

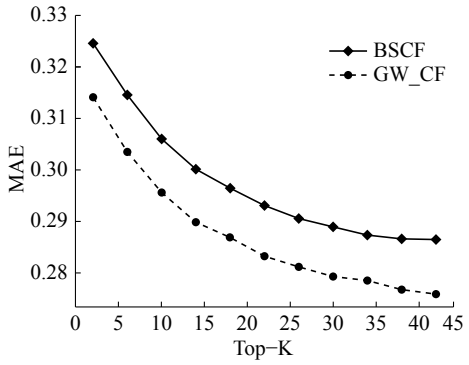

图 9　图游走效果图
Fig. 9　Random Walk Effect Graph

图 10 中生成推荐列表时推荐项目数为 10，随着近邻用户个数 Top-*K* 的增加，两条曲线呈上升趋势，GW_CF 准确率曲线趋势更明显并且高于 BSCF 曲线。实验结果表明，在一般情况下，GW_CF 比 BSCF 拥有更高的推荐准确率。

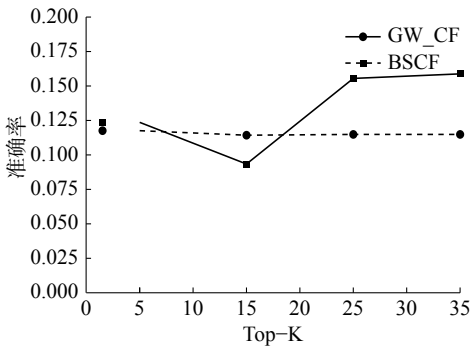

图 10　准确率对比图
Fig. 10　Accuracy comparison chart

5.3　基于图游走的并行协同过滤推荐算法可扩展性实验

为了验证基于图游走的并行协同过滤推荐算法的可扩展性，使用 Movielens-1M 和 Movielens-100k 数据集在 Spark 平台进行实验。其中 1M 数据集包含 6 040 个用户和 3 952 个项目，共计 100 万条评分记录；100k 数据集包含 943 用户，1 682 项目，共 10 万条评分记录。实验在 Spark 集群上实现，集群环境包括 6 个节点，一个 Master 节点，5 个 worker 节点，每个节点的配置相同，且处在同一个局域网内，操作系统为 CentOs6.5，CPU 为 E5-2620 v4，核心频率 2.10 GHz，节点内存 32 GB。加速比结果如图 11。

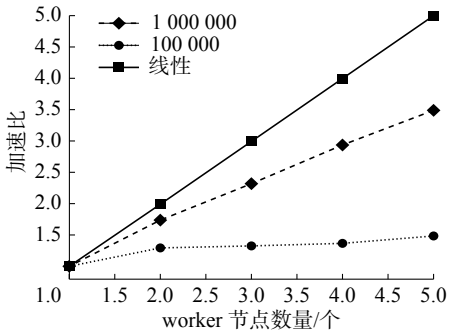

图 11　加速比示意图
Fig. 11　Speed-up ratio graph

从图 11 中可以看出，随着节点个数的增加，加速比呈现上升趋势，100 万数据集更逼近线性加速比。实验结果表明，并行协同过滤推荐算法在大规模数据集的情况下有较好的可扩展性。

6　结束语

本文针对协同过滤推荐算法中的数据稀疏性问题和可扩展性问题进行研究。针对稀疏性问题，在基于用户的协同过滤推荐算法的基础上，首先为传统的皮尔逊相关相似度引入交占比系数来计算用户的直接相似度，其次提出一种基于图游走方法来计算用户间接相似度，并重建相似度矩阵和进行推荐。针对可扩展性问题，在 Spark 平台上实现本文方法的并行化。通过在 Movielens 数据集和 IPTV 数据集上进行实验，先后验证了加入交占比系数和基于图游走的方法在提高推荐准确度上的有效性，以及本文方法的可扩展性。实验结果表明，本文的方法在提高推荐准确度上是有效的，并且在大规模数据上拥有较好的可扩展性。

参考文献：

[1] 黄立威,江碧涛,吕守业,等.基于深度学习的推荐系统研究综述 [J].计算机学报,2018,41(7):1619-1647.
HUANG Liwei, LIU Yanbo, LI Deyi. Deep learning based recommender systems[J].Chinese journal of computers. 2018,41(07):1619-1647.

[2] 孙光福, 吴乐, 刘淇, 等. 基于时序行为的协同过滤推荐算法 [J]. 软件学报, 2013, 24(11): 2721–2733.
SUN Guangfu, WU Le, LIU Qi, et al. Recommendations based on collaborative filtering by exploiting sequential behaviors[J]. Journal of software, 2013, 24(11): 2721–2733.

[3] 许智宏, 蒋新宇, 董永峰, 等. 一种基于 Spark 的改进协同过滤算法研究 [J]. 计算机应用与软件, 2017, 34(5): 247–254, 278.
XU Zhihong, JIANG Xinyu, DONG Yongfeng, et al. An improved collaborative filtering algorithm based on Spark[J]. Computer applications and software, 2017, 34(5): 247–254, 278.

[4] SUN Baoshan, DONG Lingyu. Dynamic model adaptive to user interest drift based on cluster and nearest neighbors[J]. IEEE access, 2017, 5: 1682–1691.
[5] 彭宏伟, 靳远远, 吕晓强, 等. 一种基于矩阵分解的上下文感知 POI 推荐算法 [J/OL]. 计算机学报: (2018-05-14)[2018-05-30]. http://kns.cnki.net/kcms/detail/11.1826.TP.20180512.2150.008.html.
PENG Hongwei, JIN Yuanyuan, LÜ Xiaoqiang, et al. Context-aware POI recommendation based on matrix factorization[J]. Chinese Journal of Computers: (2018-05-14)[2018-05-30]. http://kns.cnki.net/kcms/detail/11.1826.TP.20180512.2150.008.html.
[6] WU Jibing, YU Lianfei, ZHANG Qun, et al. Multityped community discovery in time-evolving heterogeneous information networks based on tensor decomposition[J]. Complexity, 2018, 2018: 9653404.
[7] MA Wenping, WU Yue, GONG Maoguo, et al. Local probabilistic matrix factorization for personal recommendation[C]//Proceedings of the 13th International Conference on Computational Intelligence and Security. Hong Kong, China, 2017: 97–101.
[8] 杨志文, 刘波. 基于 Hadoop 平台协同过滤推荐算法 [J]. 计算机系统应用, 2013, 22(7): 108–112.
YANG Zhiwen, LIU Bo. Hadoop-based collaborative filtering recommendation algorithm[J]. Computer systems and applications, 2013, 22(7): 108–112.
[9] LU F, HONG L, CHANGFENG L. The improvement and implementation of distributed item-based collaborative filtering algorithm on Hadoop[C]//Proceedings of the 34th Chinese Control Conference. Hangzhou, China, 2015: 9078–9083.
[10] KUPISZ B, UNOLD O. Collaborative filtering recommendation algorithm based on Hadoop and Spark[C]// Proceedings of 2015 IEEE International Conference on Industrial Technology. Seville, Spain, 2015: 1510–1514.
[11] 冷亚军, 陆青, 梁昌勇. 协同过滤推荐技术综述 [J]. 模式识别与人工智能, 2014, 27(8): 720–734.
LENG Yajun, LU Qing, LIANG Changyong. Survey of recommendation based on collaborative filtering[J]. Pattern recognition and artificial intelligence, 2014, 27(8): 720–734.
[12] 范波, 程久军. 用户间多相似度协同过滤推荐算法 [J]. 计算机科学, 2012, 39(1): 23–26.
FAN Bo, CHENG Jiujun. Collaborative filtering recommendation algorithm based on user's multi-similarity[J]. Computer Science, 2012, 39(1): 23–26.
[13] 徐堃, 朱小柯, 荆晓远. 基于改进协同过滤的个性化 web 服务推荐方法研究 [J]. 计算机技术与发展, 2018, 28(1): 64–68.
XU Kun, ZHU Xiaoke, JING Xiaoyuan. Research on personalized web service recommendation based on improved collaborative filtering[J]. Computer technology and development, 2018, 28(1): 64–68.
[14] WU Xiaokun, CHENG Bo, CHEN Junliang. Collaborative filtering service recommendation based on a novel similarity computation method[J]. IEEE transactions on services computing, 2017, 10(3): 352–365.
[15] 肖春景, 夏克文, 乔永卫. 基于时序逆影响的随机游走推荐算法 [J]. 计算机应用研究, 2018, 35(8): 2304–2307.
XIAO Chunjing, XIA Kewen, QIAO Yongwei. Temporal inverse influence based recommendation method by using random walk[J]. Application research of computers, 2018, 35(8): 2304–2307.
[16] 王鹤, 邬春学. 基于图结构和项目类型的协同过滤推荐算法 [J]. 数据通信, 2016(5): 44–47.
WANG He, WU Chunxue. Collaborative filtering recommenddation algorithm based on graph structure and item type[J]. Data communications, 2016(5): 44–47.
[17] 宫秀文, 张佩云. 基于 PageRank 的社交网络影响最大化传播模型与算法研究 [J]. 计算机科学, 2013, 40(S1): 136–140.
GONG Xiuwen, ZHANG Peiyun. Research on propagation model and algorithm for influence maximization in social network based on pageRank[J]. Computer science, 2013, 40(S1): 136–140.
[18] HU Yan, PENG Qimin, HU Xiaohui, et al. Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering[J]. IEEE transactions on services computing, 2015, 8(5): 782–794.
[19] 黄筱云, 董国海, 常佳夫, 等. Level set 函数快速步进重构并行算法的改进 [J]. 哈尔滨工程大学学报, 2017, 38(6): 836–842.
HUANG Xiaoyun, DONG Guohuai, CHANG Jiafu, et al. Improvement of parallel fast marching method for reconstruction of level set function[J]. Journal of Harbin Engineering University, 2017, 38(6): 836–842.
[20] LIU Tiantian, FANG Zhiyi, ZHAO Chen, et al. Parallelization of a series of extreme learning machine algorithms based on spark[C]//Proceedings of the IEEE/ACIS 15th International Conference on Computer and Information Science. Okayama, Japan, 2016.
[21] 顾军华, 官磊, 张建, 等. 基于 Hadoop 的 IPTV 隐式评分模型 [J]. 计算机应用, 2017, 37(11): 3188–3193.
GU Junhua, GUAN Lei, ZHANG Jian, et al. IPTV implicit scoring model based on Hadoop[J]. Journal of computer applications, 2017, 37(11): 3188–3193.

作者简介:

顾军华, 男, 1966 年生, 教授, 博士生导师, CCF 会员, 中国离散数学学会常务理事, 河北省计算机学会副理事长。主要研究方向为数据挖掘、智能信息处理等。完成科研项目 30 余项, 发表学术论文 50 余篇。

谢志坚, 男, 1995 年生, 硕士研究生, 主要研究方向为数据挖掘与机器学习。

武君艳, 女, 1994 年生, 硕士研究生, 主要研究方向为数据挖掘与计算机仿真。