

基于加权二部图的 Slope One 推荐算法

王 冉¹, 徐 怡^{1,2}, 胡善忠¹, 何明慧¹

(1 安徽大学 计算机科学与技术学院, 安徽 合肥 230601;

2 计算智能与信号处理教育部重点实验室(安徽大学), 安徽 合肥 230039)

摘 要: 相对协同过滤算法, Slope One 算法在执行速度上更加迅速, 并且易于实现. 但是算法没有考虑项目之间的推荐关系, 故提出了一种基于加权二部图的 Slope One 推荐算法. 利用加权二部图推荐算法计算项目之间的推荐程度, 并用计算得到的项目之间的推荐程度对 Slope One 预测评分过程进行加权处理, 由于二部图推荐算法计算得到的项目之间的推荐程度是非对称的, 从而使推荐结果更加多样化. 在 MovieLens 数据集上利用 5-折交叉验证以及 4 种评价指标对算法进行验证, 实验表明改进后的算法在提高了推荐准确性的同时也提高了推荐的多样性.

关键词: 推荐系统; Slope One; 加权二部图; 5-折交叉验证

中图分类号: TP319

文献标识码: A

文章编号: 1000-7180(2018)03-0093-06

Slope One Recommendation Algorithm Based on Weighted Bipartite Graph

WANG Ran¹, XU Yi^{1,2}, HU Shan-zhong¹, HE Ming-hui¹

(1 Department of Computer Science and Technology, Anhui University, Hefei 230601, China;

2 Key Lab of IC&SP, Ministry of Education(Anhui University), Hefei 230039, China)

Abstract: Compared with the collaborative filtering algorithm, the Slope One algorithm is faster and easy to implement, however, the algorithm does not consider the recommended level between items. Therefore, this paper proposes a Slope One recommendation algorithm based on weighted bipartite graph. The weighted bipartite graph algorithm calculates the recommended level between the items and uses the recommended level to weight the prediction of the Slope One. Since the recommended level between the items calculated by the bipartite graph recommendation algorithm is asymmetric, the results are more diversified. We test the proposed method by 5-fold cross-validation on the MovieLens with four metrics: ranking score, Recall, Precision and diversification, and the experimental results show that the improved algorithm improves the recommendation accuracy and improves the recommendation diversity.

Key words: recommend system; Slope One; weighted bipartite graph; 5-fold cross-validation

1 引言

随着信息时代的飞速发展,数据指数型的增长带来的最主要的问题就是:信息过载,用户从海量的信息中迅速找到自己需要的内容是十分困难的行为.为

了解决这类问题,推荐系统由此诞生,不同于搜索引擎,推荐系统不需要用户有意识地输入关键字信息,而是利用用户的历史行为数据发现用户兴趣,从而主动给用户推荐感兴趣的内容.目前很多电子商务网站,都在推荐系统中做出了各种尝试^[1-2].目前主要的

收稿日期: 2017-06-01; 修回日期: 2017-07-10

基金项目: 国家自然科学基金项目(61402005);安徽省自然科学基金项目(1308085QF114);安徽省高等学校省级自然科学基金项目(KJ2013A015, KJ2011Z020);安徽大学计算智能与信号处理教育部重点实验室课题项目;国家留学基金委员会资助项目

推荐算法有基于协同过滤的推荐算法、基于内容的推荐算法、基于图的推荐算法以及混合推荐算法等^[3], 推荐算法的好坏直接影响推荐系统的性能. Slope One 算法可以看成线性回归模型 $y=x+b$, 利用所有用户的评分数据进行预测的算法, 由于不考虑用户或项目之间的相似关系, 所以运行速度快, 可以用来进行实时推荐, 具有简洁、高效以及易于实现等特点; 不过也正是因为没有考虑用户之间以及项目之间的相似性, 推荐效果有待提高^[4]. 因此, 学者们提出了很多改进算法. 文献[5]提出了一种基于图书推荐的阈值优化 Slope One 推荐算法, 通过计算相似度选取邻居来对 Slope One 进行加权, 同时加入了 KNN 算法选择最近邻居, 从而提高推荐的准确率. 文献[6]提出了一种混合的 Slope One 推荐算法, 将用户相似度、汉明距离、聚类 and Slope One 算法进行融合, 从而提高评分预测的效果. 文献[7]提出了一种改进的 Slope One 协同过滤算法, 结合了项目语义相似度算法和 Slope One 算法, 并将各自算法计算得到的预测评分进行加权处理, 从而得到最终预测评分, 有效提高了推荐精度.

以上方法虽然在一定程度上提高了推荐精度, 但是没有考虑项目之间的推荐关系, 并且这种关系是种不对称的关系, 即项目 i 对项目 j 的推荐程度和项目 j 对项目 i 的推荐程度是不同的, 而基于二部图的推荐算法的资源分配过程计算得到的项目之间的资源分配是不对称的, 同时可以体现项目之间的推荐关系. 故本文提出一种基于加权二部图的 Slope One 推荐算法. 在数据集 MovieLens 数据集上进行 5-折交叉验证, 通过平均分排名、准确率、召回率, 以及多样性 4 种指标进行检验, 实验表明本文提出的算法提高了推荐精度的同时也提高了推荐的多样性.

2 Slope One 算法

Slope One 算法思想很简单, 主要基于评分差异, 在预测评分时可以看成一个线性函数 $y=x+b$, 首先找出同时对项目 x 和项目 y 评分的所有用户, 然后利用这些用户对 b 进行线性拟合, 最后预测目标用户对 y 的评分时直接带入目标用户已有的评分 x 即可^[8]. 例如: 考虑以下示例(表 1), 预测目标用户 $target$ 对 $Item\ z$ 的评分, 根据 Slope One 算法, $Item\ x$ 和 $Item\ z$ 有两次共同评分, 对于 $User\ a$, 与 $Item\ x$ 相比, $Item\ z$ 打高了 2 分($4-2=2$), 对于 $User\ b$, 与 $Item\ x$ 相比, $Item\ z$ 打低了 3 分($1-4=-3$), 因此 $Item\ z$ 和 $Item\ x$ 的平均评分差是 $(2+(-3))/2=-0.5$, 由于目标用户 $target$ 对 $Item\ x$ 的评分是 3,

所以目标用户 $target$ 对 $Item\ z$ 的评分是 $3+(-0.5)=2.5$; 同理需要计算 $Item\ y$ 和 $Item\ z$ 的平均评分差, 因为只有 $User\ a$ 对 $Item\ y$ 和 $Item\ z$ 有共同评分, 故 $Item\ z$ 和 $Item\ y$ 的平均评分差为 $4-5=-1$, 由于目标用户 $target$ 对 $Item\ y$ 的评分为 5, 所以目标用户对 $Item\ z$ 的预测评分为 $5+(-1)=4$, 最后对两次预测评分求平均值得到 $target$ 对 $Item\ z$ 最终的预测评分 $(2.5+4)/2=3.25$.

表 1 预测 $target$ 对 $Item\ z$ 的评分

	$Item\ x$	$Item\ y$	$Item\ z$
$target$	3	5	?
$User\ a$	2	5	4
$User\ b$	4	—	1

算法模型如下所示.

首先建立一个评分集合 R , 包含训练集中所有用户对项目的评分, 定义 u_i 和 u_j 分别是评分集合 R 中用户 u 对项目 i 以及用户 u 对项目 j 的评分, $C_{ij}(R)$ 表示在评分集合 R 中, 同时对物品 i 和物品 j 评分的用户集合, 定义项目 j 和项目 i 的平均评分差值 dev_{ji} 的计算如式(1) 所示.

$$dev_{ji} = \sum_{u \in C_{ji}(R)} \frac{u_j - u_i}{|C_{ji}(R)|} \tag{1}$$

利用式(1) 得到的 dev_{ji} 来预测用户 u 对项目 j 的评分 $pred(u, j)$, 公式如下:

$$pred(u, j) = \frac{1}{|S(u, j)|} \sum_{i \in S(u, j)} (dev_{ji} + u_i) \tag{2}$$

式中, $S(u, j) = \{i | i \in C(u), i \neq j, |C_{ij}(R)| > 0\}$; $C(u)$ 表示用户 u 评分过的项目集合; $S(u, j)$ 表示的含义是用户 u 已评分的项目中与项目 j 的共同评分次数大于等于 1 的项目集合; u_i 表示用户 u 对项目 i 的评分; dev_{ji} 表示 j 和 i 项目的评分差.

3 基于加权二部图的 Slope One 推荐算法

Slope One 算法虽然简单且高效, 但是由于所有用户在预测评分时同等利用评分差, 没有考虑项目之间内在的推荐程度不同, 导致推荐的准确率和多样性不高, 故本文提出了一种基于加权二部图的 Slope One 推荐算法, 利用二部图资源扩散计算得到项目之间的推荐程度, 从而对预测评分差进行加权, 以提高推荐的准确率和多样性.

3.1 基于二部图的推荐算法

二部图将用户集合和项目集合分别看成两类抽象节点, 用户对项目的购买或评分等行为作为二部

图中的边. 二部图算法的主要思想是:假设每个类型的节点具有一定的初始资源,以项目节点为例,如果要计算项目节点 j 分配给项目 i 节点的资源,则首先从项目 j 节点开始经过与其相连的边将资源平均分配到用户节点,然后用户节点再将接收到的资源通过边再次平均分配给项目节点 i ,此时项目节点 i 的

值就是项目 j 分配的资源,用 res_{ij} 表示, res_{ij} 也可以理解为项目节点 j 对项目节点 i 的推荐程度或者影响程度. 由于连接项目节点 i 和项目节点 j 的用户节点数量可能不同,所以一般 res_{ij} 和 res_{ji} 是不相等的^[9-10]. 图1表示资源在项目之间分配的过程.

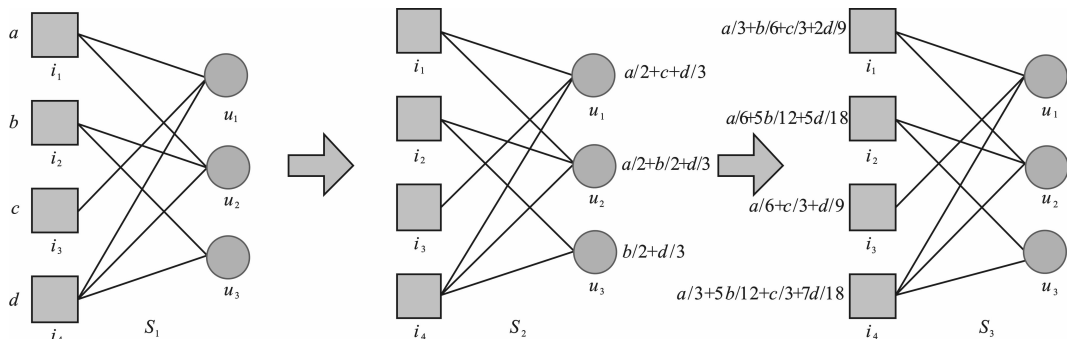


图1 二部图资源分配过程

图中 u_1, u_2, u_3, u_4 是用户节点, i_1, i_2, i_3, i_4 是项目节点, a, b, c, d 是项目节点的初始资源. 通常二部图的初始资源都为1, 这里为了使分配过程更加清晰, 故用 a, b, c, d 表示. 图1中 S_1 表示为项目节点分配初始资源, 二部图的资源分配一共分为两个步骤: 第一步为图1中 S_2 表示的过程, 从项目节点将资源通过边平均分配到用户节点; 第二步为图1中 S_3 表示的过程, 用户节点将接收到的资源再次通过边平均分配给项目节点.

一个二部图表示为 $G(U, I, E)$, 其中 $U = \{u_1, u_2, u_3, \dots, u_m\}$ 是用户集合, $I = \{i_1, i_2, i_3, \dots, i_n\}$ 是项目集合, E 是二部图中的边构成的集合. 对于任意两个项目节点 i 和 j , 从项目节点 j 分配给项目节点 i 的资源 res_{ij} 的计算公式如下:

$$res_{ij} = \frac{1}{deg(j)} \sum_{u \in U} \frac{r_{ui} r_{uj}}{deg(u)} \quad (3)$$

式中, U 是用户集合; $deg(j)$ 表示项目 j 的度; 也就

是连接项目 j 的边的个数; $deg(u)$ 表示用户 u 的度; 也就是连接用户 u 的边的个数. r_{ui} 和 r_{uj} 分别是 $m \times n$ 的邻接矩阵 M 的第 u 行第 i 列以及第 u 行第 j 列值, 其取值如下:

$$r_{ui} = \begin{cases} 1, & ui \in E \\ 0, & ui \notin E \end{cases} \quad (4)$$

式中, 当用户 u 节点存在连接项目 i 节点的边时, r_{ui} 为1; 反之, r_{uj} 为0.

3.2 基于加权二部图的推荐算法

由于二部图的边只有0和1两种情况, 没有考虑权值, 所以不论是用户节点还是项目节点资源都是平均分配的, 而对于加权的二部图, 其资源分配是根据某一节点的当前边的权值占该节点的所有边权值之和的比重, 按比例进行分配^[11], 具体过程如图2所示.

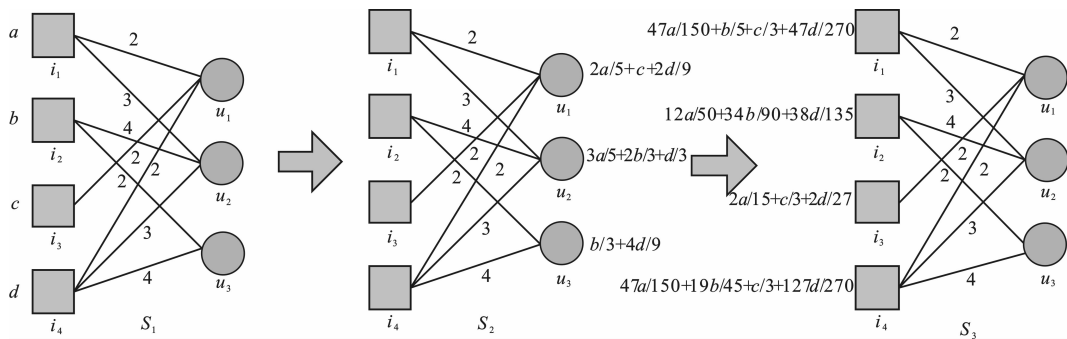


图2 加权二部图资源分配过程

在加权二部图中,对于任意两个项目节点 i 和 j ,从项目节点 j 分配给项目节点 i 的资源 res_{ij} 的计算公式如下:

$$res_{ij} = \sum_{u \in U} \frac{r_{ui} r_{uj}}{deg(j) deg(u)} \quad (5)$$

式中, U 是用户集合; $deg(j)$ 表示连接项目节点 j 的所有边的权值之和; $deg(u)$ 表示连接用户 u 的所有边的权值之和; r_{ui} 和 r_{uj} 分别是 $m \times n$ 的邻接矩阵 M 的第 u 行第 i 列以及第 u 行第 j 列值,其取值如下:

$$r_{ui} = \begin{cases} E_{ui}, & ui \in E \\ 0, & ui \notin E \end{cases} \quad (6)$$

式中, E_{ui} 表示用户 u 节点与项目 i 节点边的权值。

3.3 基于加权二部图 Slope One 推荐算法

由于传统的 Slope One 算法对所有用户在预测评分时同等地利用评分差,没有考虑不同项目之间的推荐程度不同. 比如当预测用户 u 对项目 i 的评分时,首先找出用户 u 已评分过的项目集合,分别通过式(1)计算得到的评分差来预测项目 i 的评分,最后取平均值. 然而,不同的项目之间的推荐程度不同. 例如,在购物行为中,当大多数用户在购买商品 a 的同时也购买商品 b ,那么商品 a 推荐商品 b 的程度也就越高,如果只有极少数用户在购买商品 a 的同时选择购买商品 c ,那么商品 a 对商品 c 的推荐程度也就越低.

基于上述分析,本文在 Slope One 算法预测评分时,利用式(5)计算加权二部图中项目之间的资源分配,从而得到项目之间的推荐度. 改进后的 Slope One 评分预测公式如下:

$$Newpred(u, j) = \frac{\sum_{i \in S(u, j)} (dev_{ji} + u_i) \cdot res_{ji}}{\left| \sum_{i \in S(u, j)} res_{ji} \right|} \quad (7)$$

式中, $S(u, j) = \{i \mid i \in C(u), i \neq j, |C_{ij}(R)| > 0\}$; $C(u)$ 为用户 u 已评分的项目集合; $S(u, j)$ 表示的含义是用户 u 已评分的项目中与项目 j 共同评分次数不小于 1 次的项目集合; dev_{ji} 表示 j 和 i 项目的评分差; res_{ji} 是通过加权二部图资源分配得到的项目 i 对项目 j 的推荐程度;注意,这里不能写成 res_{ij} ,因为需要预测的是 j 项目的评分,所以计算的是已有评分的项目 i 对待预测项目 j 的推荐程度,而 res_{ij} 表示的是项目 j 对项目 i 的推荐程度.

基于加权二部图 Slope One 推荐算法如下所示.

算法 1: 基于加权二部图 Slope One 推荐算法

输入 用户 - 项目评分矩阵 R

输出 目标用户 u 对目标项目 i 的预测评分 $NewPred(u, i)$

Step1. 通过评分矩阵 R 得到项目之间的共同评分用户集合 $comUsers$, 利用式(1) 计算 $comUsers$ 对目标用户已评分项目和待预测项目的评分差 dev_{ij} .

Step 2. 通过评分矩阵构建加权二部图, 利用式(5) 计算项目 j 分配给项目 i 的资源 res_{ij} , 也就是项目 j 对项目 i 的推荐程度.

Step 3. 将 Step 1 计算得到的 dev_{ij} 和 Step 2 计算得到的 res_{ij} 带入式(7), 得到最终的预测评分 $NewPred(u, i)$.

Step 4. 结束.

4 实验结果与分析

本文通过实验比较基于加权二部图 Slope One 推荐算法和传统的 Slope One 算法的推荐准确度和推荐多样性.

4.1 数据集

实验采用 GroupLens 小组收集和整理的 MovieLens 100k 数据集, 包含 943 名用户对 1 682 部电影的 100 000 多个评分, 并且每个用户至少有 20 个评分. 评分值大小在 1~5 之间, 值越大表示用户对电影的评价越高. 本文使用 5-折交叉验证的方法对提出的改进算法进行验证.

4.2 评价标准

本文采用的两种评价标准对实验结果进行对比, 分别是平均分排名 ($ARankS$), 准确率 ($Precision$), 召回率 ($Recall$) 以及多样性 ($Diversification$).

(1) 平均分排名 ($ARankS$): 分值排名 ($RankS$) 是用来描述推荐列表中的项目在未选择项目的位置. 比如用户 u 有 1 000 个未选择的项目, 测试集中的项目 i 出现在推荐列表的第 10 个位置, 则分值排名 $P_{ui} = 10/1\,000 = 0.01$. 显然, 当项目 i 位置越靠前, 则分值排名的值越低, 推荐效果越好; 反之, 分值排名值越高, 则推荐效果越差. 那么 u 用户测试集中所有项目 P_{ui} 的平均分值排名如下:

$$ARankS = \frac{\sum_{i \in T(u)} P_{ui}}{|T(u)|} \quad (8)$$

式中, $T(u)$ 表示 u 的测试集; P_{ui} 表示测试集中的项目 i 在排序后的推荐列表中的分值排名.

(2) 准确率 ($Precision$): 表示对于目标用户给定的推荐列表中, 在测试集上出现的物品个数占推荐列表大小的比例, 准确率依赖于推荐列表的长度, 且值越大推荐效果越好, 公式如下:

$$Precision = \frac{\sum_{u \in U} |L(u) \cap T(u)|}{\sum_{u \in U} |L(u)|} \quad (9)$$

(3) 召回率(*Recall*):表示对于目标用户给定的推荐列表中,在测试集上出现的物品个数占测试集大小的比例,召回率也依赖于推荐列表的长度,召回率的值越大推荐越精确,公式如下:

$$Recall = \frac{\sum_{u \in U} |L(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (10)$$

(4) 多样性(*Diversification*):用来衡量不同用户推荐列表的之间的差异性,可以理解为用户内在的多样性,公式如下:

$$Diversification = \frac{2}{n(n-1)} \sum_{i \neq j} (1 - \frac{|L_i \cap L_j|}{|L_i|}) \quad (11)$$

式中, L_i 表示用户 U_i 的推荐列表集合; L_j 表示用户 U_j 的推荐列表集合;多样性的值越大,则推荐的个性化程度越高,反之,值越小,推荐的个性化程度越低。

4.3 实验结果对比

为了使实验验证更加充分,本文采用 5-折交叉验证,将 MovieLens 数据集随机划分为 5 个子数据,每次选取其中 1 个作为测试集,剩下的 4 个作为训练集,重复进行 5 次,涵盖所有子数据。相当于 5 次实验使用的都是不同的数据集,分别用 data1, data2, data3, data4, data5 表示。实验对比了传统的 Slope One 推荐算法(SO),基于项目个数加权的 Slope One 推荐算法(WSO)以及本文提出的基于加权二部图的 Slope One 推荐算法(WNBISO)的平均分值排名、准确率、召回率,以及多样性 4 种指标。

图 3 表示的是平均分排名 (ARankS) 在 data1~data5 不同数据集下的对比实验,横坐标代表的是数据集的编号,纵坐标是 ARankS 的值。

由于 ARankS 的值越小推荐的准确率越高,从图 3 可以看出在 data1~data5 不同数据集下,本文提出的 WNBISO 算法的推荐效果均好于传统的 SO 算法和 WSO 算法。

图 4 表示的是不同推荐步长 s 的情况下, data1~data5 所有数据集的平均准确率的实验对比,横坐标为推荐的步长 s ,取值为 10~100,以 10 为间隔,纵坐标是 data1~data5 所有数据集下的平均准确率的值。

由图 4 可知,在不同步长 s 下,本文提出的

WNBISO 算法在给定的所有步长 s 下的平均准确率均高于 SO 算法和 WSO 算法。

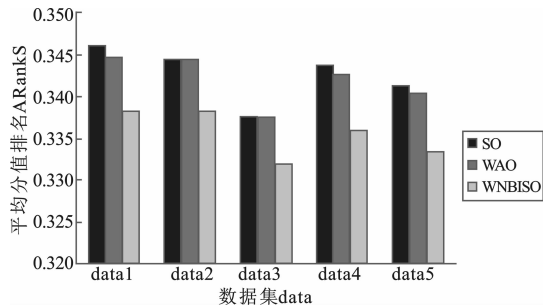


图3 不同数据集下的平均分排名

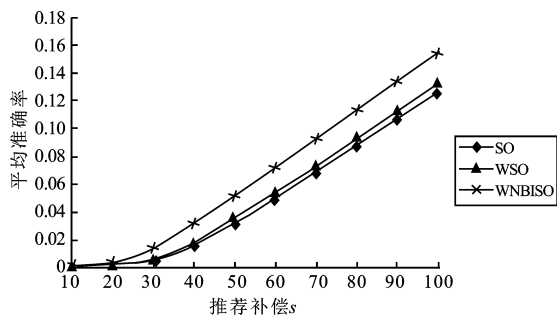


图4 不同推荐步长 s 下的平均准确率

图 5 所示为不同推荐步长 s 的情况下, data1~data5 所有数据集的平均召回率的实验对比,横坐标为推荐的步长 s ,取值为 10~100,以 10 为间隔,纵坐标分别是 data1~data5 数据集下的平均召回率的值。

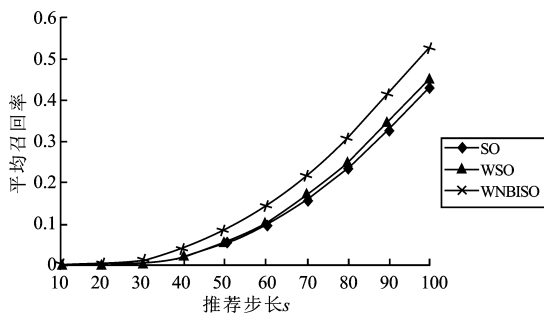


图5 不同推荐步长 s 下的平均召回率

由图 5 可以看出,本文提出的 WNBISO 算法在给定的所有步长 s 下的平均召回率均高于 SO 算法和 WSO 算法。

图 6 分别是不同推荐步长 s 的情况下, data1~data5 所有数据集的平均多样性的实验对比,横坐标均为推荐的步长 s ,取值为 10~100,以 10 为间隔,纵坐标分别是 data1~data5 数据集下的平均多样性的值。

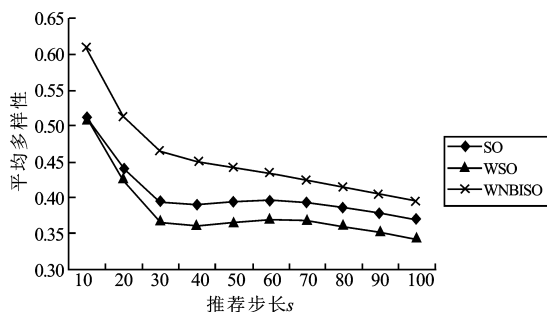


图6 不同推荐步长 s 下的平均多样性

由图6可以看出,在不同步长 s 下,本文提出的 WNBISO 算法在给定的所有步 s 下的平均多样性均高于 SO 算法和 WSO 算法,而且随着步长 s 的增加,多样性降低,这是由于推荐的项目个数越多,不同用户的推荐列表中则有更大概率出现相同的项目。另外结合图3~5可知,本文提出的 WNBISO 算法,不仅能提高推荐的准确度,还能增加推荐的多样性,给用户提供更丰富、个性化的推荐,避免推荐趋于同质化。

5 结束语

本文提出了一种基于加权二部图的 Slope One 推荐算法(WNBISO),通过利用加权二部图的资源分配来计算项目之间的推荐程度,基于得到的项目之间推荐值来对 Slope One 算法的评分预测过程进行加权,从而获得更加准确并且多样化、个性化的推荐。通过实验证明了算法的有效性,在平均分排名、准确率、召回率以及多样性4个评价指标中,均好于加权的 Slope One 算法(WSO)以及传统的 Slope One(SO)算法。下一步的工作是考虑在本文的基础上加入标签等信息,从而进一步提高推荐的效果。

参考文献:

- [1] 王志虎,黄曼莹. 基于用户历史行为的协同过滤推荐算法[J]. 微电子学与计算机, 2017, 34(5): 132-136.
- [2] 孟祥武,刘树栋,张玉洁,等. 社会化推荐系统研究

[J]. 软件学报, 2015, 26(6): 1356-1372.

- [3] 黄光球,靳峰,彭绪友. 基于兴趣度的协同过滤商品推荐系统模型[J]. 微电子学与计算机, 2005, 22(3): 5-8.
- [4] 李淋淋,倪建成,于苹苹,等. 一种基于聚类和 Spark 框架的加权 Slope One 算法[J]. 计算机应用, 2017, 37(5): 1287-1291.
- [5] Li Q, Han S, Li Y, et al. Threshold optimized slope one algorithm based book recommendation[C]// International conference on Communication Technology and Application. IEEE, 2014: 281-288.
- [6] Zhao Zhengde, Li Jianjun. Based on slope-one hybrid recommendation[C]// Advanced Research and Technology in Industry Applications. IEEE, 2014: 203-205.
- [7] 王毅,楼恒越. 一种改进的 Slope One 协同过滤算法[J]. 计算机科学, 2011, 38(b10): 192-194.
- [8] 张鹏,葛小青. 融合标签相似度的 k 近邻 Slope One 算法[J]. 重庆邮电大学学报(自然科学版), 2016, 28(4): 518-524.
- [9] 张怡文,王冉,程家兴. 基于用户兴趣度的改进二部图随机游走推荐方法[J]. 计算机应用与软件, 2015, 32(6): 76-79.
- [10] 刘洪,陈恩红. 结合二部图投影与排序的协同过滤[J]. 小型微型计算机系统, 2010, 31(5): 835-839.
- [11] 张新猛,蒋盛益. 基于加权二部图的个性化推荐算法[J]. 计算机应用, 2012, 32(3): 654-657.

作者简介:

王冉 男, (1993-), 硕士研究生. 研究方向为个性化推荐、深度学习.

徐怡(通讯作者) 女, (1981-), 博士, 副教授. 研究方向为智能信息处理、粒计算、粗糙集理论、个性化推荐.

E-mail: xuyi1023@126.com.

胡善忠 男, (1990-), 硕士研究生. 研究方向为粗糙集理论、个性化推荐.

何明慧 男, (1991-), 硕士研究生. 研究方向为神经网络、个性化推荐.