

# 移动环境下情境导向的推荐系统<sup>①</sup>

李旭, 李长云, 张清清, 胡淑新, 周玲芳

(湖南工业大学 计算机与通信学院, 株洲 412007)

**摘要:** 本文提出了一种在移动情境下基于协同过滤算法的推荐模型. 针对现有算法与情境之间融合程度不足, 在传统协同过滤算法基础上设计了情境导向的协同过滤推荐系统模型框架. 采用 MAE 和 F1 评价指标对本文所提出的模型与情境过滤以及普通情境建模的方法之间的推荐质量进行评估, 实验表明本文提出的方案能在推荐质量上拥有更好的表现.

**关键词:** 移动情境; 情境相似度; 协同过滤算法; 推荐质量评估

## Context-Oriented Recommender System in Mobile Circumstances

LI Xu, LI Chang-Yun, ZHANG Qing-Qing, HU Shu-Xin, ZHOU Ling-Fang

(College of Computer and Communication, Hunan University of Technology, Zhuzhou 412007, China)

**Abstract:** This paper puts forward a mobile recommender system model based on collaborative filtering algorithm under the mobile circumstances. We aim at the insufficient degree in integration of the existing algorithm and the instant context, and design a circumstance oriented collaborative filtering recommendation system framework model based on the traditional collaborative filtering algorithm. We use MAE and F1 evaluation index to evaluate the commendation quality between our proposed model, pre-filtering model and the general context modeling. Through a series of experiments, it shows that the proposed scheme has the better performance in recommendation quality.

**Key words:** mobile context; context similarity; collaborative filtering algorithm; the evaluation of recommendation

在互联网技术不断发展的背景下, 网络中各类资源总量正在迅速地增长, “信息过载”等问题已经开始制约用户对信息资源的利用. 而信息推荐技术正是解决这类问题的有效方法, 通过用户兴趣爱好以及个人特点对信息资源进行整理分类, 进而向用户推荐符合其兴趣爱好或需要的信息<sup>[1]</sup>. 目前推荐系统在电子商务、互联网新闻、信息检索等领域均取得进展<sup>[2]</sup>. 此外, 随着移动互联网的迅速发展, 移动推荐系统(mobile recommender system)<sup>[3]</sup>逐渐得到越来越多的关注, 它利用移动互联网的资源信息为移动用户提供满足其需求的结果. 其中用户所处的情境信息逐渐成为影响用户选择与决策中的重要因素, 因此推荐技术需要生成符合用户当前情境的个性化信息推荐.

本文将移动用户的情境因素融合在传统推荐系统

中, 使用最广泛的协同过滤算法构建出移动环境下情境导向的推荐模型: 在引入情境的过程中, 如何有效解决数据稀疏性问题; 按照协同过滤算法的特点, 将情境信息与协同过滤算法结合.

## 1 概述

推荐系统作为缓解“信息过载”的一种有效手段, 是通过建立用户与项目之间的二元关系, 根据用户的兴趣特点及爱好, 利用相似性关系来推测每个用户潜在感兴趣的对象, 从而实现信息资源的个性化推荐. Adomavicius 等人<sup>[4]</sup>给推荐系统提出通用的形式化定义: 假设  $C$  表示全部用户集合,  $S$  表示需要推荐给用户的项目集合(如新闻、书籍、电影等), 同时使用效用函数  $u$ (效用函数用来计算推荐项目  $s$  对于用户  $c$  的推

<sup>①</sup> 基金项目: 2013 年度科技部科技支撑计划课题(2013BAJ10B14-5)

收稿时间: 2015-10-27; 收到修改稿时间: 2015-12-12 [doi: 10.15888/j.cnki.csa.005205]

荐度). 因此使用效用函数计算项目  $s$  对用户  $c$  的相关度, 如  $u: C \times S \rightarrow R$ . 其中  $R$  为一个有序集合(如一定范围的正实数). 那么推荐系统就是要将项目集合中推荐度  $R$  最大的项目推荐给用户, 即找到使效用函数  $u$  最大的那些项目. 那么, 推荐可以表达为:  $\forall c \in C, s'_c = \arg \max_{s \in S} u(c, s)$ , 即为用户推荐效用最大的项目. 推荐系统中的最主要部分就是推荐算法的选择以及使用, 一般来说可以将推荐算法分为协同过滤算法(Collaborative Filtering Recommendation)、基于内容的推荐(Content-based Recommendation)、基于知识的推荐(Knowledge-based Recommendation)以及组合推荐(Hybrid Recommendation)<sup>[5]</sup>.

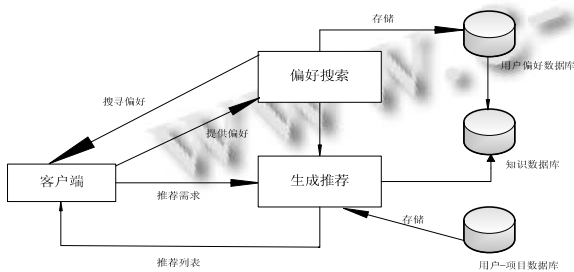


图 1 推荐系统基本框架图

随着移动互联网的应用领域不断扩展, 移动互联网逐渐成为发展的热点. 对于推荐系统来说, 移动推荐系统需要克服移动环境所带来的困难: 移动设备的多样性、用户地理位置的不确定性、网络环境及周围环境的影响.

根据移动情境以及移动推荐系统的结合方式, 可将其分为移动情境过滤<sup>[6]</sup>和移动情境建模<sup>[7]</sup>两个部分. 首先, 情境过滤是指使用情境对待推荐的项目集合进行过滤, 将与目标用户当前情境不相关的项目过滤之后, 对过滤后的项目集使用传统的推荐技术(如协同过滤推荐、基于内容的推荐等)完成推荐. 另外, 移动情境建模是指通过设计推荐算法将目标用户当前的情境信息融入到推荐过程当中, 从而产生更加符合用户情境及用户喜好的推荐结果. 融入移动用户情境因素的推荐系统通过实时获取用户移动端设备提供的情境信息, 如时间、气象、温度、网络环境、地理位置信息等, 从而更加准确掌握用户当前综合情境信息. 与传统的“用户-项目”二维推荐系统相比, 融入移动用户情境因素的推荐系统可以根据用户当前情境信息为用户

提供更加准确、更加符合用户需求的推荐.

如基于位置的移动推荐系统, 利用不同地理位置情境更及时准确的获得用户偏好并提供对应的推荐. Yang 等人<sup>[8]</sup>通过分析移动用户访问过的网上商家来获得移动用户的偏好特征向量, 利用余弦相似度来衡量用户描述文件与商家网页的相似度来衡量用户对商家的偏好. 基于时间的移动推荐系统通常是鉴于用户在不同时间点的偏好不同这一特点, 利用时间情境来获取用户偏好. Lee<sup>[9]</sup>等人考虑移动用户最近的购买记录, 对于较早时间的购买记录, 在用户偏好挖掘时所占的权重会逐渐降低, 使用余弦相似度计算用户间的相似度, 利用协同过滤算法进行推荐.

如果推荐系统从多维情境(日期、时间点、天气情况、同行人员等因素)出发, 考虑这些情境综合对用户选择物品时的影响, 则会形成“情境-项目-评分”的多维矩阵. Gallego 等人<sup>[10]</sup>提出情境移动推荐模型, 先考虑到用户当前的移动状态以及地理位置、随行朋友等情境的综合权重, 若权重达到设定的阈值, 利用协同过滤方法进行推荐.

## 2 情境导向的推荐模型

### 2.1 整体处理方案

本文研究主要涉及推荐系统通用模型的推荐模块部分, 推荐模块包括对用户喜好的识别以及使用相应的推荐算法而生成推荐结果. 在移动情境导向的推荐模型中, 主要问题就是解决如何将移动情境融入到推荐系统之中并且能够降低项目-评分-情境的复杂度. 这里所提出的推荐模型分别解决以下问题:

首先是情境过滤处理, 即针对情境数据引入后的数据稀疏性问题. 在本文基础上, 可采用情境过滤处理方法, 通过设定的情境相似度阈值, 将与当前情境相似度较低的数据集中的评分项目过滤. 同时可以将某些特定的情境因素作为主要参考, 如时间、地点、气候等属性, 对评分项目数据集进行过滤筛选从而降低数据稀疏性. 其次是推荐模块, 在融入情境信息之后与传统协同过滤算法相结合来完成推荐, 包括有: 用户情境相似度计算、用户间相似度计算、邻居用户集的确定和预测用户对项目的评分等.

### 2.2 情境导向的协同过滤推荐

本文将情境过滤处理与协同过滤算法相结合, 构建出情境导向的协同过滤推荐算法. 通常在推荐算法

中,邻居用户与目标用户之间相似度,即用户的相似度经常使用 Pearson 相关系数计算:

$$Sim_{m,u} = \frac{\sum_{i \in I_m \cap I_u} (r_{m,i} - \bar{r}_m)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I_m \cap I_u} (r_{m,i} - \bar{r}_m)^2 \sum_{i \in I_m \cap I_u} (r_{u,i} - \bar{r}_u)^2}}$$

$Sim_{m,u}$ 表示目标用户  $m$  与其他用户  $u$  之间的相似度;  $i$  表示目标用户与其他用户  $u$  共同参与过评分的项目.  $r_{m,i}$  则是目标用户  $m$  对项目  $i$  的评分:

$$P_{m,i} = r_m + \frac{\sum_{u \in N} (r_{u,i} - \bar{r}_u) \times Sim(m,u)}{\sum_{u \in N} Sim(m,u)}$$

其中  $N$  是目标用户的邻居用户集合, 而  $Sim(m,u)$  则表示目标用户与邻居用户之间的相似度.

然而, 上述所提到的用户相似度的计算并没有考虑情境因素的影响, 考虑到不同情境条件下用户评价标准不同, 推荐时若能获取与目标用户当前情境  $C$  相似度较高的情境条件下该用户对评价过的项目的平均分  $\bar{r}_{m,c}$ , 以及与目标用户当前情境  $C$  相似度较高的情境条件下其他用户  $u$  对评价过的项目的平均分  $\bar{r}_{u,c}$ , 可以解决不同情境条件下用户评价标准不一致的问题:

$$Sim_{m,u,c} = \frac{\sum_{i \in I_m \cap I_u} (r_{m,i} - \bar{r}_{m,c})(r_{u,i} - \bar{r}_{u,c})}{\sqrt{\sum_{i \in I_m \cap I_u} (r_{m,i} - \bar{r}_{m,c})^2 \sum_{i \in I_m \cap I_u} (r_{u,i} - \bar{r}_{u,c})^2}}$$

上述表达式的重点在于如何判断情境是否具有高的相似度, 本文定义情境相似度来解决这一问题. 首先通过计算用户之间相似度, 将用户相似度最高的  $N$  个用户将选作邻居用户. 再考虑到邻居用户中的平均评分  $\bar{r}_u$  与目标用户的平均评分  $\bar{r}_m$  均是随情境变化而发生改变的, 因此将评分项目限制在那些情境与当前情境  $C$  相匹配的评分中, 而并非是所有项目的评分. 在进行情境相似度计算后, 进而选取  $M$  个具有较高相似度情境下的项目评分平均值作为用户  $u$  在当前情境  $C$  下的平均评分.

### 2.3 情境相似度

获得情境相似度的目的是确定哪些评分项目的情境与目标用户当前情境具有相似性. 例如, 十一长假很多用户选择浏览与旅游相关的资讯, 那么在关于旅游地点的推荐中, “十一长假”这个情境条件下对旅游地点的评分就比其他情境条件下的评分更能反应用户偏好.

我们将用户所处的情境定义成一组由  $k$  种不同情境信息表示的情境变量, 即将 Context 表示为:  $C=(C_1, C_2, \dots, C_k)$ . 其中,  $C_t$  ( $t \in 1 \dots k$ ) 表示其中一种情境

类型(比如季节、地理位置、天气等),  $C_t$  则表示情境类型  $C_t$  的属性值. 若  $C_t$  为天气情境类型, 那么  $C_t$  属性值可以为晴天、多云、阴天、雨天等. 对于每种单维度情境类型  $t$ , 使用情境相似度计算函数  $sim_t(x,y)$  计算出在该情境类型下的情境值  $x$  与  $y$  之间相似度的值 ( $x,y \in C$ ).

我们假设用户对项目的喜好随着情景变化而出现较大改变的情况并不明显<sup>[1]</sup>, 那么在某个情境下用户对项目的评分也可用到其他情境条件. 我们采用 Pearson 相关系数来衡量情境变量之间的相似度. 这里  $r_{u,i,\alpha}$  表示用户  $u$  在  $\alpha$  的情境条件下对项目  $i$  的评分; 同样,  $r_{u,i,\beta}$  则是表示用户  $u$  在  $\beta$  的情境条件下对项目  $i$  的评分,  $t$  代表情境在  $C_t$  维度下. 那么, 在项目  $i$  的背景下, 两个不同情境变量之间的相似度权重可以表示为:

$$sim_t(\alpha, \beta, i) = \frac{\sum_{u=1}^n (r_{u,i,\alpha} - \bar{r}_i) \cdot (r_{u,i,\beta} - \bar{r}_i)}{\sigma_{\alpha} \cdot \sigma_{\beta}}$$

还可以根据  $sim_t(\alpha, \beta, i)$  的计算来定义项目  $i$  在情境  $C$  下的全局情境相似度. 即:

$$S_{C,\alpha,i} = \frac{\sum_k^{t=1} sim_t(c, \alpha, i)}{k}$$

$sim_t(c, \alpha, i)$  表示在单维度情境  $t$  下, 当前情境  $C$  与对项目  $i$  评分时的情境之间的情境相似度. 全局情境相似度可选择与当前情境相似度较高的情境条件下的项目评分作为预测目标用户评分的基础.

在情境化的协同过滤推荐中, 情境之间的相关程度会影响在该情境下用户对项目评分之间的相关程度. 使用给  $r_{a,i}$  赋权重的方式来对该用户评分进行调整, 首先定义  $R$  为在情境条件  $C$  下, 根据情境相关程度调整后用户  $a$  对项目  $i$  的评分, 公式如下:

$$R_{a,i,C} = k \sum_{x \in C} \sum_{t=1}^k r_{a,i,C} \cdot sim_t(c, x, i)$$

这里定义  $k$  为一个权重系数,  $sim_t(c, x, i)$  表示在  $t$  维度情境条件下, 基于项目  $i$  的情境  $x$  与当前情境  $c$  之间的相似度. 引入基于情境的用户相似度计算, 将考虑情境因素的用户相似度  $sim_{m,u,C}$  替代  $sim(m,u)$ , 那么基于情境的目标用户对于项目的预测评分如下:

$$R = \bar{r}_{m,C} + \frac{\sum_{u \in N_C} (R_{u,i,C} - \bar{r}_{m,C}) \times sim_{a,u,C}}{\sum_{u \in N} sim_{a,u,C}}$$

### 2.4 情境导向的协同过滤算法

本文中情境相似度的计算是根据用户-项目-情境的三维向量空间, 即  $\{User, Item, Context\}$  模型为基础, 我们将用户、项目以及情境和用户喜好作为记录进行

存储形成数据集。其中，我们使用向量模型表示所涉及到的情境数据，即  $C=(C_1,C_2,\dots,C_k)$ 表示由  $k$  个情境类型所组成的情境信息。

(1) 情境相似度的算法如下，

输入：{User, Item, Context} 模型，目标用户  $a$  的当前情境  $C$ ；

输出：在项目  $I$  背景下，当前情境  $C$  与其他情境  $\alpha$  的综合相似度  $S_{C,\alpha,i}$  以及两个不同情境属性值的相似度；

① 得到用户对项目  $I$  的评分记录，并计算均值  $\bar{r}_I$ ；

② 获取用户  $u$  对项目  $I$  评分时的情境  $A$ ，其各情境属性值为  $A=(a_1,a_2,\dots,a_k)$ ，另外取得目标用户当前情境  $C$  以及该情境属性值  $C=(c_1,c_2,\dots,c_k)$ ；

③ 计算  $k$  维度下不同情境属性值的相似度：

$$\text{sim}_k(a,c,I) = \frac{\sum_{u=1}^n (r_{u,I,\alpha_k} - \bar{r}_I) \cdot (r_{u,I,c_k} - \bar{r}_I)}{\sigma_{\alpha_k} \cdot \sigma_{c_k}}$$

④ 计算目标用户当前情境  $C$  与对项目  $I$  评价过的用户  $u$  的全局情境相似度  $S_{C,A,I}$ ：

$$S_{C,A,I} = \frac{\sum_{i=1}^k \text{sim}_k(c,\alpha,i)}{k}$$

(2)在得到情境相似度之后，进行邻居用户的确定从而形成目标用户的最近邻：

输入：全局情境相似度  $S_{C,A,I}$ ，当前情境  $C$ ；

输出：用户  $u$  与目标用户  $m$  在当前情境下的相似度，目标用户的最近邻居用户集  $N$ ；

① 根据情境相似度算法，得到目标用户当前情境  $C$  与用户  $u$  评分项目集合中每个情境  $A$  的全局情境相似度  $S_{C,A,I}$ ，并通过设定阈值  $L$  来确定情境相似度较高的项目评分集合，得到项目平均分  $\bar{r}_{u,c}$ ；

② 同样计算得到目标用户评分中，与当前情境相似度较高的项目评分集合，并计算  $\bar{r}_{m,c}$ ；

③ 获取目标用户  $m$  所评价过的项目  $i$ ，得到目标用户评分  $r_{m,i}$ ，同时获取对项目  $i$  有评价的用户  $u$ ，得到用户  $u$  对项目评分  $r_{u,i}$ ；

④ 重复步骤 2 与 3 对目标用户与用户  $u$  都评论过的项目  $i$  分别获取  $r_{m,i}$  与  $r_{u,i}$ ；

⑤ 计算用户  $u$  与目标用户  $m$  在当前情境  $C$  下相似度：

$$\text{Sim}_{m,u,C} = \frac{\sum_{i \in I_m \cap I_u} (r_{m,i} - \bar{r}_{m,C})(r_{u,i} - \bar{r}_{u,C})}{\sqrt{\sum_{i \in I_m \cap I_u} (r_{m,i} - \bar{r}_{m,C})^2 \sum_{i \in I_m \cap I_u} (r_{u,i} - \bar{r}_{u,C})^2}}$$

⑥ 获取情境相似度相对较大的用户评分作为邻

居用户集  $N$ ；

(3) 在确定邻居用户之后，通过邻居用户对待推荐项目进行评分预测，计算出目标用户在情境  $C$  下对项目预测评分：

输入：邻居用户集合  $N$ ； $k$  维度情境类型下，项目  $I$  的不同情境属性  $a, c$  之间的相似度  $\text{sim}_k(a,c,I)$ ，用户  $u$  与目标用户  $m$  在当前情境下的相似度  $\text{Sim}_{m,u,C}$ ；

输出：待推荐项目集的预测评分集合；

① 得到邻居用户  $u$  对一个待推荐项目  $I$ ，在情境条件  $X$  下的评分  $r_{u,I,X}$ ；

② 计算当前情境  $C$  下用户  $u$  对项目  $I$  偏好  $R_{u,I,C}$ ：

$$R_{u,I,C} = k \sum_{x \in C} \sum_{i=1}^k r_{u,I,x} \cdot \text{sim}_k(a,c,I)$$

③ 计算所有邻居用户对当前情境  $C$  下，对项目  $I$  的评分；

④ 计算情境  $C$  下目标用户对项目  $I$  的预测评分：

$$R = \bar{r}_{m,C} + \frac{\sum_{u \in N} (R_{u,I,C} - \bar{r}_{u,C}) \times \text{Sim}_{m,u,C}}{\sum_{u \in N} \text{Sim}_{m,u,C}}$$

为目标用户生成在当前情境下的项目推荐列表。

### 3 实验

本文的实验是在配置为 Intel 酷睿双核 2.1GHz, 4G 内存, 320G 内存的 PC 上进行,操作系统版本是 Windows7, 集成开发平台为 Myeclipse8.6, JDK 版本为 1.6. 以下所有实验都重复三次, 取平均结果。

#### 3.1 目标用户邻居数量 $n$

本实验采用邻居数量范围为 6 到 18, 设定  $H$  为 0.2,  $k=0.5$ , 对传统的协同过滤算法、Pre-CF、以及本文所提出的情境导向的协同过滤算法进行比较, 实验结果如图 2 所示。

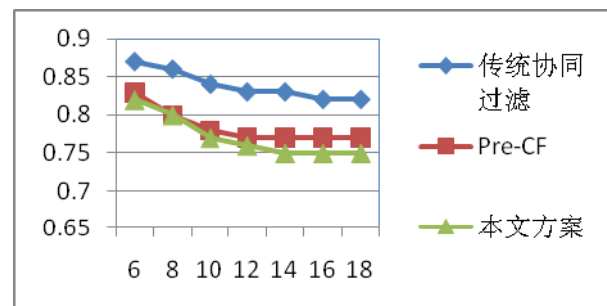


图 2 邻居用户数对算法的 MAE 值的影响对比

根据实验结果可以得到，随着邻居用户的增长，初始阶段较为利于 MAE 值的降低，在  $n>12$  之后，不

同方案对邻居用户的生长而产生的变化逐渐降低。

### 3.2 权重系数 $K$

权重系数  $k$  是在计算预测目标用户对项目评分时, 对邻居用户的评分参考程度的平衡细数.  $K$  值大小影响 MAE 值的大小, 我们用  $k$  对本文所提出的算法的 MAE 值影响进行评估:

从实验结果看来, 在  $k=0.6$  时, 本文提出的算法的 MAE 值最小.

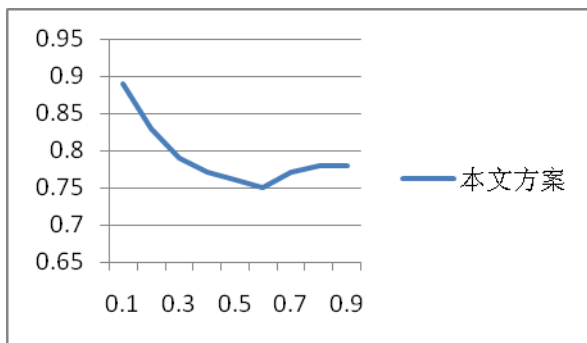


图3 权重系数  $k$  对本文方案的 MAE 值影响

### 3.3 算法的 MAE 值与 F1 值

对传统的协同过滤算法、Pre-CF、以及本文所提出的情境导向的协同过滤算法的 MAE 值与 F1 值进行比较, 将邻居用户值  $n=12$ , 权重系数  $k=0.6$  用到本实验中. 结果如图 4 所示.

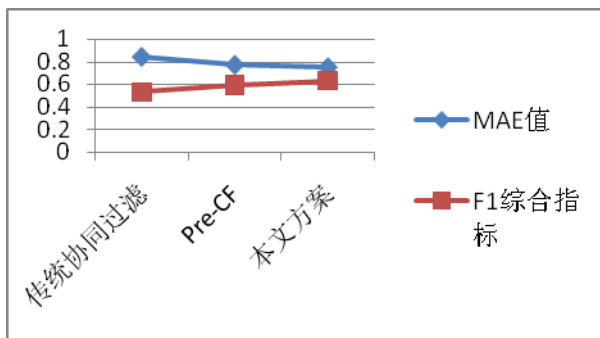


图4 文件读取时间对比

从 MAE 值来看, 本文所提出的情境导向的协同过滤算法在 MAE 值上有一定提升. 从 F1 指标来说, 与 MAE 值的情况类似, 本文提出的算法与其他两种算法相比提高了推荐质量.

## 4 结束语

在移动环境下的推荐系统中, 情境因素的影响仍然是需要解决的主要问题. 本文所提出情境相似度计

算以及用户相似度计算, 能够对情境导向的推荐系统的推荐质量进行提升. 然而, 还是需要更好地解决情境条件下协同过滤推荐系统的稀疏性问题, 今后研究可以根据移动用户环境特点, 通过挖掘更多用户日常活动及心理等方面提高推荐系统质量.

### 参考文献

- 1 许海玲, 吴潇, 李晓东, 阎保平. 互联网推荐系统比较研究. 软件学报, 2009, 20(2): 350-362.
- 2 王立才, 孟祥武, 张玉洁. 上下文感知推荐系统. 软件学报, 2012, 23(1): 1-20.
- 3 Ricci F. Mobile recommender systems. Information Technology & Tourism, 2010, 12(3): 205-231.
- 4 Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowledge and Data Engineering, 2005, 17(6): 734-749.
- 5 王国霞, 刘贺平. 个性化推荐系统综述. 计算机工程与应用, 2012, 7: 66-76.
- 6 Girardello A, Michahelles F. AppAware: Which mobile applications are hot? Proc. of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services. ACM. 2010. 431-434.
- 7 Yu Z, Zhou X, Zhang D, et al. Supporting context-aware media recommendations for smart phones. Pervasive Computing, IEEE, 2006, 5(3): 68-75.
- 8 Yang WS, Cheng HC. A location-aware recommender system for mobile shopping environments. Expert Systems with Applications, 2008, 34(1): 437-445.
- 9 Lee TQ, Park Y, Park YT. A time-based approach to effective recommender systems using implicit feedback. Expert Systems with Applications, 2008, 34(4): 3055-3062.
- 10 Vico DG, Woerndl W, Bader R. A study on proactive delivery of restaurant recommendations for android smartphones. Proc. of the RecSys2011 Workshop on Personalization in Mobile Applications. Chicago. 2011.
- 11 Chen A. Context-aware collaborative filtering system: Predicting the user's preference in the ubiquitous computing environment. Location and Context Awareness. Springer Berlin, 2005: 244-253.