DOI:10.13245/j.hust.2013.s2.013

一种基于扩展 FP-TREE 的服务推荐方法

莫 同 褚伟杰 李伟平 吴中海

(北京大学软件与微电子学院, 北京 100871)

摘要 针对协作过滤推荐的矩阵稀疏性与扩展膨胀问题,提出一种基于扩展 FP-TREE 的改进方法.将用户的情境取值抽象为情境空间状态,通过挖掘情境状态与服务的关联进行服务推荐.引入倒排索引扩展 FP-TREE 频繁项头表,建立状态-状态与服务-服务关联;通过索引树表示状态-服务关联,给出扩展 FP-TREE 与协作过滤矩阵的映射机制,在继承协作过滤的基础上极大地压缩了过滤矩阵.仿真实验表明:与传统的协作过滤推荐算法相比,该推荐方法具有更高的效率.

关键词 服务推荐; 情境感知; 关联挖掘; 协作过滤; FP-TREE

中图分类号 TP393 **文献标志码** A **文章编号** 1671-4512(2013)S2-0081-07

A service recommended method based on extended FP-TREE

Mo Tong Chu Weijie Li Weiping Wu Zhonghai

(School of Software and Microelectronics, Peking University, Beijing 100871, China)

Abstract In order to solve the sparsely and expansion problems of collaborative filtering, an improved method based on the extension of FP-TREE was proposed. States in context space was used to present context values, and service was recommended by mining association between states and service. Inverted index was involved into frequent item list to present states-states and service-service relations, and states-service was shown in index tree. Mapping mechanism from collaborative filtering matrix to extended FP-TREE was also discussed. The method could inherit compression filter and compress the matrix extremely. Simulation result shows that, compared with the traditional collaborative filtering method, the method has a higher efficiency.

Key words service recommendation; context aware; association mining; collaborative filtering; FP-TREE

随着物联网与情境感知技术的发展,现实世界逐渐变得可感知和互联互通[1].信息采集、传递和感知处理为主动智能服务提供了强有力的技术支撑.智能服务已被应用于交通、水处理、供电等多个领域,正在逐步改善人们的生活[2].

智能服务能够根据用户的环境变化,实时分析用户服务需求,并及时主动地提供合适的服务[3].基于情境的服务推荐是智能的重要体现.服务的长尾效应十分显著,随着"长尾"中服务数量的快速增长,如何帮助用户及时准确找到所需的服务成为制约服务水平提升和消费需求扩大的瓶颈.因此,推荐技术在电子商务、社交网络、信息检索等领域受到广泛的重视.

推荐技术主要可以分为基于规则的推荐、基于内容的推荐和协作过滤推荐三类.

基于规则的推荐:推荐规则其实质是一个相关属性与推荐项目之间的 if-then 语句.规则通过预定义定制或基于关联规则挖掘技术发现[4].基

收稿日期 2013-07-25.

作者简介 莫同(1981-),男,副教授,E-mail: motong@ss.pku.edu.cn.

基金项目 国家自然科学基金资助项目(61033005);国家科技支撑计划资助项目(2012BAH06B01);高等学校博士学科点专项科研基金资助项目(20120001120119, 20120001110086);深圳市科技研发资助项目(CXY201107010258A).

于规则的推荐效率和准确性依赖于规则的质量与数量.

基于内容的推荐:通过抽取特征表述被推荐的项目,根据项目特征与用户兴趣特征的相似程度进行过滤[5].特征选取是基于内容推荐的核心,选取的特征集合既要尽可能小,又要最大化地保证区分度.特征的抽取方法主要有TF-IDF[6]、信息熵[7]和潜语义分析[8]等.考虑用户兴趣特征可能随时间发生变化,有学者通过机器学习的方法对用户兴趣特征进行自适应更新,以提高推荐算法的精度[9].基于内容推荐的系统效率较高、简单易实现.但受内容过滤条件限制,仅适用于过滤具有标准描述格式的对象.依赖大量的历史数据,特征提取能力有限且过分细化,容易导致"过拟合"现象.同时,基于内容推荐适用于发现已有兴趣相似的项目,无法产生新的兴趣项目.

协作过滤推荐:协作过滤通过挖掘用户-项目关联矩阵中的对象间相似性,找到与目标对象相似的其他对象进行推荐[9].根据对象的不同,协作过滤主要分为基于用户的推荐[10]和基于项目的推荐[11].基于用户的推荐通过找到与目标用户偏好相似的其他用户,将他们感兴趣的内容推荐给目标用户.当用户量变化较大而推荐项目相对稳定时,可以通过挖掘项目间相似性和关联关系,进行基于项目的推荐.与传统的文本过滤相比,协作过滤能够发现对象间未被描述的关联,可以对难以描述的对象(如服务)进行过滤,且能够产生新的兴趣项目.协作过滤是目前使用较为广泛的推荐方法.但是,随着矩阵维度和项目数量的增加,协作过滤关联矩阵的膨胀速度成几何数量增长,严重影响推荐算法的效率.为提高协作过滤的可扩展性,可以对用户或项目进行排序,采用Top-N方法对关联矩阵进行剪裁[12],也有学者采用分类[13]、线性回归[14]等机器学习方法对矩阵进行压缩,以提高算法效率.

三类推荐方法均存在难以弥补的问题,根据应用领域特点,对协作过滤进行扩展,引入其他方法混合使用以取长补短是常用的方式[15].情境感知环境下的服务推荐一方面由于服务缺乏标准化的描述,难以抽取统一普适的特征进行过滤查找,协作过滤通过关联性进行推荐较基于内容的方式有更好的适应性;另一方面由于情境环境描述变量较多,而推荐需要能够根据用户的情境变化做出及时响应,保障及时性,协作过滤矩阵过于膨胀,效率不如基于规则或内容的推荐.根据上述分析,须要综合三类推荐方法的优势,建立一种既不依赖标准化描述特征,又不会对情境环境变量数量敏感的推荐方法.

用户的服务需求与情境环境变量的特定取值组合相关,该组合标志了用户在情境空间中的特定状态,可以通过挖掘状态-服务的关联进行服务推荐.状态的转换由情境变量取值变化触发,根据用户行为习惯,状态间存在某种偏序关系,并根据状态-服务关联使得服务之间也存在偏序关系.根据偏序关系可以进行关联状态或服务的发现.

综上所述,本文在协作过滤方法基础上引入关联规则挖掘与倒排索引,提出一种基于扩展FP-TREE的情境感知环境下服务推荐方法,将用户的情境取值抽象为情境空间状态,根据状态的跃迁对情境空间进行约减;通过FP-TREE挖掘情境空间状态与服务之间的关联关系;通过倒排索引树压缩表示状态-状态之间以及服务-服务之间的偏序关联关系.

1 协作过滤推荐

协作过滤推荐是目前应用较为广泛的一种推荐方法,其核心思想是基于历史记录建立N维($N-1$个相关对象+推荐项目)协作过滤矩阵,矩阵的每个元素表示$N-1$个相关对象与推荐项目的关系,针对某一维对象的目标实例值(或若干维对象的实例值集合)通过计算协作过滤矩阵发现与目标实例值(或实例值集合)相似的邻居实例值(或实例值集合),根据目标实例值(或实例值集合)可能对邻居实例值(或实例值集合)的推荐项目也感兴趣的假设进行推荐[10].

以常见的用户-项目评价协作过滤为例,协作过滤矩阵如表1所示.

表1 用户-项目评价协作过滤矩阵

	$\boldsymbol{i}_1$	$\boldsymbol{i}_2$	…	$\boldsymbol{i}_n$
$\boldsymbol{u}_1$	r_{11}	r_{12}	…	r_{1n}
$\boldsymbol{u}_2$	r_{21}	r_{22}	…	r_{2n}
…	…			
$\boldsymbol{u}_m$	r_{m1}	r_{m2}	…	r_{mn}

在表1中,协作过滤矩阵有两个维度,横向是推荐项目,如商品、音乐、电影、文件等;纵向是用户.协作过滤矩阵中的元素r_{ij}表示用户$\boldsymbol{u}_i$对项目$\boldsymbol{i}_j$的评价值.协作过滤矩阵可以看作是由m个在n维项目空间的用户评价向量组成,反之亦然.通过计算向量相似度进行基于用户的协作过滤或基于项目的协作过滤.

以基于用户的协作过滤为例,对用户$\boldsymbol{u}_i$,计算

其他用户评价向量与评价向量 v_{ui} 在 n 维项目空间的相似度，找到邻居用户集合 N_{u_i}，根据 N_{u_i} 的评价向量集合，计算各个项目对 u_i 的推荐度，根据推荐度进行项目推荐.

1.1　用户相似度度量

常用的用户向量相似度计算方法包括余弦相似度度量、Spear 相似度度量和泊松相似度度量.

a. 余弦相似度度量

用户 u_i 与用户 u_j 的相似度通过协作过滤矩阵的用户评价向量 v_{ui} 和 v_{uj} 的余弦夹角表示，余弦值越大表示两个用户的相似程度越高. 余弦相似度度量为

$$\mathrm{sim}(\boldsymbol{u}_i,\boldsymbol{u}_j)=\cos(\boldsymbol{u}_i,\boldsymbol{u}_j)=\frac{\boldsymbol{u}_i\cdot\boldsymbol{u}_j}{\|\boldsymbol{u}_i\|*\|\boldsymbol{u}_j\|},$$

式中：$u_i\cdot u_j$ 为两个用户评价向量的内积；$\|u_i\|*\|u_j\|$ 为两个用户评价向量模的乘积.

b. Spear 相似度度量

对单个用户而言，其评价过的项目数通常远少于项目总数，导致协作过滤矩阵非常稀疏，使得余弦相似度主要由用户是否有共同评价项决定，而忽略了用户对项目评价的差异. 此外，相同评价体系下，用户的评分尺度差异也会对相似性度量的客观性造成较为严重的影响. Spear 相似度度量综合上述因素，计算式为

$$\mathrm{sim}(u_i,u_j)=\sum_{k\in I_{ij}}(r_{ik}-\bar{r_i})(r_{jk}-\bar{r_j})\Big/\sqrt{\sum_{p\in I_i}(r_{ip}-\bar{r_i})^2}\sqrt{\sum_{q\in I_j}(r_{jq}-\bar{r_j})^2},$$

式中：I_i，I_j 为用户 u_i 和用户 u_j 评价过的项目集合；I_{ij} 为用户 u_i 和用户 u_j 共同评价过的项目集合；$\bar{r_i}$，$\bar{r_j}$ 为用户 u_i 和用户 u_j 的平均评价结果.

c. 泊松相似度度量

基于用户的协作过滤本质是找到对推荐项目有相近兴趣的邻居用户. 兴趣的近似程度主要与用户对共同评价项目的评价一致性有关，而与用户间的共同评价项目多少关联较弱. 基于该假设，泊松相似度度量的计算式为

$$\mathrm{sim}(u_i,u_j)=\sum_{k\in I_{ij}}(r_{ik}-\bar{r_i})(r_{jk}-\bar{r_j})\Big/\sqrt{\sum_{p\in I_{ij}}(r_{ip}-\bar{r_i})^2}\sqrt{\sum_{q\in I_{ij}}(r_{jq}-\bar{r_j})^2}.$$

泊松相似度度量与 Spear 相似度度量的主要区别在于泊松相似度度量基于用户间的共同评价项，用户平均评价结果和评分尺度度量均以用户间的共同评价项为基础.

1.2　邻居用户选取

根据用户间相似度度量结果确定邻居用户的方法主要有阈值法或 TOP-N 法. 其中阈值法是根据预先设定的邻居用户相似度阈值，选择相似度大于阈值的用户成为邻居用户；TOP-N 法是根据预先确定的邻居数 N，对用户相似度进行排序，取相似度最大的 N 个用户成为邻居用户.

项目推荐度计算如下：根据邻居用户计算项目推荐度的方法主要有简单平均法、相似加权平均法和关系加权平均法等.

a. 简单平均法. 用户 u_i 的邻居用户集合 N_{u_i}，项目 i_j 的推荐度 $r(i_j)$ 是所有邻居用户对 i_j 的评价的算数平均值，

$$r(i_j)=\sum_{k\in N_{u_i}}r_{kj}\Big/n(N_{u_i}),$$

式中 $n(N_{u_i})$ 为用户 u_i 的邻居用户数.

b. 相似加权平均法. 由于邻居用户的相似度不同，相似度越高的邻居用户的评价对推荐度的影响越大，根据该假设，基于用户相似度对评价进行加权平均，

$$r(i_j)=\sum_{k\in N_{u_i}}\omega_{sk}r_{kj}\Big/n(N_{u_i}),$$

式中 ω_{sk} 为相似度权重，常见的权重包括直接使用相似度或使用相似度与平均相似度的比值等.

c. 关系加权平均法. 现实世界中，关系紧密的人较无关的人对需求产生有更为重要的影响，根据该假设，通过计算用户之间的关系远近，如基于社交网络，作为推荐度计算的权重对评价进行加权平均，

$$r(i_j)=\sum_{k\in N_{u_i}}\omega_{rk}r_{kj}\Big/n(N_{u_i}),$$

式中 ω_{rk} 为关系权重.

项目推荐方法为：根据项目推荐度计算结果，使用阈值法或 TOP-N 法进行项目推荐.

2　情境感知环境下的服务推荐

情境感知服务是指通过传感器采集/感知被服务对象的情境信息，根据情境信息分析判断被服务对象当前的状况，然后选择并提供适当服务的一种新的服务模式. 情境感知服务的核心是建立情境到服务的映射，进而根据采集的情境进行服务推荐. 该映射可以通过协作过滤矩阵方式建立，以用户、服务和各个相关情境为矩阵维度，表示用户在不同情境取值下需求的服务. 情境感知服务通常需要多个情境共同刻画业务环境，导致协作过滤矩阵过于膨胀，且存在较为严重的稀疏性问题，须要从情境过滤和推荐方法两方面入手

提高计算效率.

2.1 基本概念

为描述服务推荐的应用环境,须要建立并定义服务推荐的情境概念空间. 如图 1 所示.

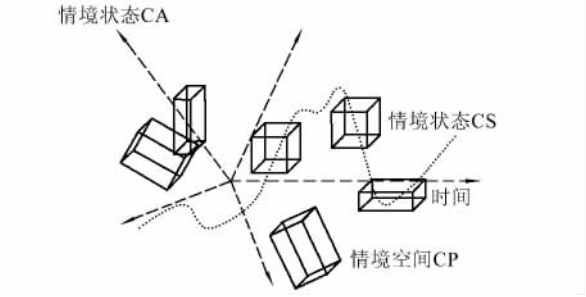

图 1 情境空间示意图

定义 1 情境属性(context attribute). C_A 是对服务推荐应用环境中的实体的某类相关特征信息的表征,比如温度、位置或者年级、学分等.

定义 2 情境空间(context space). $C_P=\langle C_{A1}, C_{A2}, \cdots, C_{An}\rangle$,情境空间定义和情境属性选取与服务领域密切相关.

定义 3 情境状态(context state). $C_S=\langle C_{A1.v}, C_{A2.v}, \cdots, C_{An.v}\rangle$($C_{Ai}\in C_P$),$C_{Ai.v}$ 表示情境属性 C_{Ai} 的取值,情境空间中若干情境属性的取值集合定义了该对象在情境空间中的状态.

定义 4 状态跃迁. $\tau=C_{Si}\rightarrow C_{Sj}$,状态跃迁表示由于一个或若干情境属性的取值发生变化,导致情境空间状态发生变化. 状态跃迁的业务含义为用户的应用环境发生改变.

2.2 情境空间约减

情境空间中情境属性的增长会导致空间膨胀,增加计算复杂度,影响计算效率. 状态跃迁时,仅须考虑取值发生改变的情境. 可以根据跃迁发生的频率和情境在跃迁中的作用对情境进行评级,进而对情境空间进行约减,保留导致核心跃迁的关键情境.

定义 5 情境状态评级. R_{C_S} 衡量情境空间中情境状态的重要程度,$R_{C_S}=\omega_{C_S}f_{C_S}$,其中:$f_{C_S}$ 为情境状态 C_S 在情境空间中出现的频率;ω_{C_S} 为情境状态 C_S 的业务权重. 为简化计算,可设所有情境状态的业务权重均为 1,此时 $R_{C_S}=f_{C_S}$.

定义 6 情境评级 R_{C_A} 衡量情境空间中情境属性的重要程度,

$$R_{C_A}=\sum_{C_S}\Big[R_{C_S}\sum_{\tau_{C_S}}f_{\tau_{C_S}^{C_A}}\Big/\Big(\sum_{\tau_{C_S}}f_{\tau_{C_S}^{C_A}}n_{\tau_{C_S}}\Big)\Big]\quad(C_A=C_S)$$

式中:$f_{\tau_{C_S}}$ 为以情境状态 C_S 为初始顶点的跃迁中跃迁 τ 发生的频率;$n_{\tau_{C_S}}$ 为跃迁 τ 中发生变化的情境属性数;$f_{\tau_{C_S}^{C_A}}n_{\tau_{C_S}}$ 为以情境状态 C_S 为初始顶点的跃迁中,情境属性 C_A 发生变化的跃迁 τ 发生的频率.

根据情境评级可以对情境空间进行约减,情境空间约减算法如下.

算法 1 情境空间约减算法

输入 情境空间,情境属性评级

输出 约减后的情境空间

步骤 1 约减所有评级为 0 的情境.

步骤 2 对情境进行评级排序,按照评级从低到高的次序进行约减.

步骤 3 对每个待约减的情境属性,查找是否有情境状态仅包含该属性,判断删除该情境状态是否会对业务逻辑产生影响:若有影响,则保留该情境属性;若没有影响,则删除该情境状态,并删除所有与该状态相关的跃迁.

步骤 5 在所有情境状态中删除该情境属性.

步骤 6 遍历所有情境属性,直至所有情境属性都无法被约减.

3 扩展 FP-TREE 推荐方法

3.1 扩展 FP-TREE

传统协作过滤方法通过协作过滤矩阵表示特征-项目的关联,协作过滤矩阵具有直观、简单、易于建立和挖掘关联等优点,但存在稀疏性问题. 为此,可以借鉴关联规则挖掘的 FP-TREE 方法和搜索中倒排索引方法,构建协作过滤索引树,将协作过滤矩阵改进为索引树结构,提高推荐规则抽取效率.

协作过滤索引树(CFI). CFI 树是满足下列条件的一个树结构:它由一个根节点、特征属性值前缀项(树干)、推荐项叶子节点、若干特征索引表和一个推荐项索引表组成,其中:根节点为 null;特征属性值前缀项包括特征属性值 $C_{Ai.vj}$、节点子路径数 n 和相同特征属性值的链接指针;推荐项叶子节点包括推荐项 s、节点子路径数 n 和相同推荐项链接指针;特征索引表是特征属性的倒排索引;推荐项索引表是推荐项的倒排索引.

CFI 树可以根据服务选用记录,通过综合 FP-TREE 建立算法与倒排索引建立算法建立.

3.2 基于扩展 FP-TREE 的情境感知服务

CFI 树表示推荐项目及多个相关对象(特征)之间的关联关系,其中特征索引表表示各推荐特

征与推荐项目的相关性. 从根节点到推荐项叶子节点的特征路径表示推荐项的推荐特征，当某些特征满足时推荐项的推荐度即是叶子节点路径数与该路径上被满足的最末端特征节点路径数的比值. CFI 树可用来表示情境感知服务，情境空间的各个情境作为特征，服务作为推荐项，CFI 树的一条路径表示一个情境到服务的映射.

通过 CFI 树实现情境感知服务的过程如下：a. 根据情境感知服务历史记录建立情境感知服务 CFI 树；b. 通过传感器对情境特征索引表中的情境进行物联网环境数据采集，得到情境属性取值；c. 根据情境特征索引表的情境排序依次判断情境特征取值，根据取值在 CFI 树中选择相应路径前进；d. 根据路径前进结果进行服务推荐.

3.3　协作过滤矩阵与扩展 FP-TREE 映射

协作过滤矩阵的 N 个维度可以映射为 CFI 树的 N 个倒排索引，N 维关联矩阵对应为 CFI 树的关联树. 本研究的重点是情境感知环境下的服务推荐，图 2 展示了一个二维协作过滤矩阵到 CFI 树的映射.

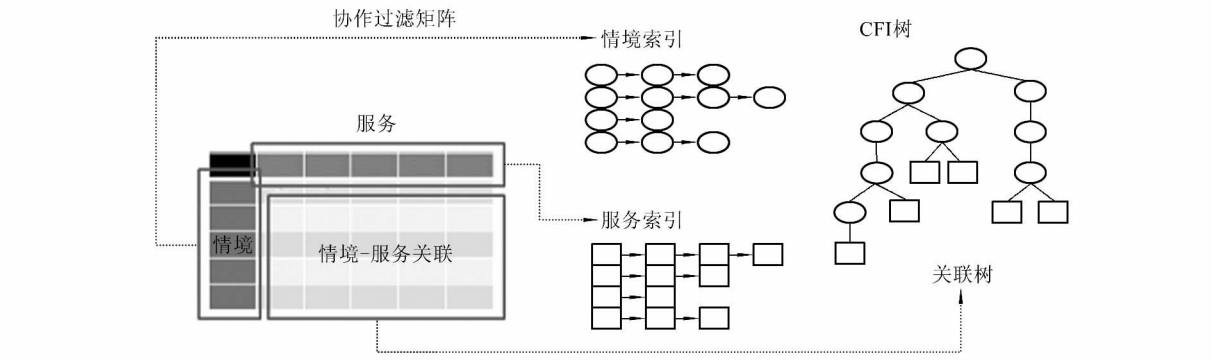

图 2　CFI 树与协作过滤矩阵映射关系

在图 2 中，协作过滤矩阵包含情境和服务两个维度. 情境状态之间可以通过情境属性取值建立情境倒排索引. 同理，服务之间根据服务特征建立服务索引. 情境属性取值与服务的关联矩阵可以压缩成关联树存储.

协作过滤矩阵的大小是各维度项数目的笛卡尔积，若情境属性取值总数目为 m，服务数目为 n，则情境与服务关联矩阵的大小为 $m\times n$. 在实际业务场景中，情境属性根据其取值不同导致 m 可能非常大，所须提供的服务数量 n 也可能较多，使得关联矩阵十分庞大，但关联矩阵中情境属性值与服务有关联的项极少，导致关联矩阵极为稀疏，使得存储和计算的效率较低. 而通过关联树形式表示则仅须要保存这些关联项，可有效地压缩关联矩阵.

3.4　扩展 FP-TREE 建立方法

CFI 树可以根据服务选用记录，即情境属性取值与服务的关联记录，通过综合 FP-TREE 建立算法与倒排索引建立算法建立. 具体算法如下.

算法 2　CFI 树建立算法

输入　服务历史记录集，频繁度阈值

输出　CFI 树

步骤 1　扫描历史记录集，得到属性值和服务的频繁集及频繁度，根据频繁度进行降序排序，过滤掉频繁度低于阈值的属性值和服务，得到频繁集.

步骤 2　统计各情境属性在频繁集中的频繁度并进行排序，每个情境属性对应的属性值按照频繁集中的频繁度进行排序，建立情境索引.

步骤 3　统计各服务特征在频繁集中的频繁度并进行排序，具有该特征的服务按照频繁集中的频繁度进行排序，建立服务索引.

步骤 4　创建关联树的根结点 root，以“null”标记.

步骤 5　再次扫描服务历史集，对每条记录的情境属性值按照频繁集进行排序和过滤，然后将服务放在序列末尾，设排序后的序列为[$p|P$]，其中 p 是第一项，P 是剩余项. 调用 insert_tree([$p|P$], root)，执行过程如下.

若 root 有子女 child 使 child. name = p. name，则 child. count 加 1；否则创建 root 的一个子女 child，child. name = p，child. count = 1，链接 child 到它的父结点 root；在频繁集中查找名称为 p 的频繁项，依次找到 p 指针的末尾，将其指向 child.

若 P 非空，递归地调用 insert_tree(P, child).

3.5　基于扩展 FP-TREE 的服务推荐方法

基于扩展 FP-TREE，根据用户的当前情境属性取值，查找用户可能需求的服务，并计算用户需求服务的置信度，通过置信度排序根据不同的推荐策略进行服务推荐. 具体算法如下.

算法3 服务推荐算法

输入 CFI树,用户情境属性值

输出 服务推荐集合,服务置信度

步骤1 直接关联服务查找.根据用户情境属性的当前取值,在索引树中寻找相应的分支,该分支下的叶子节点集合即为直接关联服务集.服务的置信度为 $\theta=\sum(n_l/n_t)$,其中:n_l 是叶子节点子路径数;n_t 是分支终端树干节点子路径树.

步骤2 间接关联服务查找.

a. 情境关联:根据用户当前情境属性取值,通过查找CFI树的情境状态索引,找出与用户当前情境状态相似的状态,并将该状态的对应服务作为间接关联服务进行推荐,核心要点包括如下几点.

相似情境状态查找:采用基于倒排索引的相似度比较算法计算情境状态的相似度 φ,通过设定相似度阈值,取高于阈值的状态.

情境状态对应服务查找:根据情境状态的情境属性和取值,采用直接关联服务查找方法,找出对应服务集并计算服务置信度 θ.

服务置信度计算:间接关联服务的置信度为 $\theta^t=\theta\varphi$.

b. 服务关联:根据已有的服务候选集,通过查找CFI树的服务索引,找出与之相似的服务进行推荐,核心要点包括如下几点.

相似服务查找:采用基于倒排索引的相似度比较算法计算服务的相似度 φ,通过设定相似度阈值,取高于阈值的服务.

服务置信度计算:间接关联服务的置信度同样为 $\theta^t=\theta\varphi$.

服务关联的迭代:可以通过设定迭代次数或置信度阈值作为迭代终止条件.

生成:服务推荐结果根据服务置信度排序生成.根据不同的应用场景,依照排序选择服务项进行推荐.

4 仿真实验与对比分析

4.1 实验条件

硬件:CPU 1.87 GHz,内存 2.00 GB,硬盘 300 GB.软件环境:操作系统 Windows 7,运行环境 Myeclipse 9.0 + Tomcat 7.0.编程语言:JAVA,基于 Apache Mahout 的 taste 系统实现.

实验数据采用的餐饮行业数据 Entree Chicago Recommendation Data Set,共有 $5.067\ 2\times10^6$ 条实例数据.数据来源:http://archive.ics.uci.edu/ml/datasets/Entree+Chicago+Recommendation+Data.实验通过分析用户情境对进行推荐.本文选用 90% 数据作为训练集,10% 数据作为测试集.

4.2 实验结果与对比分析

实验针对相同数据集分别采用CFI树与协作过滤算法(CF)进行推荐.两种算法的推荐时间对比如图3所示.由图3可知:由于协作过滤矩阵使用用户-项目评分矩阵取得用户对项目的偏好,通常系统中项目数量很多,而用户对项目的操作

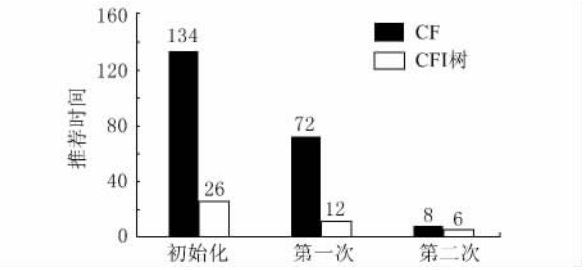

图3 推荐时间对比

很少,导致用户-项目评分矩阵具有很大的稀疏性,在这种数据量大而又稀疏的情况下计算最近邻用户的时间耗费很大.而CFI树通过挖掘最大频繁项集较好地解决了数据的稀疏性并减少了数据处理量,运算效率较协作过滤有很大提升.当推荐次数较多后,两种方法都能根据推荐结果对算法进行优化,耗费时间均能快速收敛.当重复推荐次数达到两次以上时,二者时间差距已经较小.

两种方法的推荐准确率与查全率对比如图4所示.由图4可知:由于CFI树通过设定支持度

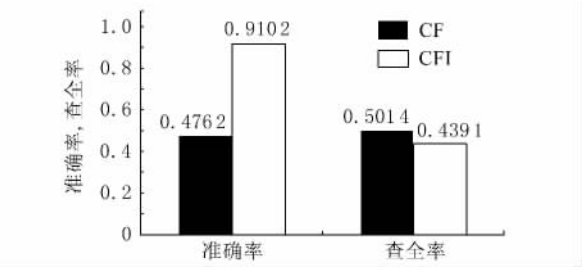

图4 准确率与查全率对比

阈值对树进行过滤剪枝,因而具有较高的推荐准确率,但是阈值设置需要与业务相关,不合适的阈值可能导致信息的丢失,可能将符合用户需求的项目舍弃,导致查全率较协作过滤稍差.

5 结论

提出了一种基于扩展FP-TREE的服务推荐方法,该方法通过服务历史记录建立情境状态与服务的关联FP-TREE,并倒排索引扩展FP-

TREE 频繁项头表，根据用户的当前情境值，通过直接关联和间接关联搜索用户可能需求的服务并计算置信度. 通过案例分析可见，该方法在不影响服务查准率和查全率的前提下能够有效地提高推荐效率.

接下来的工作包括：继续完善扩展 FP-TREE 的建立，使之能够根据新的服务记录进行更新；结合领域本体与用户特征本体，研究用户自身特征和用户间的间接关联对服务推荐的影响.

参　考　文　献

[1] Thompson C W. Smart devices and soft controllers [J]. Internet Computing, 2005, 9(1): 82-85.

[2] 莫同，李伟平，吴中海，等. 一种情境感知服务系统框架[J]. 计算机学报, 2010(11): 2084-2092.

[3] Kortuem G, Kawsar F, Fitton D, et al. Smart objects as building blocks for the Internet of things[J]. Internet Computing, 2010, 14(1): 44-51.

[4] 王玉祥，乔秀全，李晓峰，等. 上下文感知的移动社交网络服务选择机制研究[J]. 计算机学报, 2010, 33(11): 2126-2135.

[5] Balabanovic M, Shoham Y. Fab: content-based collaborative recommendation [J]. Communications of the ACM, 1997, 40(3): 66-72.

[6] Lee D L, Chuang H, Seamons K. Document ranking and the vector-space model [J]. IEEE Software, 1997, 14(2): 67-75.

[7] Yang Y, Pedersen J O. A comparative study on feature selection in text categorization[C]// Proceedings of the 14th International Conference of Machine Learning. San Francisco: Morgan Kaufmann Publishers, 1997: 412-420.

[8] Hofmann T. Latent semantic models for collaborative filtering[J]. ACM Transaction Information Systems, 2004B, 22(1): 89-115.

[9] Chang Yein, Shen Junhong, Chen T L. A data mining-based method for the incremental update for supporting personalized information filtering[J]. Journal of Information Science and Engineering, 2008, 24(1): 129-142.

[10] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]// Process of the 14th Conference on Uncertainty in Artificial Intelligence. San Francisco: Morgan Kaufmann Publisher, 1998: 43-52.

[11] Linder G, Smith B, York J. Amazon. com recommendations: item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1): 76-80.

[12] Deshpande M, Karypis G. Item-based top-N recommendation algorithms[J]. ACM Transaction on Information Systems, 2004, 22(1): 143-177.

[13] Chen Y H, George E I. A Bayesian model for collaborative filtering[C]// Process of the 7th International Workshop on Artificial Intelligence and Statistics. Fort Lauderdale: Morgan Kaufmann Publisher, 1999: 187-192.

[14] Sarwar B, Karypis G, Konstan J. Item-based collaborative filtering recommendation algorithms[C]// Process of the 10th International Conference on World Wide Web. New York: ACM, 2001: 285-295.

[15] 张少中，陈德人. 面向个性化推荐的两层混合图模型[J]. 软件学报, 2009, 20(S): 123-130.