

基于评分预测与排序预测的协同过滤推荐算法

李改¹, 陈强³, 李磊²

(1. 顺德职业技术学院电子与信息工程学院, 广东顺德 528333; 2. 中山大学数据科学与计算机学院, 广东广州 510006; 3. 广东第二师范学院计算机科学系, 广东广州 510303)

摘要: 协同过滤推荐算法在电子商务领域运用广泛. 之前的研究要么仅从评分预测的角度来研究, 要么仅从排序预测的角度来研究. 为了兼顾这两个方面, 本文在传统的基于评分预测的 PMF (Probabilistic Matrix Factorization) 算法和基于排序预测的 xCLiMF (Extended Collaborative Less-is-More Filtering) 算法的基础上提出了一种基于评分预测与排序预测的协同过滤推荐算法 URA (Unified Recommendation Algorithm), 该方法通过在 PMF 和 xCLiMF 算法中共享用户和推荐对象的特征空间, 利用 PMF 算法来学习高精度的用户和推荐对象的特征向量, 从而进一步增强排序推荐性能. 实验验证, 该方法在评价指标 NDCG 和 ERR 下均优于 PMF 和 xCLiMF 算法, 且复杂度与评分点个数线性相关. URA 算法可运用于互联网信息推荐领域的大数据处理.

关键词: 推荐系统; 协同排序; 协同过滤; 评分预测; 排序预测

中图分类号: TP302 **文献标识码:** A **文章编号:** 0372-2112 (2017)12-3070-06

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2017.12.033

Collaborative Filtering Recommendation Algorithm Based on Rating Prediction and Ranking Prediction

LI Gai¹, CHEN Qiang³, LI Lei²

(1. Department of Electronic and Information Engineering, Shunde Polytechnic, Foshan, Guangdong 528300, China;

2. School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, Guangdong 510006, China;

3. Department of Computer Science, Guangdong University of Education, Guangzhou, Guangdong 510303, China)

Abstract: Collaborative filtering (CF) recommendation algorithm is widely used in the field of e-commerce. The previous researches on CF focused on either rating prediction or ranking prediction. In order to take into account these two aspects, a collaborative filtering recommendation algorithm based on rating prediction and ranking prediction (Unified Recommendation Algorithm, URA) is proposed. URA shares common latent features of users and items in PMF (Probabilistic Matrix Factorization, rating-oriented) and xCLiMF (Extended Collaborative Less-is-More Filtering, ranking-oriented) algorithms, and PMF learns improved latent features of users and items in URA, so that URA improves the performance of ranking recommendation. Experimental results showed that our proposed URA Algorithm outperformed PMF and xCLiMF algorithms over evaluation metrics NDCG and ERR, and that the complexity of URA is shown to be linear with the number of observed ratings. URA is suitable for big data processing in the field of internet information recommendation.

Key words: recommended systems; collaborative ranking; collaborative filtering; rating prediction; ranking prediction

1 引言

随着互联网以及电子商务的迅猛发展, 互联网上的信息量爆炸性增长, 如何准确而快速地从浩如烟海的数据海洋中找到用户所需的信息, 成为了当前工业界和学术界的一大挑战. 在此背景下, 信息推荐系统应

运而生. 协同过滤算法是信息推荐系统中运用最广泛且最成功的核心推荐技术. 当前使用最广泛的是基于模型的协同过滤算法^[1,2]. 基于模型的协同过滤算法按照机器学习方法的不同, 主要分为两种: 一种是基于评分预测的协同过滤推荐算法^[3-6], 另一种是基于排序预测的协同过滤推荐算法^[7-16]. 基于评分预测的协同过

收稿日期: 2016-09-13; 修回日期: 2017-02-28; 责任编辑: 覃怀银

基金项目: 国家自然科学基金 (No. 61370186, No. 61640222); 广东省自然科学基金项目 (No. 2016A030310018); 广东省科技计划项目 (No. 2014A010103040, No. 2014B010116001); 广州市科技计划项目 (No. 201604010049, No. 201510010203); 广东第二师范学院教授博士科研专项 (No. 2015ARF25)

滤推荐算法首先预测用户对所有推荐对象的评分,再通过这些评分值的大小来对推荐对象进行排序.基于排序预测的协同过滤推荐算法通过某种优化算法来直接给出推荐对象的偏序关系,而不需要预测用户对推荐对象的评分值作为中介.

之前的研究工作要么仅从评分预测的角度来研究^[3-6],要么仅从排序预测的角度来研究^[11-16].而基于评分预测的协同过滤推荐算法和基于排序预测的协同过滤推荐算法均存在缺陷^[17].为了克服现有研究的缺陷,本文在传统的 PMF 算法(基于评分预测)^[3]和 xCLiMF 算法(基于排序预测)^[14]的基础上提出了一种新的基于评分预测与排序预测的协同过滤推荐算法 URA;URA 算法通过在 PMF 和 xCLiMF 算法中共享用户和推荐对象的特征空间,利用 PMF 算法来学习高精度的用户和推荐对象的特征向量,从而进一步增强 xCLiMF 算法的排序推荐性能.文中给出了 URA 算法的一种有效学习算法.在真实的数据集上实验验证,该算法在各个评价指标下性能均优于几个经典的协同过滤推荐算法(基于评分预测和基于排序预测),且复杂度低、运算时间与评分点个数线性相关.

2 基本定义

本文中,矩阵 \mathbf{R} 表示评分矩阵(1~5 分),该矩阵具有 M 个用户、 N 个对象; $\mathbf{U} \in \mathbb{C}^{K \times M}$, $\mathbf{V} \in \mathbb{C}^{K \times N}$, \mathbf{U} 和 \mathbf{V} 分别表示用户和推荐对象的特征矩阵, K 表示特征个数,一般 $K \ll r$, r 表示矩阵 \mathbf{R} 的秩, $r \leq \min(M, N)$.

3 基于评分预测与排序预测的协同过滤推荐算法

3.1 基于评分预测与排序预测的协同过滤推荐算法简介

由文献[3]、[14]知道 PMF 算法和 xCLiMF 算法采用不同的机器学习方法来学习用户和推荐对象的潜在特征向量. PMF 算法通过矩阵分解模型来学习用户对推荐对象的评分模型,进而产生推荐序列. xCLiMF 算法是通过矩阵分解模型来直接优化评价指标 ERR,进而产生推荐序列.基于评分预测与排序预测的协同过滤推荐算法(URA 算法)的核心思想是:通过整合 PMF 算法和 xCLiMF 算法的优点,利用 PMF 算法挖掘隐含在评分数据中的信息以进一步提高 xCLiMF 算法在学习用户和推荐对象的特征向量时的精度,从而进一步提高协同排序算法的推荐性能.也就是说在 URA 算法中引入 PMF 算法的目的是为了利用 PMF 算法来为 xCLiMF 算法学习高精度的用户和推荐对象的特征向量,从而进一步从增强 xCLiMF 算法性能的角度来增强基于排序预测的协同过滤算法的推荐性能.最新研究成果表

明基于排序预测的协同过滤推荐算法的性能要优于基于评分预测的协同过滤推荐算法.故本文所提出的 URA 算法仍采用和 xCLiMF 算法一样的优化学习算法. URA 算法的输入输出流程图如图 1 所示^[17].

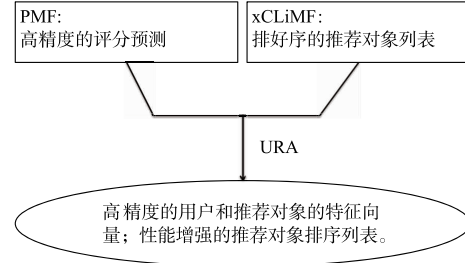


图1 URA算法的输入输出流程图

由于 PMF 算法和 xCLiMF 算法均是基于矩阵分解的算法,本文假设两种算法共用相同的用户和推荐对象的特征空间,同时引入折中参数 α 以控制 PMF 算法和 xCLiMF 算法对整个 URA 算法性能的相对贡献度.整合 PMF 算法和 xCLiMF 算法的目标优化函数所得到 URA 算法的目标优化函数如下:

$$\begin{aligned}
 L(\mathbf{U}, \mathbf{V}) = & (1 - \alpha) \times \left(- \sum_u^M \sum_{i=1}^N r_{ui} \left[\ln g(\mathbf{U}_u^T \mathbf{V}_i) \right. \right. \\
 & \left. \left. + \sum_{j=1}^N \ln(1 - r_{uj} g(\mathbf{U}_u^T \mathbf{V}_j - \mathbf{U}_u^T \mathbf{V}_i)) \right] \right) \\
 & + \alpha \times \frac{1}{2} \sum_u^M \sum_{i=1}^N (\mathbf{I}_{ui} (\mathbf{R}_{ui} - g(\mathbf{U}_u^T \mathbf{V}_i))^2) \\
 & + \frac{\lambda}{2} (\|\mathbf{U}\|^2 + \|\mathbf{V}\|^2) \quad (1)
 \end{aligned}$$

这里, r_{ui} 表示用户 u 对推荐对象 i 的满意概率,其中 \mathbf{I} 为一个指示矩阵, PMF 算法和 xCLiMF 算法的目标优化函数及其变量的定义详见文献[3, 14].正如上面所分析的,本文在 URA 算法中引入 PMF 算法的目的是为了进一步增强 xCLiMF 算法的推荐性能.因此折中参数 α 的值相对较小, $0 \leq \alpha \leq 1$, 当 $\alpha = 1$ 时 URA 算法退化为 PMF 算法, 当 $\alpha = 0$ 时 URA 算法退化为 xCLiMF 算法. 本文将在实验部分的 4.4.1 节分析折中参数 α 的值对 URA 算法的性能影响.最小化式(1)以优化 URA 算法,进而返回特征矩阵 \mathbf{U} 、 \mathbf{V} . 返回的特征矩阵 \mathbf{U} 、 \mathbf{V} 不仅能够用于给出性能优越的推荐对象序列,而且能够较准确的预测用户对推荐对象的评分值,从而增强了推荐的实用性.

3.2 算法优化求解与形式化描述

以特征矩阵 \mathbf{U} 、 \mathbf{V} 为参数的目标函数(1)是个非凸函数,因此本算法采用通过梯度下降法来迭代更新特征矩阵 \mathbf{U} 、 \mathbf{V} ,以找到目标函数(1)的局部最小值.对最终目标函数(1)使用梯度下降法分别对 \mathbf{U} 、 \mathbf{V} 求偏导得到如下公式:

$$\frac{\partial L}{\partial \mathbf{U}_u} = (1 - \alpha) \times \left(- \sum_{i=1}^N r_{ui} \left[g(-f_{ui}) \mathbf{V}_i \right. \right.$$

$$+ \sum_{j=1}^N \frac{r_{uj}g(f_{uj}-f_{ui})(V_i-V_j)}{1-r_{uj}g(f_{uj}-f_{ui})} \Big] \Big) \\ + \alpha \sum_{i=1}^N I_{ui}(g(U_u^T V_i) - R_{ui})g(U_u^T V_i) V_i + \lambda U_u \quad (2)$$

$$\frac{\partial L}{\partial V_i} = (1-\alpha) \times \left(-r_{ui} \left[g(-f_{ui}) + \sum_{j=1}^N r_{uj}g(f_{uj}-f_{uj}) \right. \right. \\ \left. \left. \cdot \left(\frac{1}{1-r_{uj}g(f_{uj}-f_{uj})} - \frac{1}{1-r_{ui}g(f_{ui}-f_{uj})} \right) \right] U_u \right) \\ + \alpha \sum_{u=1}^M I_{ui}(g(U_u^T V_i) - R_{ui})g(U_u^T V_i) U_u + \lambda V_i \quad (3)$$

其中 $g'(x)$ 表示 $g(x)$ 的导数, $f_{ui} = \langle U_u, V_i \rangle = (U_u)^T V_i$. URA 算法的形式化描述如算法 1 所示.

算法 1 基于评分预测与排序预测的协同过滤推荐算法 (URA 算法)

输入: 训练数据矩阵 R , 学习率 β , 正则化参数 λ , 折中参数 α , 特征数 K , 最大迭代轮数 $itermax$.
输出: 特征矩阵 U, V
for $u = 1, 2, \dots, M$ do
 % 索引用户 u 赋予过显式评分的所有推荐对象;
 $N(u) = \{i | R_{ui} > 0, 0 \leq i \leq N\}$;
end
初始化特征矩阵 $U^{(0)}, V^{(0)}$;
设置变量 $t = 0$; % 变量 t 用于记录算法迭代的轮数;
repeat
 for $u = 1, 2, \dots, M$ do
 % 利用式(2)更新 U_u ;
 $U_u^{(t+1)} = U_u^{(t)} + \beta \frac{\partial L}{\partial U_u^{(t)}}$;
 for $i \in N(u)$ do
 % 利用式(3)更新 V_i ;
 $V_i^{(t+1)} = V_i^{(t)} + \beta \frac{\partial L}{\partial V_i^{(t)}}$;
 end
 end
 $t = t + 1$;
until $t \geq itermax$;
 $U = U^{(t)}, V = V^{(t)}$;
返回特征矩阵 U, V .

3.3 算法计算复杂度分析

本文所提出的 URA 算法的计算复杂度主要与偏导项 $\frac{\partial L}{\partial U_u}$ 和 $\frac{\partial L}{\partial V_i}$ 的计算复杂度相关. 分析式(2)、(3), 得到偏导项 $\frac{\partial L}{\partial U_u}$ 和 $\frac{\partial L}{\partial V_i}$ 的计算复杂度分别为 $O(Kp^2M + KpM + KM)$ 和 $O(Kp^2M + KpM)$. 因此整个算法迭代一轮的时间复杂度是 $O(Kp^2M)$, 其中 p 表示用户给过评分值的推荐对象的个数的平均值, p 是一个比较小的数值, 满足 $p^2 \ll M$, 因此算法运行一轮的时间与用户的个数 M 成正比. 注意到数据集中评分点的总数 $s = pM$, 因此整

个算法迭代一轮的时间复杂度是 $O(Kps)$. 假定算法在迭代 d 轮后收敛, 则整个算法的复杂度为 $O(dKps)$. d, K, p 均是较小的常数, 从上述分析可知算法复杂度与 s 成正比, 而在实际的应用中, 数据集往往非常稀疏, 也就是 s 很小, 因此该算法非常适合处理大数据.

4 实验结果及分析

4.1 实验数据集

在本实验中, 一共使用了 2 个数据集.

Epinions 数据集^[11]: 这个数据集总共包含了 22,166 个用户对接近 296,277 件商品的 922,267 个评分, 其中评分数据是用 1 到 5 分来表示的, 分数越高表示用户对该商品越喜欢. 这个数据集的稀疏度是 99.98%.

Netflix 比赛数据集^[5]: 这个数据集包含了 480,189 个用户对 17,770 部电影 100,000,000 条评分记录. 该数据集的稀疏度达到了 99.88%. 为了实验方便, 随机挑选了 10,000 个用户对其中 10,000 部电影的评分数据作为实验数据集, 其中每个用户至少对 100 部电影有过评分.

4.2 实验的评价标准

文献报道显示 NDCG 值^[16] 和 ERR 值^[14] 非常适合评价处理显式评分数据的协同排序推荐算法的性能, 本文同样采用这两个性能评价标准来作为实验评价指标. 两个评价指标的详细说明见相应参考文献, 下面给出计算公式:

$$NDCG_u = \frac{DCG_u}{IDCG_u} \quad (4)$$

IDCG_{*u*} 表示理想状态下, 即所有推荐项目均按用户 u 的喜欢程度排序时的 DCG_{*u*} 值. NDCG 就是所有用户 NDCG_{*u*} 值的平均值.

计算对用户 u 所产生的推荐序列的 ERR_{*u*} 值的公式如下:

$$ERR_u = \sum_{i=1}^n \frac{r_{ui}}{R_{ui}} \prod_{j=1}^n (1 - r_{uj} H(R_{uj} < R_{ui})) \quad (5)$$

ERR 就是所有用户 ERR_{*u*} 值的平均值. 其中 $H(x)$ 是一个指示函数:

$$H(x) = \begin{cases} 1, & \text{if } x \text{ 为真;} \\ 0, & \text{else} \end{cases} \quad (6)$$

NDCG@5 和 ERR@5 就是给所有用户返回 5 个推荐对象时的 NDCG 值和 ERR 值.

对每个实验均反复运行 10 次, 构成新的训练集和测试集, 最终结果取 10 次运算结果的平均值.

4.3 实验设置

对不同数据集, 本文将分别计算在给出不同数量用户打分数据的情况下, 各个算法的表现. 例如条件 "Given 10" 表示对于每个用户随机挑选 10 个他们打过

分的推荐对象作为训练集,剩下没选中的数据作为测试集.对于 Epinions 数据集和 Netflix 数据集均给出:“Given 10”,“Given 20”和“Given 30”这三个条件.

实验中随机挑选用户打过的 5 个推荐对象和用户没打分的 1000 个推荐对象合在一起作为用户的测试集.对于每个数据集中给定每个条件,分别对每个算法在各个评价标准下进行了 10 次实验,实验的最终结果是这 10 次实验的平均得分.对于矩阵分解的方法,设置的特征个数 K 为 10.通过交叉确认实验,URA 算法的学习率 β 设置为 0.001、正则化参数 λ 设置为 0.001 时的性能最好.其他算法的参数均参照相应参考文献中的设置.

4.4 实验结果

各个算法的程序代码用 Java 开发工具 Eclipse 编写.所有算法均在装有 Linux 系统环境的计算机下运行,计算机配置环境是内存是 7.6GB,4 核 CPU,每个核主频是 3.2GHz.

4.4.1 折中参数 α 对 URA 算法性能的影响

图 2 给出了折中参数 α 对 URA 算法性能的影响,其中横轴表示折中参数 α 的值,纵轴表示评价指标 ERR@5 的值.图 2(a) 表示在 Epinions 数据集的子数据集“Given 30”下的实验结果,图 2(b) 表示在 Netflix 数据集的子数据集“Given 30”下的实验结果.从图 2 可以看出,在两个数据集下,URA 算法的性能随着折中参数 α 值的增加先快速提升后逐渐下降.在 Epinions 数据集的子数据集“Given 30”下,当折中参数 $\alpha = 0.15$ 时 URA 算法的性能最佳.在 Netflix 数据集的子数据集“Given 30”下,当折中参数 $\alpha = 0.1$ 时 URA 算法的性能最佳.分析 3.1 节 URA 算法的目标优化函数可知, $0 \leq \alpha \leq 1$,当 $\alpha = 1$ 时 URA 算法退化为 PMF 算法,当 $\alpha = 0$ 时 URA 算法退化为 xCLiMF 算法.本节实验结果表明当折中参数 α 的值较小时,URA 算法的性能最佳.实验结果与本文设计 URA 算法的初衷一致:在 URA 算法中,引入 PMF 算法的目的是为了利用 PMF 算法来学习高精度的用户和推荐对象的特征向量来进一步增强 xCLiMF 算法的排序推荐性能,PMF 算法在 URA 算法中起辅助作用;因此折中参数 α 的值相对较小.

实验发现结合 PMF 和 xCLiMF 算法的折中参数 α 的值会随数据集的变化而变化,但当折中参数 α 取值较小时($0 < \alpha < 0.35$),URA 算法在各个数据集的条件子数据集上的性能均优于 xCLiMF 算法,这也进一步说明了 URA 算法的实用性.接下来的实验中,对于 URA 算法,在 Epinions 数据集的各个条件子数据集上设置折中参数 $\alpha = 0.15$,在 Netflix 数据集的各个条件子数据集上设置折中参数 $\alpha = 0.1$.因为实际应用的信息推荐系统中的数据集规模是随时间变化而不断变化的,因此

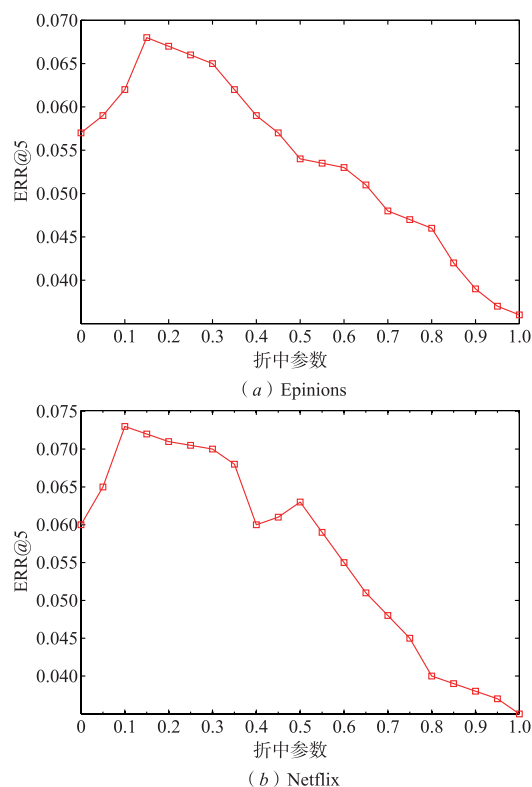


图2 折中参数 α 对URA算法性能的影响

研究固定折中参数值下的 URA 算法的性能更具实际应用价值.

4.4.2 URA 算法的可扩展性分析

图 3 给出了在 URA 算法的可扩展性分析.其中纵轴表示算法的运行时间,单位为分钟.横轴表示在训练 URA 算法时,训练集中用户占全部用户数的比例.分别给出了在 Epinions 数据集的 3 个条件子数据集:“Given 10”,“Given 20”和“Given 30”上的实验结果.

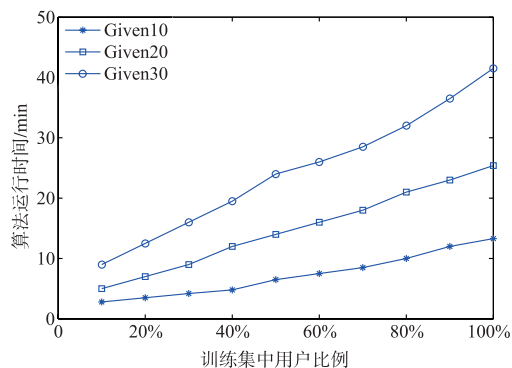


图3 URA算法中用户比例变化对算法运行时间的影响

从图 3 可以看出,随着训练集中用户占全部用户数的比例值的线性增加,在各个子数据集下 URA 算法的训练时间也线性增加.也即当固定用户的平均评分点

的个数时,每一轮迭代时间就和用户数目接近线性相关,因此本文所提出的 URA 算法的时间复杂度与数据集中可观测到的评分点的个数成线性关系.这也进一步证明了本文所提 URA 算法的可扩展性.

4.4.3 URA 算法和几个经典的协同过滤推荐算法的性能比较

本节将在 2 个数据集上把本文所提出的 URA 算法与如下 4 个经典的协同过滤推荐算法作比较:

ItemCF^[4]:这是一个传统的广泛使用的基于推荐对象之间相似度的协同过滤推荐算法.

PMF^[3]:这是一种最新的基于评分预测的协同过滤推荐算法.该算法等价于折中参数 $\alpha = 1$ 时的 URA 算法.在本实验中,PMF 算法采用和 URA 算法一样的特征维数和规范化参数 λ .

CofiRank^[12]:这是一种最新的基于排序预测的协同过滤推荐算法,实验中采用公用的软件包来实现该算法.

xCLiMF^[14]:这也是一种最新的基于排序预测的协同过滤推荐算法,该算法等价于折中参数 $\alpha = 0$ 时的 URA 算法.在本实验中,xCLiMF 算法采用和 URA 算法一样的特征维数和规范化参数 λ .

图 4~7 给出了 URA 算法与其他几个经典的协同过滤推荐算法的性能比较,图中的纵轴表示相应的评价指标的值,横轴表示 Epinions 数据集或 Netflix 数据集的条件子集.图 4 和图 5 的实验结果使用的数据集是 Epinions 数据集.图 6 和图 7 的实验结果使用的数据集是 Netflix 数据集.从图 4~7 可以看出,在各个数据集下以评价指标 NDCG@5 和 ERR@5 作为性能评价标准,URA 算法的性能均远优于其他几个经典的协同过滤推荐算法,并且这种优势随着训练集中用户评分数据量的增加越发明显;本文把 URA 算法性能的显著提高归功于:在 URA 算法中利用 PMF 算法来学习高精度的用户和推荐对象的特征向量从而进一步增强了原始 xCLiMF 算法的排序推荐性能.此外,xCLiMF 算法的性

能优于 CofiRank 算法,这与参考文献[14]中报道的实验结果一致.CofiRank 算法的性能优于 PMF 算法,这也与参考文献[12]中报道的实验结果一致,这说明:对于排序推荐问题,优化排序目标函数的 CofiRank 算法优于优化评分误差函数的 PMF 算法.xCLiMF 算法和 CofiRank 算法的性能优于 PMF 算法,这也与参考文献[12]、[14]中报道的实验结果一致.

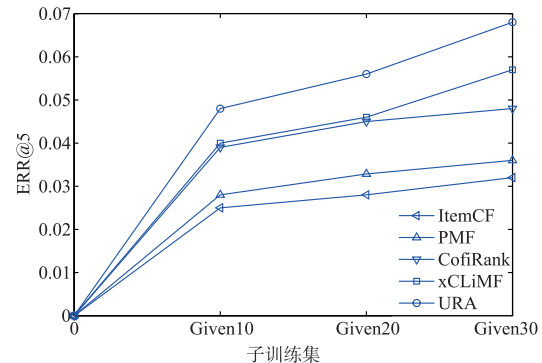


图5 URA算法与其它经典协同过滤推荐算法的性能比较 (Epinions)

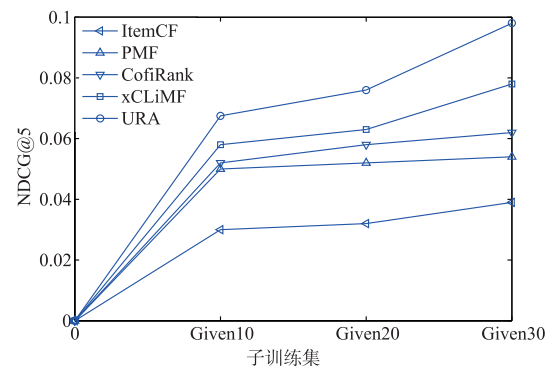


图6 URA算法与其它经典协同过滤推荐算法的性能比较 (Netflix)

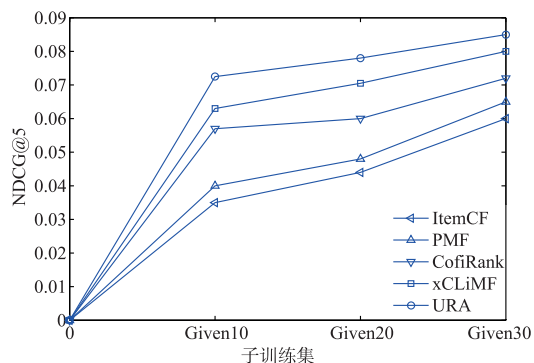


图4 URA算法与其它经典协同过滤推荐算法的性能比较 (Epinions)

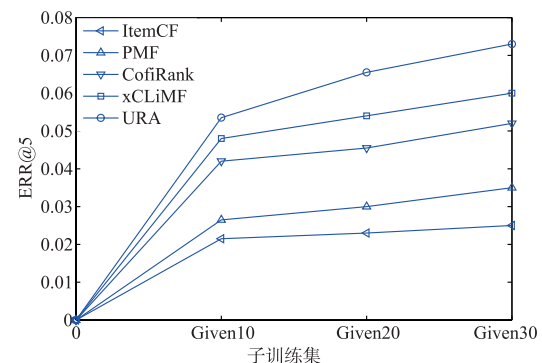


图7 URA算法与其它经典协同过滤推荐算法的性能比较 (Netflix)

5 结论

本文在传统的 PMF 算法和 xCLiMF 算法的基础上提出了一种新的 URA 算法. 在真实的数据集上实验验证, URA 算法推荐精度高、复杂度与评分点个数线性相关, 适合处理大数据, 在互联网信息推荐领域具有很高的工程应用价值, 可以应用于云端大数据分析. 在以后的工作中还将考虑 URA 算法在实际应用的大数据环境下的并行化问题、实时推荐问题, 以及 URA 算法和社交网络信息、文本描述信息的融合问题.

参考文献

- [1] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2005, 17(6): 734 – 749.
- [2] Ricci F, Rokach L, Shapira B, et al. *Recommender System Handbook*[M]. New York, USA: Springer, 2011.
- [3] Salakhutdinov R, Mnih A. Probabilistic matrix factorization [A]. *Proceedings of the 21st Annual Conference on Neural Information Processing Systems*[C]. Vancouver B C, Canada: ACM, 2007. 252 – 260.
- [4] Deshpande M, Karypis G. Item-based top-n recommendation algorithms[J]. *ACM Transactions on Information Systems*, 2003, 22(1): 143 – 177.
- [5] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model [A]. *Proceedings of the 25th International Conference on Knowledge Discovery and Data Mining*[C]. Las Vegas: ACM, 2008. 426 – 434.
- [6] Zhu X W, Ming Z Y, Hao Y, et al. Tackling data sparseness in recommendation using social media based topic hierarchy modeling [A]. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*[C]. Buenos Aires, Argentina: ACM, 2015. 2415 – 2421.
- [7] Li G, Ou W H. Pairwise probabilistic matrix factorization for implicit feedback collaborative filtering [J]. *Neurocomputing*, 2016, 204: 17 – 25.
- [8] Li G, Wang L Y, Ou W H. Robust personalized ranking from implicit feedback [J]. *International Journal of Pattern Recognition and Artificial Intelligence*, 2015, 30(1): 1 – 28.
- [9] Li G, Chen Q. Exploiting explicit and implicit feedbacks for personalized ranking [J]. *Mathematical Problems in Engineering*, 2016, 2016: 1 – 11.
- [10] Liu T Y. *Learning to Rank for Information Retrieval*[M]. Berlin: Springer, 2011.
- [11] Yao W L, He J, Huang G Y, et al. SoRank: Incorporating social information into learning to rank models for recommendation [A]. *Proceedings of the 23th ACM International Conference on World Wide Web*[C]. Seoul, Korea: ACM, 2014. 409 – 410.
- [12] Weimer M, Karatzoglou A, Le Q V, et al. CofiRank-maximum margin matrix factorization for collaborative ranking [A]. *Proceedings of the 21th Conference on Advances in Neural Information Processing Systems*[C]. Vancouver B C, Canada: Curran Associates, Inc, 2007. 79 – 86.
- [13] Liu N N, Yang Q. Eigenrank: a ranking-oriented approach to collaborative filtering [A]. *Proceedings of the 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*[C]. Singapore: ACM Press, 2008. 83 – 90.
- [14] Shi Y, Karatzoglou A, Baltrunas L, et al. xCLiMF: Optimizing expected reciprocal rank for data with multiple levels of relevance [A]. *Proceedings of the Sixth ACM Conference on Recommender Systems*[C]. Hongkong: ACM Press, 2013. 431 – 433.
- [15] Liu N N, Zhao M, Yang Q. Probabilistic latent preference analysis for collaborative filtering [A]. *Proceedings of ACM International Conference on Information and Knowledge Management*[C]. Hong Kong, China: ACM, 2009. 759 – 766.
- [16] Shi Y, Larson M, Hanjalic A. List-wise learning to rank with matrix factorization for collaborative filtering [A]. *Proceedings of the Fourth ACM Conference on Recommender Systems*[C]. New York, USA: ACM, 2010. 269 – 272.
- [17] Shi Y, Larson M, Hanjalic A. Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation [J]. *Information Sciences*, 2013, 229(6): 29 – 39.

作者简介



李 改 男, 1981 年生于湖北松滋. 现为顺德职业技术学院电子与信息工程学院副教授. 主要研究方向为数据挖掘, 推荐系统, 机器学习.
E-mail: ligai999@126.com



陈 强 男, 1963 年生于广东化州. 现为广东第二师范学院计算机系教授, 主要研究领域为数据挖掘, 知识表示.
E-mail: 21115903@qq.com