

基于交通网络数据优化的地理信息推荐系统

邵阔义, 班晓娟[✉], 王笑琨, 李 斌

北京科技大学计算机与通信工程学院, 北京 100083

✉ 通信作者, E-mail: banxj@ustb.edu.cn

摘 要 针对目前上下文感知推荐系统主要研究方向为用户和系统, 而没有结合实际交通网络位置特点进行研究的问题, 本文提出了一种基于交通网络数据优化的地理信息推荐系统。该系统在协同过滤推荐模型基础上结合交通网络数据的地理信息对推荐算法进行改进。实验结果显示推荐质量获得明显提升。

关键词 地理信息系统; 推荐系统; 交通; 网络数据; 协同过滤; 优化

分类号 TP393

Geographic information recommender system optimized by transportation network data

SHAO Kuo-yi, BAN Xiao-juan[✉], WANG Xiao-kun, LI Bin

School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

✉ Corresponding author, E-mail: banxj@ustb.edu.cn

ABSTRACT In view of the problem that current context-aware recommender systems take users and systems as the main research direction and do not consider actual transportation network location characteristics, this article introduces a geographic information recommender system optimized by transportation network data. The system combines with transportation network data of geographic information in the context-aware recommender system to optimize the recommended results. Experimental results show that the recommended quality get improved significantly.

KEY WORDS geographic information systems; recommender systems; transportation; network data; collaborative filtering; optimization

随着互联网以及各种信息通信技术的飞速发展, 人们逐渐从一个信息匮乏的时代迈入了一个信息过载的时代。为了解决信息过载的问题, 许多科学家和工程师提出不同的解决方案。近年来, 搜索引擎(如谷歌和百度)成为帮助人们获取信息的主要工具, 但仍然不能满足用户的个性化信息需求。由此, 个性化推荐系统应运而生。推荐系统通过挖掘用户和项目之间的关系, 找出用户可能感兴趣的项目推荐给用户, 从而满足用户获取所需信息。从技术实现角度来看, 推荐流程包括用户偏好提取、个性化内容推荐、推荐效果评价与修正。从信息过滤角度来看, 推荐系统分为协同过滤、基于内容的过滤和混合式过滤^[1]。近年来, 推荐系

统在电子商务(如亚马逊、京东、淘宝和豆瓣)、信息检索(如谷歌、雅虎和百度)以及移动应用、互联网广告、旅游、社交网络等众多应用领域取得十分巨大的进展。

上下文感知推荐系统最早由 Adomavicius 和 Tuzhilin 等^[2]提出, 旨在将上下文环境信息融入到推荐系统中来提高推荐系统的精度。其中, 上下文信息可以包括时间、地点、天气、周围环境、用户意图等各种方面的因素。例如, 一个移动端的餐馆推荐系统, 应当向用户推荐其目前所处位置附近的餐馆, 而不能推荐一个远在几十公里外的餐馆。目前, 国内外许多研究人员对上下文感知推荐系统的理论、方法和应用展开广泛的研究。Baltrunas 和 Amatriain^[3]以隐式的用户反馈

收稿日期: 2014-10-09

基金项目: 国家自然科学基金资助项目(61272357, 61300074, 61572075)

为基础研究时间依赖的推荐系统,通过预过滤的方法构建推荐系统. Ricci^[4]提出一种基于移动端的推荐系统,以提供一种能够推动旅游行业发展的推荐系统. Panniello 等^[5]则致力于对上下文感知系统的预过滤、后置过滤等过滤方式加以研究. Karatzoglou 等^[6]、Hidasi 和 Tikk^[7]则使用张量分解的方式研究如何使上下文感知推荐系统中多个上下文信息分量的计算更加高效. Buriano 等^[8]研究的是基于本体知识库的上下文感知推荐系统算法. Palmisano 等^[9]讨论的是上下文感知推荐系统的建模技术. 王立才等^[1,10]对上下文感知推荐系统进行了综述性的探讨,对一个基于心理的上下文感知推荐系统进行了矩阵稀疏性等方面的研究. 徐月美等^[11]则研究移动设备中上下文感知推荐系统的应用场景.

不同地区的用户兴趣不同,用户在不同的地方兴趣也不同. 地点上下文(即位置上下文)作为一个重要的空间特征,往往对于用户行为产生重要的影响. 尤其是对于持卡人消费而言,一般都具有活动本地化的特点,即用户一般只会在某个活动范围内发生消费行为. 在本文中,将利用交通网络数据实现引入地点上下文的持卡人消费推荐系统.

1 模型与算法

地点是最为重要的上下文信息之一,对于用户兴趣有着十分深入的影响. 当前,智能手机已经成为普通百姓的基本配置,而智能手机上的 LBS 应用(如 Foursquare 和大众点评)也越来越多,根据用户所在位置进行推荐也成为移动应用所应具备的重要功能. 然而,目前的移动应用中,针对地点的推荐往往只能以用户所在位置为圆心、某个距离为半径的方式进行距离的测算,并且绝大部分移动应用并没有使用个性化的推荐方法,而是针对某个位置只有千篇一律的推荐结果.

为了给用户提供更好的推荐结果,一些研究人员已经开始对地点上下文感知推荐系统进行研究.

来自明尼苏达大学的科研人员提出一种地点上下文感知推荐系统 LARS(location aware recommender system)^[12]. 它将物品分为两大类:一类是没有空间属性的物品(如图书、电影和音乐),一类是有空间属性的物品(如书店和饭店). 同时将用户也分为有空间属性和没有空间属性的两大类. 这样,数据集就有以下三种表达形式.

(用户,用户位置,物品,评分):这种表达形式下,可以通过用户的位置来预测其可能喜欢的物品,如不同国家的人喜欢不同的音乐风格.

(用户,物品,物品位置,评分):这种表达形式下,用户对于物品的位置十分敏感,即用户只会在某个位

置范围内进行活动.

(用户,物品,用户位置,物品位置,评分):这种表达形式下,无论是用户还是物品的位置都变得至关重要,相当于前两种表达形式的综合.

那么,针对第一种数据集,LARS 将其根据用户的位置划分成多个子集,形成一种树状结构,根据每个用户的位置,只需对在同一位置的用户的行为数据集进行推荐算法,即可获得推荐结果. 但是,这种方式存在叶子节点上的用户数量过少的缺点,会导致相关用户行为数据过于稀疏,在该系统中又提出了一种金字塔模型,对树中不同深度的节点进行加权处理,得到最终的推荐列表. 针对第二种数据集,LARS 中使用了后置过滤的方法,首先忽略地点信息,得到 ItemCF 算法的推荐结果,并根据位置的远近进行一个加权处理. 这里可以使用欧式距离^[13]度量地图上两点的距离,也可以使用交通网络数据^[14]来度量地图上两点之间的距离.

如图1所示,本文提出一种基于交通网络数据优化的地理信息推荐算法,首先利用传统的协同过滤算法对数据集进行预处理,而后将地点上下文加入进来,生成包含基于交通网络数据地点上下文的新的数据集,将其划分为训练集和测试集,进行实验分析. 下面将分别讨论基于协同过滤的推荐模型以及地理信息的距离计算问题,其中地理信息的距离计算是实现推荐算法衰减函数的基础.

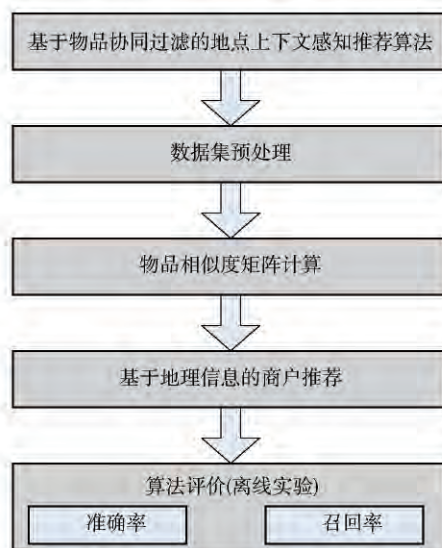


图1 地点上下文感知推荐系统流程图

Fig. 1 Flow chart of the context-aware recommender system

1.1 基于协同过滤的推荐模型

随着推荐系统研究的不断发展,涌现出大量不同的推荐思想. 这些思想可以分为两类:一类是根据信息过滤的方式来分类,可以分为协同过滤、基于内容的过滤、社会化过滤和混合式过滤等;另一类是根据所建

立的模型来分类,可以分为基于领域的模型、基于图的推荐模型和基于矩阵分解的推荐模型等。

本文采用的是协同过滤算法,其推荐思想是通过挖掘用户的历史行为数据记录来给用户推荐他们可能感兴趣的物品。文献[15]第一次提出了协同过滤的推荐思想,并提出基于用户的协同过滤思想。在此基础上,文献[16]进一步提出基于物品的协同过滤思想。

基于物品的协同过滤思想指出,用户会喜欢跟他以前喜欢的物品相似的物品。因此,根据该思想给用户推荐他们感兴趣的物品,则必须先获得他以前喜欢的物品信息,然后找到和这些物品相似的物品并推荐给该用户。与基于用户的协同过滤不同,该算法思想要求计算两个(或多个)物品的相似度。但与之相似的是,该算法思想通过两个(或多个)物品被同时喜欢的用户数的多少来衡量相似度的大小。假设给定物品 i 和 j ,令 $N(i)$ 为喜欢物品 i 的用户集, $N(j)$ 为喜欢物品 j 的用户集,则物品 i 和物品 j 的相似度可以通过如下公式衡量:

$$\text{sim}(i, j) = \frac{|N(i) \cap N(j)|}{|N(i)|}. \quad (1)$$

用户在间隔很短的时间内产生的行为往往具有更高的相似度,因此本文中使用的物品相似度计算方法采用下式^[17]:

$$\text{sim}(i, j) = \frac{\sum_{u \in N(i) \cap N(j)} \frac{1}{1 + \alpha |t_{ui} - t_{uj}|}}{\sqrt{|N(i)| |N(j)|}}. \quad (2)$$

式中: α 是时间衰减参数,其取值根据推荐系统的特点不同而不同,如果系统中用户的兴趣变化快,则应当取较大的 α ,否则取较小的 α ; $|t_{ui} - t_{uj}|$ 表示用户 u 对于产品 i 和产品 j 产生行为的时间间隔。

计算出物品之间的相似度后,基于物品的协同过滤算法通过如下公式来衡量用户 u 对物品 j 的兴趣大小:

$$p(u, j) = \sum_{i \in S(j, k) \cap N(u)} \text{sim}(j, i). \quad (3)$$

其中, $N(u)$ 是用户 u 喜欢的物品数, $S(i, k)$ 表示与物品 i 最相似的 k 件物品, r_{ui} 表示用户 u 对物品 i 的兴趣偏好。

类似式(2)近期的行为更能体现用户当前的兴趣,应当在推荐时对其增加一定的权重^[17],则修正公式(3)为

$$p(u, j) = \sum_{i \in S(j, k) \cap N(u)} \text{sim}(j, i) \frac{1}{1 + \beta |t_0 - t_{ui}|}. \quad (4)$$

式中 t_0 是当前时刻, β 是衰减系数。

1.2 基于交通网络数据优化的地理信息推荐算法

1.2.1 基于经纬度的距离计算

基于地点上下文的推荐利用交通网络数据来度量地图上两点之间的距离,因此需要利用各个地点的经

纬度进行计算(本文使用百度地图提供的 Geocoding API 进行地理信息编码的操作)。在进行空间中不同点之间的距离计算后,推荐算法会根据两个地点间的距离远近对推荐结果进行调整。首先,使用最基本的两点间距离计算的方法,直接度量两点间的欧几里德距离。由于地球是一个椭球体,而不是平面,因此不能直接对经纬度进行欧几里德距离计算。式(5)所示为地球上两点间经纬度距离计算的公式:

$$D = 2 \arcsin \sqrt{\sin^2 \frac{a}{2} + \cos \varphi_1 \cos \varphi_2 \sin^2 \frac{b}{2}} \times 6378.137. \quad (5)$$

式中 φ 为纬度, a 为两点之间的纬度的差值, b 为两点之间经度的差值,而常数 6378.137 为地球半径(单位为 km),这里所有的经纬度值都需要转化为弧度制。

根据式(5)可以对使用经纬度表示的任意两点进行欧几里德距离的计算。然而,直线距离更近的两个点并不能够代表这两个点的真实距离更近。例如,如图2所示,图中两条横向直线为道路两侧,纵向图形为行人天桥,除该人行天桥外用户无法穿过该道路, A、B 和 C 为三个商户位置,用户当前处于 A 处。可以看出用户在 A 点时距离 C 更近,应向其推荐 C 处的商户。然而,从 A 到 C 需先经过 B 之后过天桥才能到达,远超过 A 到 B 的距离。因此,向用户推荐 C 处的商户排名不应该高于 B 处商户的排名。

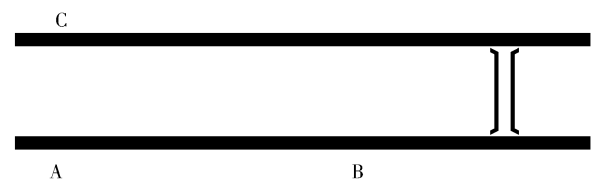


图2 直线距离非最近距离的示意图

Fig.2 Schematic diagram of closest distance

通过图2所示的简单示例可以看出,对于推荐系统而言,应当向用户推荐实际距离更近的物品,而不是直线距离最近的物品。在实际生活中,这种实际距离远大于直线距离的情况十分常见。

1.2.2 基于交通网络数据的距离计算

基于交通网络数据进行距离计算可以提高推荐系统推荐结果的准确性,为用户提供更好的使用体验。因此,在本节中将给出利用交通网络数据进行距离计算的方法。

为了使用交通网络数据进行距离计算,必须借助于成熟的地理信息系统,使用其关于两点间距离计算或路径规划的 API 来获取到它们的实际距离。使用百度地图所提供的 Direction API 来进行距离计算,该 API 可以进行步行、驾车以及公交换乘三种形式的路径查询,并返回 json 格式或者 xml 格式的查询结果数据。

Direction API 可以直接使用经纬度作为起终点来进行检索。

Direction API 是通过程序构造 HTTP 请求 URL 的方式向地图 API 提供方请求数据,由于用户具有活动本地化的特点,其活动范围不会过大,因此使用步行模式进行检索。表1所示是 Direction API 的部分重要请求接口参数。

表1 Direction API 请求参数

Table 1 Request parameters of Direction API

参数	是否必须	默认值	含义
origin	是		起点地址或经纬度信息
destination	是		终点地址或经纬度信息
mode	否	driving	导航模式
region	是		城市
output	否	xml	结果输出形式(json/xml)
tactics	否	11(最少时间)	导航策略

根据本文中实验的具体情况,设置导航模式 mode 为 walking(步行),导航策略为 12(最短路径)。

表2所示为使用了步行导航模式后返回的 JSON 格式的数据中部分重要的字段信息。

表2 Direction API 返回结果字段

Table 2 Return resultfields of Direction API

字段名	含义	备注
status	状态码	0: 成功
message	状态码对应信息	“ok”: 成功
type	返回数据类型	1: 模糊 2: 精确
distance	距离	单位: m
duration	耗时	单位: s

实际上,除了表2中提到的一些返回结果字段以外,Direction API 返回的结果 JSON 中还包含了大量的信息,如路线方案、起终点具体信息和出租车信息。不过,对于本文而言,这些信息并没有太大的意义,只需要得到路线方案的总距离即可。对于步行而言,耗时与距离往往是成正比的关系,而如果采用公交或驾车模式的话,耗时和距离可能会因为路况不同等原因造成不成比例。

1.2.3 地理信息推荐算法

两个地点之间的实际距离获得后,把地点上下文信息加入到推荐算法当中。算法的流程如下所示:

①构建上下文无关的物品相似度矩阵;

②构建物品距离矩阵;

③对于每个用户 u , 根据式(6)计算该用户对各物品的兴趣度,选取兴趣度最高的 N 个作为对其的推荐结果。

$$p(u, i) = \sum_{j \in N(u) \cap S(i, k)} \frac{\text{sim}(i, j)}{1 + \gamma d_{ij}} \quad (6)$$

式中 γ 表示针对该系统而言距离属性所占的比重,不同的系统 γ 值不同, γ 值越大则表示距离属性在该系统中所占有的地位越低,值越小表明该系统对于地点上下文的依赖性越大。 d_{ij} 表示物品 i 和物品 j 的交通网络距离。从该式中可以看出,两点间的实际距离越远,经过衰减函数后用户对其产生行为的可能性越小。

通过式(6),可以计算出对于每个用户的推荐列表。其具体思路为:对该用户的所有既往行为,分别从相似度矩阵中找到与该行为最相似的 N 个物品,求得用户 u 对这些物品的兴趣度,再排序得到这些物品中兴趣度最高的 N 个,作为推荐结果列表。

对于本算法的步骤①来说,构造上下文无关的物品相似度矩阵,使用协同过滤推荐算法;式(2)时间衰减参数 α 取 0.1;式(4)衰减系数 β 取 0.5。步骤②构造物品距离矩阵的目的是为了减少实验中可能会出现的大量重复的 API 的请求。由于本实验中采用离线实验的方式,通过分析训练集中该用户的行为,找出与这些行为距离相近的地点,因此本实验中地点数量有限,也不会出现新地点,可以通过构造距离矩阵的方式事先保存每两点之间的实际距离,在后续的实验中将不再需要向地图 API 提供商发送请求,大大降低了实验所需耗费的时间。

下面是基于地理信息推荐算法的伪代码,其中 W 是上下文无关的物品相似度矩阵, D 是物品距离矩阵, γ 是衰减系数。

```
def recommendation( train, uid, W, D, n, gamma ) {
    rank = dict()
    for( i in train[uid] ) {
        sorted_W = sorted( W[i].items(), key = lambda
            tran: tran[1], reverse = True ) [0:n]
        for( j, wij in sorted_W ) {
            if( j in train[uid] )
                continue
            decay = 1/(1 + gamma * D[i][j])
            rank[j] += wij * decay
        }
    }
    rank = sorted( rank.items(), key = lambda tran:
        tran[1], reverse = True ) [0:n]
    return rank
}
```

2 实验结果与分析

2.1 数据集的选取与处理

本文所使用的数据集来源于北京银联商务提供的

银行卡持卡人消费数据. 这组数据中包含有如下一些列: MERCHID(商户号)、TRANDATE(交易日期)、TRAN TIME(交易时间)、CARDNO(银行卡号) 和 TRANAMT(交易金额). 此外, 还有一个商户地址对应表, 每列包含 MERCHID(商户号)、MERCHNAME(商户名称) 和 ADDRESS(商户地址). 商户地址对应表将作为地理信息使用.

首先将这个数据集进行预处理, 得到一个包含有如下项目的新数据集.

ItemID: 对商户号 MERCHID 进行处理, 生成一个 ItemID-MERCHID 对应字典, 即每遇到一个新的商户号, 生成一个自增 1 的 ItemID, 将 ItemID 与 MERCHID 的对应关系添加到字典中. 进行此步操作的原因是 MERCHID 是一个根据特定格式表示的纯数字 ID, 其位数较长, 转化后可以减少实验执行和数据保存时所占用的空间大小.

UserID: 对银行卡号进行处理, 生成一个 UserID-CARDNO 对应字典, 即每遇到一个新的银行卡号, 生成一个自增 1 的 UserID, 将 UserID 与 CARDNO 的对应关系添加到字典中. 此操作和 ItemID 操作的意义相同.

由于对于不同的 ItemID 拥有不同的商户名称和商户地址, 因此可以认为 ItemID 中已经包含有地点上下文, 但并没有包含可以用于计算的地点上下文. 如果将商户的地理信息放置于数据集中, 会造成数据集中存在大量无关的冗余项, 因此生成一个 ItemID 与地理信息(经纬度) 对应的新的字典 geo_dict. 在 geo_dict 中的每一行包含如下三列.

ItemID: 与数据集中 ItemID 对应;

lat: 纬度;

lng: 经度.

这里的经纬度是将原商户地址对应表中的商户地址信息编码而成的. 对于那些置信度过低的地址信息和无法取得有效结果的地址信息, 采取手工补充的方式填入该矩阵.

经过上述处理后, geo_dict 字典可以用于生成距离矩阵, 结合新的数据集进行商户推荐.

对于训练集和测试集的生成而言, 用户行为的先后顺序不会对地点上下文感知推荐系统造成影响, 因此本文中的实验使用一般形式的交叉验证来进行. 进行多次实验, 每次实验中选取 90% 的行为作为训练集, 剩余 10% 的行为作为测试集. 根据训练集生成每个用户的推荐列表, 在测试集上验证得出准确率和召回率, 并通过多次实验后求出其均值, 作为该实验的结果.

2.2 评价指标

平均绝对误差 MAE(mean absolute error) 是评估

推荐系统质量的重要标准, 它通过计算预测结果与实际结果之间的偏差来度量预测的准确性, 平均绝对误差的值越小, 推荐的准确性越高^[18]. 其计算方法如下:

$$MAE = \frac{\sum_{j=1}^{|CI_j|} |p_{ij} - r_{ij}|}{|CI_i|}. \quad (7)$$

式中 p_{ij} 为预测的当前用户 i 对候选项目 j 的评分(即式(6) 计算所得预测结果) r_{ij} 为测试集中用户 i 项目 j 的实际评分, $|CI_i|$ 为对用户 i 进行推荐的候选项目数.

覆盖率是指算法向用户推荐的商品能覆盖全部商品的比例. 如果推荐系统的覆盖率较低则系统很可能由于其推荐范围的局限性降低用户的满意度^[19]. 其定义为

$$COV = \frac{N_d}{N}. \quad (8)$$

式中 N_d 为系统可以预测的商品数目, N 为所有商品数目.

此外, 由于本文中提供的是 TopN 类型的推荐系统, 而 TopN 推荐质量的衡量常采信息检索领域评价系统的标准, 即准确率和召回率.

准确率 R_p 是指推荐正确的物品占推荐列表中物品的比例, 可以用下式表示:

$$R_p = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|}. \quad (9)$$

召回率 R_r 是指推荐正确的物品占测试集该用户产生行为的物品列表, 可以用下式表示:

$$R_r = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}. \quad (10)$$

式中 μ 为用户, $R(u)$ 是根据用户在训练集上的行为给用户作推荐列表, 而 $T(u)$ 是用户在测试集上的行为列表.

2.3 结果分析

首先, 分析在不同 γ 值下本系统的准确率和召回率指标($N=5$), 如表 3 所示. 通过表 3 可以看出, 选用不同的 γ 值对于本系统的实验结果有着比较重要的影响. γ 值越大, 衰减函数的值造成的影响也就越小, 推荐系统的推荐结果准确性的变化幅度也越小. 对于本系统而言, 当 γ 值为 0.0001 时, 推荐结果最好, 而当 γ 值逐渐增大时, 由于距离信息对于公式的影响逐渐变小, 其推荐结果也会逐渐降低准确度. 为了保证变量的唯一性, 下文中将 γ 值固定为 0.0001, 并且将相似度矩阵取为未经过时间优化的相似度矩阵.

图 3 所示为上下文无关的推荐系统与基于地理信息优化后的推荐系统的准确率-召回率曲线对比, N 取值分别为 5、10、20、30、40 和 50. 从图 3 中可以看出,

表3 不同 γ 值下的准确率和召回率

Table 3 Accuracy and recall of different gamma values

γ	准确率 R_p	召回率 R_r
0.00001	0.0423	0.2075
0.0001	0.0441	0.2161
0.001	0.0435	0.2131
0.01	0.0435	0.2130
0.1	0.0404	0.2129

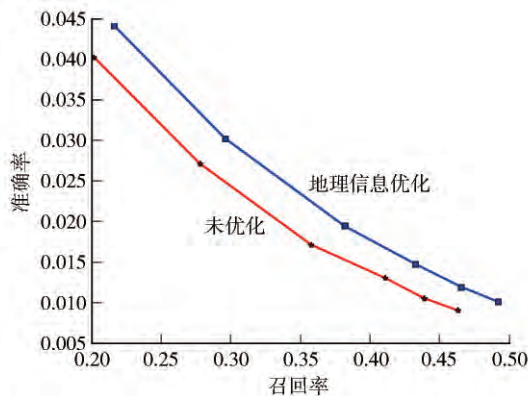


图3 两种推荐系统准确率-召回率曲线

Fig. 3 Accuracy-recall curves of 2 kinds of recommender systems

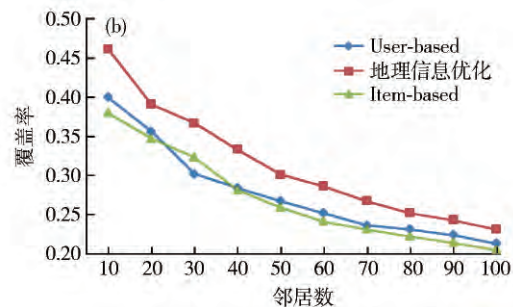
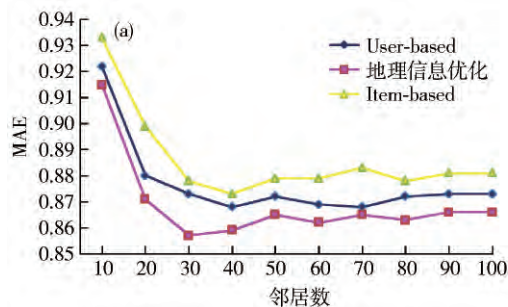


图4 三种算法 MAE(a) 和覆盖率(b) 比较

Fig. 4 MAE(a) and coverage(b) comparison between three algorithms

3 结论

基于协同过滤算法,结合地点上下文,利用真实数据,提出一种基于交通网络数据优化的地理信息推荐算法。首先利用传统的协同过滤算法对数据集进行预处理,而后将地点上下文加入进来,生成包含基于交通网络数据地点上下文的新的数据集,再结合地理信息推荐算法进行实验。实验结果表明,结合交通网络数据的地理信息进行优化的推荐系统推荐质量获得明显提升。

参考文献

- [1] Wang L C, Meng X W, Zhang Y J. Context-aware recommender systems. *J Software*, 2012, 23(1): 1
(王立才,孟祥武,张玉洁. 上下文感知推荐系统. 软件学报,

包含了地点上下文信息的推荐系统明显优于未包含地点上下文的推荐系统。

为了检验优化算法的效果,以传统的 User-based 过滤推荐算法和 Item-based 过滤算法^[20]为参照,以余弦相似性和相关相似性作为相似度量标准,分别计算其 MAE 和覆盖率,然后与本文算法进行比较。在实验中,选取推荐数目 $N=20$,邻居数目取 10~100,不同邻居数情况下的实验对比结果如图 4 所示。从图 4 可以看出,在相同度量方法以及相同邻居数的条件下,本文方法与传统的 User-based 过滤推荐算法和 Item-based 过滤算法相比,平均绝对偏差低且覆盖率高,说明本文优化算法预测质量比前两种算法在整个区间内都稍好一些。另外,从图 6(a)中还可以看出,平均绝对偏差(MAE)变化(减少)的非常有限,并且曲线趋于平滑,体现了本文算法具有较好的稳定性和鲁棒性。

由于人们的活动范围往往局限于一个不大的圈子中,较远的距离对于用户消费而言有着比较大的时间开销,对于一些在多个地点均可以进行的消费来说,用户更倾向于选择距离自己最近的地点进行消费。实验结果表明,经过交通网络距离优化后的地点上下文感知推荐系统各项指标相比于传统的方法以及未优化系统有所提高,推荐质量更佳。

2012, 23(1): 1)

- [2] Adomavicius G, Tuzhilin A. Personalization technologies: a process-oriented perspective. *Commun ACM*, 2005, 48(10): 83
- [3] Baltrunas L, Amatriain X. Towards time-dependant recommendation based on implicit feedback // *Workshop on Context-aware Recommender Systems (CARS'09)*. New York, 2009
- [4] Ricci F. Mobile recommender systems. *Inf Technol Tour*, 2010, 12(3): 205
- [5] Panniello U, Tuzhilin A, Gorgoglione M, et al. Experimental comparison of pre- vs. post-filtering approaches in context-aware Recommender Systems // *Proceedings of the Third ACM Conference on Recommender Systems*. New York, 2009: 265
- [6] Karatzoglou A, Amatriain X, Baltrunas L, et al. Multiverse recommendation: n -dimensional tensor factorization for context-aware collaborative filtering // *Proceedings of the Fourth ACM Conference on Recommender Systems*. New York, 2010: 79

- [7] Hidasi B, Tikk D. Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback//*Machine Learning and Knowledge Discovery in Databases*. Bristol 2012: 67
- [8] Buriano L, Marchetti M, Carmagnola F, et al. The role of ontologies in context-aware recommender systems//*7th International Conference on Mobile Data Management*. Nara, 2006: 80
- [9] Palmisano C, Tuzhilin A, Gorgoglione M. Using context to improve predictive modeling of customers in personalization applications. *IEEE Trans Knowl Data Eng*, 2008, 20(11): 1535
- [10] Wang L C. *Research on Key Technologies for Context-aware Recommender Systems* [Dissertation]. Beijing: Beijing University of Posts and Telecommunications, 2012
(王立才. 上下文感知推荐系统若干关键技术研究[学位论文]. 北京: 北京邮电大学, 2012)
- [11] Xu Y M, Jiang W, Wang Y C. Personalized recommendation of mobile devices in context-aware application. *Microcomput Inf*, 2009, 21: 186
(徐月美, 姜薇, 王溢策. 移动设备的个性化推荐在上下文感知应用. 微计算机信息, 2009, 21: 186)
- [12] Yang W S, Cheng H C, Dia J B. A location-aware recommender system for mobile shopping environments. *Expert Syst Appl*, 2008, 34(1): 437
- [13] Hurley N J. Robustness of recommender systems//*Proceedings of the Fifth ACM Conference on Recommender Systems*. New York, 2011: 9
- [14] Bao J, Chow C Y, Mokbel M F, et al. Efficient evaluation of k -range nearest neighbor queries in road networks//*11th International Conference on Mobile Data Management*. Kansas City, 2010: 115
- [15] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews//*Proceedings of ACM Conference on Computer Supported Cooperative Work*. New York, 1994: 175
- [16] Linden G, Smith B, York J. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput*, 2003, 7(1): 76
- [17] Shao K Y, BanX J, Li B. Context-aware recommender systems optimized by item periodicity. *J Comput Inf Syst*, 2014, 10: 867
- [18] Zhang X W. *The Research on Collaborative Filtering Algorithm in Artificial Recommendation System* [Dissertation]. Shanghai: Shanghai Jiao Tong University, 2008
(张雪文. 智能推荐系统中协同过滤算法的研究[学位论文]. 上海: 上海交通大学, 2008)
- [19] Zhu Y X, Lü L Y. Evaluation metrics for recommender systems. *J Univ Electron Sci Technol China*, 2012, 41(2): 163
(朱郁筱, 吕琳媛. 推荐系统评价指标综述. 电子科技大学学报, 2012, 41(2): 163)
- [20] Lu W. *Study of Recommendation System Based on Collaborative Filtering Algorithm* [Dissertation]. Beijing: Beijing University of Posts and Telecommunications, 2007
(鲁为. 协作过滤算法及其在个性化推荐系统中的应用[学位论文]. 北京: 北京邮电大学, 2007)