

文章编号: 1003-0077(2016)02-0090-09

基于用户相似性传递的跨平台交叉推荐算法

李超^{1,2}, 周涛^{1,2}, 黄俊铭³, 程学旗³, 沈华伟³

(1. 电子科技大学 互联网科学中心, 四川 成都 611731;

2. 北京百分点信息科技有限公司, 北京 100080;

3. 中国科学院 计算技术研究所, 中国科学院网络数据科学与技术重点实验室, 北京 100190)

摘要: 个性化推荐系统在电子商务领域中的广泛应用带来了巨大的经济效益和良好的用户体验。由于用户数据往往分布在多个不同的网站, 单个网站的推荐系统受制于数据稀疏性的限制, 难以获得准确的推荐效果。该文提出了一种基于传递相似性的交叉推荐系统算法, 可以利用多个网站平台数据计算不同网站中的用户的相似度, 从而很大程度上克服了推荐系统中的数据稀疏性以及冷启动问题。结果显示, 该交叉推荐算法与传统的针对单个数据集的推荐算法相比, 推荐的精确性有一至两倍的提高。

关键词: 个性化推荐系统; 协同过滤; 多源数据; 稀疏性; 冷启动

中图分类号: TP391

文献标识码: A

Transfer with Shared Users: A Cross-platform Recommender System with Transferred Similarity

LI Chao^{1,2}, ZHOU Tao¹, HUANG Junming³, CHENG Xueqi³, SHEN Huawei³

(1. University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China;

2. Beijing Baifendian Information Technology Co., Ltd. Beijing 100080, China;

3. CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy Sciences, Beijing 100190, China)

Abstract: The widely use of personalized recommender systems on online shopping websites results in great profits and enhanced user experiences. However, since a user's behaviors usually scatter cross multiple different websites, it becomes difficult to provide accurate recommendations when a recommender system sees a section of his behaviors on a single website. We propose a new recommendation algorithm that transfers behaviors across different websites to calculate similarities between users on different websites. Our algorithm overcomes the sparsity and cold-start problem in recommender systems with a significant accuracy improvement, outperforming traditional algorithms that applied on a single website only.

Key words: personalization recommender systems; collaborative filtering; multiple source datasets; sparsity; cold-start problem

1 引言

随着互联网的迅猛发展, 用户在互联网上可接触到的信息也与日俱增, 用户不得不花费大量的时间在信息海洋里挑选出对自己有用的信息, 这种现象被称之为信息过载。个性化推荐系统旨在帮助用户走出这一困境, 在为用户提供良好体验的同时, 也

可以为电子商务带来更多的经济效益。目前, 个性化推荐系统已经被广泛地部署并应用在各个互联网领域^[1], 例如, Amazon 的推荐系统将用户可能感兴趣的商品推荐给用户, Amazon 的 35% 的销售额均来自推荐系统^[2]。MovieLens 是一个基于研究目的发布的电影推荐系统, 根据用户的历史信息提供其可能喜欢的电影^[3]。其他的推荐系统还包括 Last.fm 音乐推荐系统, Jester^[4]笑话推荐系统等等。

主流推荐系统都使用用户历史数据预测用户隐藏的兴趣分布,其性能对用户数据规模敏感。事实上,大部分实际使用的推荐系统都面临数据缺失的问题,只观测到用户的一小部分历史行为,从而难以作出准确的推荐预测。数据缺失通常表现为数据稀疏^[5]和冷启动^[6]。前者指单个用户只能对系统中的海量物品中的一小部分做出浏览、购买、评价等行为,后者指新注册用户的行为记录较少。

数据缺失的一个重要原因是网站的多样化与细分性导致用户行为数据分散在不同网站中。例如,一个用户对电影的偏好只记录在豆瓣网上,而其购买软件的行为只记录于苹果 App Store。因为每个网站都只能利用用户的一部分数据,而传统推荐算法只能利用单一数据源的用户历史行为估计用户兴趣,因此其准确性深受稀疏数据困扰。虽然利用电子邮件地址匹配等简单技术已经可以在不侵犯隐私的前提下识别多个网站中的同一用户,但若直接将多个网站数据简单合并,扩大的物品空间将导致更严重的数据稀疏问题。如何有效地利用多个网站的用户行为数据克服数据稀疏性,准确估计用户的兴趣,获得更优质的推荐效果,这一问题具有重要的学术意义和应用价值。

我们提出了一种利用多个网站数据的推荐系统算法,其核心思想是采用一种传递策略计算不同网站的用户之间的相似度。具体来说,对于网站 A 的一个新用户 u ,我们希望计算他与网站 A 中现有用户的相似度,以便利用协同过滤算法估计他的兴趣并推荐物品。如果该网站中任一用户在任一其他网站上均与 u 没有交集,则传统算法无法计算其与用户 u 的相似度。我们寻找一组中间用户,他们与 u 存在交集可计算相似度,同时亦与网站 A 中某些用户存在交集可计算相似度。根据社会平衡理论,我们可以通过中间用户与 u 的相似度及其与网站 A 用户的相似度的传递关系推断 u 与网站 A 用户的相似度。从而可以利用协同过滤算法对 u 在网站 A 上的兴趣作出估计。这一算法具有理论上的收敛性和实证上的高效性。实验发现,这一算法能比较准确地推断隐藏的用户相似度,有效地提高协同过滤算法对于冷启动和稀疏用户的推荐性能。

本文余下部分包括:第二部分介绍推荐系统及其数据缺失问题的相关工作,第三部分是算法描述,第四部分介绍所采用的评价指标,第五部分介绍实验结果及分析,第六部分总结探讨该算法的有效性和不足并建议未来工作方向。

2 相关工作

推荐系统的任务是为用户提供相关商品的推荐,一般被形式化成为矩阵填充或矩阵缺失值预测问题。协同过滤^[1]是应用较早的推荐算法,假设相似的用户会有相同的偏好,利用相似的行向量来进行缺失值的填充。协同过滤这一类基于历史记录信息^[7]的推荐算法会受到数据稀疏性和冷启动问题的困扰,数据过于稀疏的话,不能有效地进行相似度的计算,新的用户登录到推荐系统中来,由于其历史记录信息为零,所以无法利用协同过滤来对其进行推荐。

另外一种较为常见的推荐算法是基于矩阵分解^[8-11]的推荐算法,假设用户和商品都有自己特定的潜在特征向量,将目标用户与目标商品的特征向量进行点积运算得到的值即为该用户对该商品的评分。基于隐变量^[12-13]和矩阵分解的推荐算法虽然在评分预测的准确性^[14]上效果较好,但是由于其计算和实现的复杂性和缺乏可解释性,该类算法并不适用于大规模数据上的实际应用。

关于数据缺失问题,迁移学习^[15]是一种较为有效的解决方式,迁移学习利用不同领域之间共同的部分来相互促进各个领域内的学习任务。面对将迁移学习应用到推荐系统^[16-19]中的数据缺失问题时,一般也是采用矩阵分解技术,从隐变量的层面来建立起不同学习任务之间的联系,从而提高不同领域内的推荐准确度。例如,文献^[16]在数据较为稀疏的情况下,利用书籍和电影之间的潜在的共同主题来相互促进各自的推荐准确度。在将迁移学习应用到推荐系统中的数据缺失和稀疏性问题时,与普通的基于隐变量和矩阵分解的方法一样,计算复杂度较高,难以应用到大规模数据的实际应用中。另外,不同领域之间,知识的可迁移程度是不同的,一种迁移方案难以满足不同领域之间知识迁移的需求,需要针对各个领域的具体情况来进行迁移方案的设计^[17]。

3 基于用户相似性传递的交叉推荐算法

本文提出一种基于用户传递相似性的跨电商交叉推荐算法,利用系统已经获得的目标用户在目标电商外的历史行为信息,来解决目标用户在目标电商网站初次登录无法进行推荐的冷启动问题,以及在目标电商网站中的历史行为较少的数据稀疏性问题。

下面以两个电商为例对该算法进行说明,不失一般性,该算法可以推广到两个以上的多电商交叉推荐情形。

假设有两个电商 x_1 和 x_2 ,我们将只在电商 x_1 中有过历史行为的用户集合定义为 U_1 ,只在电商 x_2 中有过历史行为的用户集合定义为 U_2 ,类似 U_1 , U_2 这种在且只在其中某一家电商有过历史行为的用户,我们称之为非交叉用户。将在 x_1, x_2 两个电商中均有过历史行为的用户集合定义为交叉用户,用 U_c 表示。用户行为矩阵如图 1 所示。

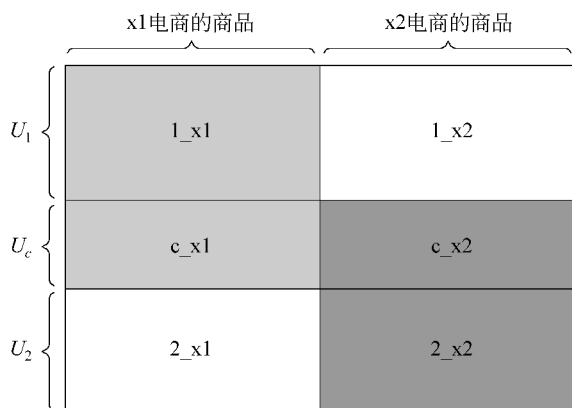


图 1 评分矩阵示例

将普通的 UCF (User-based Collaborative Filtering) 算法^[20]应用到图 1 所示的交叉推荐的情形时,由于非交叉用户 U_1 只能通过自身在电商 x_1 中的行为信息(图 1 中 1_x1 部分)和交叉用户 U_c 在电商 x_1 中的行为信息(图 1 中 c_x1 部分)来与交叉用户 U_c 建立相似性的联系,所以我们只能利用交叉用户 U_c 的历史行为信息来对非交叉用户 U_1 推荐电商 x_2 中的商品。同理,也只能利用交叉用户 U_c 对非交叉用户 U_2 推荐电商 x_1 中的商品。直观地来看, U_1 与 U_2 之间是没有相似性联系的,因为他们之间没有任何的历史行为的交集。

在实际情况下,非交叉用户的数量是远远高于交叉用户的,传统的 UCF 算法只能利用所占比例较少的交叉用户的历史行为信息对所占比例较大的非交叉用户进行推荐,由于数据稀疏性问题,传统的 UCF 很难提供较为理想的推荐结果。

我们提出基于用户传递相似性的推荐算法,利用所占比例较少的交叉用户的历史行为信息作为纽带,将两个不同电商的非交叉用户 U_1 和 U_2 建立起相似性上的联系,从而达到交叉推荐的目标。

3.1 传统的协同过滤推荐算法 (UCF)

传统的基于用户的协同过滤算法,假设用户之

间是有相似性的,相似的用户对于同一个商品会有同样的喜好程度。用户之间的相似性是根据各自的历史行为信息的相似程度来定义的。

将系统中所有用户的历史行为信息看作一个矩阵,每一行代表一个用户,每一列代表一个商品。该用户对应的那一行即为该用户的行为向量 u 。定义用户行为向量之间的相似性,即为用户之间的相似性。

本文采用的相似性为 Jaccard 相似性:

$$S_{uv} = \frac{|u \cap v|}{|u \cup v|} \quad (1)$$

UCF 对目标用户 u 进行推荐时,首先找到与目标用户 u 最为相似的若干个用户 N_u , N_u 被称之为目标用户的邻居,邻居用户 N_u 对该目标商品 i 的评分进行加权平均作为用户 u 对商品 i 的预测评分,邻居用户 N_u 对目标商品 i 的评分的权值即为 N_u 与目标用户的相似性。然后将预测评分按从大到小排序,选择预测评分最高的前若干个商品作为最终的推荐列表。

UCF 中,计算目标用户 u 对于目标商品 i 的预测评分公式为式(2)。

$$R_{ui} = \frac{\sum_{v \in N_u} R_{vi} \cdot S_{uv}}{\sum_{v \in N_u} S_{uv}} \quad (2)$$

3.2 基于用户传递相似性的交叉推荐算法 (TSUCF)

协同过滤算法认为用户的偏好信息都体现在其对应的行为向量中,如图 1 所示,非交叉用户 U_1 和 U_2 之间是无法直接计算相似性的。但是, U_1 和 U_2 分别均在各自行为所发生的电商与 U_c 有相似性。我们将交叉用户 U_c 作为纽带来建立 U_1 与 U_2 之间的相似性。

交叉用户不仅体现出了其在电商 x_1 中的行为偏好信息,因此,也体现出了其在电商 x_2 中的行为偏好信息。那么交叉用户则可以用来建立用户在两个电商 x_1 和 x_2 中的行为偏好联系。我们认为交叉用户的数量达到一定的比例后,其在电商 x_1 中的行为和电商 x_2 中的行为可以体现出一种潜在的模式关联,即在电商 x_1 中有某些特定的购物行为模式后,也会在电商 x_2 中有类似的购物行为模式,反之亦然。在对用户 U_1 推荐电商 x_2 中的商品时,我们建立 U_1 与 U_c 之间的相似性,然后计算 U_c 与 U_2 之间的相似性,然后以 U_c 用户的行为向量所体现出的行为模式来建立起 U_1 与 U_2 之间的相似性。

U_1 与 U_2 之间的相似性通过 U_c 传递的策略为,首先找出 U_c 中与 U_1 和 U_2 均有相似性的用户 U_c' ,

然后分别计算出 U_1 与所有 U_c ' 的相似性, 记录到向量 \vec{S}_{1c} 中, 再计算 U_2 与所有 U_c ' 的相似性, 记录到向量 \vec{S}_{c2} 中。那么我们会得到两个相似性向量, \vec{S}_{1c} 和 \vec{S}_{c2} , 之后我们求这两个向量的内积 $S_{1c} \cdot S_{c2}$, 即为 U_1 与 U_2 的传递相似性。

如图 2 所示, $S_1 - S_{10}$ 分别是 U_1 与 U_c , U_c 与 U_{21} , U_{22} 之间的相似性。下面以图 2 为例说明传递相似性的计算过程。计算 U_1 与 U_{21} 的传递相似性, 找到 U_c 中与 U_1 和 U_{21} 均有相似性的用户 U_{c1}, U_{c2}, U_{c4} , 则 U_1 与 U_{21} 的传递相似性即为: $S_1 S_5 + S_2 S_6 + S_4 S_7$ 。计算 U_1 与 U_{22} 的传递相似性, 找到 U_c 中与 U_1 和 U_{22} 均有相似性的用户 U_{c2}, U_{c3}, U_{c4} , 则 U_1 与 U_{22} 的传递相似性即为: $S_2 S_8 + S_3 S_9 + S_4 S_{10}$ 。

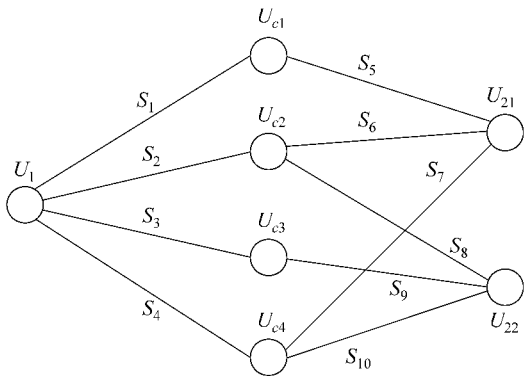


图 2 相似性的传递

U_1 和 U_2 之间的传递相似性计算可以形式化为式(3):

$$S_{U_1 U_2} = S_{U_1 U_c} \cdot S_{U_c U_2} \quad (3)$$

$S_{U_1 U_2}$ 表示 U_1 和 U_2 之间的传递相似性矩阵^[21], $S_{U_1 U_c}$ 表示 U_1 和 U_c 之间的相似性矩阵, $S_{U_c U_2}$ 表示 U_c 和 U_2 之间的相似性矩阵。

得到传递相似性矩阵 $S_{U_1 U_2}$ 之后, 我们就可以利用式(2)来进行评分预测和推荐了。

表 1 TSUCF 推荐算法

```

计算  $U_1$  和  $U_c$  之间的相似性矩阵  $S_{U_1 U_c}$ 
计算  $U_c$  和  $U_2$  之间的相似性矩阵  $S_{U_c U_2}$ 
计算  $U_1$  和  $U_2$  之间的传递相似性矩阵  $S_{U_1 U_2}$ :
 $S_{U_1 U_2} = 0$ 
for  $u_1 \in U_1, u_2 \in U_2$ 
  for  $u_c \in U_c$  do
    if  $S_{U_1, U_c} \neq 0 \ \&\& \ S_{U_c, U_2} \neq 0$  then
       $S_{U_1 U_2} += S_{U_1, U_c} \cdot S_{U_c, U_2}$ 
    end if
  end for
end for
利用传递相似性矩阵  $S_{U_1 U_2}$  结合 User-based CF 算法进行推荐

```

4 评价指标

评价指标^[22] 我们采用准确率 (Precision) 和召回率 (Recall)^[23]:

$$Precision = \frac{1}{U} \sum_u \frac{|R_u \cap T_u|}{|R_u|} \quad (4)$$

$$Recall = \frac{1}{U} \sum_u \frac{|R_u \cap T_u|}{|T_u|} \quad (5)$$

其中, 对于用户 u 来说, R_u 是推荐列表中的商品集合, T_u 是测试集中用户 u 选过的商品集合, U 是测试用户数。

5 实验及结果分析

5.1 数据集及划分方式

我们采用百分点推荐引擎提供的两个电商 x1 和 x2 中的交叉用户数据, 共有 27 899 个交叉用户, 28 617 个商品, 其中电商 x1 中有 8 372 个商品, 电商 x2 中有 20 245 个商品。由于交叉用户的测试集合只能从交叉用户中选取, 所以我们只在这些交叉用户的数据上进行实验。实验时, 分别屏蔽掉对应用户在电商 x1 中或电商 x2 中的行为, 分别以此来仿真模拟电商 x2 中的非交叉用户和电商 x1 中的非交叉用户。

下面以对用户 U_1 推荐电商 x2 中的商品为例, 说明交叉推荐的实验方案。

如图 1 所示, 1_x2 是我们要进行推荐的部分, 我们目前的任务是利用 2_x2 部分的信息来填充 1_x2 部分的信息。在预测用户 U_1 对 1_x2 部分的评分时, 我们将 2_x1 部分的数据屏蔽掉, 以此来模拟真实情况下的非交叉用户, 即 U_2 。在对 U_2 推荐 x1 中的商品时, 采用同样的方案。

我们采用传统的 UCF 算法, 利用交叉用户 U_c 中的行为信息来对 1_x2 以及 2_x1 部分进行预测, 作为我们提出的交叉推荐算法的测试基准。

5.2 实验结果

我们分别考察推荐列表长度 (RL), 邻居数目 (NS), 以及交叉用户所占比例 (PCD, Percent of Cross Data) 对推荐 precision 和 recall 的影响。

关于交叉用户占比 PCD, 我们分别取 PCD = 10, 20, 30, 40, 50; 当 PCD = 10 时, 即 U_c 占有用户比例的 10%, U_1 U_2 的比例相等均为 45%, 其他 PCD 取值时, U_1 U_2 的比例以此类推。UCF 算法只利用 U_c 部分的用户数据进行推荐。TSUCF 为我们

提出的算法。

给定邻居数目 $NS=50$ 的情况下,推荐列表长度 RL 以及不同的交叉用户占比 PCD 对 $Precision$,

$Recall$ 两个指标的影响。以下各图中 1_x2 表示将图 1 所示的 1_x2 部分作为测试集, 2_x1 表示将图 1 所示的 2_x1 部分作为测试集。

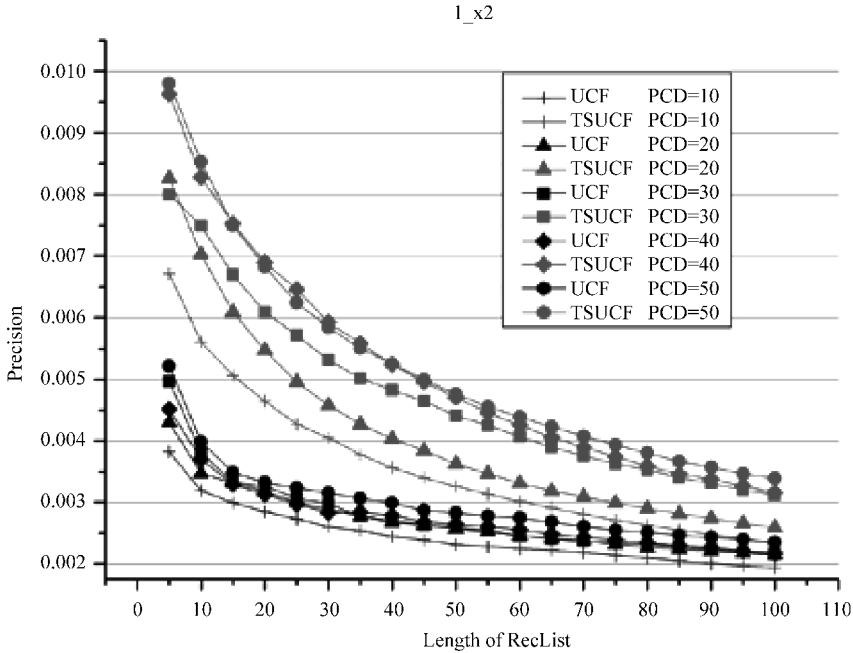


图 3 准确率 1_x2 为测试集

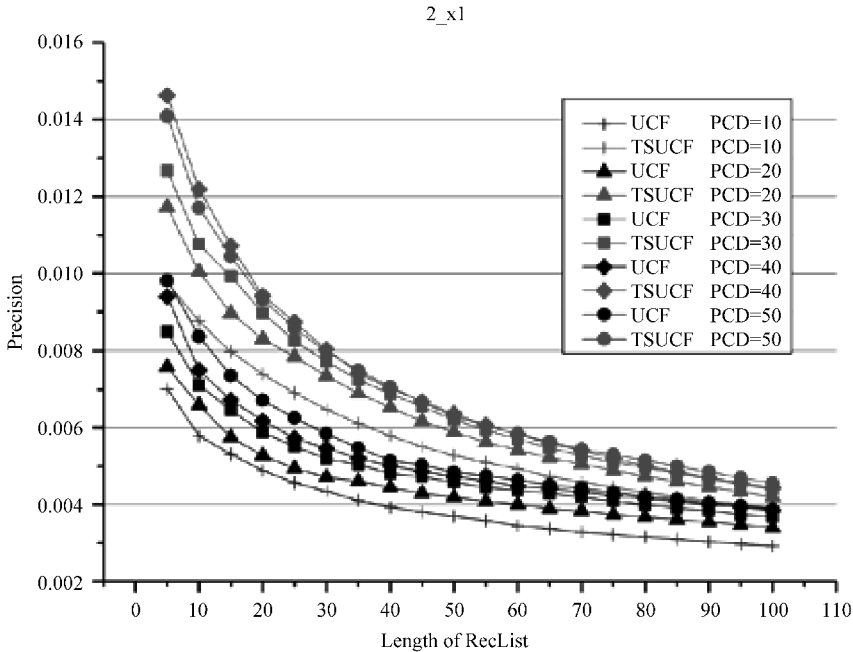


图 4 准确率 2_x1 为测试集

图 3、图 4 分别是测试集为 1_x2 和 2_x1 时,测试结果的 $Precision$ 指标,图 5、图 6 分别是测试集为 1_x2 和 2_x1 时,测试结果的 $Recall$ 指标。由以上评价指标的测试结果可以得出结论,TSUCF 算法较之于传统的 UCF 在准确率和召回率上均有巨大的提高。在推荐列表长度在 50 以内时,无论交叉用

户的所占比例如何,TSUCF 的准确性均要好于 UCF。随着交叉用户数量的逐渐增加 (TSUCF $PCD=10, 20, 30, 40, 50$),TSUCF 的准确性也逐步提高,这说明我们的 TSUCF 算法对于交叉用户的数量有一定程度上的依赖。

在一般的推荐系统应用场景中,推荐列表长度

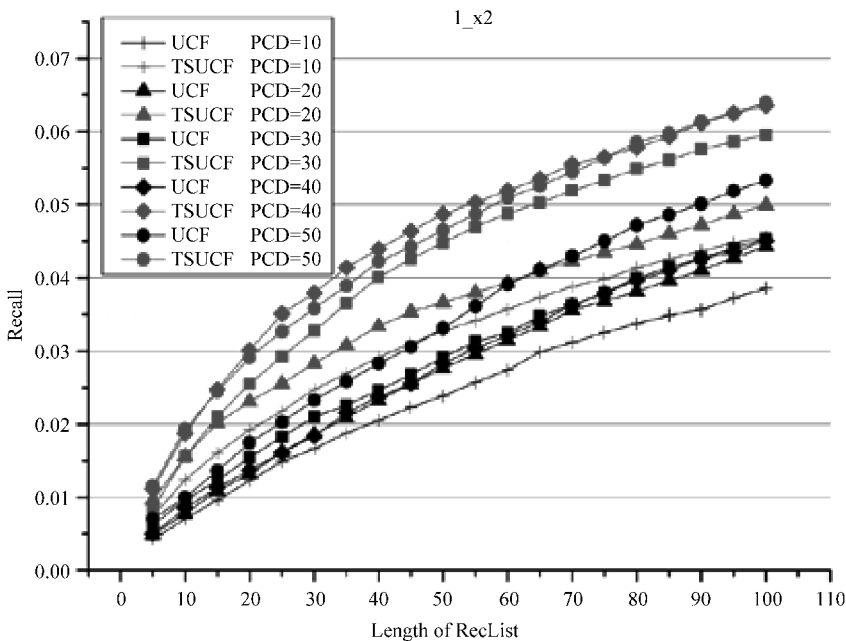


图5 召回率 1_x2 为测试集

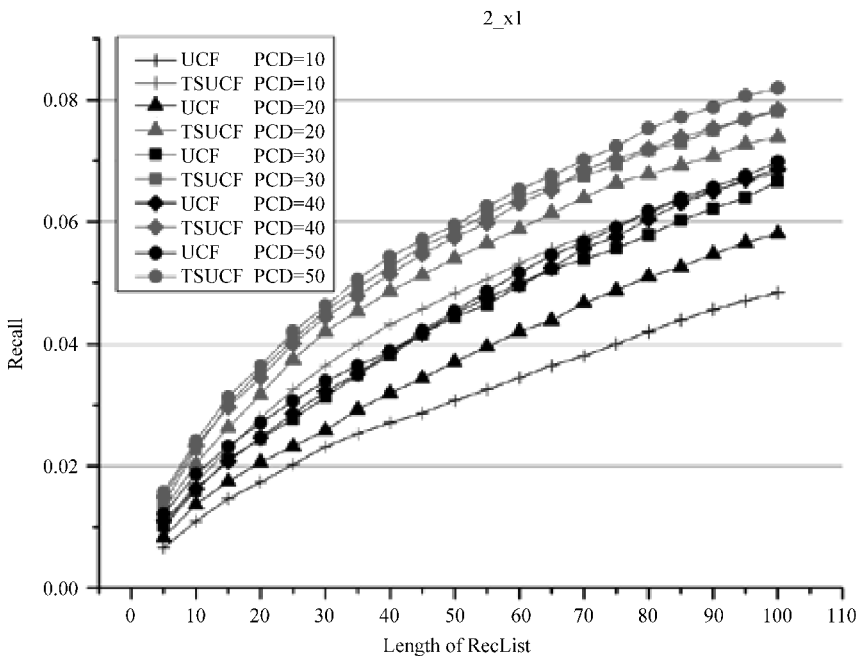


图6 召回率 2_x1 为测试集

一般最大取 20 就可以满足需要了,接下来的实验我们考察在推荐列表固定为 20 的时候,邻居数目 NS 对于 UCF 和 TSUCF 的影响。

给定推荐列表长度 $RL=20$ 的情况下,邻居数目 NS 以及不同的交叉用户占比 PCD 对 Precision, Recall 两个指标的影响。以下各图中 1_x2 表示将图 1 所示的 1_x2 部分作为测试集,2_x1 表示将图 1 所示的 2_x1 部分作为测试集。

图 7、图 8 分别是测试集为 1_x2 和 2_x1 时,测试结果的 Precision 指标,图 9、图 10 分别是测试集

为 1_x2 和 2_x1 时,测试结果的 Recall 指标。由以上评价指标的测试结果同样可以得出与上一小节相同的结论,TSUCF 算法较之于传统的 UCF 在准确率和召回率上均有巨大的提高。在邻居数目大于 30 的时候,无论交叉用户的所占比例如何,TSUCF 的准确性均要好于 UCF。在邻居数目小于 30 时,TSUCF 在交叉用户比例较少(如 TSUCF PCD=10)而 UCF 算法中的训练集 U_c 部分占比较高(如 UCF PCD=50)的情况下,表现会不如普通的 UCF。这说明我们的 TSUCF 算法在邻居数目大于

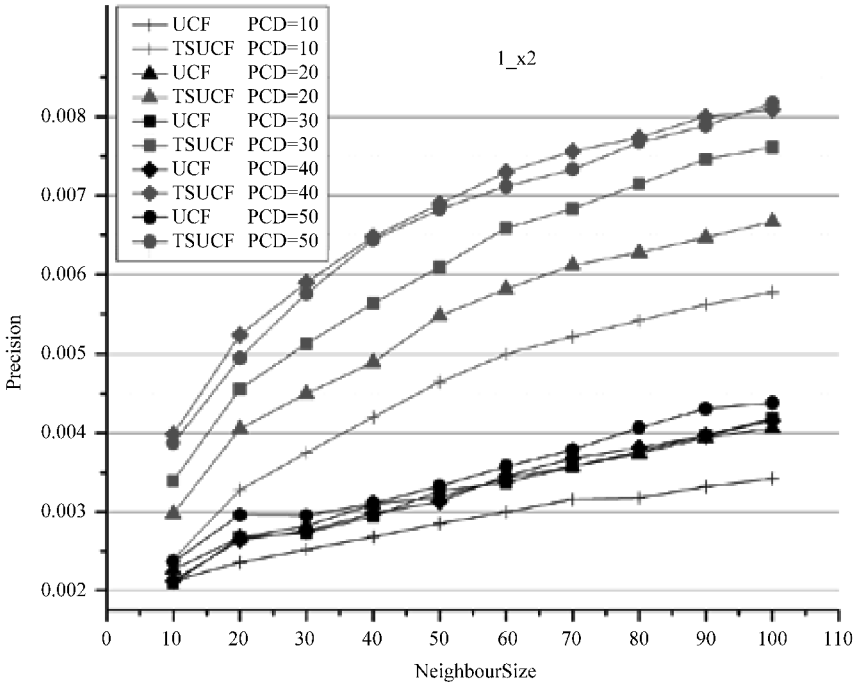


图 7 准确率 1_x2 为测试集

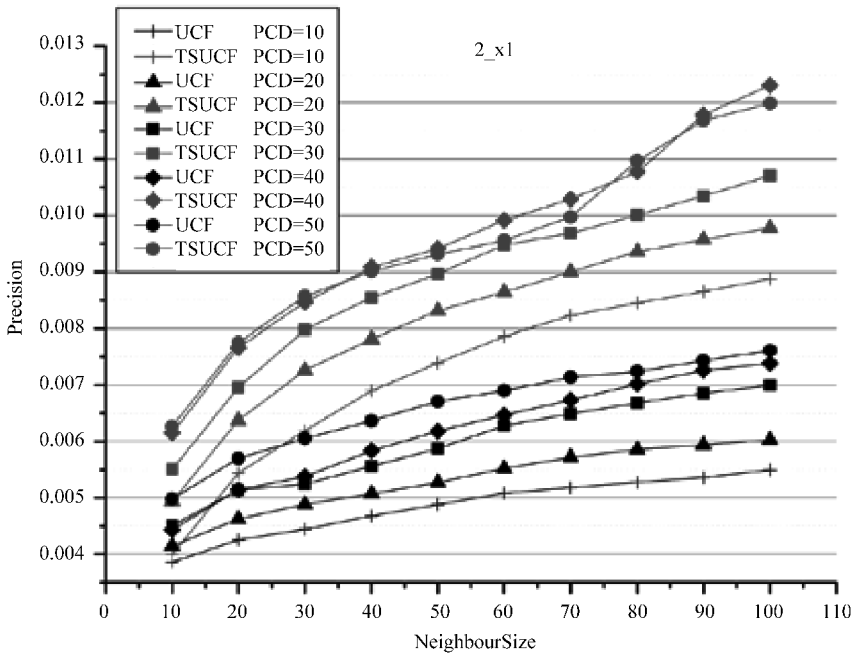


图 8 准确率 2_x1 为测试集

30 的时候会取得更好的效果。

6 结论

我们提出了一种基于用户传递相似性的跨电商交叉推荐算法,在百分点推荐引擎提供的两个电商的交叉用户数据上,验证了该算法的有效性。该算法与传统的 UCF 相比,在推荐准确性上有巨大的

提高,在不同的参数(推荐列表长度,邻居数目,交叉用户占比)配置下能得到 1 至 2 倍的提高。

我们算法对于交叉用户的比例有一定程度上的依赖。这也与实际情况较符合,因为只有当交叉用户达到一定数量的情况下,交叉用户在不同电商之间行为模式的关联才能够有效地体现出来,交叉用户的行为作为非交叉用户相似性之间的纽带才能够足以健壮,从而建立起更加可信的传递相似性。

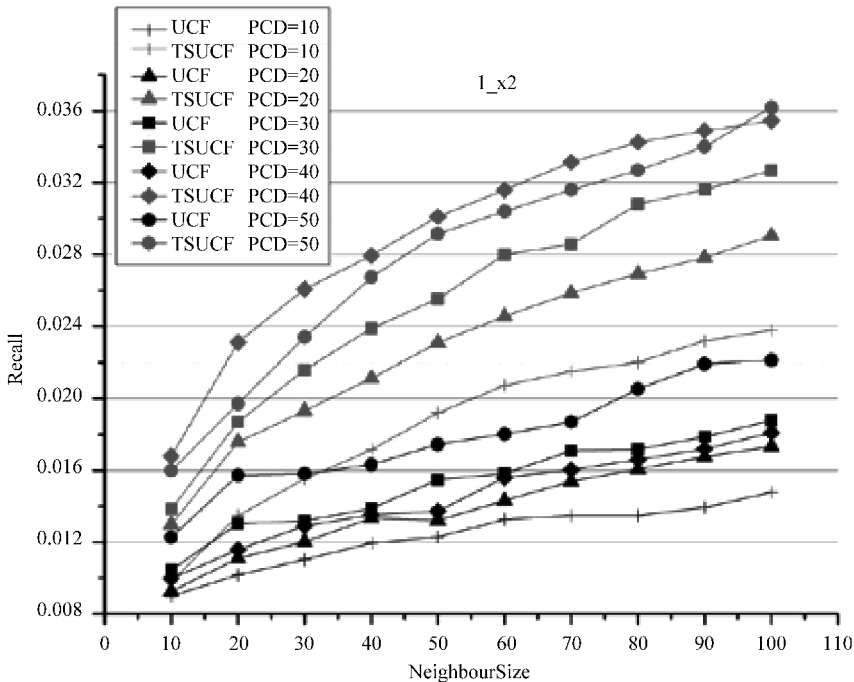


图 9 召回率 1_x2 为测试集

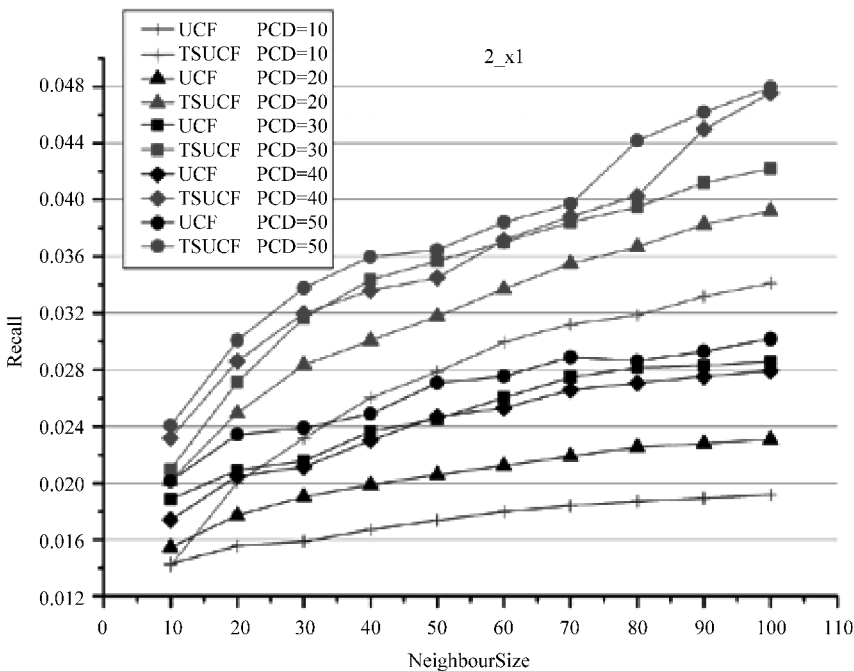


图 10 召回率 2_x1 为测试集

未来相关的工作可以在如下方面展开,如挑选出可信度较高的交叉用户,来作为更加优质的相似性传递的纽带;从隐变量和数据降维的角度来考虑相似性的传递,可能会有更好的推荐效果。

参考文献

[1] Adomavicius G, Tuzhilin A, Toward the next genera-

tion 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] Linden G, Smith B, York J, Amazon. com Recommendations Item-to-Item Collaborative Filtering[J]. IEEE Internet Computing, 2003, 7(1): 76-80.

[3] Dahlen B J, Konstan J A, Herlocker J L, et al. Jump-starting movielens: user benefits of starting a collabo-

- rative filtering system with “dead data”[R]. TR 98-017. University of Minnesota, March 1, 1998.
- [4] Goldberg K, Roeder T, Gupta D, et al. A Constant time collaborative filtering algorithm[J]. Information Retrieval, 2001, 4(2): 133-151.
- [5] Huang Z, Chen H, Zeng D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filter[J]. IEEE Trans Information Systems. 2004, 22(1): 116-142.
- [6] Zhang ZK, Liu C, Zhang YC, et al. Solving the Cold-Start Problem in Recommender Systems with Social Tags[J]. EPL. 2010, 9228002.
- [7] Su X, Khoshgoftaar T. A Survey of Collaborative Filtering Techniques[J]. Advances in Artificial Intelligence, 2009.
- [8] Hofmann T. Latent semantic models for collaborative filtering[J]. ACM Trans Inf. Syst, 2004, 22: 89-115.
- [9] Koren Y. Collaborative filtering with temporal dynamics[J]. Commun. ACM, 2010, 53: 89-97.
- [10] Koren, Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. IEEE Computer, 2009, 42: 30-37.
- [11] Srebro N, Rennie JDM, Jaakkola, T. Maximum-margin matrix factorization[C]//Proceedings of the 17th Advances in Neural Information Processing Systems (NIPS'04), 2004: 1329-1336.
- [12] Salakhutdinov R, Mnih A, Probabilistic matrix factorization[C]//Proceedings of the 21st Advances in Neural Information Processing Systems (NIPS'08), 2008: 1257-1264.
- [13] Salakhutdinov R, Mnih, A. Bayesian probabilistic matrix factorization using markov chain monte carlo [C]//Proceedings of the 25th International Conference on Machine Learning. New York, NY, USA: ACM, ICML '08, 2008: 880-887.
- [14] Y Zhou, et al. ,Large-Scale Parallel Collaborative Filtering for the Netflix Prize[C]//Proceedings of 4th International Conference on Algorithmic Aspects in Information and Management, LNCS 5034, Springer, 2008: 337-348.
- [15] S J Pan, Q Yang. A survey on transfer learning [C]//Proceedings of the IEEE Transactions on Knowledge and Data Engineering, 2010: 22(10): 1345-1359.
- [16] B Li, Q Yang, X Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction[C]//Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09), 2009: 2052-2057.
- [17] B Li. Cross-Domain Collaborative Filtering-A Brief Survey[C]//Proceedings of 23rd IEEE International Conference on Tools with Artificial Intelligence, 2011.
- [18] W Pan, E W Xiang, N N Liu, et al. Transfer learning in collaborative filtering for sparsity reduction[C]//Proceedings of the 26th in AAI, 2010: 230-235.
- [19] B Li, Q Yang, X Xue. Transfer learning for collaborative filtering via a rating-matrix generative model, in ICML, 2009, pp. 617-624.
- [20] Resnick P, Iacovou N, Suchak M, et al, GroupLens: An Open Architecture for Collaborative Filtering of Netnews[C]//Proceedings of the 1994 ACM conference on Computer Supported Cooperative Work. New York, ACM, 1994: 175-186.
- [21] Duo Sun, Tao Zhou, Jian-Guo Liu, et al. Information filtering based on transferring similarity[J]. Phys. Rev. E, 2009, 80, 017101.
- [22] 刘建国,周涛,郭强,等. 个性化推荐系统的评价方法综述[J]. 复杂系统与复杂性科学, 2009, 6(3): 1-8.
- [23] George K. Evaluation of item-based top-n recommendation algorithms[C]//Proceedings of the 10th International Conference on Information and Knowledge Management ACM, New York, 2001: 247-254.



李超(1988—), 硕士, 主要研究领域为社交媒体分析、数据挖掘、机器学习。

E-mail: xunhuan_lc@gmail.com



周涛(1982—), 博士, 教授, 主要研究领域为统计物理与复杂性科学。

E-mail: zhutou@ustc.edu



黄俊铎(1984—), 博士, 主要研究领域为信息传播、社交网络分析。

E-mail: mail@junminghuang.com