

# 基于路径融合的多图层推荐算法<sup>①</sup>

李宏恩, 肖如良, 陈洪涛, 赵 婷, 李源鑫

(福建师范大学 软件学院, 福州 350108)

**摘 要:** 图推荐算法中资源分配矩阵的计算和存储占用大量开销, 为了提高基于图推荐的效率, 提出一种基于路径融合的多图层混合策略推荐算法(PIML). 该算法中, 为了提高推荐精度并综合考虑多种因素以给出全面推荐, 基于路径融合方法, 利用分流策略优化资源分配, 并对时间和评分因素加权, 将人口统计学和物品内容信息融入到多图层, 实现基于二部图推荐. 实验结果表明: 该算法没有增加时间开销, 提高了推荐精度, 使推荐更全面更灵活, 并可实时推荐.

**关键词:** 路径融合; 多图层; 资源分配; 分流策略

## Path Integration-Based Multiple Layers Recommendation

LI Hong-En, XIAO Ru-Liang, CHEN Hong-Tao, ZHAO Ting, LI Yuan-Xin

(Faculty of Software, Fujian Normal University, Fuzhou 350108, China)

**Abstract:** In graph-based recommendation, the calculation and storage of resource allocation matrix take up a lot of overhead, in order to improve efficiency of algorithm. Path Integration-Based Multiple Layers Recommendation(PIML) was proposed. In the algorithm, path integration-based shunt strategy was used to dynamically allocate resources, and timestamp and ratings were added to the graph. On the other hand, demographic information and item content information were added to the multiple layers structure, considering a variety of factors, it gave more comprehensive recommendation. Experimental results show that the algorithm improves the accuracy without increasing time cost; and makes recommendation more comprehensive and can give real-time recommendation.

**Key words:** path integration; multiple layers; allocation of resources; shunt strategy

## 1 引言

21 世纪电子商务的飞速发展, “信息过载”问题给人们带来很大的负担. 推荐系统<sup>[1]</sup>为用户推荐感兴趣的商品或信息, 在一定程度上满足了用户的需求, 但是随着互联网信息日益加大, 传统推荐系统在对大量用户做推荐时仍面临诸多问题, 所以, “更高效的推荐系统”成为一个更具意义的研究课题.

目前, 主要的推荐算法<sup>[2]</sup>有协同过滤推荐<sup>[3-4]</sup>、基于内容的推荐<sup>[5]</sup>、基于关联规则推荐<sup>[6-7]</sup>和基于图的推荐. 传统的基于图的推荐算法<sup>[8]</sup>主要有基于资源分配矩阵的二部图算法、随机游走的二部图算法<sup>[9]</sup>和路径融合算法, 基于图的推荐算法在做 Top-N 推荐时利用

的信息隐含在用户和物品的选择关系之中, 必须分析图结构才能给出合理的推荐. 给用户  $u$  推荐物品的任务就可以转化为度量用户节点  $vu$  和物品节点  $vi$  在图上的相关性, 相关性越高的物品在推荐列表中的权重就越高.

本文引入二部图理论和基于路径融合方法, 利用分流策略优化资源分配, 通过学习电流在电网中的分配来模拟用户对物品喜好值的分配, 提出一种基于路径融合的多图层算法(PIML), 主要应用在隐反馈数据集的 Top-N 推荐问题上, 旨在提高二部图推荐算法的推荐精度和推荐广度. 第 2 节介绍图推荐算法的相关研究工作, 第 3 节提出基于路径融合的多图层算法,

① 基金项目: 教育部规划基金(11YJA860028); 福建省自然科学基金(3013J01219)

收稿时间: 2013-09-27; 收到修改稿时间: 2013-11-11

并应用到基于图的推荐中,第 4 节讨论该算法分别在 MovieLens 和 Netflix 数据集上做 Top-N 推荐的实验过程和结果分析,第 5 节总结全文。

### 2 相关工作

基于图的模型是推荐系统的重要内容。二部图是一种特殊的网络,包含两类节点:一类节点是用户,例如读者、观众等;另一类节点是物品,例如论文、电影等,仅允许不同类的节点间相连。

文献[10-11]提出一种基于资源分配矩阵的二部图推荐算法,假设每个物品均有一定的资源,一个物品节点占有资源的多少代表了目标用户对该物品喜好程度的大小,定义边权  $w_{ij}$  表示物品  $j$  分配给物品  $i$  的资源权重。比如在一个“用户-电影”的二部图中,有两部电影  $i$  和  $j$ ,如果一个用户已经观看了电影  $j$ ,那么  $w_{ij}$  表示电影  $j$  推荐电影  $i$  的强度。借鉴动态网络资源分配过程,该方法的推荐过程如下:

(1) 建立二部图。推荐系统包含  $n$  个用户和  $m$  个物品,若用户  $i$  对物品  $j$  产生过行为,则令  $a_{ij}=1$ ,否则令  $a_{ij}=0$ ,建立一个包含  $n+m$  个节点的二部图。

(2) 计算资源分配矩阵  $W$ 。物品  $j$  到物品  $i$  的资源分配权重  $w_{ij}$  计算如下( $D_j$  为节点  $j$  的度,即在二部图中与节点  $j$  直接相连顶点的个数):

$$W_{ij} = \frac{1}{D_j} \sum_{k=1}^n \frac{a_{ik} a_{jk}}{D_k} \tag{1}$$

(3) 令  $f_i=(a_{i1}, a_{i2}, \dots, a_{im})$  表示用户  $i$  对  $m$  个物品的初始资源分配,表示在对用户  $i$  做推荐之前,用户  $i$  对所有物品喜好程度的分配;  $f'_i$  表示用户  $i$  对  $m$  个物品的最终资源分配,表示对用户  $i$  做推荐之后,用户  $i$  对所有物品喜好程度的分配,则有  $f'_i = Wf_i$ 。

(4) 根据  $f'_i$  产生推荐列表。

图 1 表示一个由 5 个物品和 3 个用户组成的二部图,  $x$  表示物品,  $y$  表示用户,  $y_1$  为目标用户。

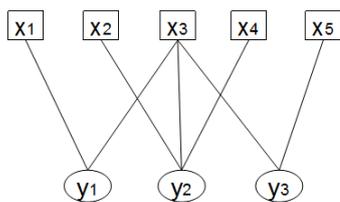


图 1 推荐系统对应的二部图

由图可知,  $y_1$  的初始资源分配为:  $f_1=(1,0,1,0,0)^T$ ,由公式(1)可计算出项目间的资源分配矩阵:

$$H_{k,n} = U_{k,n} S_{k,n} V_{k,n}^H = \sum_{i=1}^M u_{k,n}^i s_{k,n}^i (v_{k,n}^i)^H \tag{2}$$

目标用户  $y_1$  的最终资源分配如下:

$$\begin{cases} \sum_{i=1}^K \rho_{k,n} = 1 \\ \sum_{k=1}^K \sum_{n=1}^{N_c} \rho_{k,n} = N_c \end{cases} \tag{3}$$

所以用户  $y_1$  推荐的列表为  $(x_4, x_2, x_5)$ , 由于  $x_1, x_3$  是  $y_1$  用户使用过的物品, 所以不出现在推荐列表中。

Taher H.Haveliwala 的“Topic-Sensitive PageRank”提出一种基于随机游走的二部图 PersonalRank[12-13]算法。假设要给用户  $u$  推荐, 从用户  $u$  对应的节点  $v_u$  开始在二部图上进行随机游走。经过很多次游走后, 每个物品节点被访问到的概率会收敛到一个数, 即最终的推荐列表中物品的权重, 表示成公式:

$$P_T = \min \sum_{n=1}^{N_c} \sum_{k=1}^K \sum_{i=1}^M \rho_{k,n} P_{k,n}^i \tag{4}$$

虽然 PersonalRank 算法可以通过随机游走得到比较好的理论解释, 但在对每个用户推荐时, 都需要在整个二部图上进行迭代, 直到整个图上的每个顶点的 PR 值收敛, 这一过程时间杂度高, 不仅无法在线实时推荐, 而且离线生成推荐结果也相当耗时。

而基于资源分配矩阵算法的资源分配矩阵大、实时更新数据困难、忽略时间因素、没有考虑用户人口统计学信息和物品内容信息等, 有必要采用一种新颖的基于路径融合的多图层混合推荐算法, 该方法结合了分流策略优化资源分配, 旨在不增加时间开销的同时, 减少空间的开销, 加权时间和评分因素, 结合人口统计学和物品内容信息, 不仅提高推荐的效率, 还可以给出不同的推荐策略或给出策略组合推荐。在数据集不断更新情况下可给出实时推荐。

### 3 基于路径融合的多图层混合策略推荐算法

这里引用文献[8]的图层结构, 首先需要将用户行为数据、人口统计学数据和物品特征数据表示成二部图的形式. 如图 2 所示, 多层图  $G(V, E)$  由三个图层  $G_1$ 、 $G_2$  和  $G_3$  组成( $V$  为顶点集,  $E$  为边集). 其中根据人口统计学信息构成了用户-特征图层  $G_1(V_1, E_1)$ ; 根据用户历史行为产生用户-物品图层  $G_2(V_2, E_2)$ ; 根据物品特征分类信息构成物品-特征图层  $G_3(V_3, E_3)$ . 而评分信息, 时间戳信息等作为边的权重.

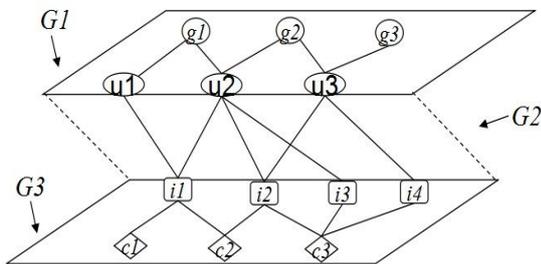


图 2 多层图结构  $G$

这里的引入分流策略主要是通过电流在电网中的分配来模拟图模型中用户对物品喜好值的分配, 可以有效地优化路径融合的资源分配并应用到 Top-N 推荐中, 我们称这种算法为基于路径融合的多图层混合策略推荐算法, 英文缩写为 PIML 算法. PIML 算法包括两个部分, 一部分是基于路径融合方法, 利用分流策略优化资源分配, 第二部分是多层图混合加权策略.

#### 3.1 结合分流策略的路径融合方法

根据路径融合方法, 图上两点相关性较高一般有如下特征: 1) 两点有很多路径相连; 2) 两点之间路径较短; 3) 两点之间路径不经过出度较大的点.

分流策略是指在基于图结构的推荐中, 假设图中每个节点都有一定的资源, 节点  $i$  分配给节点  $j$  的资源代表节点  $j$  与节点  $i$  的相关度大小, 而分配原则是: 对任一节点  $j$ , 流入  $j$  的资源等于流出  $j$  的资源, 即节点  $j$  从上层节点得到一定的资源并把全部资源按不同比例分配给下层节点.

##### 3.1.1 结合分流策略的路径融合规则

在对用户  $u$  推荐中, 利用分流策略的路径融合规则定义: 1) 单条路径从根节点  $v_u$  出发, 终止于一个物品节点  $v_i$ ; 2) 单条路径总是向远离根节点  $v_u$  的方向遍历,

即流向下层; 3) 单条路径不构成回路; 4) 遍历形成以一个  $v_u$  为根节点的树形结构, 叶节点均为物品节点; 5) 在同一图层中, 任一节点所在的层次  $h$  表示该节点与根节点的最短距离, 即树的高度. 图 3(a) 是以用户  $u_1$  遍历形成的树形结构图, 图 3(b) 是以  $u_1$  为根节点的资源流向图.

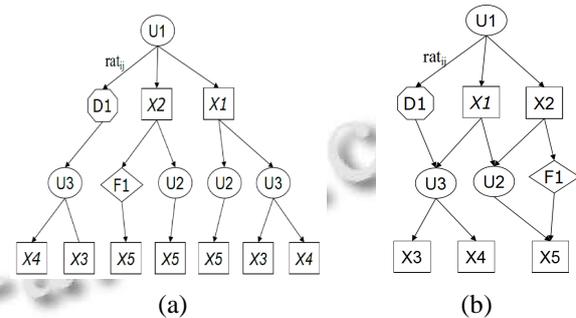


图 3 (a) 树形结构图, (b) 资源流向图

##### 3.1.2 利用分流策略的资源分配计算和推荐

在单一图层上利用分流策略的路径融合理论计算图中资源分配中, 在对用户  $u_1$  做推荐时遍历计算过程如图 3(a) 树状结构,  $u_1$  的资源按不同权重分配给第二层节点, 第二层节点再把资源按不同权重分配给下一层节点, 我们设定一个阈值  $h_{max}$ , 当树的深度达到  $h_{max}$  时停止分配. 树的第  $h$  层的节点表示为  $v(h)$ , 其分配得资源权重  $w_{v(h)}$  是从  $h-1$  层获得, 表示为:

$$\begin{cases} \sum_{n=1}^{N_k} \sum_{i=1}^M \rho_{k,n} b_{k,n}^i = R_k \\ \sum_{i=1}^k \rho_{k,n} = 1 \\ R_1 : R_2 : \dots : R_k = \gamma_1 : \gamma_2 : \dots : \gamma_k \end{cases} \quad (5)$$

$In(v)$  为节点  $v$  上层关联边集合,  $Out(v)$  为  $v$  节点下层关联边的集合,  $w_{v(h-1)}$ ,  $e$  是边  $e$  对应  $h-1$  层的顶点  $v(h-1)$  已有的资源权重,  $rat_e$  为边  $e$  上的评分权值.

加权用户行为的时间信息对图模型的优化, 又考虑热门物品的惩罚, 物品节点  $v_i$  获得的资源做为推荐的度量值  $R_{ui}$ :

$$P_{i,j} = \max \Delta P_{i,j} [n] + \max \Delta P_{j,i} [n] \quad (6)$$

其中,  $D_{max}$  为所有节点中最大的度,  $d$  为当前节点的度,  $\Delta T_{max}$  和  $\Delta t$  分别所有行为时间差的最大值和当前两次行为的时间差值,  $b, c$  为关系参数.

图 4, 利用上述路径融合算法为用户  $u_1$  推荐物品.

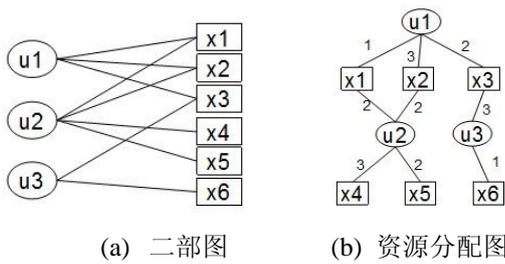


图 4 示例图

图 4(a)为用户-物品二部图, 图 4(b)为算法计算的过程图, 其中边上数字为评分权重, 计算如下:

$$\begin{cases} \rho_{i,n_i,j} = 0, \rho_{i,n_i,j} = 1 \\ \rho_{j,n_j,i} = 0, \rho_{j,n_j,i} = 1 \end{cases} \quad (7)$$

$$\Delta P_{k,n}^i = \frac{f_k(b_{k,n}^i + 1) - f_k(b_{k,n}^i)}{(s_{k,n}^i)^2} \quad (8)$$

$$\begin{cases} b_{k,n}^i = 0 \\ \Delta P_{k,n}^i = \frac{f(1) - f_k(0)}{(s_{k,n}^i)^2} \end{cases} \quad (9)$$

因此为用户 u1 推荐的列表为(x4, x6, x5). 但为了简化计算, 把评分项作为路径衰减项代入计算:

$$i' = \arg \min_{i \in \{1, \dots, M\}} \Delta P_{k,n}^i \quad (10)$$

$$b_{k,n}^{i'} = b_{k,n}^{i'} + 1 \quad (11)$$

$$\Delta P_{k,n}^{i'} = \frac{f_k(b_{k,n}^{i'} + 1) - f_k(b_{k,n}^{i'})}{(s_{k,n}^{i'})^2} \quad (12)$$

因此简化后的算法对 u1 的推荐列表为(x4, x5, x6), 其中  $w_0$  表示一个单位的资源权值.

### 3.2 多图层混合加权策略

基于多图层混合策略推荐算法可以在图结构上给出不同策略的推荐, 如一种推荐策略就是根据用户统计学特征图层给用户同推荐同一年龄段的用户喜欢的物品; 如另一种策略就是根据物品-特征图层给用户推荐刚加入的新物品, 当然也可以结合不同策略给出更组合推荐.

用户-物品图层中, 为简化计算, 可以把上述计算近似分解成单条路径求解后再叠加,  $w_{ui}$  为用户 u 对项目 i 的单条路径资源分配权值,  $R_{ui}$  为推荐度量值:

$$w_{ui} = \prod_{j=1}^m a \cdot rat_j \prod_{i=1}^n \frac{D_{\max}}{D_{\max} + b \cdot d_i} \prod_{j=1}^{m-1} \frac{\Delta T_{\max}}{\Delta T_{\max} + c \cdot \Delta t_j} \quad (13)$$

$$R_{ui} = \sum w_{ui} \quad (14)$$

其中,  $rat_j$  为该条路径上第 j 条边的评分,  $D_{\max}$  为所有节点中最大的度,  $d_i$  为第 i 节点的度,  $\Delta T_{\max}$  和  $\Delta t$  分别所有行为时间差的最大值和当前两次行为的时间差值, n 为该条路径包含节点数(不包含根节点), m 为该条路径包含的边数, a, b, c 为权重参数.

一种混合策略是先分别在三个图层各自计算出用户 u 对项目 i 的喜好度  $R_{ui}^1$ ,  $R_{ui}^2$  和  $R_{ui}^3$  然后再加权计算,  $\alpha, \beta, \gamma$  分别表示不同策略推荐的加权系数:

$$R_{ui} = \alpha R_{ui}^1 + \beta R_{ui}^2 + \gamma R_{ui}^3 \quad (15)$$

另一种混合策略是多个图层压缩成一个大图层, 在计算过程中, 当某边属于不同的子图层时, 权重系数  $\lambda$  也取不同的值,  $R_{ui}$  表示为:

$$R_{ui} = w_{vi} = \frac{D_{\max}}{D_{\max} + b \cdot d_{vi}} \cdot \frac{\Delta T_{\max}}{\Delta T_{\max} + c \cdot \Delta t_{vi}} \lambda \sum_{e \in ID(vi)} w_e \quad (16)$$

式中  $\lambda$  为权重系数,  $E_1, E_2, E_3$  分为三个图层边的集合.

$$\lambda = \begin{cases} \lambda_1, e \in E_1 \\ \lambda_2, e \in E_2 \\ \lambda_3, e \in E_3 \end{cases} \quad (17)$$

## 4 实验与结果分析

### 4.1 实验设计

本实验主要包含两部分: 第一部分实验了时间上下文信息对 PIML 算法效率的影响, 采用的数据集是 Netflix, 该数据集是从 Netflix prize 推荐算法大赛上所用的大数据集上抽样得到的, 在不破坏其原有数据分布和结构的完整性的情况下按照均匀分布随机抽样, 得到 6000 个用户对 4000 部电影的近 100 万条评分记录, 包含较完整可靠的用户行为的时间信息. 第二部分实验了 PIML 算法在结合人口统计学和物品特征信息后不同策略的推荐效率, 采用的是标准 MovieLens 数据集, 该数据集包含 6040 个用户对 3952 部电影的 100 多万条评分记录、6040 个用户的人口统计学信息(包括性别、年龄、职业和邮编)和 3952 部电影的类型信息(共 18 中类型, 如动作片, 喜剧片等).

表 1 实验所用样本数据集

数据集	用户数量	电影数量	评分数量
Movielens	6040	3952	1,000,209
Netflix	6000	4000	987,015

本实验用 Java 语言编写, 将数据集随机拆分 1/8 为测试集, 余下为训练集, 然后在训练集上建立图模型, 并在测试集上进行预测. 为了保证评测指标不是过拟合的结果, 每个实验都需要进行 8 次, 使用相同随机数种子 seed, 保证每次用不同测试集, 8 次实验各指标的平均值作为最终评测指标. 在实验的设计上, 首先, 使用图的数据结构, 以每一个用户节点为根节点, 树形向下遍历, 每遍历一层就计算一次下层节点分配资源权重  $w$ ; 当遍历的深度达到设定的阈值  $h_{max}$  时, 停止本次遍历; 最后, 根据该层节点分配的资源作为度量值对该用户生成推荐列表, 并且设定这样的计算只在同一图层内进行, 然后多个图层各自计算后进行策略加权.

实验对比如下算法(括号内均为算法的英文缩写):

- 1) Random(Rand) 随机推荐算法;
- 2) MostPopular(MostP) 热门物品推荐算法;
- 3) PersonalRank(PerP) 文献中随机游走算法;
- 4) UserCF 文献中基于用户的协同过滤算法;
- 5) PIML1 忽略时间、评分和热门因素的 PIML 算法;
- 6) PIML2 考虑评分和热门因素而没有考虑时间的 PIML 算法;
- 7) PIML3 考虑了所有因素的 PIML 算法;
- 8) ContentItem(ContI) 基于物品内容信息利用相似度矩阵的算法;
- 9) PIML-Item(P-Item) 基于物品内容的 PIML 算法;
- 10) PIML-Demo(P-Demo) 基于人口统计学信息的 PIML 算法;
- 11) PIML-Rat(P-Rat) 基于用户评分的 PIML 算法;
- 12) PIML-All(P-All) 加权了所有策略的 PIML 算法.

#### 4.2 实验评测指标

本次实验最终所给出的是 Top-N 推荐列表, 采用的评测指标是准确率、召回率、覆盖率和流行度.

准确度描述了推荐系统正确推荐的物品占推荐列

表的总物品集合的比例, 定义为:

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (18)$$

式中,  $R(u)$  为用户  $u$  在训练集上的推荐列表,  $T(u)$  测试集上的行为列表.

召回率描述了推荐系统正确推荐的物品占测试集物品集合的比例, 定义为:

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (19)$$

覆盖率描述了推荐系统能够推荐出来的物品占总物品集合的比例, 定义为:

$$Coverage = \frac{|U_{u \in U} R(u)|}{|I|} \quad (20)$$

流行度描述了一个物品的热门程度, 定义为对一个项目产生过行为的用户总数.

#### 4.3 实验结果分析

##### 4.3.1 Netflix 数据集上加权时间和评分的实验

这里没有对比不同参数下各算法的性能, 实验结果在最优参数下获得, 实验设置 PIML 算法只在用户-物品图层进行, 因为 Netflix 数据集没有用户人口统计学信息, 设置树的最大深度  $h = 4$ , 推荐长度为 10.

图 5 显示了在 Netflix 数据集上考察了时间, 评分和热门因素对 PIML 算法推荐结果影响; 实验结果表明 PIML 算法的准确率和召回率明显高于 Random 算法、MostPopular 算法、PersonalRank 算法和 UserCF 算法; PIML 算法在考虑时间、评分和热门因素后, 其准确率、召回率有显著提高; 图 6 所示, 结果表明 PIML 算法覆盖率高于 PersonalRank 算法, 结合时间信息后, 覆盖率明显提高; 考虑了时间、评分、热门因素后的 PIML 算法推荐物品的流行度呈明显降低趋势.

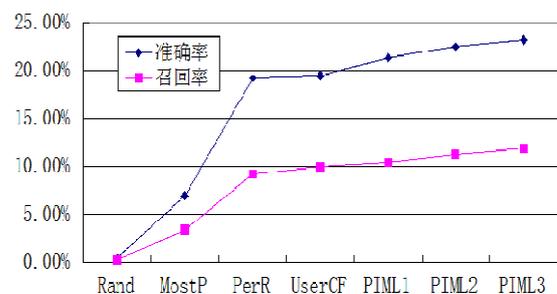
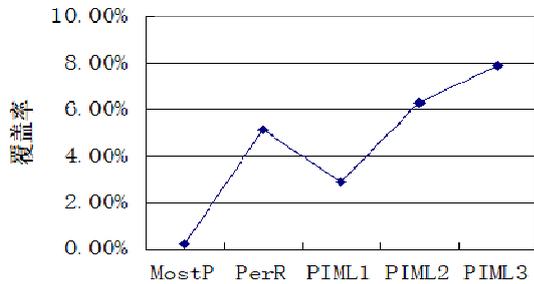
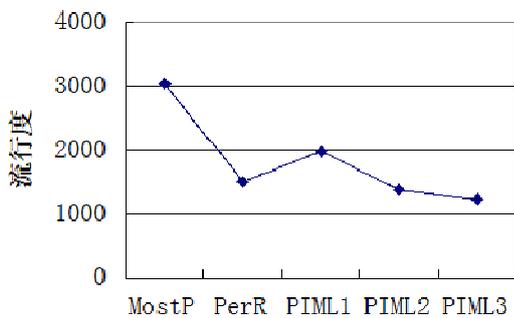


图 5 Netflix 上几种算法准确率和召回率对比

注: Rand, MostP, PerP 分别为算法 Random, MostPopuar, PersonalRank 的缩写.



(a) 覆盖率



(b) 流行度

图 6 Netflix 上几种算法覆盖率和流行度对比

### 4.3.2 MovieLens 数据集上多图层混合推荐的实验

第二组实验在 MovieLens 数据集上实现, 考察了 PIML 算法的不同推荐策略的效率, 并且对比了其他算法的效率. 实验没有对比不同参数下各算法的性能, 实验结果在最优参数下获得, 对应树的最大深度  $h = 4$ , 推荐长度为 10.

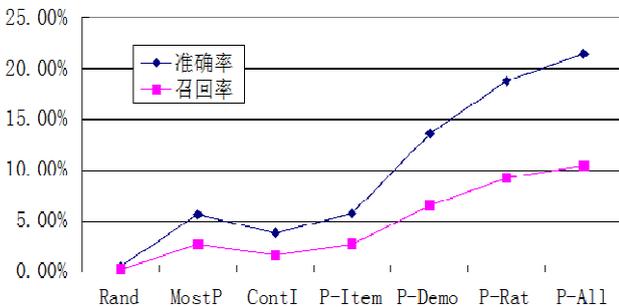


图 7 MovieLens 数据集上几种算法准确率和召回率对比

注: ContI, S-Demo, P-Rat, P-All 分别为算法 ContentItemKNN, PIML-Item, P-Demo, P-Rat, P-All 的缩写.

实验结果证明了: 同样基于物品内容信息的推荐, PIML 算法的推荐策略 PIML-Item 的效率明显高于 ContentItem 算法的推荐效率, 基于评分行为的推荐策略 PIML-Rat 的效率要明显高于基于人口统计学的推荐策略 PIML-Demo 的效率, 而基于人口统计学推荐策略 PIML-Demo 的效率又远高于基于物品内容的推荐策略 PIML-Item 的效率; 最后基于多策略综合的推荐效率要明显高于单一策略的推荐效率. 实验证明了 PIML 算法的优势以及推荐的灵活性.

## 5 结语

由于图推荐算法中资源分配矩阵的计算和存储的开销量大, 又忽略了时间信息对推荐效率的影响, 也没能结合用户人口统计学信息和物品内容信息, 因此难以做出更有效更全面的推荐. 本文针对二部图资源分配, 空间开销等问题, 提出了基于路径融合的多图层推荐算法, 由于舍弃了资源分配矩阵的计算形式, 有效地降低空间的开销, 把抽象的图模型看作实际中的电网, 充分利用电学理论, 来计算图的更接近理论上的资源分配, 仿真实验结果也表明, 这种结合了时间上下文信息的多图层推荐算法效率和灵活性更高, 更有利于动态的推荐; 在结合人口统计学和项目特征信息后, PIML 算法可以从不同的角度给出推荐策略, 也可以从多角度给出组合推荐. 接下来的工作将会考虑把图推荐结合协同过滤算法进一步增加 Top-N 推荐的覆盖率, 也将增强算法对大数据的处理能力, 在 MapReduce 分布式云计算的框架上实现.

## 参考文献

- 1 刘建国,周涛,汪秉宏.个性化推荐系统的研究进展.自然科学进展,2009,19(1):1-15.
- 2 Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowledge and Data Engineering, 2005, 17(6): 734-749.
- 3 高滢,齐红,刘杰,等.结合似然关系模型和用户等级的协同过滤推荐算法.计算机研究与发展, 2008,45(9):1463-1469.
- 4 Choi K, Suh Y. A new similarity function for selecting

- neighbors for each target item in collaborative filtering. Knowledge-Based Systems, 2012, 37(1): 146–153.
- 5 Chen JL, Rowan N, Les N, et al. Short and tweet: Experiments on recommending content from information streams. Proc. of the SIGCHI Conference on Human Factors in Computing Systems. New York, ACM Press. 2010. 1185–1194.
- 6 Li J, Xu Y, Wang YF, et al. Strongest association rules mining for efficient applications. Proc. of the Fourth IEEE Conference on Service Systems and Service Management. Piscataway, NJ, IEEE Press. 2007. 502–507.
- 7 李杰,徐勇,王云峰,等.面向个性化推荐的强关联规则挖掘. 系统工程理论与实践,2009,29(8):144–152.
- 8 张少中,陈德人.面向个性化推荐的两层混合图模型. 软件学报,2009,20(zk):123–130.
- 9 Li X, Su X, Wang MY. Social network-based recommendation: a graph random walk kernel approach. Proc. of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries. New York, ACM Press. 2012. 409–410.
- 10 Zhou T, Ren J, Medo M. Bipartite network projection and personal recommendation. Physical Review E, 2007, 76(4): 6116–6123.
- 11 Zhou T, Zoltan K, Liu J. Solving the apparent diversity-accuracy dilemma of recommender systems. Process of the National Academy of Sciences, 2010, 107(10): 4511–4517.
- 12 Fancois F, Alain P, Marco S, et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. Knowledge and Data Engineering, IEEE Trans., 2007, 19(3): 355–369.
- 13 Shang S, Kulkarni SR, et al. Random walk based model incorporating social information for recommendation. IEEE International Workshop on Machine Learning for Signal Processing. 2012. 23–26.