Journal of Computer Applications
计算机应用,2013,33(11):3062-3066,3106
ISSN 1001-9081
CODEN JYIIDU
2013-11-01
http://www.joca.cn

文章编号:1001-9081(2013)11-3062-05
doi:10.11772/j.issn.1001-9081.2013.11.3062

考虑项目属性的协同过滤推荐模型

杨兴耀[1*],于　炯[1,2],吐尔根·依布拉音[1],钱育蓉[2],孙　华[2]
(1.新疆大学 信息科学与工程学院,乌鲁木齐 830046;　2.新疆大学 软件学院,乌鲁木齐 830008)
(*通信作者电子邮箱 yangxy@xju.edu.cn)

摘　要:针对传统的基于用户的协同过滤(UCF)模型在相似性度量过程中没有充分考虑项目属性的问题,提出了两种考虑项目属性的协同过滤推荐模型。模型首先对用户评分相似性进行优化;然后从项目属性的角度统计用户关于不同项目的评价次数,获得优化的基于项目属性的用户相似性;最后通过自适应平衡因子协调处理两方面的相似性结果进行项目预测与推荐。实验结果表明,在不同的数据集中,新提出的模型不仅时间花费较为合理,而且评分预测准确性明显提高,平均提高了5%,从而证明了模型在改进用户相似性度量精度方面的有效性。

关键词:推荐系统;协同过滤;评分相似性;项目属性;相似性模型
中图分类号:TP311.13;TP391.3　　**文献标志码**:A

Collaborative filtering recommendation models considering item attributes

YANG Xingyao[1*], YU Jiong[1,2], Turgun IBRAHIM[1], QIAN Yurong[2], SUN Hua[2]
(1. *College of Information Science and Engineering, Xinjiang University, Urumqi Xinjiang* 830046, *China*;
2. *School of Software, Xinjiang University, Urumqi Xinjiang* 830008, *China*)

Abstract: The traditional User-based Collaborative Filtering (UCF) models do not consider the attributes of items fully in the process of measuring the similarity of users. In view of the drawback, this paper proposed two collaborative filtering recommendation models considering item attributes. Firstly, the models optimized the rating-based similarity between users, and then summed the rating numbers of different items by users according to item attributes, in order to obtain the optimized and attribute-based similarity between users. Finally, the models coordinated the two types of similarity measurements by a self-adaptive balance factor, to complete the item prediction and recommendation process. The experimental results demonstrate that the newly proposed models not only have reasonable time costs in different data sets, but also yield excellent improvements in prediction accuracy of ratings, involving an average improvement of 5%, which confirms that the models are efficient in improving the accuracy of user similarity measurements.

Key words: recommender system; collaborative filtering; rating similarity; item attribute; similarity model

0　引言

近年来,随着Web 2.0技术的日益发展与成熟,信息量的爆炸式增长成为最明显的时代特征,它在使人们生活变得丰富多彩的同时,也让人们逐渐步入了"信息过载"的时代。在浩瀚的信息海洋中,普通用户如何更快更准确地从中检索到自己感兴趣的信息是一件非常困难的事情,而且可能出现的一种结果是,花费了大量的时间却没有找到自己想要寻找的信息,即出现所谓的"资源迷向"问题。同时作为信息提供商,如何从海量的信息中获取用户的偏好特征让自己提供的信息内容脱颖而出,满足用户的个性化需求,同样是一件非常困难的事情。在这种背景下,推荐系统[1]作为建立在海量数据挖掘平台基础上的一种高级智能推手,它能够根据用户的特征、用户所处的情景信息和历史记录等,帮助信息提供商自动地为用户提供个性化的决策支持和信息推荐,从而具有巨大的应用潜力和商业空间。相关数据[2]表明,Amazon将推荐系统应用到电子商务中,通过分析用户的购买、浏览行为,预测用户可能感兴趣的商品,从而将销售额成功地提高了35%。类似的例子还有美国著名的网上零售商Overstock,采用个性化推荐方案后,公司的广告点击率是以前的两倍,伴随而来的销售增长也高达20%至30%。

推荐模型作为个性化推荐系统的核心组成部分,其性能的高低直接决定着系统性能的好坏。为了获得较好的系统性能,在不同的应用环境下涌现出了各种各样的推荐模型。根据推荐信息产生原理的不同,大致可以分为:协同过滤模型[3]、内容过滤模型[4]、网络结构模型[5]、基于规则的过滤模型[6]等。其中针对协同过滤模型的研究最为热门和深入[7-8],其原理是根据与目标用户相似的用户的兴趣来预测目标用户可能感兴趣的信息内容,并将最终的预测结果推荐给目标用户。目前,基于协同过滤的推荐模型已经广泛应用到各个领域,如社交网络、音乐视频点播、电子商务等[7],但在具体的应用中还存在一些问题,系统性能有待进一步提高。

传统的协同过滤模型主要依赖于用户关于项目的评分来度量对象之间的相似性,在相似性度量过程中却较少考虑不

收稿日期:2013-05-24;**修回日期**:2013-07-21。
基金项目:国家自然科学基金资助项目(61262088,61063042,61063026);新疆大学优秀博士创新项目基金资助项目(XJUBSCX-2011007);新疆维吾尔自治区自然科学基金资助项目(2011211A011);新疆高校重大科研项目(XJEDU2012I10)。
作者简介:杨兴耀(1984-),男,湖北襄阳人,博士研究生,CCF会员,主要研究方向:推荐系统、网格计算与云计算、可信计算;　于炯(1964-),男,新疆乌鲁木齐人,教授,博士生导师,主要研究方向:网络安全、网格与分布式计算;　吐尔根·依布拉音(1958-),男,新疆乌鲁木齐人,教授,博士生导师,主要研究方向:自然语言处理、软件工程;　钱育蓉(1980-),女,山东武城人,博士,主要研究方向:遥感图像处理、模式识别、数据挖掘;孙华(1977-),女,山东烟台人,博士,主要研究方向:信息安全、信誉管理。

同对象的类别属性等特征,这在很大程度上影响了推荐系统性能。此外,对象相似性的优化同样是一个值得关注的问题,因为相似性模型本身会存在一定的缺陷。针对这些问题,本文提出了两种考虑项目属性的协同过滤推荐模型,在改进用户评分相似性的过程中,同时考虑项目属性方面的用户相似性,使得目标用户的近邻用户集合更加合理,最终提高推荐系统的性能。

1　传统的协同过滤推荐模型

协同过滤推荐模型作为当前研究最为深入、应用最为广泛的个性化推荐技术,它根据收集相似对象模式的不同,通常可以分为两种:基于内存的协同过滤和基于模型的协同过滤。前者应用较为普遍,但由于在使用过程中需要将用到的所有数据装入内存,所以它并不适用于超大规模数据的环境。同时根据相似对象类型的不同,基于内存的协同过滤又可以进一步细分为基于用户的协同过滤(User-based Collaborative Filtering,UCF),如用户间多相似度协同过滤推荐算法(Collaborative Filtering recommendation algorithm based on User's Multi-similarity, UMCF)[9];和基于项目的协同过滤(Item-based Collaborative Filtering, ICF),如基于项目属性和云填充的协同过滤推荐算法(Collaborative Filtering recommendation algorithm based on Item Attribute and cloud model filling,IACF)[10]。两者的主要区别在于,前者从用户的角度出发,采用相似度度量模型得到具有相似爱好或者兴趣的用户,该方法适用于用户数目变化不大、项目数量远多于用户的情况;而后者则从项目的角度出发,一般适用于用户数量非常多、而项目的数量相对用户数目较少的情况。与基于内存的协同过滤不同,基于模型的协同过滤[11]并不直接进行对象间的相似度计算,而是在某种统计模型或机器学习方法的基础上,将用户的项目评分编译成预测模型,再利用该预测模型为目标用户进行预测。目前,基于模型的协同过滤广泛使用的技术包括潜在语义检索、人工神经网络、贝叶斯网络模型、聚类等模型。

本文重点研究基于内存的协同过滤模型,下面将详细介绍其工作流程、在实际中遇到的一些问题及相应的解决方案。

1.1　评分矩阵

推荐系统中各种类型的数据很多,大致可以分为两类:1)隐式数据,如用户收藏、浏览日志等;2)显示数据,如用户和项目的属性表等,其中的核心数据是用户关于所有项目的评分数据,它可以用一个 $m \times n$ 维的矩阵 $\boldsymbol{R}$ 来表示,其中:m 和 n 分别为系统中用户和项目的数目;矩阵中的元素 $r_{v,i}$ 表示用户 v 对项目 i 的评价,通常用一个离散的或连续的数值表示,数值越大表示用户对项目的偏好程度越高。表 1 给出了一个评分矩阵 $\boldsymbol{R}$ 样例,其中的评价值为 1 ~ 5 的整数,元素值 $r_{v,i}$ = 0 表示用户未对项目进行评价。

表 1　评分矩阵样例

用户	项目					
	1	2	…	i	…	n
1	1	3	…	0	…	5
⋮	⋮	⋮		⋮		⋮
v	4	4	…	$r_{v,i}$	…	0
⋮	⋮	⋮		⋮		⋮
m	3	0	…	2	…	$r_{m,n}$

1.2　近邻集合

近邻集合是协同过滤模型中评分预测的基础,其质量的高低直接影响着预测的准确度,因此一直是各种模型围绕研究的焦点。以 UCF 为例,用户近邻集合的构造首先需要度量不同用户与目标用户之间的相似性,然后找出与目标用户 v 最为相似的 k 个用户来构造近邻集合 $K_v = \{v_1, v_2, \cdots, v_k\}$。常用的相似性度量模型有 4 种:余弦相似性、修正的余弦相似性、Pearson 相关相似性以及受约束的 Pearson 相关相似性模型。本文选择第 3 种模型,因为它在同等情况下具有更加优秀的性能[3],下面给出用户 u、v 的 Pearson 相关相似性表达式,如式(1):

$$sim(u,v) = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \overline{r_u})(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \overline{r_u})^2 \sum_{i \in I_{u,v}} (r_{v,i} - \overline{r_v})^2}} \tag{1}$$

其中:$I_{u,v} = \{i \in I \mid r_{u,i} \neq 0 \wedge r_{v,i} \neq 0\}$ 表示用户 u、v 的公共评分项目集合;$\overline{r_u}$ 和 $\overline{r_v}$ 为用户的评分均值;$sim(u,v)$ 的取值范围为[-1,1],正值表示正相关性,值越大表明相似性程度越高,负值表示负相关性。不过通常的研究仅考虑相似性值大于等于 0 的用户,值越大表明用户的兴趣偏好愈有可能趋于相同,0 表示没有相似性,本文也采取这种做法。

1.3　预测与推荐生成

基于近邻集合 K_v,可以利用 K_v 中的相似用户为目标用户未评分的项目进行评分预测。方法如下:针对目标项目 i 即需要获得预测的项目,首先在 K_v 中选取对其进行过评价的用户构造用户近邻子集合 $k_v = \{x \in K_v \wedge r_{x,i} \neq 0\}$;然后利用预测函数以相似性值 $sim(u,v)$ 为权重,加权它们关于项目 i 的评分;最终生成用户 v 关于 i 的预测评分 $P_{v,i}$。常用的预测函数有三种:均值函数、权重函数和改进型的权重函数,其中最后一种函数模型在通常情况下性能较优且应用较为广泛[3,9-10],因此选择它作为本文的预测函数,表达式如式(2)所示:

$$P_{v,i} = \overline{r_v} + \frac{1}{\sum_{x \in k_v} sim(x,v)} \sum_{x \in k_v} sim(x,v)(r_{x,i} - \overline{r_x}) \tag{2}$$

其中:$sim(x,v)$ 为通过各种相似性度量模型获得的用户相似性,Pearson 相关相似性是其中之一。显然 $P_{v,i}$ 存在的条件是 k_v 不为空集;否则无法为 i 进行评分预测。

这样进行完用户 v 关于所有未评分项目的评分预测,最后选取预测评分值最高的 N 个项目向 v 进行推荐,完成系统整个的预测与推荐过程。ICF 的预测与推荐过程与 UCF 大致相同,限于篇幅,这里不再赘述。

1.4　问题与解决方案

协同过滤推荐模型相比其他模型虽然拥有更加广泛的应用空间,但在实际应用中还是会遇到不少问题,例如:

1)数据稀疏性与冷启动问题。推荐系统中用户和项目的数量巨大,而每个用户关于项目的评分数目又很少,相对比例不足 1%,这样便产生了评分数据的稀疏性问题。对此,一些研究成果[12]首先利用其他方法如均值模型来对未评分项目进行评分预测,然后将获得的预测评分填充到原始矩阵中,在此基础上进行项目预测与推荐,可以较大程度上缓解数据稀疏性问题。此外,随着系统中新用户、新项目的不断增加,由于关于它们没有任何评分记录,这样在利用传统基于评分的协同过滤模型进行推荐时便出现了问题,即所谓的冷启动问题[8]。对此通常的解决方案是采用基于用户、项目属性的评分预测模型,因为在此情况下,用户项目的属性资料是已知

的且容易获得,所以可以利用这些信息来构造近邻集合。

2)相似性度量问题。通常的 UCF 和 ICF 模型主要基于公共评分来度量对象相似性,事实上这样做具有很多的局限性。从 UCF 的角度来说,受数据稀疏性的影响,相似性度量的偶然性因素是很大的,例如通过两三个相同的评分就得出 $sim(u,v)=1$ 的结果显然是不合理的。很多学者也注意到了这一问题,并对此提出了若干解决思路,典型的如公共评分对象数目阈值法[13]和 Gaussian 相似度支持度模型[14],不过本文将采用另外一种方法。其实从项目属性出发同样是一个有意义的尝试,因为项目的属性特征具有自然稳定性的特点,不过通常的研究成果仅是研究基于项目属性的项目相似性[10,15],并未真正发掘出项目属性对于用户相似性的影响和意义。本文正是基于这一点提出了新的协同过滤推荐模型,实验结果证明了所提模型的研究价值。

2 考虑项目属性的协同过滤推荐模型

从用户的角度,本文将结合项目属性介绍两种考虑项目属性的协同过滤推荐模型,以较好地应对在传统协同过滤模型中所遇到的相似性度量问题。

2.1 基于评分的用户相似性

Pearson 相关相似性基于公共项目评分来计算用户间的相似性值,一般来说,公共评价项目数目越多这样获得的相似性越可靠。但是在 Pearson 相似性的度量过程中并未反映这一点,于是在评分相似或相同的情况下,便出现了公共项目数目很小而相似性值却很高的不合理现象。因此,在计算用户相似性的同时,还必须考虑公共评价项目数目对相似性的影响作用。通常的做法之一[13]是取一个整数阈值 μ,当公共评价项目数目超过 μ 时,相似性值保持不变;否则相似性值会随公共评价项目数目的减小而减小,以 $sim(u,v)$ 为例,如式(3):

$$sim(u,v)=\frac{\min(\mu,|I_{u,v}|)}{\mu}\times sim(u,v) \tag{3}$$

式(3)中的做法存在一些问题,主要在于阈值 μ 值需要预先设定,而这在一般情况下是比较困难的。为此本文引入 Jaccard 系数来自适应调节用户间的相似性值,如用户 u、v 的 Jaccard 系数值如式(4)所示:

$$Jaccard(u,v)=\left|r_u\cap r_v\right|/\left|r_u\cup r_v\right| \tag{4}$$

其中:r_u 和 r_v 分别为用户的评分项目集合,| | 表示集合中的项目数目。可以看出,$Jaccard(u,v)$ 的取值范围为[0,1],当两个用户拥有完全相同的评分项目集合时,值为 1;而当拥有完全不同的项目集合时,值为 0。它较好地反映了用户在评价项目方面的重叠情况,并且可以与 $sim(u,v)$ 相结合达到修正 $sim(u,v)$ 的效果,最终获得更加准确的用户相似性,用 $sim_r(u,v)$ 表示,如式(5):

$$sim_r(u,v)=Jaccard(u,v)\times sim(u,v) \tag{5}$$

2.2 基于项目属性的用户相似性

通过评分固然可以度量用户相似性且简洁明了,但用户的相似性却不仅限于此,它更深层次地反映在项目的类型上。不同类型的项目具有不同的主题和功能,它能够首先满足用户的偏好取向,获得用户的关注,在此基础上用户才会进一步去了解它最终给出一个评价,评价高的表示认可。反过来同样好理解,大多数人在不同时期里的兴趣偏好大致是稳定的,这样它自然会对与之偏好相一致的一个或若干个类型的项目表示关注并给出自己的评价,评价的多少不论高低表明了关注的程度。因此可以从项目的类型入手来深挖用户之间的相似性,这一点合理且在现有的条件下容易实现。

项目在录入到系统中的数据库时,会登记项目的基本属性信息,然后按照分类标准将不同的项目归入到不同的类型中,便于管理且方便用户浏览。因此通过对项目属性数据的整理,可以很容易地获得一张项目属性表,如表 2。

表 2 项目属性表样例

项目	属性					
	a_1	a_2	…	a_l	…	a_p
1	1	0	…	0	…	1
2	1	0	…	1	…	0
⋮	⋮	⋮		⋮		⋮
n	0	1	…	0	…	1

表 2 列出了 n 个项目其中的 p 个属性,属性值均用 0 或 1 表示,1 表示项目具有该属性;0 表示没有。这种属性的数值化表示方法在程序上很好实现,而且可以简化系统对于不同类型项目的管理。

以项目属性表为基础,本文可以统计出用户在不同类型项目上的评价次数,评价次数越多表明用户对于该类项目越关注,从而可以了解用户大致的兴趣偏好。这样获得不同用户关于不同类型项目的评价次数向量,其维度与项目的属性个数相同均为 p,例如对于用户 u、v 来说:

$$\begin{cases}\boldsymbol{A}_u=(f_{u,1},f_{u,2},\cdots,f_{u,p})\\ \boldsymbol{A}_v=(f_{v,1},f_{v,2},\cdots,f_{v,p})\end{cases} \tag{6}$$

其中元素 $f_{u,1}$ 为用户 u 在项目第 1 个属性上的评价次数。但这样做存在不妥,因为绝对的评价次数大小并不足以表明一个用户对某类项目的关注程度。为克服这一问题,本文采用相对评价次数,即对 $\boldsymbol{A}_u$ 向量进行归一化处理,例如对于 $f_{u,1}$:

$$f_{u,1}'=f_{u,1}/(f_{u,1}+f_{u,2}+\cdots+f_{u,p})\in[0,1] \tag{7}$$

这样得到新的用户评价项目次数向量:

$$\begin{cases}\boldsymbol{A}_u'=(f_{u,1}',f_{u,2}',\cdots,f_{u,p}')\\ \boldsymbol{A}_v'=(f_{v,1}',f_{v,2}',\cdots,f_{v,p}')\end{cases} \tag{8}$$

对于两个向量,如何度量两者之间的相似性,目前较为成熟的处理方法是式(1)中的 Pearson 相关相似性法用 $\Delta A_1(u,v)$ 表示,表达式如式(9):

$$\Delta A_1(u,v)=\frac{\sum_{i=1}^{p}(f_{u,i}'-\overline{f_u})(f_{v,i}'-\overline{f_v})}{\sqrt{\sum_{i=1}^{p}(f_{u,i}'-\overline{f_u})^2\sum_{i=1}^{p}(f_{v,i}'-\overline{f_v})^2}} \tag{9}$$

其中 $\overline{f_u}$ 和 $\overline{f_v}$ 为分别为用户相对评价次数均值。

除式(9)中的处理方法外,近年来流行的“云模型”同样可以为本文提供一种好的处理思路。“云模型”通过期望 E_x、熵 E_n 和超熵 H_e 三个数字特征来描述一个数据统计对象 $\boldsymbol{D}$,记为 $\boldsymbol{D}(E_x,E_n,H_e)$,称为 $\boldsymbol{D}$ 的特征向量。借助该模型,每个用户的向量 $\boldsymbol{A}_u'$ 可以视为一朵“云”,其中每个非零元素被视为“云”中一个“云滴”,这样可以利用逆向云生成算法实现每朵“云”从定量值到云的特征向量值的转化,具体转化公式见文献[10],这里不多作介绍。

通过云模型获得向量 $\boldsymbol{A}_u'$ 的云特征向量 $\boldsymbol{D}_u(E_x,E_n,H_e)$ 后,便可以利用余弦相似性来度量两个向量之间的相似性,用 $\Delta A_2(u,v)$ 表示,如式(10):

$$\Delta A_2(u,v)=\frac{\boldsymbol{D}_u\cdot\boldsymbol{D}_v}{\|\boldsymbol{D}_u\|\,\|\boldsymbol{D}_v\|}\in[-1,1] \tag{10}$$

式(9) ~ (10) 中的 $\Delta A_1(u,v)$、$\Delta A_2(u,v)$ 同样面临评价的公共项目类型数目很少而相似性值却很高的问题,受前面的 Jaccard 系数启发,定义基于用户评价项目次数的 Jaccard 系数表达式,如式(11):

$$Jaccard_A(u,v) = \frac{|\boldsymbol{A}_u''\&\boldsymbol{A}_v''|}{|\boldsymbol{A}_u''|\boldsymbol{A}_v''|} \in [0,1] \tag{11}$$

其中:$\boldsymbol{A}_u''$ 和 $\boldsymbol{A}_v''$ 分别为与向量 $\boldsymbol{A}_u'$、$\boldsymbol{A}_v'$ 对应的布尔向量,1 表示所对应的元素不为 0,否则为 0;"&" 和"|" 分别为向量二进制位运算的"按位与" 和"按位或" 运算符;"| |" 用来统计运算结果中元素为 1 的数目。

$Jaccard_A(u,v)$ 反映了用户间评价的项目在类型方面的重叠情况,它可以与 $\Delta A_j(u,v)\{j=1,2\}$ 结合获得更加准确的基于项目属性的用户相似性,用 $sim_j(u,v)\{j=1,2\}$ 表示,取值范围定为[0,1],如式(12):

$$sim_j(u,v) = \Delta A_j(u,v) \times Jaccard_A(u,v) \tag{12}$$

2.3　用户相似性

式(5) 中的 $sim_r(u,v)$ 和式(12) 中 $sim_j(u,v)\{j=1,2\}$ 分别从不同角度度量了用户间的相似性,综合两方面便可以获得更为全面的用户相似性度量模型,用 $Sim_j(u,v)\{j=1,2\}$ 表示,如式(13):

$$Sim_j(u,v) = \lambda_j \times sim_j(u,v) + (1-\lambda_j) \times sim_r(u,v) \tag{13}$$

其中 λ_j 为平衡因子用作协调两方面相似性度量的结果,取值范围为[0,1]。关于 λ_j 的取值,通常的做法[10] 是在[0,1] 中取一系列值观察不同的 λ_j 值对预测准确度的影响,然后从中选择较优的值。这种做法仅适合于研究,在实际运行中并不可取,因为随着条件的改变需要不停的设值。

这里本文采用自适应平衡因子模型,因为它可以动态结合 $sim_r(u,v)$ 和 $sim_j(u,v)\{j=1,2\}$,使最终的用户相似性达到一个良好的平衡,从而为用户近邻选择提供较好的依据,具体表达式如式(14):

$$\lambda_j = \frac{sim_j(u,v)^2}{sim_j(u,v)^2 + sim_r(u,v)^2};\ j=1,2 \tag{14}$$

可以看出,当 $sim_j(u,v)$ 为 0 时,λ_j 为 0,此时完全按照 $sim_r(u,v)$ 来计算用户相似性;反之当 λ_j 为 1 时,则完全按照项目属性相似性来计算用户相似性。

根据经验,不同用户的 $sim_j(u,v)$ 不同,它对最终用户相似性的影响也应该不同,λ_j 的值反映了这一点。而且当 $sim_j(u,v)$ 较高时,说明用户此时在项目属性方面具有更高的相似性,用户间的相似性应该更多地从这方面来考虑,λ_j 的值同样反映了这一点,从而相当程度上说明了 λ_j 取值的合理性。

接下来,本文便可以根据相似性值选取相似性较大的用户为目标用户 v 构造近邻集合 K_v,然后通过1.3 节中介绍的预测与推荐过程进行评分预测与项目推荐。

2.4　时间复杂度分析

关于模型的时间复杂度,应该以预测与推荐为界限,划分为两个阶段。

1) 离线阶段。在 $m \times n$ 维的矩阵 $\boldsymbol{R}$ 中,对于两个用户来说,$sim_r(u,v)$ 模型需要 $c_1 n$ 次的加法、乘法操作来获得两个用户的相似性。而在 $sim_j(u,v)$ 模型中,它首先需要进行 $c_2 n$ 次的加法乘法来获得用户基于项目属性的评价次数向量 $\boldsymbol{A}_u'$ 和 $\boldsymbol{A}_v'$,然后通过 $c_3 p$ 次操作得到与 $\boldsymbol{A}_u'$、$\boldsymbol{A}_v'$ 对应的 $\boldsymbol{A}_u''$ 和 $\boldsymbol{A}_v''$,以及 $c_4 p$ 次操作得到 $Jaccard_A(u,v)$。接下来的分析应该分为两部分:1) $\Delta A_1(u,v)$ 模型需要进行 $c_5 p$ 次的操作;2) $\Delta A_2(u,v)$ 模型首先需要进行 $c_6 p$ 次的操作来获得 $\boldsymbol{D}_u(E_x,E_n,H_e)$ 和 $\boldsymbol{D}_v(E_x,E_n,H_e)$,然后进行 $c_7 p$ 次的操作获得相似性值 $\Delta A_2(u,v)$,其中 $c_i\{i=1,2,\cdots,7\}$ 为某一常数。因此关于 $Sim_1(u,v)$ 模型,此时加乘法操作总数为 $c_1 n + c_2 n + c_3 p + c_4 p + c_5 p = (c_1+c_2)n + (c_3+c_4+c_5)p$。同理,$Sim_2(u,v)$ 模型的操作数为 $(c_1+c_2)n + (c_3+c_4+c_6+c_7)p$。由于系统中 $p \ll n$,因此上述的操作总数为可以统一为 $c_0 n$,c_0 为一常数,两种模型的时间复杂度为 $O(n)$。关于用户间相似性的对称性问题,无论相似性是否具有对称性,均按照不对称性原则来计算,这样可以获得最大的计算量,便于统计。这样由于需要计算系统中所有用户之间的相似性,总共需要计算 $m(m-1)$ 次,因此此阶段模型的复杂度为 $O(m^2 n)$。该阶段需要花费很长的时间,尤其是在 m、n 比较大时,不过庆幸的是,这些操作可以在后台预先完成。

2) 在线阶段即预测与推荐阶段。根据用户相似性值的大小选取一定数量 T(T 为常数且 $T \ll m$) 的用户为目标用户构造近邻集合,然后预测函数通过常数次的加法乘法操作计算获得项目的预测评分,这个过程最多需要运行 n 次,所以此阶段模型的时间复杂度为 $O(n)$。该阶段的时间花费很小却对用户来说非常敏感,在实际中这段时间即为用户为获得推荐项目而需要在线等待的时间,后文会专门对这段时间进行记录以比较不同推荐模型的时间性能。

3　实验评估

3.1　评价标准

推荐系统中,关于性能评价的标准通常分为两类:1) 决策支持精度标准,如决策树过程;2) 统计精度标准,如平均绝对误差(Mean Absolute Error, MAE)、均方根误差(Root Mean Square Error, RMSE) 等。其中 MAE 比较直观且应用最为广泛,所以本文选用它作为各种推荐模型性能比较的评价标准。MAE 的原理是统计用户关于项目的实际评分与预测评分之间的绝对差值来直接衡量模型评分预测的准确性,其值越小表明预测准确性越高。假设系统中项目的预测评分集合为 $\{P_1,P_2,\cdots,P_q\}$,q 为评分数目,而对应的实际项目评分集合为 $\{r_1,r_2,\cdots,r_q\}$,则 MAE 的表达式如式(15):

$$MAE = \frac{1}{q}\sum_{i=1}^{q} |r_i - P_i| \tag{15}$$

3.2　实验数据集

实验采用两个著名的电影评分数据集:一个是大家目前广泛采用的 Movielens 100k,里面包含了 943 个用户关于1 682 部电影的100 000个匿名评分;另一个是 Movielens 1M,里面包含了6 040个用户关于3 706部电影的1 000 209个评分。两个数据集均由美国 Minnesota 大学 GroupLens 研究项目组提供(http://www.grouplens.org),供学习研究推荐系统使用。数据集中每个用户至少对其中的 20 部电影进行了评分,评分范围为[1,5]区间的整数:5 评价最高,1 评价最低,0 表示未给出评分。除此之外,数据集中还包含了用户的特征信息,如用户的性别、年龄、职业等,以及电影的属性信息,如上映时间、风格等,其中电影风格有科幻、冒险、动作、喜剧等,这些信息会在本文实验中用到。

需要说明的是,本文实验引入 Movielens 1M 的目的在于:1) 数据集中包含了电影的属性信息,同样可以满足实验中模型对于项目属性信息的需求;2) 数据量更大,可以更充分地

对模型的性能进行验证。在具体的实验中,本文会将整个的评分数据集划分为两部分,其中的 80% 用作训练集,主要用于构建模型,剩下的 20% 用作测试集,用于验证模型的实际性能。

3.3 比较实验

为了验证推荐模型在不同用户近邻数目条件下的具体性能,本节选取 UMCF[9] 和 IACF 模型[10] 与本文提出的相似性模型:Sim_1 和 Sim_2,分别基于数据集 Movielens 100k 和 Movielens 1M 进行实验比较,如图 1 ~2。

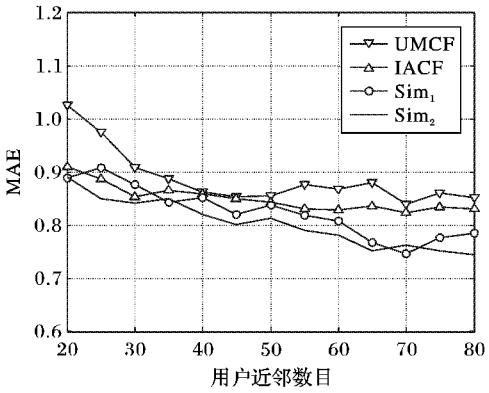

图 1 基于 MovieLens 100k 的性能比较

通过图 1 可以看出,Sim_1 和 Sim_2 则均表现出了不错的性能,改进程度在 5% 左右,主要原因在于 $\Delta A_1(u,v)$ 利用 Pearson 相关相似性和 $\Delta A_2(u,v)$ 利用云模型较好地度量了用户基于项目属性的相似性,并基于 $Jaccard_A(u,v)$ 对相似性作了优化,从而能够与优化的 $sim_r(u,v)$ 结合获得更加准确的用户相似性,最终提高了模型的 MAE 性能。这其中自适应平衡因子的作用不容忽视,IACF 模型的大致出发点与本文类似,却需要不断地根据实验条件来设置经验值,这会在很大程度上影响模型的性能,尤其是在实验条件不断变化的情况之下。相比之下,UMCF 模型仅通过无优化的 Pearson 相似性和余弦相似性模型来获得用户间的相似性,其性能表现就要更差一些。

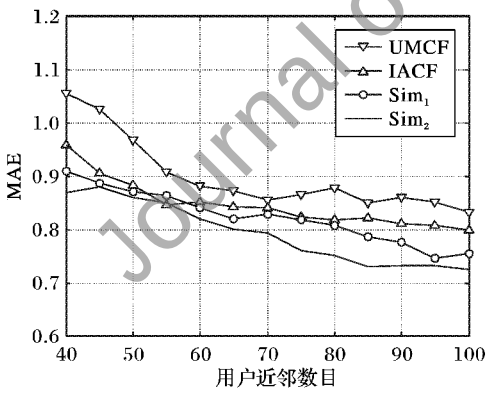

图 2 基于 MovieLens 1M 的性能比较

图 2 的实验结果表明,在数据量更大的情况下,各种模型的性能比较结果与图 1 相比并没有大的改变,Sim_1 和 Sim_2 模型仍旧拥有相当的性能优势,只是表现出了些许小变动:IACF 与 UMCF 模型的性能差距变得更加明显。这种情况应该与 IACF 中的云模型评分填充密切相关,因为随着数据量的增大,云模型能够更准确地获得关于用户评分的云特征向量,从而能够较好地度量项目之间的相似性,获得更准确的评分预测进行评分填充。而同样情况下,UMCF 模型却不能充分利用数据量增大带来的好处。

3.4 运行时间比较

本节基于两个数据集,分别从测试集中随机选取 80% 的评分数据对各种模型在线阶段的时间花费进行记录,记录次数为 10 次,对记录结果取均值作为模型的运行时间,如表 3 所示。

表 3 运行时间比较 s

数据集	UMCF 模型	IACF 模型	Sim_1 模型	Sim_2 模型
MovieLens 100k	31.5	19.1	22.3	20.7
MovieLens 1M	273.4	210.4	207.5	212.8

表 3 的数据显示,相同数据集中不同模型的运行时间除了 UMCF 模型以外,基本上都相差不大。这是因为对于不同模型来说,用户或者项目的相似性值都是预先计算出来的,在评分预测阶段仅需要提取出来即可。UMCF 模型的特殊之处在于,它的用户相似性是以项目属性为标准分开存放的,因此需要在不同的用户表之间来回切换,从而相当程度上增加了运行时间。总的来说,本文模型的时间花费基本上还是较为合理的。

4 结语

本文分析了传统协同过滤推荐模型的工作原理及通常面临的问题,提出了两种考虑项目属性的协同过滤推荐模型,验证实验基于 Movielens 100k 和 Movielens 1M 数据集进行。实验结果表明,与现有基于用户和基于项目的改进型协同过滤模型相比,本文模型具有预测准确度方面的性能优越性。下一步的研究工作是建立一种合理的评分预测模型来对评分矩阵进行预填充,以较好地缓解数据稀疏性问题,提高模型的预测推荐质量。此外,推荐系统除评分数据外还包含很多隐性数据,通过整理和分析这些数据来对当前各种基于用户和基于项目的协同过滤模型进行改进,同样是一个很有前景的研究方向。

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[J]. 自然科学进展,2009,19(1):1 -15.

[2] MARSHALL M. Aggregate knowledge raises $5M from Kleiner, on a roll [EB/OL]. (2006-11-10)[2013-05-20]. http://venturebeat.com/2006/12/10/aggregate-knowledge-raises-5m-from-kleiner-on-a-roll.

[3] BOBADILLA J, ORTEGA F, HERNANDO A. A collaborative filtering similarity measure based on singularities [J]. Information Processing and Management, 2011, 48(2): 204 -217.

[4] NADSCHLAGER S, KOSORUS H, BOGL A, *et al.* Content-based recommendations within a QA system using the hierarchical structure of a domain-specific taxonomy [C]// Proceedings of the 23rd International Workshop on Database and Expert Systems Applications. Washington, DC: IEEE Computer Society, 2012: 88 -92.

[5] JIA C X, LIU R R, SUN D, *et al.* A new weighting method in network-based recommendation [J]. Physica A—Statistical Mechanics and Its Applications, 2008, 387(23): 5887 -5891.

[6] CHO Y S, MOON S, RYU K H. Mining association rules using RFM scoring method for personalized u-commerce recommendation system in emerging data [C]// Proceedings of the 2012 International Conference on Computer Applications for Modeling, Simulation, and Automobile. Berlin: Springer, 2012: 190 -198.

[7] 许海玲,吴潇,李晓东,等. 互联网推荐系统比较研究. 软件学报[J]. 软件学报,2009,20(2):350 -362.

(下转第 3106 页)

是影响算法的计算效率的一个重要因素。在本算法中,由于PSAGSO 算法采用自适应的方法来动态调整比例参数,使群组在进化过程中能够在一个较好解的附近进行搜索,扩大了搜索空间,提高了算法加速比,提升了算法的收敛速度。与其他五种算法相比,PSAGSO 算法在解决高维复杂数值优化问题具有良好的性能和竞争力。最后,通过对 PSAGSO 算法加速比对比分析,进一步说明了倒序变异机制和动态自适应机制的引入使 GSO 算法的性能得到了一定程度的提高。

6 结语

PSAGSO 算法在发现者的进化过程中加入预选择机制的小生境实现方法,可以动态自适应分配追随者和游荡者的分布比例,避免了根据人为经验设定算法参数,经过单峰、多峰、高维函数的不同测试,表明对算法的改进是有效的,特别是对多峰高维函数的优化问题效果明显。进一步对 GSO 算法的研究及分析表明:影响该算法的一些重要参数选取和设置,对算法的性能有较大的影响。因此,在优化问题时,如何自适应地调整算法参数选取,使其能够更有效地指导算法的进行;同时,根据智能综合集成的观点,改进算法的相应算子,提高算法的搜索能力;在未来的研究将考虑实际路径优化问题的特性,推广该算法在解决实际应用问题的能力。

参考文献:

[1] COLORNI A, DORIGO M, MANIEZZO V, *et al.* Distributed optimization by ant colonies [C]// Proceedings of the 1st European Conference on Artificial Life. Amsterdam: Elsevier, 1991: 134 – 142.

[2] KARABOGA D. An idea based on honey bee swarm for numerical optimization [R]. Kayseri: Erciyes University, 2005.

[3] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceeding of IEEE International Conference on Neural Networks. Piscataway: IEEE Press, 1995: 1942 – 1948.

[4] HE S, WU Q H, SAUNDERS J R. A novel group search optimizer inspired by animal behavioral ecology [C]// Proceedings of the 2006 IEEE Congress on Evolutionary Computation. Piscataway: IEEE Press, 2006: 1272 – 1278.

[5] HE S, WU Q H, SAUNDERS J R. Group search optimizer: an optimization algorithm inspired by animal searching behavior [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 973 – 990.

[6] CHEN D B, WANG J T, ZOU F, *et al.* An improved group search optimizer with operation of quantum-behaved swarm and its application [J]. Applied Soft Computing, 2012, 12(2): 712 – 725.

[7] FANG J Y, CUI Z H, CAI X J, *et al.* A hybrid group search optimizer with metropolis rule [C]// Proceeding of 2010 International Conference on Modelling, Identification and Control. Piscataway: IEEE Press, 2010: 556 – 561.

[8] 姚健. 群搜索算法与二次插值法的混合算法及其应用研究[D]. 太原: 太原科技大学, 2010.

[9] 罗磊, 谢静, 周晖. 一种新的群搜索优化实现算法[J]. 南通大学学报: 自然科学版, 2012, 11(2): 1 – 8.

[10] 刘锋, 覃广, 李丽娟. 快速群搜索优化算法及其应用研究[J]. 工程力学, 2010, 27(7): 38 – 44.

[11] 汪慎文, 丁立新, 谢大同. 应用反向学习策略的群搜索优化算法[J]. 计算机科学, 2012, 39(9): 183 – 187.

[12] KANG Q, LAN T, YAN Y, *et al.* Group search optimizer based optimal location and capacity of distributed generations [J]. Neurocomputing, 2012, 78(1): 55 – 63.

[13] ZARE K, HAQUE M T, DAVOODI E. Solving non-convex economic dispatch problem with valve point effects using modified group search optimizer method [J]. Electric Power Systems Research, 2012, 84(1): 83 – 89.

[14] XIE H B, LIU F, LI L J, *et al.* Research on topology optimization of truss structures based on the improved group search optimizer [C]// Proceedings of the 2nd International Symposium on Computational Mechanics and the 12th International Conference on the Enhancement and Promotion of Computational Methods in Engineering and Science. Melville: AIP, 2010: 707 – 712.

[15] ZHONG G Q, LIU F. Optimal design of plate structures with discrete variables by group search optimizer [J]. Advanced Science Letters, 2011, 4(3): 1057 – 1061.

[16] HE S, WU Q H, SAUNDERS J R. Breast cancer diagnosis using an artificial neural network trained by group search optimizer [J]. Transactions of the Institute of Measurement and Control, 2009, 31(6): 517 – 531.

[17] ZENG S K, LI L J. The particle swarm group search optimization algorithm and its application on structural design [J]. Advanced Science Letters, 2011, 4(3): 900 – 905.

[18] WANG L, ZHONG X, LIU M. A novel group search optimizer for multi-objective optimization [J]. Expert Systems with Applications, 2012, 39(3): 2939 – 2946.

[19] CAVICCHIO D J. Reproductive adaptive plans [C]// Proceedings of the Association for Computing Machinery Annual Conference. New York: ACM Press, 1972: 60 – 70.

[20] CAVICCHIO D J. Adaptive search using simulated evolution [D]. Ann Arbor: University of Michigan, 1970.

[21] 江巧永, 高岳林. 融合差分进化和倒序变异扩展蚁群算法[J]. 计算机应用, 2010, 30(9): 2283 – 2285.

[22] YAO X, LIUY, LIU G. Evolutionary programming made faster [J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82 – 102.

[23] YAO X, LIU Y. Fast evolution strategies [C]// Proceedings of the 6th International Conference on Evolution Programming. Berlin: Springer-Verlag, 1997: 151 – 161.

[24] RAHNAMAYAN S, TIZHOOSH H R, SALAMA M M A. Opposition-based differential evolution [J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 64 – 79.

(上接第 3066 页)

[8] 孙冬婷, 何涛, 张福海. 推荐系统中的冷启动问题研究综述[J]. 计算机与现代化, 2012(5): 59 – 63.

[9] 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.

[10] 孙金刚, 艾丽蓉. 基于项目属性和云填充的协同过滤推荐算法[J]. 计算机应用, 2012, 32(3): 658 – 660.

[11] 董文远. 基于混合过滤的推荐系统开发研究[D]. 长春: 吉林大学, 2011.

[12] 刘庆鹏, 陈明锐. 优化稀疏数据集提高协同过滤推荐系统质量的方法[J]. 计算机应用, 2012, 32(4): 1082 – 1085.

[13] 黄创光, 印鉴, 汪静, 等. 不确定近邻的协同过滤推荐算法[J]. 计算机学报, 2010, 33(8): 1369 – 1377.

[14] 罗辛, 欧阳元新, 熊璋, 等. 通过相似度支持度优化基于 K 近邻的协同过滤算法[J]. 计算机学报, 2010, 33(8): 1437 – 1445.

[15] 吴月萍, 郑建国. 改进相似性度量方法的协同过滤推荐算法[J]. 计算机应用与软件, 2011, 28(10): 7 – 9.