Journal of Computer Applications
计算机应用,2013,33(12):3354-3358
ISSN 1001-9081
CODEN JYIIDU
2013-12-01
http://www.joca.cn

文章编号:1001-9081(2013)12-3354-05
doi:10.11772/j.issn.1001-9081.2013.12.3354

综合用户和项目预测的协同过滤模型

杨兴耀[1*],于　炯[1,2],吐尔根·依布拉音[1],廖　彬[1,2]
(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:针对基于用户和基于项目的协同过滤模型存在推荐质量不高等问题,提出一种综合用户和项目预测的协同过滤模型。该模型同时考虑用户和项目两方面,首先对性能优秀的相似性模型进行自适应的优化;然后根据相似性值分别选取相似用户和相似项目为目标对象构造近邻集合,并利用预测函数得到基于用户和基于项目的预测结果;最后通过自适应平衡因子的协调处理获得最终预测结果。比较实验在不同的评估标准下进行,结果表明,与目前典型的模型如RSCF、HCFR和UNCF相比,新提出的协同过滤模型不仅在项目预测准确性方面拥有出色的表现,而且在推荐准确性和全面性方面同样表现优秀。

关键词:推荐系统;协同过滤;近邻集合;相似性模型;平均绝对偏差
中图分类号:TP311.13　　**文献标志码**:A

Collaborative filtering model combining users' and items' predictions

YANG Xingyao[1*], YU Jiong[1,2], TURGUN Ibrahim[1], LIAO Bin[1,2]
(1. *School of Information Science and Engineering, Xinjiang University, Urumqi Xinjiang* 830046, *China*;
2. *School of Software, Xinjiang University, Urumqi Xinjiang* 830008, *China*)

Abstract: Concerning the poor quality of recommendations of traditional user-based and item-based collaborative filtering models, a new collaborative filtering model combining users' and items' predictions was proposed. Firstly, it considered both users and items, and optimized the similarity model with excellent performance dynamically. Secondly, it constructed neighbor sets for the target objects by selecting some similar users and items according to the similarity values, and then obtained the user-based and item-based prediction results respectively based on some prediction functions. Finally, it gained final predictions by using the adaptive balance factor to coordinate both of the prediction results. Comparative experiments were carried out under different evaluation criteria, and the results show that, compared with some typical collaborative filtering models such as RSCF, HCFR and UNCF, the proposed model not only has better performance in prediction accuracy of items, but also does well in the precision and recall of recommendations.

Key words: recommender system; collaborative filtering; neighbor set; similarity model; Mean Absolute Error (MAE)

0　引言

目前推荐系统的应用领域已非常广泛,尤其是Web 2.0技术的成熟,推荐被作为一个独立的概念提出来并得到深入研究[1]。根据推荐信息产生原理的不同,推荐模型可以分为多种[2-5],其中研究最为深入并取得长足进展的是协同过滤推荐模型,它开始于1992年提出的Goldberg系统,目前已在各大推荐系统中得到广泛应用[6]。这主要是由于它能够利用各种性能优秀的相似性分析技术来更好地寻找与目标用户有着相同或者相似兴趣爱好的用户,然后根据这些用户的兴趣爱好来向目标用户进行信息推荐。

但通常的协同过滤模型往往只考虑单一群体的影响,忽视了其他关联群体的作用,如基于用户的协同过滤[7]和基于项目的协同过滤[8],预测信息不够全面,这在一定程度上影响了最终的推荐质量。针对此问题,目前多数的研究者一般倾向于两种解决途径:1)混合过滤模型[9],即将不同的推荐模型通过某种方式与协同过滤模型结合起来,以弥补各自的缺陷;2)综合不同的数据源进行协同过滤,从而获得较好的推荐效果,如文献[10]中基于评分支持度的最近邻协同过滤推荐算法(Collaborative Filtering Recommendation Algorithm Based on Nearest-neighborhood and Rating Support,RSCF)利用评分支持度,来自适应地调节基于用户和基于项目两方面的评分预测结果获得最终推荐。同样的还有一种综合用户和项目因素的协同过滤推荐算法(Collaborative Filtering Recommendation Algorithm Based on Both User and Item,HCFR)[11],以及不确定近邻的协同过滤(Uncertain Neighbors' Collaborative Filtering,UNCF)推荐算法[12]等。

为了获得用户和项目两方面的统计信息,本文首先利用性能优秀的相似性模型来度量对象之间的相似性,然后基于预测函数分别获得来自于用户方面和项目方面的预测结果,整个计算过程没有预先设置其他参数。在此基础上,本文利用自适应平衡因子综合了两方面的预测结果,提出了综合用户和项目预测的协同过滤模型(Collaborative filtering Model combining Users' and Items' predictions,CMUI),介绍了系统的

收稿日期:2013-07-10;**修回日期**:2013-08-14。　**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)。

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、云计算、可信计算;　于炯(1964-),男,新疆乌鲁木齐人,教授,博士,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,主要研究方向:自然语言处理、软件工程;　廖彬(1986-),男,四川内江人,博士研究生,主要研究方向:网格计算、云计算。

预测与推荐过程。

1　问题描述

1.1　用户—项目评价数据

推荐系统的评价数据中包含 m 个用户和 n 个项目,这些用户和项目分别构成一个用户集合 $U = \{user_1, user_2, \cdots, user_m\}$ 和一个项目集合 $I = \{item_1, item_2, \cdots, item_n\}$,其中用户关于项目的评价可以看成一个 $m \times n$ 维的用户—项目评分矩阵 $\boldsymbol{R}(m,n)$,见表 1。

表 1　评分矩阵

用户	项目			
	$item_1$	$item_2$	…	$item_n$
$user_1$	$r_{1,1}$	$r_{1,2}$	…	$r_{1,n}$
$user_2$	$r_{2,1}$	$r_{2,2}$	…	$r_{2,n}$
…	…	…		…
$user_m$	$r_{m,1}$	$r_{m,2}$	…	$r_{m,n}$

矩阵 $\boldsymbol{R}(m,n)$ 中的元素如 $r_{2,2}$ 表示用户 2 对项目 2 的评价值,没有评价值则用 $r_{2,2}$ = null 表示,评价值的大小体现了用户对该项目感兴趣的程度,通常值越大,表明用户对项目感兴趣的程度越高。

1.2　相似性度量模型

为方便说明,本文将需要获得项目推荐的用户 u 称为目标用户,而将需要进行评分预测的项目 i 称为目标项目,两者合起来称为目标对象。

根据评分矩阵 $\boldsymbol{R}(m,n)$,可以利用不同的相似性模型获得对象之间的相似性。目前常用的模型有:皮尔逊相关模型(Pearson Correlation,PC)、余弦模型(Cosine,COS)、受约束的皮尔逊相关模型(Constrained Pearson Correlation,CPC)和斯皮尔曼等级相关模型(Spearman Rank Correlation,SRC),限于篇幅,这些模型的表达式不再列出,详见文献[13]。对象间相似性度量的准确性会直接关系到推荐模型的推荐质量,因此为了较好地度量对象之间的相似性,必须寻求一种性能优秀的相似性模型。PC 模型由于在同等的条件下拥有更好的性能(见图 1),因此本文选择 PC 作为用户间的相似性度量模型,表达式如下:

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

其中:$r_{u,i}$ 和 $r_{v,i}$ 为用户 u、v 关于项目 i 的评分;I_{uv} 表示用户的公共评价项目集合;$\overline{r_u}$ 和 $\overline{r_v}$ 为用户的评价项目均值。$PC(u,v)$ 的取值区间为[-1,1],值越大表明用户间的相似性程度越高,兴趣偏好愈加趋于一致,而当值小于等于 0 时,一般认为对象之间便没有什么相似性了。

从项目的角度,项目相似性反映了两个项目同时满足用户兴趣偏好的可能性大小,它通过公共评价用户的评分来体现,评分越相近表明这种可能性越大。因此,本文同样可以利用 PC 模型来度量项目之间的相似性,表达式如下:

$$PC(i,j) = \frac{\sum_{u \in U_{ij}} (r_{u,i} - \overline{r_i}) \times (r_{u,j} - \overline{r_j})}{\sqrt{\sum_{u \in U_{ij}} (r_{u,i} - \overline{r_i})^2} \sqrt{\sum_{u \in U_{ij}} (r_{u,j} - \overline{r_j})^2}} \tag{2}$$

其中:$r_{u,i}$ 和 $r_{u,j}$ 为项目 i、j 获得某用户 u 的评分;U_{ij} 表示项目共同获得用户评价的公共用户集合;$\overline{r_i}$ 和 $\overline{r_j}$ 为项目获得的评分均值。$PC(i,j)$ 的取值区间为[0,1],值越大表明项目间的相似性程度越高。

1.3　相似性度量的进一步优化

式(1)中的用户相似性计算取决于用户间的公共评分项目数目,由于实际推荐系统中评分数据的稀疏性问题,用户公共评分项目的数目差异非常大,这样相似性的计算便存在较大的偶然性。同时根据日常生活经验,用户的相似性并不能通过一两次的公共项目评分来衡量,因为即使两次的用户评分完全相同,此时相似性值非常大,也有可能出于偶然。因此,有必要利用公共评分项目数目来对用户相似性值作一定程度的改进。通常的处理方法是取一个整数常数阈值 μ,让它与公共评分项目数目 $|I_{uv}|$ 作比较,如下:

$$PC(u,v) = \frac{1}{\mu}\min(\mu, |I_{uv}|) \times PC(u,v) \tag{3}$$

这种方法存在不足之处(后文 3.3 节的实验会证明这一点),主要原因在于常数 μ 需要预先设定。为此,本文采用性能较好的 Jaccard 系数来对用户之间的相似性值进行优化,它反映了两个用户在评价项目属性上的重叠度,重叠度的大小反映了用户在该属性上的相似性。Jaccard 系数即为两个用户间的评价项目交集与评价项目并集的项目数目之比,表示为 $Jaccard(u,v)$:

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

其中 r_u 和 r_v 分别为用户 u、v 的评价项目集合。可以看出,当两个用户拥有完全相同的评价项目集合时,它们的 Jaccard 值等于 1;而当两个用户拥有完全不同的项目集合时,它们的 Jaccard 值为 0;其余情况介于两者之间。这样 Jaccard 系数便较好地反映了两个用户间的评价项目情况,并且可以与相似性模型结合,更好地度量用户间的相似性来为用户寻找近邻。最终,优化后的相似性模型如下:

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

同理,项目之间的相似性度量也存在同样的问题,即项目的相似性也不能通过一两次的公共用户评分来衡量。因此同样有必要利用 Jaccard 系数来对项目相似性值进行优化,这样得到优化后的项目相似性计算模型如下:

$$PC(i,j) = Jaccard(i,j) \times PC(i,j) \tag{6}$$

其中 $Jaccard(i,j)$ 为两个项目间的 Jaccard 系数值,它反映了两个项目在获得用户评价方面的相对差异程度。

1.4　对象近邻的选取

获得对象间的相似性后,便可以选取一定数目的相似对象为目标对象构造近邻集合。通常的选取标准是预先设定一个阈值常数,相似性大于此阈值的便被选取,否则不予选取。这种做法存在一些弊端:1)评价数据的稀疏性问题会导致对象的相似性度量存在很多偶然性;2)评价数据的大规模性使得不同对象的相似对象数目差异很大。这种情况下设定一阈值,既显得比较困难,同时也显得比较僵硬,难以反映实际数据的变化。

为此本文采用相似对象数目百分数 *Percentage* 作为对象近邻选取的标准,例如当 *Percentage* = 30% 时,便按照相似性大小选取前 30% 的相似对象来构成近邻集合。这样做的好处是:1)可以满足相似对象数目参差不齐的要求;2)适用于相似对象数目不高的情况;3)考虑了相似对象数目对预测结果的影响,因为在实际中,目标对象的相似对象越多,所提

供的推荐信息就越全面,这样获得的预测结果相对就越准确。

根据 *Percentage* 标准便得到了目标用户的近邻集合 K_u 和目标项目的近邻集合 K_i,集合中用户和项目的相似性相比其他对象与目标用户 u 和目标项目 i 具有更大的相似性,于是有下面的命题成立:

1) $\forall x \in K_u, \forall y \in U - K_u \wedge y \neq u: PC(u,y) \leqslant PC(u,x)$;

2) $\forall x \in K_i, \forall y \in I - K_i \wedge y \neq i: PC(i,y) \leqslant PC(i,x)$。

2 预测与推荐过程

基于目标用户 u 的近邻集合 K_u 和目标项目 i 的近邻集合 K_i,便可以利用预测函数收集集合中对象的评分来对目标用户未评分的项目进行评分预测,然后选取预测值较高的项目来向用户进行推荐。下面详细介绍整个预测与推荐过程。

输入:评分矩阵 $\boldsymbol{R}(m,n)$,百分数 *Percentage*;

输出:目标用户 u 的 N 个推荐项目。

步骤 1 利用矩阵 $\boldsymbol{R}(m,n)$ 和 PC 相似性模型计算用户间的相似性和项目间的相似性。

时间复杂度分析:对于 $m \times n$ 维的矩阵 $\boldsymbol{R}(m,n)$ 来说,PC 模型计算每对用户的相似性总共需要进行 $3n$ 次乘法操作和 $4n$ 次加法操作。由于该相似性具有对称性,即 $PC(u,v) = PC(v,u)$,且系统中共有 m 个用户,所以这样的相似性计算过程总共需要进行 $m(m-1)/2$ 次,因此该阶段用户方面的模型时间复杂度为 $O(m^2n)$;同理,项目方面的模型复杂度为 $O(mn^2)$。可以看出,此阶段需要花费很长的时间,尤其是在 m 和 n 较大的情况下,不过该阶段在系统中可以通过离线的方式在后台完成。

步骤 2 根据 *Percentage* 值选取相似性较大的用户来为目标用户 u 构造近邻用户集合 K_u,然后通过 K_u 中用户对目标项目 i 的评分,利用预测函数计算用户 u 关于项目 i 的预测评分 $P^u_{u,i}$,通常性能较优的预测函数为改进型的权重函数,如下所示:

$$P^u_{u,i} = \bar{r}_u + \frac{1}{\sum_{x \in G_u} PC(u,x)} \sum_{x \in G_u} PC(u,x)(r_{x,i} - \bar{r}_x) \tag{7}$$

其中 $G_u = \{x \in K_u \wedge r_{x,i} \neq \text{null}\}$ 为 u 的近邻集合 K_u 中对项目 i 评分过的用户子集合。此时,$P^u_{u,i}$ 存在的条件是 $G_u \neq \varnothing$,否则 $P^u_{u,i}$ 为空值。

步骤 3 根据 *Percentage* 选取相似性较大的项目为目标项目 i 构造近邻集合 K_i,然后通过 K_i 中项目获得目标用户 u 的评分,利用预测函数从项目的角度计算用户关于项目的预测评分 $P^i_{u,i}$,如下所示:

$$P^i_{u,i} = \bar{r}_i + \frac{1}{\sum_{y \in G_i} PC(i,y)} \sum_{y \in G_i} PC(i,y)(r_{u,y} - \bar{r}_y) \tag{8}$$

其中 $G_i = \{y \in K_i \wedge r_{u,y} \neq \text{null}\}$ 为 i 的近邻集合 K_i 中获得过 u 评价的项目子集合。此时,$P^i_{u,i}$ 存在的条件是 $G_i \neq \varnothing$,否则 $P^i_{u,i}$ 为空值。

步骤 4 根据集合 G_u 中的用户个数 $|G_u|$ 和集合 G_i 中的项目个数 $|G_i|$,计算自适应平衡因子 λ 和 $1-\lambda$,分别作为预测评分 $P^u_{u,i}$ 和 $P^i_{u,i}$ 的权重,如下所示:

$$\begin{cases} \lambda = \dfrac{|G_u|}{|G_u| + |G_i|} \\ 1 - \lambda = \dfrac{|G_i|}{|G_u| + |G_i|} \end{cases} \tag{9}$$

其中 λ 和 $1-\lambda$ 的取值范围为 $[0,1]$,这样综合用户和项目两方面的预测结果,得到最终用户 u 关于项目 i 的预测评分 $P_{u,i}$:

$$P_{u,i} = \lambda \times P^u_{u,i} + (1-\lambda) \times P^i_{u,i} \tag{10}$$

可以看出,当 $\lambda = 1$ 时,此时将基于用户的 $P^u_{u,i}$ 作为最终预测结果;而当 $\lambda = 0$ 时,则将基于项目的 $P^i_{u,i}$ 作为最终预测结果。

步骤 5 重复执行步骤 2 ~ 4,直到用户 u 的所有未评分项目预测完毕,然后执行步骤 6。

步骤 6 从用户 u 未评分但可以获得评分预测的项目集合中,选取预测评分值最大的 N 个项目向 u 进行推荐。

时间复杂度分析(步骤 2 ~ 6):根据对象间的相似性值选取数量 *Percentage* 的相似用户和相似项目为目标用户和目标项目构造近邻集合,然后经过常数次的加法乘法操作综合用户和项目两方面的预测结果来获得用户关于未评分项目的评分预测。由于系统中共有 n 个项目,所以此过程最多需要运行 n 次,然后为目标用户选取 N 个推荐项目,因此该阶段模型的时间复杂度为 $O(n \cdot Percentage)$。该阶段的时间花费即为用户在实际中为了获得项目推荐而需要在线等待的时间。

3 实验评估与分析

3.1 实验数据集

实验采用目前衡量推荐模型质量广泛常用的 MovieLens 数据集,它是由美国 Minnesota 大学 GroupLens 研究实验室在线公布的一个标准数据集(http://www.grouplens.org),里面记录了 6 040 个用户关于 3 706 部电影的 1 000 209 个评分,且每个用户至少对其中的 20 部电影进行了评分。数据集的评分值范围为 1 ~ 5 的整数,其中 5 级别最高表示用户最喜爱,1 则级别最低为差评,0 表示用户未对相应项目评分。另外,数据集的数据稀疏性程度为:1 - 1 000 209/(6 040 × 3 706) = 95.53%,可以看出数据集中的用户评分数据相当稀疏。

实验过程中,本文会将整个数据集评分数据的 80% 用作训练集,主要用于模型的构建,剩下的 20% 用作测试集,用于检验模型的具体性能。

3.2 评估标准

推荐系统中的评估标准主要包括统计精度评价方法和决策支持精度评价方法两类,本文采用统计精度评价方法中广泛使用的平均绝对偏差(Mean Absolute Error,MAE)作为评估标准之一,用来统计项目的预测评分与实际评分之间的差值,*MAE* 值越小表明预测的准确性越高。假设测试集中用户的预测评分集合为 $H = \{P^{(1)}_{u,i}, P^{(2)}_{u,i}, \cdots, P^{(v)}_{u,i}\}$,对应用户的实际评分集合为 $L = \{r^{(1)}_{u,i}, r^{(2)}_{u,i}, \cdots, r^{(v)}_{u,i}\}$,$v$ 为评分数目,则 *MAE* 为:

$$MAE = \frac{1}{v} \sum_{l=1}^{v} |P^{(l)}_{u,i} - r^{(l)}_{u,i}| \tag{11}$$

接下来,本文定义关于推荐质量的 *F*1 标准,它是通常的荐准率和荐全率两种评估标准的综合,用来对项目推荐的准确性和全面性进行评估,*F*1 值越大表明推荐的质量越高,计算公式如下:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

其中的 *Precison* 和 *Recall* 即为荐准率和荐全率的标准,表达式如下:

$$\begin{cases} Precison = |V|/|T| \\ Recall = |V|/|W| \end{cases} \tag{13}$$

其中:T 为测试数据集中获得推荐的项目集合,$V = \{i \in T | r_{u,i} \geqslant \varepsilon\}$ 为推荐集合中推荐正确的项目集合,ε 为一个阈值常数,用来判定推荐的项目是否确为用户所喜欢的,$W = \{i \in H | r_{u,i} \geqslant \varepsilon\}$ 为测试集中用户喜欢的所有项目集合。

3.3　优化的相似性模型比较

为检验不同模型的具体性能,本节选取 PC、COS、CPC 和 SRC 四种模型,从用户的角度分别利用阈值法 ($\mu = 40$)[14] 和 Jaccard 系数对它们进行优化,然后采用 *MAE* 标准进行性能比较,结果如图 1 所示。

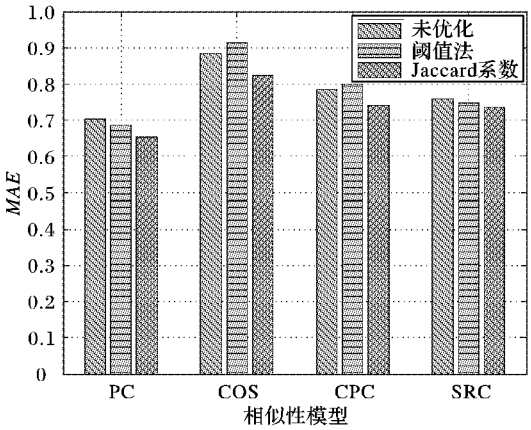

图 1　相似性模型的 *MAE* 性能比较

从图 1 可以看出,就四种模型本身而言,PC 模型表现最好,这是因为它考虑了用户评价的评分均值,从而可以有效避免评价值过分抖动的情况。同时可以看出,经过 Jaccard 系数优化的相似性模型,在 *MAE* 标准上与常数 μ 优化过的模型相比,性能表现均要优秀一些。这主要是由于前者可以根据两个用户评价项目的具体情况,对相似性值进行自适应地调整,从而有助于模型更好地度量用户相似性。而 μ 值则必须预先设定,这在通常情况下是比较困难的,而且用户间公共评价项目数目的巨大差异性和动态性特征很可能会使所设定的值无效,以致于在某种程度上会制约模型的性能,从而证明了本文选择 Jaccard 系数的有效性。

3.4　推荐模型比较

为检验 CMUI 模型的性能,选取目前比较典型的 RSCF[10]、HCFR[11] 和 UNCF[12] 协同过滤推荐模型作为参照对象,利用 *MAE* 和 *F*1 标准分别对它们在不同 *Percentage* 条件下进行性能比较,*Percentage* 取值范围为 [0.1,0.8],步长为 0.05,实验结果见图 2 和图 3。

图 2 的实验结果表明,在不同的 *Percentage* 值条件下,现有的各种模型均表现出了不同程度的性能波动,相比之下,本文的 CMUI 模型表现更出色。值得注意的是,随着 *Percentage* 取值的变大(大于 0.55 以后),模型性能逐渐变优并趋向于稳定,*MAE* 值在 0.65 上下波动。这主要是由于 CMUI 模型利用 Jaccard 系数对 PC 模型进行了自适应的优化,获得了较为准确的用户项目相似性值。在此基础上,本文还通过 *Percentage* 充分考虑了相似对象数目对预测结果的影响,并利用自适应平衡因子较好地统一了来自用户和项目方面的预测结果。这样,CMUI 模型才获得了比较准确的用户关于未评分项目的评分,最终改善了模型的 *MAE* 性能。比较而言,其他模型则要繁琐得多,这期间均涉及到了至少 2 个参数阈值的设置,比如 UNCF 模型需要预先设置包括 μ、相似性选择阈值等在内的 6 个参数,导致模型难以适应实际数据的变化,包括 *Percentage* 带来的好处。此外,这些研究成果还需要具体讨论这些参数对模型性能的影响,在很大程度上影响了模型的运行性能和实际应用价值。

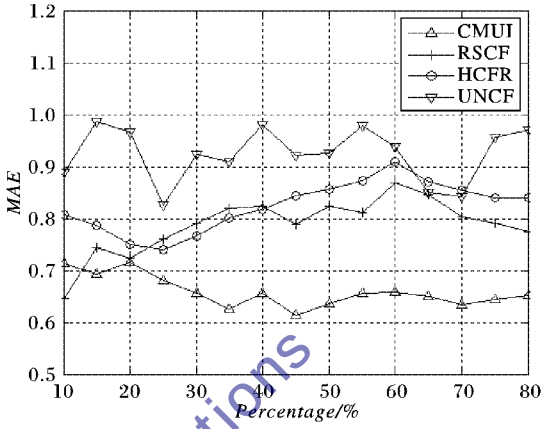

图 2　推荐模型的 *MAE* 性能比较

图 3 的实验结果表明,CMUI 模型在不同的 *Percentage* 取值条件下,相比其他模型在 *F*1 值和表现稳定性方面同样具有明显的优势。综合前面的图 2,*Percentage* 的建议取值范围为 [0.55,0.8],因为这样可以获得较为优异且稳定的性能。不过需要说明的是,CMUI 的 *F*1 优势与图 2 中的 *MAE* 优势密切相关,因为当模型在 *MAE* 方面具有优势时,可以更加准确地预测用户关于项目的评分,这样用户真正喜欢的项目便可以更好地被预测出来向用户进行推荐,从而同等条件下提高了模型在 *Precision* 和 *Recall* 方面的性能,最终表现为 *F*1 方面的优势。

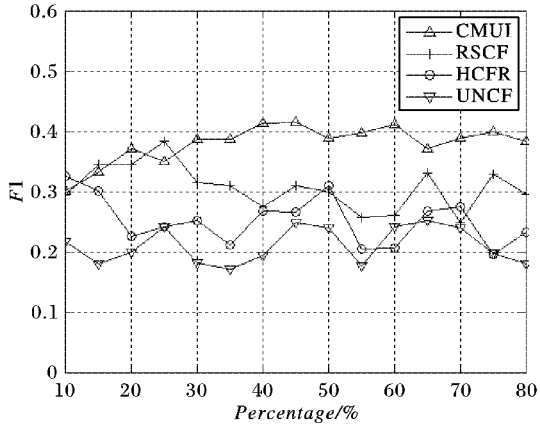

图 3　推荐模型的 *F*1 性能比较

4　结语

推荐系统中,从不同的角度出发可以获得不同的推荐结果,对此需要统计来自于不同数据源的数据信息。这个过程非常复杂,并且很多时候需要根据人为设置的经验值进行处理,这样便造成了信息统计结果不准确的问题。在此情况下,本文通过平衡因子自适应地统一了用户和项目方面的预测结果,整个过程没有人为设置其他参数。在此基础上,本文提出了综合用户和项目预测的协同过滤推荐模型,并基于 MovieLens 数据集将其与其他推荐模型进行实验比较,结果证

明了文中所提模型的优越性。

未来的研究工作包括:1)对用户项目的相关属性信息进行深入研究,并将其与评分数据结合获得合理的对象相似性模型,来更好地度量不同对象之间的相似性;2)协同过滤模型还面临着数据稀疏性问题、冷启动问题以及可扩展问题等挑战,如何获得一种较好的方法来缓解甚至于克服这些问题,同样是一个值得研究的方向。

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] 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.

[3] 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 International Conference on Modeling and Simulation. Berlin: Springer-verlag, 2012: 190-198.

[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] 杨兴耀,于炯,吐尔根·依布拉音,等.融合奇异性和扩散过程的协同过滤模型[J].软件学报,2013,24(8):1868 – 1884.

[6] KWON H J, HONG K S. Personalized smart TV program recommender based on collaborative filtering and a novel similarity method[J]. IEEE Transactions on Consumer Electronics, 2011, 57(3): 1416 – 1423.

[7] MU X W, CHEN Y, YANG J A, *et al*. An improved similarity algorithm based on hesitation degree for user-based collaborative filtering [C]// Proceedings of the 5th International Symposium on Intelligence Computation and Applications. Berlin: Springer-verlag, 2010: 261 – 271.

[8] SARWAR B, KARYPIS G, KONSTAN J, *et al*. Item-based collaborative filtering recommendation algorithms[C]// Proceedings of the 10th International Conference on World Wide Web. New York: ACM, 2001: 285 – 295.

[9] YAMASHITA A, KAWAMURA H, SUZUKI K. Adaptive fusion method for user-based and item-based collaborative filtering [J]. Advances in Complex Systems, 2011, 14(2): 133 – 149.

[10] 陶维安,范会联.基于评分支持度的最近邻协同过滤推荐算法[J].计算机应用研究,2012,29(5):1723 – 1728.

[11] 黄裕洋,金远平.一种综合用户和项目因素的协同过滤推荐算法[J].东南大学学报:自然科学版,2010,40(5):917 – 921.

[12] 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.

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

[14] 汪静,印鉴,郑利荣,等.基于共同评分和相似性权重的协同过滤推荐算法[J].计算机科学,2010,37(2):99 – 104.

(上接第 3344 页)

参考文献:

[1] 吴吉义,傅建庆,平玲娣,等.一种对等结构的云存储系统研究[J].电子学报,2011,39(5):1100 – 1107.

[2] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters [J]. Communications of the ACM, 2008, 51(1): 107 – 113.

[3] The Apache Software Foundation. Apache Hadoop [EB/OL]. [2012 – 10 – 18]. http://hadoop.apache.org/.

[4] GHEMAWAT S, GOBIOFF H, LEUNG S. The google file system [C]// SOSP 2003: Proceedings of the nineteenth ACM Symposium on Operating Systems Principles. New York: ACM Press, 2003: 29 – 43.

[5] ZHAO W Z, MA H F, HE Q. Parrallel k-means clustering based on MapReduce [C]// Proceedings of the 1st International Conference on Cloud Computing. Berlin: Springer-Verlag, 2009: 674 – 679.

[6] CATANZARO B, SUNDARAM N, KEUTZER K. Fast support vector machine training and classification on graphics processors [C]// ICML 2008: Proceedings of the 25th International Conference on Machine Learning. New York: ACM, 2008: 104 – 111.

[7] NORSTAD J. A MapReduce algorithm for matrix multiplication [EB/OL]. [2012 – 11 – 02]. http://www.norstad.org/matrix-multiply/index.html.

[8] LIN C, HUANG Z H, YANG F, *et al*. Identify content quality in online social networks [J]. IET Communications, 2012, 6(12): 1618 – 1624.

[9] SUN Z G, LI T, RISHE N. Large-scale matrix factorization using MapReduce [C]// ICDMW'10: Proceedings of the 2010 IEEE International Conference on Data Mining Workshops. Washington, DC: IEEE Computer Society, 2010: 1242 – 1248.

[10] LIU C, YANG H-C, FAN J L, *et al*. Distributed nonnegative matrix factorization for Web-scale dyadic data analysis on MapReduce [C] // WWW 2010: Proceedings of the 19th International Conference on World Wide Web. New York: ACM, 2010: 681 – 690.

[11] GEMULLA R, HAAS P, NIJKAMP E, *et al*. Large-scale matrix factorization with distributed stochastic gradient descent [C]// KDD 2011: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2011: 69 – 77.

[12] GUNTHER J H, HOFFMAN K H. [J]. Numerische Mathematik, 1991, 60: 354 – 356.

[13] STEWART G W. Jampack: a Java package for matrix computations [EB/OL]. [2012 – 10 – 20]. ftp://math.nist.gov/pub/Jampack/Jampack/AboutJam pack.html.

[14] JOE H, CLEVE M, PETER W. JAMA: a Java matrix package [EB/OL]. [20130 – 10 – 15]. http://math.nist.gov/javanumerics/jama/.

[15] FILIPPONE S, COLAJANNI M. PSBLAS: a library for parallel linear algebra computation on sparse matrices [J]. ACM Transactions on Mathematical Software, 2000, 26(4): 527 – 550.

[16] The Apache Software Foundation. Apache Hama [EB/OL]. [2012 – 07 – 12]. http://hama.apache.org/.

[17] PAPADIMITRIOU S, SUN J M. DisCo: distributed co-clustering with MapReduce: a case study towards petabyte-scale end-to-end mining [C]// ICDM 2008: Proceedings of the Eighth IEEE International Conference on Data Mining. Washington, DC: IEEE Computer Society, 2008: 512 – 521.

[18] KANG U, TSOURAKAKIS C E, FSLOUTSOS C. PEGASUS: a peta-scale graph mining system implementation and observations [C]// ICDM 2009: Proceedings of the 2009 Ninth IEEE International Conference on Data Mining. Washington, DC: IEEE Computer Society, 2009: 229 – 238.