

基于用户多属性与兴趣的协同过滤算法^{*}

赵文涛, 王春春, 成亚飞, 孟令军, 赵好好

(河南理工大学 计算机科学与技术学院, 河南 焦作 454000)

摘要: 传统的协同过滤算法广泛应用于推荐系统领域, 但该算法仍存在用户冷启动和数据稀疏性问题, 造成算法的推荐质量较差。对此, 提出一种基于用户多属性与兴趣的协同过滤算法 AICF(attributes and interests collaborative filtering)。首先通过对多种用户属性分配权重计算出用户多属性相似度。其次利用改进的 Slope One 算法填充用户—项目评分矩阵, 然后计算基于隐性标签的用户兴趣相似度。最后基于两种相似度的组合进行推荐。实验结果表明, AICF 算法不仅明显提高了推荐结果的准确性, 同时也改善了用户冷启动和数据稀疏性问题。

关键词: 协同过滤; 冷启动; 数据稀疏性; 用户多属性; 隐性标签

中图分类号: TP393.04

文献标志码: A

文章编号: 1001-3695(2016)12-3630-04

doi:10.3969/j.issn.1001-3695.2016.12.025

Collaborative filtering algorithm based on multiple attributes and interests of users

Zhao Wentao, Wang Chunchun, Cheng Yafei, Meng Lingjun, Zhao Haohao

(College of Computer Science & Technology, Henan Polytechnic University, Jiaozuo Henan 454000, China)

Abstract: The traditional collaborative filtering algorithms have been widely used in the field of recommender systems. However, there is a decline in the quality of recommendation due to user cold start and data sparsity. Therefore, this paper proposed a collaborative filtering algorithm called AICF(attributes and interests collaborative filtering), which was based on multiple attributes and interests of user. Firstly, AICF assigned a variety of user attributes weights to calculate the user multi-attribute similarity. Secondly, AICF applied the improved Slope One algorithm fill user-item rating matrix, and then implicit tag similarity of user interest could be worked out. Finally, AICF combined these two similarities to get recommendation results. Experimental results show that the proposed algorithm not only highly improves the accuracy of the recommendation, but also improves problems of user cold start and data sparsity.

Key words: collaborative filtering; cold start; data sparsity; user multi-attribute; implicit tag

0 引言

信息技术和互联网的快速发展,使人们从信息匮乏的时代步入信息过载的时代。推荐系统便应运而生。推荐系统的基本任务是联系用户和物品,解决信息过载问题。在电子商务、音乐、社交网络、阅读、电影和视频、广告、基于位置的服务和个性化邮件等众多领域得到广泛应用^[1]。

推荐系统的方法有基于社交网络的推荐^[2]、基于内容的推荐^[3]和基于协同过滤的推荐^[4]。基于社交网络的推荐认为用户不一定信任计算机的智能,但会信任好友的推荐。基于内容的推荐是以用户间拥有相似内容属性作为推荐的标准。基于协同过滤的推荐是目前个性化推荐系统中应用较多的一种推荐技术,主要包括基于用户的协同过滤和基于项目的协同过滤^[5]。然而随着数据量日益庞大,数据稀疏性、可扩展性以及用户冷启动等问题日益严重,导致推荐质量逐步下降。为了提高推荐质量,研究者们提出了许多改进的方法。Hariri 等人^[6]根据用户的持续性行为动态调整用户的意见来匹配用户的最

近喜好,基于此开发了一个交互式推荐系统。Sedhain 等人^[7]将人口统计学数据和社交网络信息引入到协同过滤算法中,避免了用户冷启动问题。Vermaas 等人^[8]利用余弦相似度计算标签的增量,以提高算法的可扩展性。Begelmann 等人^[9]通过对标签聚类的方法来改善用户体验。不过这些研究都只考虑了单一的相似度计算,并未全方位地考虑用户的各个行为。

因此,本文提出基于用户多属性与兴趣的协同过滤算法 AICF(attributes and interests collaborative filtering)。首先根据属性的类型计算用户单个属性的相似度,然后为每个属性分配权重计算出用户的多属性相似度;其次,利用改进的 Slope One 算法填充用户—项目评分矩阵,使用项目的类型信息挖掘出用户的隐性标签计算出用户的兴趣相似度;最后,基于两种相似度的组合找出与目标用户最相似的 k 近邻居并预测未评分项目的分数,产生 Top N 推荐。利用用户的属性信息计算相似度,避免了用户冷启动问题。通过填充用户—项目评分矩阵,有效地改善了数据稀疏性的问题。实验表明,该方法显著提高了推荐质量。

收稿日期: 2016-01-14; **修回日期:** 2016-02-29

基金项目: 河南省科技攻关资助项目(142402210435);河南省高等学校矿山信息化重点

学科开放基金资助项目(ky2012-02)

作者简介: 赵文涛(1963-),男,河南焦作人,硕士,教授,主要研究方向为数据库技术、信息系统、数据挖掘(spring_0205@sina.com);王春春(1990-),女,河南洛阳人,硕士研究生,主要研究方向为数据挖掘;成亚飞(1990-),男,河南焦作人,硕士研究生,主要研究方向为数据挖掘;孟令军(1989-),男,河南新乡人,硕士研究生,主要研究方向为数据挖掘、云计算;赵好好(1989-),女,河南汝州人,硕士研究生,主要研究方向为数据挖掘。

1 相关研究

1.1 基于用户的协同过滤算法

基于用户的协同过滤算法的基本思想:在个性化推荐中,若用户 u 需要个性化推荐,可以先找到与用户 u 兴趣相似的其他用户,然后把那些用户喜欢的、而用户 u 没有听说过的物品推荐给 u 。在用户—项目评分矩阵中存在 m 个用户 $U = \{u_1, u_2, \dots, u_m\}$ 和 n 个项目 $I = \{i_1, i_2, \dots, i_n\}$ 。用户对项目的评分矩阵 $R_{mn} = \{R_{ui}\}$,其中 R_{ui} 表示用户 u 对项目 i 的评分。使用相似性度量方法计算用户之间的相似度,找出与目标用户最相似的 k 个邻居,使用预测算法预测目标用户对未评分项目的分数,产生 Top N 推荐。

1.1.1 相似度的计算

相似度计算方法包括余弦相似性,修正的余弦相似性和皮尔逊相关相似性等。本节主要介绍修正余弦相似性(见式(1))和皮尔逊相似性(见式(2))。

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{uv}} (R_{ui} - \bar{R}_u) (R_{vi} - \bar{R}_v)}{\sqrt{\sum_{i \in I_u} (R_{ui} - \bar{R}_u)^2} \sqrt{\sum_{i \in I_v} (R_{vi} - \bar{R}_v)^2}} \quad (1)$$

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{uv}} (R_{ui} - \bar{R}_u) (R_{vi} - \bar{R}_v)}{\sqrt{\sum_{i \in I_{uv}} (R_{ui} - \bar{R}_u)^2} \sqrt{\sum_{i \in I_{uv}} (R_{vi} - \bar{R}_v)^2}} \quad (2)$$

式中: $\text{sim}(u, v)$ 表示用户 u 和 v 之间的相似度; R_{ui} 和 R_{vi} 分别表示用户 u, v 对项目 i 的评分; I_{uv} 表示用户 u 和用户 v 共同评过的项目集合; I_u 和 I_v 分别表示用户 u, v 已评分的项目集合; $\bar{R}_u$ 和 $\bar{R}_v$ 分别表示用户 u, v 对评过的项目的平均评分。

1.1.2 产生推荐

利用式(3)预测目标用户 u 对未评分项目 i 的分数,将预测分数最高的前 N 个项目推荐给目标用户 u 。

$$R_{u,i} = \bar{R}_u + \frac{\sum_{v \in S_u} (\text{sim}(u, v)) \times (R_{v,i} - \bar{R}_v)}{\sum_{v \in S_u} (\text{sim}(u, v))} \quad (3)$$

式中: $R_{u,i}$ 表示用户 u 对未评分项目 i 的预测评分; S_u 为用户 u 的最近邻集合。

1.2 Slope One 算法

Slope One 算法是一种基于线性回归模型的协同过滤算法^[10]。该算法假设用户 u 对两个项目的评分为 x, y , 则 x 和 y 之间符合线性关系 $y = x + b$ 。Slope One 算法具有简单易实现, 执行效率高等优点。Herlocker 等人^[11]使用 Slope One 算法对用户—项目评分矩阵的空值进行填充, 然后用协同过滤算法进行评分的预测。Slope One 算法最基本的表现形式为 Weight Slope One 算法。首先计算目标项目 j 与其他项目 i 之间的评分偏差 $\text{dev}_{j,i}$, 然后以共同评价过项目 j 和项目 i 的用户作为这两个项目评分偏差的权重。

$$\text{dev}_{j,i} = \sum_{u \in U_{ij}} \frac{u_j - u_i}{\text{card}(U_{ij})} \quad (4)$$

$$P^{\text{wso}}(u)_j = \frac{\sum_{i \in I_u} (\text{dev}_{j,i} + u_i) \text{card}(U_{ij})}{\sum_{i \in I_u} \text{card}(U_{ij})} \quad (5)$$

式中: U_{ij} 表示对项目 i 和 j 都评分的用户集合, $\text{card}(U_{ij})$ 表示集合 U_{ij} 中的元素个数。

2 基于多属性的用户相似度计算

传统的协同过滤算法依赖于用户对项目的评分, 当遇到新

用户或者新项目时, 由于缺乏历史评分数据导致推荐质量下降。同时在实际应用中, 用户对项目的评分缺乏主动性, 单一的依靠评分数据有很大的局限性。因此本文提出一种基于用户多属性的相似度计算方法, 来避免用户冷启动造成的推荐质量较低问题。用户的属性一般分为: 数值型(比率)属性, 如用户的年龄; 二元属性, 如用户的性别; 标称属性, 如用户的职业和邮编等。假设每个用户有 l 个属性 $(x_{j,1}, x_{j,2}, \dots, x_{j,3}, \dots, x_{j,l})$, x_j 表示用户第 j 个属性的取值。首先计算其在每一维属性上的相似度, 然后计算用户的多属性相似度。基于此, 给出了一种多属性相似度度量方法(见式(6))。

$$\text{Asim}(u, v) = \frac{w_1 \text{sim}(u_{x_1}, v_{x_1}) + w_1 \text{sim}(u_{x_2}, v_{x_2}) + \dots + w_l \text{sim}(u_{x_l}, v_{x_l})}{w_1 + w_2 + \dots + w_l = 1} \quad (6)$$

式中: l 是属性的个数; $\text{sim}(u_{x_j}, v_{x_j})$ 是用户 u 和 v 在属性 x_j 上的相似度; $w_1, w_2, \dots, w_l$ 为各个属性相似度分配的权值。

数值型属性的相似度计算以年龄为例。根据世界卫生组织(WHO)最新的年龄划分方法, 将人的年龄划分为五个阶段: 44 岁以下为青年, 分段值记为 1; 45 ~ 59 岁为中年, 分段值记为 2; 60 ~ 74 为年轻的老人, 分段值记为 3; 75 岁以上为老年, 分段值记为 4; 90 岁以上为长寿老人, 分段值记为 5。利用欧几里德距离计算用户 u 和 v 的数值型属性(年龄)相似度(见式(7))。

$$\text{sim}(u_{x_j}, v_{x_j}) = e^{-|x_{j,h} - x_{j,g}|} \quad (7)$$

式中: $x_{j,h}$ 和 $x_{j,g}$ 分别为用户 u 和 v 的年龄分段值。

二元属性分为对称的二元属性和非对称的二元属性。用户 u 和 v 取不同的二元变量值时, 可以得到一个两行两列的列联表(见表 1)。

表 1 列联表

对象 u	对象 v		
	1	0	sum
1	q	r	$q + r$
0	s	t	$s + t$
sum	$q + s$	$r + t$	p

表 1 中: q 是对象 u 和 v 都取 1 的属性数; r 是对象 u 中取 1、在对象 v 中取 0 的属性数; s 是在对象 u 中取 0、在对象 v 中取 1 的属性数; 而 t 是对象 u 和 v 都取 0 的属性数。属性的总数是 p , 其中 $p = q + r + s + t$ 。本文采用简单匹配系数计算二元属性的相似度。对于对称的二元属性, 每个状态都同样重要。以性别为例, 男性用户记为 1, 女性用户记为 0。用户 u 和 v 的对称二元属性相似度如式(8)所示。

$$\text{sim}(u_{x_j}, v_{x_j}) = \frac{q + t}{q + r + s + t} \quad (8)$$

对于非对称的二元属性, 两个状态不是同等重要的。此时, 用户 u 和 v 的非对称二元属性相似度如式(9)所示。

$$\text{sim}(u_{x_j}, v_{x_j}) = \frac{q}{q + r + s + t} \quad (9)$$

标称属性的属性值属于枚举类型, 用来表示某些符号或事物的名称。每个值代表某种类别、编码或状态, 如职业、邮编等都属于标称属性。用户 u 和 v 的标称属性相似度如式(10)所示。

$$\text{sim}(u_{x_j}, v_{x_j}) = \frac{d}{p} \quad (10)$$

式中: d 指用户 u 和 v 共同拥有的标称属性个数, p 指标称属性可选的属性总数。

3 基于隐性标签的用户兴趣相似度计算

评分使用户与项目建立直接联系,特征使用户与项目建立间接联系。标签作为一种重要的特征表现方式被广泛使用,它既描述了项目的语义,也反映了用户的兴趣。通过用户给项目打标签的行为进行个性化推荐的方法已屡见不鲜,但用户对项目打标签的主动性是不可控的,有时用户也不会按照设计者既定的想法操作,可能会打出一些奇怪的标签,导致标签数据十分稀疏。本文提出利用项目的类型信息挖掘出用户的隐性标签,以降低标签数据的稀疏性,提高推荐的准确率。

Herlocker 等人^[11]以两个用户共同评过的项目个数除以一个阈值作为相似度约束系数,并用该系数对相似度进行修正。但是阈值的获取必须通过多次验证才能获得,增加了算法的复杂度。罗琦等人^[12]参照用户间共同评分项目个数越多则用户兴趣越相似的原则,通过实验证明采用反正切函数调整用户的相似度是可行的,但是忽略了用户对项目评分的变化。

本文以用户对项目的评分为基础,利用项目的类型挖掘出用户的隐性标签,用户间共同拥有的隐性标签个数越多且被共同标记的隐性标签出现的次数越多,用户的兴趣越相似。以五分制为例,用户给某个项目打五分,认为用户对该项目非常喜欢,用户对某个项目打一分,认为用户非常不喜欢该项目。但是实际中每个用户对喜欢的衡量标准又不一样,本文引入中位数作为用户的评分阈值,用户与评分阈值一一对应,通过这个阈值来判定用户是否对某个项目感兴趣。

3.1 构建稠密矩阵

在推荐系统中,用户评分的稀疏性是造成推荐结果下降的重要原因。Slope One 算法可以充分地利用已有的评分数据快速高效地预测未评分项,大大降低了矩阵的稀疏性。但是该算法忽略了相似性,因此不能有效地提高推荐的准确率。本文将相似度引入 Slope One 算法,增加相似度约束系数以提高填充项的准确率。

$$P^{\text{swso}}(u)_j = \frac{S_{i,j} \sum_{i \in I_u - |j|} (\text{dev}_{j,i} + u_i) \text{card}(U_{ij})}{\sum_{i \in I_u - |j|} \text{card}(U_{ij})} \quad (11)$$

式中: $P^{\text{swso}}(u)_j$ 表示用户 u 对项目 j 的填充评分; $S_{i,j}$ 表示项目 i 和项目 j 的皮尔逊相似性。 $\text{dev}_{j,i}$ 如式(4)所示。

3.2 用户兴趣相似度计算

用户对所有项目的评分集合对应的中位数评分阈值集合为 $M = \{M_1, M_2, \dots, M_m\}$,项目的类型集合 $Y = \{y_1, y_2, \dots, y_e\}$ 。当 $R_{ui} \geq M_i$ 时,认为用户 u 喜欢项目 i 。此时,项目 i 对应的项目类型 y_e 被标记为用户 u 的隐性标签。根据项目的类型和用户的评分阈值提取用户的隐性标签。某个用户可能多次被同一个隐性标签标记,对次数进行累加,形成用户的隐性标签矩阵 $T_{me} = \{T_{uy}\}$, T_{uy} 表示用户 u 被项目类型 y 标记的隐性标签次数。

表2 用户—项目评分矩阵

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	5	3	1	5	1	4	3
u_2	4	2	4	4	5	3	4
u_3	4	2	4	3	2	2	3
u_4	2	1	2	5	3	4	1
u_5	5	4	3	2	5	1	3

表2中列出了用户 $u_1 \sim u_5$ 对项目 $i_1 \sim i_7$ 的评分矩阵,得到用户的中位数评分阈值的集合 $M = \{3, 4, 3, 2, 3\}$ 。将低于评

分阈值的评分记为0,结果如表3所示。

表3 用户—项目中位数评分矩阵

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	5	3	0	5	0	4	3
u_2	4	0	4	4	5	3	4
u_3	4	0	4	3	0	0	3
u_4	2	0	2	5	3	4	0
u_5	5	4	3	0	5	0	3

表4 项目—类型矩阵

	y_1	y_2	y_3	y_4	y_5
i_1	1	1	0	0	1
i_2	1	0	0	0	1
i_3	0	1	0	0	1
i_4	1	1	1	0	0
i_5	1	0	0	1	1
i_6	1	0	0	1	0
i_7	0	0	1	0	1

表4中,项目 i 属于类型 y 记为1,否则记为0。表3中不为0的评分项与表4中的项目类型对应后如表5所示。

表5 用户—隐性标签矩阵

	y_1	y_2	y_3	y_4	y_5
u_1	4	2	2	1	3
u_2	4	3	2	2	4
u_3	2	3	2	0	3
u_4	4	3	1	2	3
u_5	3	2	1	1	5

表5中用户与项目类型建立了联系。参照用户间共同的隐性标签的个数越多且被共同标记的隐性标签出现的次数越多,它们的兴趣越相似的原则,使用用户间的隐性标签作为相似度约束系数来修正用户相似度。

$$\text{Sum}(u) = \sum_{y \in T_u} T_y \quad (12)$$

$$B\text{sim}(u, v) = \arctan \left[1 - \frac{\sum_{y \in T_{uv}} \left(\frac{T_{yu}}{\text{Sum}(u)} - \frac{T_{yv}}{\text{Sum}(v)} \right)}{\text{card}(T_{uv})} \right] \times \text{sim}(u, v) \quad (13)$$

式(12)中: $\text{Sum}(u)$ 表示用户 u 的标签个数总和。式(13)中: T_{yu} 和 T_{yv} 分别表示用户 u 对应类型 y 被标记的隐性标签次数和用户 v 被标记的隐性标签次数; $\text{card}(T_{uv})$ 表示用户 u 和用户 v 共同拥有的隐性标签个数; $\text{sim}(u, v)$ 为式(1)中的修正余弦相似度; $B\text{sim}(u, v)$ 表示基于隐性标签的用户兴趣相似度。

4 基于组合相似度的推荐

组合基于多属性的用户相似度和基于隐性标签的用户兴趣相似度(式(16)),计算最终的用户相似度。

$$U\text{sim}(u, v) = (1 - \alpha)A\text{sim}(u, v) + \alpha B\text{sim}(u, v) \quad (16)$$

式中: α 的值在0~1变化, $U\text{sim}(u, v)$ 表示组合用户相似度,用式(3)预测用户未评分的项目,产生 Top N 推荐。

4.1 推荐过程

本文算法的流程如图1所示。

输入:用户—属性矩阵 $X_{j,l}$,用户—项目评分矩阵 $R_{m,n}$,项目—类型矩阵 $Z_{e,n}$,目标用户 u ,待评分项目 i ,中位数评分阈值 M 。

输出:目标用户 u 对待评分项目 i 的预测评分。

a) 针对用户—属性矩阵 $X_{j,l}$,分别计算用户各个属性的相似度 $\text{Sim}_1, \text{Sim}_2, \dots, \text{Sim}_l$;

b) 利用式(6)计算用户的多属性相似度 $A\text{Sim}$;

c) 利用式(11)得到稠密矩阵 $R'_{m,n}$;

d) 利用中位数评分阈值 M 、矩阵 $R'_{m,n}$ 和矩阵 $Z_{e,n}$ 经过筛选计算后得到用户—隐性标签矩阵 T_{me} ;

e) 利用矩阵 T_{me} 和式(1)计算用户兴趣相似度 $B\text{Sim}$;

f) 利用式(16)计算用户的组合相似度 $U\text{Sim}$,形成目标用户 u 的

候选邻居集合 S_u ;

g) 通过集合 S_u 和式(3)计算出目标用户 u 对项目 i 的评分,产生 Top N 推荐。

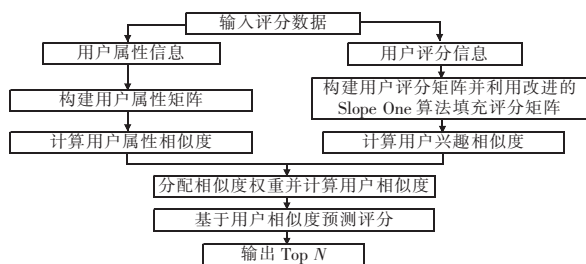


图1 算法流程图

5 实验分析与结果

5.1 数据集

本实验采用的数据来源是 MovieLens 站点提供的 100 K 的公开数据集 (<https://movielens.org/>), 由 943 个用户对 1 682 个项目打分产生的 100 000 条评分数据, 每个用户对至少 20 部电影进行评分。评分标准为 1~5 分, 打分越高, 表示对电影越感兴趣。如果用户对某部电影没有评分, 其评分值为 0。根据数据稀疏性公式可以计算该数据集的数据稀疏度。

$$S = \left(1 - \frac{N}{m \times n}\right) \times 100\% \quad (17)$$

式中: S 表示数据稀疏度, m 表示用户数, n 表示项目数, N 为总的评分数。计算得出 MovieLens 数据集的稀疏度为 93.69%。

MovieLens 数据集提供了用户的性别、年龄、职业和邮编等属性。年龄属于数值型属性, 使用式(7)计算; 性别属于对称的二元属性, 使用式(8)计算; 职业和邮编属于标称属性, 使用式(10)计算。每个属性的权重值设为 0.25。MovieLens 还提供了电影的类型, 通过电影的类型得到用户的隐性标签, 计算出第 3 章中基于隐性标签的用户兴趣相似度。

5.2 度量标准

本文评分的预测准确度采用平均绝对误差 (MAE) 计算, 预测误差越小则推荐质量越高。

$$MAE = \frac{\sum_{u,i \in H} |r_{ui} - \hat{r}_{ui}|}{|H|} \quad (18)$$

式中: r_{ui} 是用户 u 对物品 i 的实际评分, $\hat{r}_{ui}$ 是本文算法给出的预测评分, H 表示测试集中共有 H 条评分数据。

5.3 实验结果

本文的实验采用五折交叉验证的方法。将 MovieLens 数据集分成五份, 在实验中每次随机选取其中的四份作为训练集, 剩余的一份作为测试集。将本文的算法 (AICF) 与传统的基于用户的协同过滤算法 (记为 UBCF) 和基于项目的协同过滤算法^[13] (记为 IBCF) 作比较, MAE 值越小, 推荐质量越高。

为了最终能在最优的参数下进行推荐, 首先需要确定 α 的值。参照 MAE 值越小, 推荐质量越高的原则, 以最相似邻居个数 k 值为 20, 50, 80 为例, 将 α 的值从 0 开始, 每次递增 0.1 直到 1 为止。分别计算其 MAE 值。如图 2 所示, 随着 α 的值不断增加, MAE 值不断减小, 当 α 取 0.9 时 MAE 值达到最小值, 随后 α 的值继续增加, MAE 值也增大。因此, α 的最优值为 0.9。如图 3 所示, k 值以 10 为基数递增到 80, 同时与其他两种算法相比, 明显看出本文的算法优于 UBCF 算法。当 k 值取 10 时, 本文的算法推荐效果最好, 说明在用户的最近邻个数

很少时, 本文的算法也能达到很好的推荐效果, 避免了用户冷启动问题。随着 k 值的增加本文算法所得的 MAE 值略有增大, 当 k 值为 50 时, 与 IBCF 算法的 MAE 值基本持平。随后 k 值继续增加, MAE 值又逐渐减小。

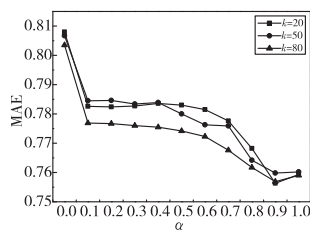


图2 $k=20, 50, 80$ 时不同 α 值下 MAE 值的比较

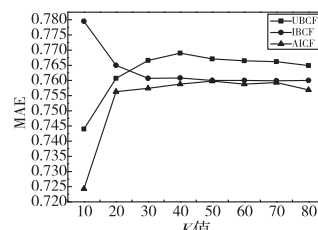


图3 不同算法的 MAE 值的比较

随着 k 值持续增大, MAE 取值最小时的 α 值有所变化。以 k 值为 200, 500, 800 为例, 将 α 的值从 0.90 开始, 每次递增 0.01 直到 1.00 为止。分别计算本文算法的 MAE 值。如图 4 所示, k 值为 200 时, α 取 0.95 时 MAE 值达到最小。随后 k 值增至 500 时, MAE 值在 α 取 0.99 和 1.00 时达到平衡。当 k 值达到 800 时, MAE 值在 α 取 1.00 时最小。说明随着 k 值的不断增大, 用户属性对推荐质量的影响越来越小, 也进一步证明了加入用户的属性因子对用户冷启动问题的改善。图 5 中, k 值从 100 开始, 每次增加 100 直到 800 为止。随着 k 值的不断增加, 在 k 值取 300 时, MAE 值最小。随后 k 值继续增加, MAE 值趋于平缓。

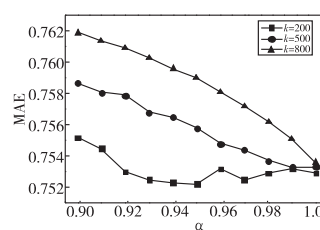


图4 $k=200, 500, 800$ 时不同 α 值下 MAE 值的比较

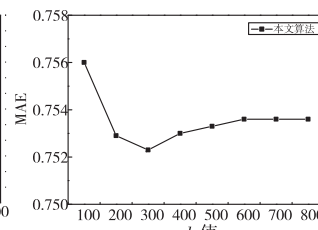


图5 不同 k 值下的 MAE 值的比较

6 结束语

本文在传统的基于用户的协同过滤算法基础上, 将用户的属性融入相似度计算中, 避免了用户冷启动问题。在用户的兴趣相似度计算上, 利用用户已经评价过项目的类型信息挖掘出用户的隐性标签, 以此作为基于用户-项目评分矩阵的相似度计算的约束条件, 提高了用户相似度的准确性。然后将用户属性相似度与用户兴趣相似度结合, 动态调整两者在相似度计算中的比例, 以达到相似度的最优组合。最终找出与目标用户最相似的 k 个邻居预测评分并进行 Top N 推荐。实验结果表明, 在遇到新用户或者新项目时由于缺乏其历史信息, 很难为其进行比较合适的推荐, 在这种情况下, 本文的方法有效提高了推荐的准确性, 并在数据稀疏的情况下, 获得更好的推荐质量。

用户的属性在一定程度上避免了用户冷启动问题, 但是在用户属性的权重分配上, 还需要进一步研究。此外, 推荐系统的数据集提供有时间属性, 怎样将时间属性融入到用户相似度的计算中, 将是下一步的研究重点。

参考文献:

- [1] 项亮. 推荐系统实践[M]. 北京: 人民邮电出版社, 2012: 3-4.
- [2] 刘玮. 电子商务系统中的信息推荐方法研究[J]. 情报科学, 2006, 24(2): 300-303.

(下转第 页)

4 结束语

灰狼优化算法是一种模拟自然界中灰狼群体社会等级机制和捕食行为的新型群体智能优化方法。本文利用反向学习策略产生初始灰狼个体位置以维持群体多样性;受 PSO 算法的启发,使收敛因子随进化迭代次数增加而非线性动态变化,以协调算法的勘探和开采能力;对当前最优灰狼个体进行变异操作以减少算法出现早熟收敛的可能性。提出一种基于非线性收敛因子的改进灰狼优化算法用于解决高维函数优化问题。通过对 10 个标准测试函数进行数值实验,结果表明,改进灰狼优化算法具有较强的寻优性能。

参考文献:

- [1] 王冰. 基于局部最优解的改进人工蜂群算法[J]. 计算机应用研究, 2014, 31(4): 1023-1026.
- [2] 井福荣, 郭肇禄, 罗会兰, 等. 应用精英反向学习的引力搜索算法[J]. 计算机应用研究, 2015, 32(12): 3638-3641.
- [3] Leung Y, Wang Yuping. An orthogonal genetic algorithm with quantization for global numerical optimization[J]. *IEEE Trans on Evolutionary Computation*, 2001, 5(1): 41-53.
- [4] Coelho L, Ayala H, Mariani V. A self-adaptive chaotic differential evolution algorithm using gamma distribution for unconstrained global optimization[J]. *Applied Mathematics and Computation*, 2014, 234(5): 452-459.
- [5] Akay B, Karaboga D. A modified artificial bee colony algorithm for real-parameter optimization [J]. *Information Sciences*, 2012, 192(1): 120-142.
- [6] Sun Yang, Zhang Lingbo, Gu Xingsheng. A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization[J]. *Neurocomputing*, 2012, 98(3): 76-89.
- [7] Gandomi A H, Yang Xinshe, Talatahari S, et al. Firefly algorithm with chaos[J]. *Communications in Nonlinear Science and Numerical Simulation*, 2013, 18(2): 89-98.
- [8] Mirjalili S, Lewis A. Adaptive gbest-guided gravitational search algorithm[J]. *Neural Computing and Applications*, 2014, 25(7): 1569-1584.
- [9] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimization[J]. *Advances in Engineering Software*, 2014, 69(7): 46-61.
- [10] Song H M, Sulaiman M, Mohamed M. An application of grey wolf optimizer for solving combined economic emission dispatch problems[J]. *International Review on Modeling and Simulations*, 2014, 7(5): 838-844.
- [11] 杨红光, 刘建生. 一种结合灰狼优化和 K-均值的混合聚类算法[J]. 江西理工大学学报, 2015, 36(5): 85-89.
- [12] El-Gaafary A, Mohamed Y, Hemeida A, et al. Grey wolf optimization for multi input multi output system[J]. *Universal Journal of Communications and Networks*, 2015, 3(1): 1-6.
- [13] Song Xianhai, Tang Li, Zhao Suta, et al. Grey wolf optimizer for parameter estimation in surface waves[J]. *Soil Dynamics and Earthquake Engineering*, 2015, 75(5): 147-157.
- [14] Madadi A, Motlagh M. Optimal control of DC motor using grey wolf optimizer algorithm[J]. *Technical Journal of Engineering and Applied Science*, 2014, 4(4): 373-379.
- [15] Komaki G, Kayvanfar V. Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time[J]. *Journal of Computational Science*, 2015, 8(3): 109-120.
- [16] Mirjalili S. How effective is the grey wolf optimizer in training multi-layer perceptrons[J]. *Applied Intelligence*, 2015, 42(2): 608-619.
- [17] Zhu Aijun, Xu Chuanpei, Li Zhi, et al. Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC[J]. *Journal of Systems Engineering and Electronics*, 2015, 26(2): 317-328.
- [18] 龙文, 赵东泉, 徐松金. 求解约束优化问题的改进灰狼优化算法[J]. 计算机应用, 2015, 35(9): 2590-2595.
- [19] Saremi S, Mirjalili S Z, Mirjalili S M. Evolutionary population dynamics and grey wolf optimizer[J]. *Neural Computing and Applications*, 2015, 26(5): 1257-1263.
- [20] Haupt R, Haupt S. Practical genetic algorithm[M]. New York: Wiley, 2004.
- [21] Tizhoosh H. Opposition-based learn; a new scheme for machine intelligence[C]//Proc of International Conference on Computational Intelligence for Modeling Control and Automation. [S. l.]: IEEE Press, 2005: 695-701.
- [22] Wang Yong, Cai Zixing, Zhou Yuren, et al. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique[J]. *Structural and Multidisciplinary optimization*, 2009, 37(1): 395-413.
- [23] Gao Weifeng, Liu Sanyang, Huang Lingling. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning[J]. *IEEE Trans on Cybernetics*, 2013, 43(3): 1011-1024.

(上接第 页)

- [3] 王国霞, 刘贺平. 个性化推荐系统综述[J]. 计算机工程与应用, 2012, 48(7): 66-76.
- [4] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry[J]. *Communications of the ACM*, 1992, 35(12): 61-70.
- [5] Karypis G. Evaluation of item-based top-n recommendation algorithms[C]//Proc of the 10th International Conference on Information and Knowledge Management. 2001.
- [6] Hariri N, Mobasher B, Burke R. Context adaptation in interactive recommender systems[C]//Proc of ACM Conference on Recommender Systems. 2014: 41-48.
- [7] Sedhain S, Sanner S, Brazianus D, et al. Social collaborative filtering for cold-start recommendations[C]//Proc of ACM Conference on Recommender Systems. 2014: 345-348.
- [8] Vermaas R, Vandic D, Frasincar F. Incremental cosine computations for search and exploration of tag spaces[M]//Database and Expert Systems Applications. Berlin Heidelberg: Springer, 2012: 156-167.
- [9] Begelman G, Keller P, PSmadja F. Automated tag clustering: improving search and exploration in the tag space[C]//Proc of the 15th Collaborative Web Tagging Workshop, International World Wide Web Conference. 2006.
- [10] Lemire D, Maclachlan A. Slope one predictors for online rating-based collaborative filtering[C]//SIAM; Proc of SIAM Data Mining Conference. 2005: 21-23.
- [11] Herlocker J L, Konstan J A, Terveen L G, et al. Evaluating collaborative filtering recommender system[J]. *ACM Trans on Information Systems*, 2004, 22(1): 5-53.
- [12] 罗琦, 缪昕杰, 魏倩. 稀疏数据集协同过滤算法的进一步研究[J]. 计算机科学, 2014, 41(6): 264-268.
- [13] 冷亚军, 陆青, 张俊岭. 结合类别偏好信息的 item-based 协同过滤算法[J]. 计算机应用研究, 2015, 33(3): 669-672.