

# 基于项目之间相似性的兴趣点推荐方法<sup>\*</sup>

邹永贵<sup>1</sup>, 望靖<sup>1</sup>, 刘兆宏<sup>1</sup>, 夏英<sup>1,2</sup>

(1. 重庆邮电大学空间信息系统研究所, 重庆 400065; 2. 重庆市地理空间信息工程技术研究中心, 重庆 401121)

**摘要:** 针对评分数据稀疏的情况下传统相似性计算的不足, 提出了一种基于项目之间相似性的协同过滤算法。该算法结合用户对项目的评分和项目之间的兴趣度进行项目之间的相似性计算, 在一定程度上减小了评分数据稀疏的负面影响。实验结果表明, 该算法在评分数据稀疏的情况下, 能使推荐系统的推荐质量明显提高。

**关键词:** 兴趣点; 推荐系统; 协同过滤; 相似性; 项目兴趣度

中图分类号: TP391 文献标志码: A 文章编号: 1001-3695(2012)01-0116-03

doi:10.3969/j.issn.1001-3695.2012.01.032

## Point of interest recommendation method based on similarity between items

ZOU Yong-gui<sup>1</sup>, WANG Jing<sup>1</sup>, LIU Zhao-hong<sup>1</sup>, XIA Ying<sup>1,2</sup>

(1. Research Center of Spatial Information System, Chongqing University of Posts & Telecommunications, Chongqing 400065, China; 2. Research Center of Chongqing Geospatial Information Engineering & Technology, Chongqing 401121, China)

**Abstract:** To solve the problems of traditional similarity measure methods with user rating data sparsity, this paper proposed a novel collaborative filtering algorithm based on item similarity, which combined user rating data with interest degree of items to calculate similarity between two items, so that it could overcome the effect of sparsity of user rating data. The experimental results show that the proposed algorithm can obviously enhance the quality of recommendation system in the case of sparsity of user rating data.

**Key words:** point of interest (POI); recommendation system; collaborative filtering; similarity; item interest degree

## 0 引言

兴趣点 (POI) 查询是指根据用户的喜好搜索他们感兴趣信息的一种基于位置的服务。随着信息技术的发展, 用户可以利用手机、互联网、报章杂志等多种途径来获得自己感兴趣的 POI 信息。但是, 查询所获得的反馈信息往往是大量的, 这使得用户难以在短时间内从中获得有效信息。为了能够向用户提供快速有效的信息, 推荐系统逐渐成为一个重要的研究内容。协同过滤被认为是推荐系统中最为有效和普遍使用的推荐技术<sup>[1]</sup>, 其基本思想是通过最近邻居对项目评分的相似性, 向用户提供推荐信息。

在协同过滤算法中, 确定最近邻居是整个算法的重点。然而, 最近邻居是通过计算目标项目与其他项目之间的相似程度而决定的。如何准确地计算项目之间的相似性, 是决定整个推荐系统效率的关键所在。但是, 由于评分数据的稀疏性、扩展性<sup>[2]</sup>等因素的影响, 加上传统相似性计算方法存在的弊端, 大大降低了计算项目最近邻居的准确性, 从而影响了推荐系统的质量。针对评分数据的稀疏性对推荐系统的影响, 本文提出了一种基于项目之间相似性的协同过滤算法。该算法在传统相似性计算只考虑用户评分的基础上, 结合项目之间的兴趣度来进行相似性的计算, 以提高推荐系统的推荐质量。

## 1 相关工作

根据事物之间的关联性, Sarwar 等人<sup>[3]</sup>将协同过滤算法分成了基于用户的协同过滤算法 (user-based) 和基于项目的协同过滤算法 (item-based)。基于用户的协同过滤算法是以用户之间行为的某种相似性为依据 (如购买行为类似的顾客会购买类似的产品); 基于项目的协同过滤算法则是基于顾客购买产品时的项目间的某种关联性 (如在购买单反相机的同时会购买内存卡和镜头)。由于项目之间的相似性相对稳定<sup>[4]</sup>, item-based 协同过滤算法相关计算都可以离线进行, 从而在用户数量远远大于项目数量时更具有优越性。而且, 基于项目的协同过滤算法在扩展性上具有优势<sup>[3]</sup>, 因此本文采用 item-based 协同过滤算法对兴趣点进行推荐。

传统的基于项目的协同过滤算法产生推荐有三个步骤<sup>[3]</sup>: a) 获取能代表项目的信息, 如用户对项目的评分信息; b) 通过用户—项目评分矩阵, 计算目标项目和其他项目之间的相似性, 并根据相似程度来选定目标项目的最近邻居; c) 根据选出来的最近邻居产生推荐。可见, 相似性的计算在整个推荐过程中起着承上启下的作用。选择适当的相似性计算方法, 可以有效地提高整个推荐系统的推荐质量。

收稿日期: 2011-06-15; 修回日期: 2011-07-26 基金项目: 重庆市科委科技项目 (CSTC 2009CB2015); 中韩国际合作项目 (C2010-02)

作者简介: 邹永贵 (1969-), 男, 四川人, 教授, 硕士, 主要研究方向为空间数据库 (zouyg@cqupt.edu.cn); 望靖 (1986-), 女, 重庆人, 硕士研究生, 主要研究方向为数据挖掘; 刘兆宏 (1975-), 男, 四川人, 讲师, 硕士, 主要研究方向为空间数据库; 夏英 (1972-), 女, 重庆人, 教授, 博士研究生, 主要研究方向为时空数据挖掘。

### 1.1 传统相似性计算方法

传统的相似性计算方法有很多,其中较典型<sup>[2]</sup>的有余弦相似性(cosine)、修正余弦相似性(adjusted cosine)和相关相似性(correlation)。其计算方式分别如下:

a) 余弦相似性。用户对项目  $i$  和  $j$  的评分被看成是  $n$  维空间上的向量,如果用户对项目没有进行评分,则将该用户对该项目的评分设为 0,项目之间的相似性通过向量之间的余弦夹角来度量。项目  $i$  与  $j$  之间的相似度为

$$\text{sim}(i, j) = \cos(i, j) = \frac{i \cdot j}{\|i\| \times \|j\|} = \frac{\sum_{u=1}^n R_{u,i} R_{u,j}}{\sqrt{\sum_{u=1}^n R_{u,i}^2} \sqrt{\sum_{u=1}^n R_{u,j}^2}} \quad (1)$$

其中:分子代表两个项目评分向量的内积,分母为两个向量模的乘积; $R_{u,i}$  和  $R_{u,j}$  分别代表用户  $u$  对项目  $i$  和  $j$  的评分。

b) 修正余弦相似性。它是在余弦相似性度量方法上的改进,通过减去用户对项目的平均评分的方法来进行改善,从而在一定程度上弥补了余弦相似性没有考虑不同用户评分尺度的缺陷。所以,项目  $i$  与  $j$  之间的相似性表示为

$$\text{sim}(i, j) = \frac{\sum_{u \in U_{ij}} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U_i} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U_j} (R_{u,j} - \bar{R}_u)^2}} \quad (2)$$

其中:对项目  $i$  和  $j$  共同评过分的用户用集合  $U_{ij}$  来表示, $U_i$  和  $U_j$  分别表示对项目  $i$  和  $j$  评过分的用户集合; $R_{u,i}$  表示用户  $u$  对项目  $i$  的评分; $\bar{R}_u$  表示用户  $u$  的平均评分。

c) 相关相似性。Pearson 相关系数常用来计算两个变量之间的线性关系。项目  $i$  和  $j$  之间的相关相似性表示为

$$\text{sim}(i, j) = \frac{\sum_{u \in U_{ij}} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U_i} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U_j} (R_{u,j} - \bar{R}_j)^2}} \quad (3)$$

其中:假设对项目  $i$  和  $j$  共同评过分的用户集合用  $U_{ij}$  表示; $R_{u,i}$  表示用户  $u$  对项目  $i$  的评分; $\bar{R}_i$  和  $\bar{R}_j$  分别表示项目  $i$  和  $j$  的平均评分。

### 1.2 传统相似性计算方法分析

传统的相似性计算方法大多是通过用户对项目的评分来计算项目之间的相似性。但是,在大多数推荐系统中,由于用户评分的项目往往不超过总项目数的 1%<sup>[5]</sup>,所以导致了评分数据的极端稀疏性,使得传统的相似性计算方法不能准确地计算出项目之间的相似性。下面详细分析传统相似性计算方法存在的问题。

余弦相似性计算方法中,将用户没有评分的项目评分值设为 0,这样做虽然有效地提高了计算性能,但事实上用户对未评分项目的喜好程度不可能全都相同且为 0。所以,在评分数据稀疏的情况下,余弦相似性计算方法就无法准确地计算项目之间的相似性。修正余弦相似性计算方法也存在同样的问题。

相关性相似性计算方法中,需要计算对项目  $i$  和  $j$  共同评过分的用户集合,然后通过这个集合里的用户对项目的评分来计算项目之间的相似性。用这种方式来处理,比余弦相似性计算方法中直接用 0 来填充具有更好的推荐质量。但是,由于评分数据稀疏,导致对项目间共同评过分的用户少之又少,这使得相关相似性计算方法难以发挥作用。而且,在相关相似性计算方式的情况下,很可能遇到分母为 0 的情况,这时计算公式就失去了意义。例如,有三个用户  $x, y, z$  对项目  $i$  和  $j$  进行评

分,并且评分都分别为 3 分。那么此时  $\bar{R}_i$  和  $\bar{R}_j$  就为 3,  $R_{u,i}$  和  $R_{u,j}$  也为 3。这种情况下,分母等于 0。在推荐系统中,由于用户评分的不确定性,这种情况会很常见。即使将分母为 0 的情况进行特殊处理(如分母为 0 时将 sim 赋为一个特殊值),这种方法也不够理想,因为不能一概而论地一旦遇到这种分母为 0 的情况时,就认为那些项目之间都有相同的相似性。

综上所述,传统的三种相似性计算方法都不能有效地度量项目之间的相似性,从而使计算出来的目标项目的最近邻居不准确,导致整个算法的推荐质量下降。为了解决以上问题对相似性计算的影响,众多学者提出了各种方法来解决这个问题。文献[6]中将位置信息作为一个关键因素引入协同过滤算法对用户 POI 进行推荐。该算法有效地提高了系统的推荐精度,但在计算相似性的时候仍然使用传统相似性计算方法,从而没有根本解决评分稀疏性对推荐质量的影响。文献[7]提出了一种结合用户和项目属性的混合协同过滤推荐算法。该算法有效地提高了在评分数据稀疏的情况下算法的推荐质量。但是在实际应用中,某个与推荐相关的应用很有可能难以采集到用户的多种属性(如手机上的推荐相关应用,不用填写用户资料就可以使用)。所以,该文献提出的方法在运用到实际生活中时就会出现时就会出现问题。

## 2 基于项目之间相似性的协同过滤算法

传统相似性计算方法存在诸多弊端,不能准确地计算出项目之间的相似性,从而降低了推荐系统的质量。现有的基于项目的协同过滤推荐算法<sup>[3,4,8]</sup>大多是从用户评分的角度来计算项目之间的相似性,而忽略了项目之间的兴趣度对相似性计算的影响。为此,本文提出了一种基于项目之间相似性的协同过滤算法,该算法将用户评分和项目之间兴趣度相结合来进行相似性的计算,以减小评分数据稀疏的情况对系统推荐质量的影响。

为了避免在传统相似性计算方法中相关相似性计算方法分母为 0 的情况,改用绝对指数相似性计算方法来计算项目的评分相似性和项目之间兴趣度的相似性。公式如下<sup>[9]</sup>:

$$\text{sim}(i, j) = e^{-\sum_{u=1}^n |R_{u,i} - R_{u,j}|} \quad (4)$$

其中: $u$  为用户; $R_{u,i}$ 、 $R_{u,j}$  分别为用户  $u$  对项目  $i$  和  $j$  的评分。其实,余弦相似性计算方法、相关相似性计算方法以及绝对指数相似性计算方法都是属于模糊聚类中,计算被分类对象间相似性程度统计量的统计方法<sup>[9]</sup>,统计方法一共有 12 种。在这 12 种统计方法中,具体用哪个方法,要根据问题的性质来选取。因为绝对指数相似性度量方法能有效地避免在相关相似性计算方法中分母为 0 的情况,所以进行替换。

### 2.1 基于项目之间兴趣度的相似性计算方法

在实际生活中,当两个项目之间拥有相似的兴趣度时,可以简单地认为它们之间存在着一定的相似性。项目之间的兴趣度特征向量用  $I_{i,k}$  表示:

$$I_{i,k} = N_{ik} / N_k \quad (5)$$

其中: $N_{ik}$  表示项目  $i$  被评过分的次数; $N_k$  表示项目  $i$  所属类别中所有项目被评过分的总次数; $k$  表示项目的类别。

将上一步算出来的项目兴趣度特征向量代入绝对指数相似性计算方法式(4),计算两个项目之间的兴趣度相似性:

$$\text{sim}_i(i, j) = e^{-\sum_{k=1}^n |I_{i,k} - I_{j,k}|} \quad (6)$$

其中:  $I_{i,k}$  表示项目之间的兴趣度向量;  $k$  代表项目的类别;  $n$  为类别数。

项目之间的评分相似性用  $\text{sim}_r$  表示:

$$\text{sim}_r(i, j) = e^{-\sum_{u \in U_{ij}} |R_{u,i} - R_{u,j}|} \quad (7)$$

其中:  $U_{ij}$  为对项目  $i$  和  $j$  共同评分过的用户集合;  $R_{u,i}$ 、 $R_{u,j}$  为用户  $u$  分别对项目  $i$  和  $j$  的评分。

本文将传统的只考虑用户对项目评分的相似性计算方法和基于项目之间兴趣度的相似性计算方法相结合的方式, 来进行项目之间相似性的计算。其计算公式表示为

$$\text{sim}(i, j) = (1 - w) \text{sim}_r(i, j) + w \text{sim}_i(i, j) \quad (8)$$

其中:  $w$  和  $(1 - w)$  表示用户对项目评分相似性计算方法和项目之间兴趣度相似性计算方法的相对重要程度。如果  $w = 0$ , 则变成了传统的只考虑用户评分的相似性计算方法; 若  $w = 1$ , 则是基于项目之间兴趣度的相似性计算方法。

### 2.2 基于项目之间相似性的协同过滤算法

**算法** 基于项目之间相似性的协同过滤算法

输入: 用户—项目评分矩阵, 项目—项目类别矩阵。

输出: 推荐项目集合。

a) 输入用户—项目评分矩阵  $R$  和项目—项目类别矩阵  $A$ :

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1i} \\ r_{21} & r_{22} & \dots & r_{2i} \\ \dots & \dots & \dots & \dots \\ r_{u1} & r_{u2} & \dots & r_{ui} \end{bmatrix}, A = \begin{bmatrix} N_{11} & N_{12} & \dots & N_{1k} \\ N_{21} & N_{22} & \dots & N_{2k} \\ \dots & \dots & \dots & \dots \\ N_{i1} & N_{i2} & \dots & N_{ik} \end{bmatrix}$$

其中: 矩阵  $R$  中, 行代表用户, 用户数为  $u$ , 列代表项目, 项目数为  $i$ ,  $r_{ui}$  代表用户  $u$  对项目  $i$  的评分, 如果用户没有对项目进行评分, 则  $r_{ui}$  为 0; 矩阵  $A$  中, 行代表项目, 项目数为  $i$ , 列则代表项目的种类, 种类数为  $k$ ,  $N_{ik}$  代表项目  $i$  被评分过的次数。

b) 对矩阵  $A$  用式(5) 计算项目的兴趣度特征向量  $I_i = (I_{i,1}, I_{i,2}, \dots, I_{i,k})$ 。

c) 将  $I_i$  代入式(6) 计算项目之间的兴趣度相似性  $\text{sim}_i$ 。

d) 对矩阵  $R$  用式(7) 来计算项目之间的评分相似性  $\text{sim}_r(i, j)$ 。

e) 计算项目的相似性矩阵。由式(8) 可知项目之间的相似性包括用户对项目的评分相似性和项目之间的兴趣度相似性。根据式(6) 和(7) 分别计算出  $\text{sim}_i(i, j)$  和  $\text{sim}_r(i, j)$ , 然后根据式(8) 计算出任意两个项目之间的相似性为  $\text{sim}(i, j)$ 。项目的相似性矩阵  $\text{sim}(i, j)$  表示为

$$\text{sim}(i, j) = \begin{bmatrix} \text{sim}(1,1) & \text{sim}(1,2) & \dots & \text{sim}(1,j) \\ \text{sim}(2,1) & \text{sim}(2,2) & \dots & \text{sim}(2,j) \\ \dots & \dots & \dots & \dots \\ \text{sim}(i,1) & \text{sim}(i,2) & \dots & \text{sim}(i,j) \end{bmatrix}$$

f) 根据相似性矩阵  $\text{sim}(i, j)$ , 选取与目标项目  $i$  相似度最高的  $s$  个项目组成最近邻居集合:  $NN_i = \{NN_1, NN_2, \dots, NN_s\}$ 。

g) 产生推荐。对最近邻居进行加权求和来预测用户  $u$  对目标项目  $i$  的预测评分, 公式如下<sup>[3]</sup>:

$$P_{u,i} = R_i + \frac{\sum_{n \in NN_i} \text{sim}(i, n) \times (R_{u,n} - \bar{R}_n)}{\sum_{n \in NN_i} (|\text{sim}(i, n)|)} \quad (9)$$

其中:  $\text{sim}(i, n)$  表示目标项目  $i$  与最近邻居  $n$  之间的相似性;  $R_{u,n}$  表示用户  $u$  对项目  $n$  的评分;  $R_i$  和  $\bar{R}_n$  分别表示对项目  $i$  和

项目  $n$  的平均评分。

## 3 实验结果及其分析

### 3.1 数据集

本文数据集由重庆“趁热网”(http://www.chenre.cn/) 提供。“趁热网”建于 2006 年, 是一个致力于向广大网友推荐重庆本土美食的网站。在接收注册用户对各个餐馆进行评分、评价和添加新餐馆的同时, “趁热网”还能根据用户对餐馆的经营菜系和所属区域的偏好选择, 对用户进行各种餐馆的推荐。到 2011 年 4 月 2 日为止, 注册用户已达 18 665 名, 用户评分过的餐馆超过 11 960 个。本文实验中选取了 129 名注册用户对 46 个不同菜系(餐馆种类)的 127 个餐馆进行的 1 000 条评分记录作为实验的数据集。此外, 实验所采用数据集的稀疏等级<sup>[3]</sup> 为  $1 - 1000 / (129 \times 127) = 0.938 96$ 。整个实验的数据集被分成训练集和测试集, 训练集占整个数据集的 80%, 测试集为 20%。

### 3.2 评价标准

衡量预测精确度的准则主要是平均绝对偏差(mean absolute error, MAE)。MAE 是用来衡量一个预测分值与真实评分分值之间的平均绝对差异。本文采用 MAE 作为评价推荐系统质量的标准。MAE 越小, 推荐质量越高<sup>[5]</sup>。其表示为<sup>[3]</sup>

$$\text{MAE} = \frac{\sum_{i=1}^N |p_i - q_i|}{N} \quad (10)$$

其中: 预测的评分集合表示为  $\{p_1, p_2, \dots, p_N\}$ , 实际用户评分集合为  $\{q_1, q_2, \dots, q_N\}$ 。

### 3.3 实验结果

由于式(8) 中的  $w$  为设定的可调节的项目兴趣度相似性的权值, 所以  $w$  的取值影响着推荐的精度。通过 2.1 节描述可知,  $0 < w \leq 1$ 。所以, 在实验中取  $w$  从 0.1 ~ 0.9, 依次每次增加 0.1, 以观察其值的变化对推荐系统效率的影响。从图 1 可以看出, 当  $w$  取 0.9 时, MAE 值最小。

为了验证本文所提出方法的有效性, 实验将传统 item-based 协同过滤算法<sup>[3]</sup> 和本文提出的基于项目之间相似性的协同过滤算法使用相同的数据集进行比较。为了避免在计算项目之间相似性程度的过程中遇到分母为 0 的情况, 在相似性计算环节均采用绝对指数相似性计算方法来计算项目之间的相似程度。作为对照, 计算各自协同过滤推荐算法的 MAE 值, 邻居个数从 5 增加到 25, 间隔为 5。

如图 2 所示, 随着邻居数的变化, 本文提出的基于项目之间相似性的协同过滤算法具有较小的 MAE。由此可以推断, 本文提出算法能有效提高推荐系统的推荐质量, 从而为用户提供更加精确的个性化推荐。

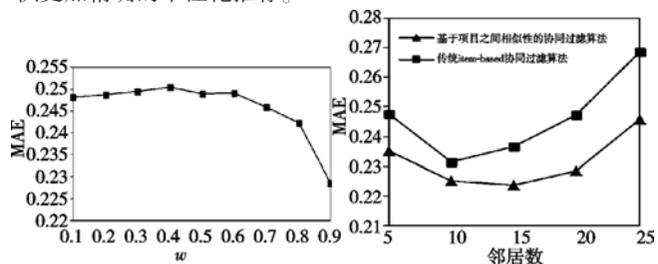


图1 权值  $w$  对 MAE 的影响

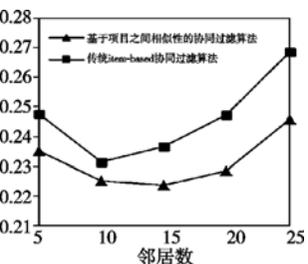


图2 本文算法与传统 item-based 算法推荐精度比较

射性气体浓度极低,对我国基本无影响。

当  $t = 120$  h、大气稳定度为 B、风速 5 m/s,风向为西偏南 20°时,放射性气体对我国东南沿海地区的影响如图 8 所示。

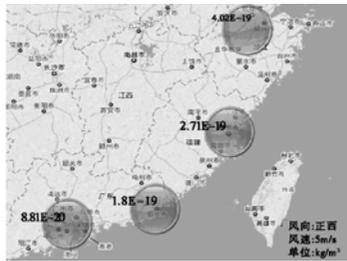


图8 福岛放射性气体 $t=120$ h对我国东南沿海地区影响

由图 8 可以看出,在该模拟条件下,日本福岛核泄漏 120 h 后,对我国东南沿海地区基本无影响。

#### 4 结束语

本文在基本 Gaussian 模型的基础上,采用了动态变化的泄漏强度,通过夹角系数的方式将风速和放射性气体自身的扩散速度合成,得到对气体扩散起到关键作用的合成扩散速度,进而更好地模拟了放射性气体扩散过程中在不同时段、不同区域、不同气流状态下的气体浓度。在气体扩散的过程中,自然界会对气体进行一定量的吸收。因此,在同一地区气体的浓度会随时间的增长而降低,并不是一成不变的。鉴于此,本文引入了放射性气体浓度衰减系数,从而能够更好地模拟实际情况中气体扩散过程的浓度变化。最后通过实验模拟,验证该模型能够帮助相关部门有效预估各种条件下的放射性气体污染范围,进而及时作出科学合理的应急决策。由于本文的预估模型是在一定假设条件的基础上构建的,并且只考虑了放射源强度由小到大的过程,并未考虑放射源强度从最大点降到 0 时的气体扩散过程的浓度变化。因此,可在该预估模型的基础上进一步引入实际影响因素,并采取一定方法来模拟放射性气体泄漏强度降低的过程,将是笔者对该模型今后努力的方向。

#### 参考文献:

[1] 杨湘山,吕焱,李冰,等.新形势下的核安全与辐射安全对策[J].

中国安全科学学报,2005,15(7):48-51.

[2] 魏东,董法军,董希琳,等.核事故中放射性核素扩散浓度的理论预测[J].中国安全科学学报,2006,16(3):107-113.  
 [3] 何宁.有毒气体扩散模型在事故救援中的应用[J].自然灾害学报,2009,18(5):197-200.  
 [4] HE Ning, WU Zong-zhi, ZHENG Wei. Simulation of an improved Gaussian model for hazardous gas diffusion [J]. Journal of Basic Science and Engineering,2010,18(4):571-580.  
 [5] 全纪龙,王浩,雷阳,等.有毒气体扩散预测法在大气环境应急监测布点中的应用[J].环境工程,2010,28(2):102-105.  
 [6] TANAKA T, AZUMA T, EVANS J A, et al. Experimental study on hydrogen explosions in a full-scale hydrogen filling station model[J]. International Journal of Hydrogen Energy,2007,32(13):2162-2170.  
 [7] ROBINS A. Wind tunnel dispersion modeling some recent and not so recent achievements[J]. Journal of Wind Engineering and Industrial Aerodynamics,2003,91(12):1777-1790.  
 [8] ROSEN M A, AO Yong-an. Feasibility of using exergy to assess air pollution levels around a smokestack [J]. Journal of Shenyang Architectural and Civil Engineering Institute,2005,21(3):73-76.  
 [9] DANDRIEUX A, DIMBOUR J P, DUSSERM G. Are diffusion models suitable for simulating small gaseous chlorine releases [J]. Journal of Loss Prevention in the Process Industries,2006,19(6):683-689.  
 [10] ROSEN M A, AO Yong-an. Using exergy to assess air pollution levels from a smokestack? Part 1: methodology [J]. International Journal of Exergy,2008,5(4):375-387.  
 [11] ESSA K S M, ELOTAIFY M S. Mathematical model for hermitized atmospheric dispersion in low winds with eddy diffusivities as linear functions of down wind distance [J]. Meteorology and Atmospheric Physics,2007,96(3-4):265-275.  
 [12] IIZUKA S, KONDO H. Large-eddy simulations of turbulent flow over complex terrain using modified static eddy viscosity models [J]. Atmospheric Environment,2006,40(5):925-935.

(上接第 118 页)

#### 4 结束语

本文首先分析了 item-based 协同过滤中传统相似性计算方法在评分数据稀疏的情况下计算项目之间相似性时存在的问题。针对上述问题,提出了一种基于项目之间相似性的协同过滤推荐算法。该算法通过结合用户对项目的评分和项目之间的兴趣度来进行项目之间相似性的计算,克服了传统相似性计算方法只依据用户评分的不足。实验结果表明,该算法弥补了传统相似性计算方法存在的弊端,有效地提高了推荐系统的推荐质量。

#### 参考文献:

[1] MICHEAL K, DAMIANOS G, ARISTIDES M. A mobile tourism recommender system [C]//Proc of the 15th IEEE Symposium on Computers and Communications. 2010: 840-845.  
 [2] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems;a survey of the state-of-the-art and possible extensions [J]. IEEE Trans on Knowledge and Data Engineering,

2005,17(6):734-749.

[3] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms [C]//Proc of the 10th International Conference on World Wide Web. New York: ACM Press, 2001:285-295.  
 [4] 李雪,左万利,赫枫龄,等.传统 item-based 协同过滤推荐算法改进[J].计算机研究发展,2009,46(z2):394-399.  
 [5] 邓爱林,朱扬勇,施伯乐.基于项目评分预测的协同过滤算法[J].软件学报,2003,14(9):1621-1628.  
 [6] HOROZOV T, NARASIMHAN N, VASUDEVAN V. Using location for personalized POI recommendations in mobile environments [C]//Proc of International Symposium on Application and the Internet. Washington DC: IEEE Computer Society, 2006:124-129.  
 [7] GONG Song-jie. Employing user attribute and item attribute to enhance the collaborative filtering recommendation [J]. Journal of Software,2009,4(8):883-889.  
 [8] 李春,朱珍民,高小芳,等.基于邻居决策的协同过滤推荐算法[J].计算机工程,2010,36(13):34-39.  
 [9] 谢季坚,刘承平.模糊数学方法及其应用[M].2版.武汉:华中科技大学出版社,2000:81-85.