

以用户为中心的移动信息服务门户系统^①

汪静甜¹, 白琳¹, 卢彬², 魏峻¹

¹(中国科学院 软件研究所, 北京 100190)

²(苏州飞讯动力信息科技有限公司, 苏州 215600)

摘要: 随着智能终端的深入普及, 用户使用移动终端的个性化需求日益增加, 在许多场景下用户需要综合大量信息去设置当前的行为目标然后再调用相关服务达成目标. 其中, 综合信息进行目标选择面临的问题是信息量很大而且信息背后隐含着关联关系, 这些因素使得用户做出合理的选择比较困难; 另一方面, 搜集调用服务遇到的问题往往是往往决策后的目标由单个服务无法实现, 需要调用并组合多个服务来达成目标. 上述两个问题影响用户需求的实现, 因此面向用户的智能化服务组合逐渐凸显其重要性和迫切性, 为了解决上述问题, 本文提出了以用户为中心的移动信息服务门户系统, 该系统围绕智能化的服务组合概念, 根据个性化信息和上下文数据帮助用户快速精准的定位目标, 并组合多个服务帮助用户达成目标.

关键词: 上下文驱动; 智能推荐; 目标分解; 服务组合; 用户行为探索

User-Centric Portal System for Mobile Information Services

WANG Jing-Tian¹, BAI Lin¹, LU Bin², WEI Jun¹

¹(Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

²(Suzhou Flycent Information Technology Limited, Suzhou 215600, China)

Abstract: Users' individual requirement has been increased with the popularity of intelligent terminal. In many scenes, users need to collect and consider much information when they make a choice, then achieve this goal by accessing web services. However, the problem for collecting information is that there are amounts of information, and the information have hidden relations, these factors make users hard to make decisions. In addition, when users search services they have to face the fact that single service always can't satisfy their needs, they need to compose many services to achieve their target. The above two questions had bad effect on users' demands satisfaction, therefore the user-centric intelligent service composition become more important. To solve this problem, we propose a user-centric portal system for mobile information services, it focus on intelligent service composition, and it can help users to find specific target and compose services to achieve the target by using individual information and context data.

Key words: context-driven; intelligent recommendation; target decomposition; service composition; user behavior exploration

根据《第 33 次中国互联网发展状况统计报告》^[1], 截至 2013 年 12 月, 手机网民规模达 5 亿, 年增长率为 19.1%, 手机继续保持第一大上网终端的地位. 手机网民规模的持续增长促进了移动终端各类应用的发展, 大量移动应用涌现市场. 虽然市场上移动应用大量涌现, 但目前大部分应用是提供围绕某一核心业务(如地

图、新闻等)的集成服务, 仅仅可以完成较单一的操作(如导航、阅读等), 用户若想完成某个复杂任务, 通常需要将多个应用或服务组合在一起. 以生活中一个常见的场景为例: 用户想合理安排候机时间; 为了达成这个目标, 用户到达机场后需要结合当前时间、航班时间决定是否立即兑换登机牌, 需要根据剩余时间以

^① 基金项目:国家自然科学基金(61173005);江苏省科技型技术创新基金(BC2013138)

收稿时间:2014-03-06;收到修改稿时间:2014-03-31

及航班是否有航餐决定是否用餐,再根据剩余时间、餐馆距离决定用餐的餐馆,用餐后还需要考虑时间、当前位置至登机口的距离决定后续的行为:是避免免税店还是在登机口附近浏览新闻.即使考虑到上述情况完成决策,但是用户仍会忽略很多信息,如选择不同餐饮类型用餐时间长短不同、航班因为天气原因延迟起飞时间、延迟起飞时间后应重新规划候机选项等等.信息量大而且信息之间隐含着关联影响,由用户收集并综合考虑信息做出合理的决策中间状态太多,另外即使做出目标决策后,单一的服务无法帮助用户完成目标,需要组合多个服务来满足用户目标,对服务的组合同样也需要综合考量上下文环境、服务信息、历史行为记录等信息.为了解决这一问题,我们提出了面向用户的智能化服务组合的概念,并围绕此概念设计实现了以用户为中心的移动信息服务门户系统.

1 相关工作

移动应用的炙热化趋势除了归功于移动终端的移动性之外,还应归功于其区别于传统终端的另一大特点:附带多种传感器;传感器和便携特征使移动终端可以获取用户的上下文环境信息(时间、位置、运动状态等),该信息使得移动应用可以提供更个性化的服务.在不同上下文环境中,用户需要完成的任务和实现的目标不同,即需要的服务组合不同,因此可以利用上下文信息来实现智能的服务组合.目前,个性化推荐技术在电子商务(如阿里巴巴、豆瓣网、当当网等)、信息检索(如 iGoogle、百度等)以及移动应用、互联网广告等等众多应用领域得以广泛运用,基于上下文感知的推荐系统在国外许多大学和研究机构已经展开了深入研究,国内也近两年也有相关工作^[2],这些研究工作强调如何更精确的找到更贴近用户喜好的某一类应用、某一属性的物品或某一类别的广告,重点在用户使用领域明确的前提下进行目标推荐.而我们关注的是利用上下文感知探索用户行为,使用推荐技术帮助用户尽快锁定目标,根据目标选择服务,并依据全局 QoS 最优的标准组合服务来达成用户的个性化目标.

针对上面提到的上下文智能推荐、服务组合,已有相关工作进行了研究:文献[2, 3]对移动推荐系统关键技术进行了详细的介绍和研究,根据移动上下文与

移动推荐系统的结合,引出了移动上下文过滤的概念,但未提出具体的推荐方法;针对面向 QoS 的 Web 服务组合,文献[4]介绍了一种解决服务聚合中服务动态选择 QoS 全局最优化问题的实现算法,文献[5]设计实现了一个中间件,其可以保证服务组合 QoS 优化的同时最大化满足用户事先对 QoS 参数设定的限制,然而上述两个工作未考虑移动上下文以及用户的个性化信息对服务选择的影响.

综上所述,本文提出了基于上下文智能化的服务组合概念,并围绕此概念设计实现了以用户为中心的移动信息服务门户系统.论文第三节首先介绍系统总体架构设计,然后详细介绍系统核心组件功能及协作关系,第四节介绍系统关键技术,第五节通过一个实际的场景对系统做出展示.

2 系统设计与实现

2.1 系统架构设计

该门户系统的功能是帮助用户明确目标并组合服务来满足这一目标.以用户为中心的移动信息门户系统总体架构由移动端应用层、服务端功能层、数据层、资源层四部分组成,如图 1 右侧所示.其中,移动端应用基于 Android 平台开发,负责与用户的交互以及与服务端的信息通信;服务端功能层部署在 Web 服务器上,负责系统核心业务的实现,如关联规则挖掘、用户目标分解、服务绑定、服务调用;数据层用于存储功能层用到的数据记录,使用关系数据库实现;系统最底层是服务资源层,负责提供统一的服务接口信息,此功能通过服务集成中间件实现.

2.2 系统实现

系统输入包括两部分,一是用户输入:a)用户模糊的目标或无目标输入,b)用户对系统提示选项的回复;二是系统搜集的信息:c)用户上下文数据,d)用户历史行为记录,e)按照 WSDL 或 WADL 标准描述的服务接口信息;系统核心业务功能包括移动端的上下文数据收集,信息交互,服务表现,服务端的智能推荐,目标分解,服务绑定,服务调用,服务集成中间件;系统输出是满足目标的具体服务展现.下面通过系统输入、核心功能组件、组件协作关系介绍系统实现.

2.2.1 系统输入

a)用户模糊目标或无目标输入: 两种类型分别适用于不同情景; 用户在有些情况下对当前要做的事有大致的规划, 即知道一个模糊的目标, 此时系统会帮

助用户进行目标分解最终明确目标; 另一种情况是用户对要做的事并没有大致的规划, 这时完全交由系统通过上下文数据、用户历史记录对目标进行推荐.

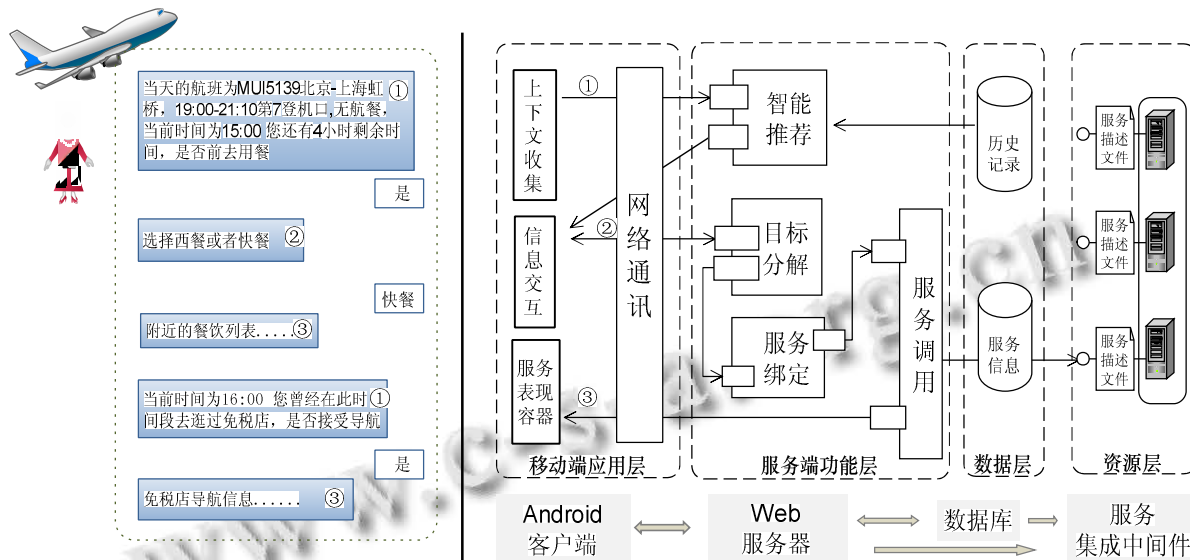


图 1 系统总体架构设计

b)用户对系统提示选项的回复: 系统在帮助用户进行目标分解时, 会给用户一些提示选项, 此时用户需要输入对选项的回复.

c)用户上下文数据: 因为系统的智能推荐、目标分解功能都需要用到上下文数据, 所以系统会在移动端自动收集用户的上下文信息, 如位置信息(经纬度)、时间.

d)用户历史行为记录: 该记录是系统实现智能化、满足用户个性化需求很关键的数据, 服务器端对每次用户选择的目标事件、上下文信息进行详细的记录并存放在关系数据库中, 其中记录的时间为用户到达该位置的起始时间; 对于位置信息的记录考虑到使用经纬度信息不利于划分地域范围, 所以我们使用百度地图提供的位置服务^[6]将经纬度信息转化为位置信息, 如经度 116.33681, 纬度 39.980186 对应的位置信息为知春路大运村; 事件序列的存储格式为[事件 1(子事件序列): 事件 2(子事件序列)]. 数据格式如表 1 所示.

e)符合标准的服务接口信息: 系统需要使用服务集成中间件将服务接口信息转化为格式一致的服务描述文件, 目前该中间件支持转化按照 WSDL 或 WADL 标准描述的服务接口信息.

表 1 历史记录数据表

ID	time	place	eventList
1	16:30	首都机场 T2 航站楼	用餐(快餐): 避免免税店(纪念品): 安检: 新闻阅读(娱乐: 经济)
2	16:00	首都机场 T2 航站楼	避免免税店(纪念品: 化妆品): 用餐 (中餐): 喝咖啡
3	15:40	首都机场 T2 航站楼	避免免税店(化妆品): 用餐(快餐): 喝咖啡
4	20:15	天通苑龙德 广场	看电影(团购): 用餐(自助餐): 打 车

2.2.2 核心功能组件

接下来对各业务功能组件做详细介绍.

上下文数据收集模块: 该模块负责收集用户的上下文环境数据.

信息交互模块: 用于接收用户文本信息的输入, 包括用户的目标或回复; 并显示目标分解的结果.

服务表现容器: 是服务集成中间件的一部分, 负责展示服务后获得的信息.

智能推荐模块: 该模块负责接收和解析客户端传来的上下文数据, 筛选历史记录, 并通过上下文敏感的关联规则挖掘算法(3.1 节做详细描述)找到用户下一步最可能的目标事件进行输出. 其中筛选历史记录将

系统收集的事件和子事件序列这一多维信息降为单维数据, 因为挖掘多维关联规则算法实现较复杂, 而且在历史记录不充足的情况下挖掘频繁项目集为空的概率很大, 所以我们将历史记录消减为单维数据, 筛选后的记录格式如 List1 所示:

List 1 规则挖掘模块筛选后的用户历史记录格式

格式: [时间 GMT+08:00 24 时]_[位置]_[事件 1: 事件 2]

示例:

16:30_首都机场 T2 航站楼_用餐: 避免税店: 安检: 新闻阅读

16:00_首都机场 T2 航站楼_避免税店: 用餐: 喝咖啡

15:40_首都机场 T2 航站楼_避免税店: 用餐: 喝咖啡

20:15_天通苑龙德广场_看电影: 用餐: 打车

目标分解模块: 该模块主要负责目标分解; 该模块的输入是规则挖掘模块输出的事件关键字或者由信息交互模块传来的用户输入的目标关键字以及系统历史记录, 构建个性化目标树并对目标进行分解(建树方法及分解方法在 3.2 节做详细描述), 并输出分解后的结果. 该模块在构建分解树之前需要格式化历史记录, 因为此模块关心的是用户行为记录中的多维事件信息, 即事件及子事件之间的关系, 所以我们需要对系统的历史记录进行格式化, 格式化后的记录如 List2 所示:

List 2 目标分解模块格式化后的用户历史记录格式

格式: [时间 GMT+08:00 24 时]_[位置]_[事件 1→子事件序列]

示例:

16:30_首都机场 T2 航站楼_用餐→快餐

16:30_首都机场 T2 航站楼_避免税店→纪念品

16:30_首都机场 T2 航站楼_安检

16:30_首都机场 T2 航站楼_新闻阅读→娱乐

16:30_首都机场 T2 航站楼_新闻阅读→经济

16:00_首都机场 T2 航站楼_避免税店→纪念品

16:00_首都机场 T2 航站楼_避免税店→化妆品

16:00_首都机场 T2 航站楼_用餐→中餐

16:00_首都机场 T2 航站楼_喝咖啡

20:15_天通苑龙德广场_看电影→团购

20:15_天通苑龙德广场_用餐→自助餐

20:15_天通苑龙德广场_打车

服务绑定模块: 该模块根据具体目标在服务集成中间件提供的服务信息列表中找到匹配的服务信息, 使用服务选择算法选择最佳服务并输出服务组合(3.3 节做详细描述).

服务调用模块: 该模块负责根据服务 ID 读取服务描述文件中的具体接口信息, 包括服务 URL、参数格式、数据输出格式等, 然后根据服务地址及附带参数访问服务, 获得服务数据, 解析后输出至服务表现容器进行展示.

服务集成中间件: 功能是将异构的服务接口信息转化成统一格式的服务描述文件, 同时提供服务表现容器. 在服务绑定之前首先面对的是服务存在异构性问题, 服务的异构性体现在 I 服务接口不同, II 数据格式不同; I 会导致无法以统一模板提取服务信息, II 影响系统对服务数据的解析进而影响服务在移动端的展示. 我们使用已有工作 SIMMA[7]服务集成中间件构建的可以适配异构服务的服务统一模型, 该模型将按照 WSDL、WADL 标准描述的服务接口信息使用服务集成描述语言 SIDL (Service Integration Description Language)描述成统一的模板, 然后我们可以对格式一致的服务描述文件进行处理. 下面通过新闻服务描述文件展示 SIDL 文件格式.

List 3 新闻服务 SIDL 文件

```
<service>
  <attributes>
    <serviceName> NewsConvergence </serviceName>
    <title>新闻服务</title>
    <tag>新闻浏览</tag>
    <params>
      <time>1</time>
      <cost>0</cost>
      <reliability>0.86</reliability>
      <duraition>10</duraition>
      <reputation>5.4</reputation>
    </params>
  </attributes>
  <binding id="getNewsListBinding" type="WADL">
    <serviceId>getNewsList</serviceId>
  </binding>
</service>
```

2.2.3 核心功能组件

图 1 结合实际航班助手的场景说明系统核心业务组件间的协作关系:

①上下文数据收集模块以固定频率收集用户当前上下文数据: 数据由网络通讯层传至服务器智能推荐模块, 此模块挖掘出用户下一步最可能的目标事件, 然后交由信息交互模块展示给用户。

②用户输入目标后, 信息交互模块将关键字传至服务器, 目标分解模块对目标进行分解然后将结果返回信息交互模块, 由用户进一步选择, 直到确定无子目标事件的具体目标。

③目标明确后, 目标分解模块将具体目标传至服务绑定模块, 该模块选择出最佳服务传给服务调用模块, 服务调用模块根据服务 ID、URL 访问服务得到服务信息, 最后交由服务表现容器展示。

3 系统关键技术

3.1 上下文驱动的目标推荐

目前个性化推荐技术在学术界和工业界已非常成熟^[8-11], 对于上下文感知的推荐概念国内也已有相关工作^[2,3], 然而对于基于上下文的推荐并未提出具体的实施方案, 因此在移动门户系统实现的过程中, 我们给出上下文驱动目标推荐的技术方案, 设计实现上下文信息敏感的关联规则挖掘算法。

上下文目标推荐功能的目的是在当前上下文环境下根据用户当前的行为对下一步行为进行探索。其中, 根据当前行为探索下一步行为的问题, 我们可以使用关联规则挖掘算法, 如 Apriori 算法、FP-tree 算法等, 此类算法支持布尔关联规则的挖掘, 规则如下所示, 通过“用餐”事件找到具有最强关联规则的“新闻浏览”事件:

$$event(\text{“用餐”}) \Rightarrow event(\text{“新闻浏览”})$$

但是根据此规则为用户推荐目标忽略了用户当前的上下文信息, 在不同的上下文环境中, 即使用户做同样的事, 下一步的行为也不一定相同。例如有的用户喜欢晚上去商业圈 A 聚餐后看浏览新闻, 而喜欢在中午去商业圈 B 聚餐后逛街。所以为了更贴近用户的需求, 需要可以融入上下文信息的关联规则:

$$event(\text{“用餐”}) \wedge time(\text{“15:00-17:00”}) \wedge place(\text{“机场”}) \\ \Rightarrow event(\text{“新闻浏览”})$$

上述规则在单维布尔规则挖掘时添加了两个属性约束: 时间 *time*、地理位置 *place*。然后我们将此属性加入关联规则挖掘算法, 实现上下文信息敏感的关联规则挖掘算法。

算法描述如下:

输入:

1. 单行为历史记录集合 *record*
2. 当前行为事件 *c_event*
3. 当前上下文数据: 时间 *curtime* + 位置 *curplace*
4. 规则约束:
 - a) $curtime - 1h < time < curtime + 1h$
 - b) $place = curplace$
 - c) $min_support = 2$

输出:

与当前行为事件同时发生的最频繁事件 *f_event*

方法:

- 1) 根据约束条件对用户行为历史记录 *record* 进行筛选;
- 2) 扫描筛选后的记录, 对事件按频数递减排序, 并删除频数小于最小支持度 *min_support* 的事件;
- 3) 对于每一条事件序列, 按照上一步中的顺序重新排序;
- 4) 使用步骤 3 的结果序列插入以 *null* 为根节点的树构建频繁树;
- 5) 建立项表, 每一项链接树中相同名称的节点, 链表的第一个元素就是项表元素;
- 6) 扫描项表的每一项 *i*, 然后在频繁树中找到所有包含 *i* 的路径;
- 7) 对于每一条路径上的节点出现的次数设置为 *i* 的次数;
- 8) 然后去掉每一条路径节点集合中的 *i*, 构建条件模式基;
- 9) 返回步骤 3 迭代执行, 直到树为 *null* 终止;
- 10) 记录频繁模式的集合, 然后扫描集合, 找到与 *c_event* 构成最强关联规则的实际 *f_event* 输出。

3.2 个性化目标分解

系统在根据上下文对用户进行目标推荐后, 还需要对目标进一步分解, 如上述场景中系统给出“用餐”这一候选项, 用户选择后需要再细化到中餐或西餐、堂食或外带等具体目标。然而, “用餐”这一目标对应的子目标有很多, 为了更快速更精确的满足用户需求, 需要对子目标集合进行筛选。

在缩减子目标集合之前, 首先要面对问题是实现目标分解的功能。目标分解基于目标树实现, 目标树的每个节点保存目标关键字。分解树的构建方法是根据用户的行为记录建立目标知识库, 然后使用在线词典数据库系统 WordNet^[12]判断词汇关系然后构建单词网络。WordNet 是一部在线词典数据库系统, WordNet 将单词组织为同义词集合, 每一个集合代表一个基本的词汇概念, 并在概念间建立了上下位关系、部分-整体关系等, 我们利用上述关系在目标关键字之间建立父子关系、兄弟关系构建目标树。目前已有相关工作进行中文 WordNet 的研究, 通过与英文 Wordnet 转换生成的方法并引入词汇消歧算法实现了中文 WordNet^[13]。同时也有相关的 WordNet 中文数据库; 在实现的过程中, 需要导入 WordNet 中文库, 并且还需要使用 JWI(Java Wordnet Interface)^[14]检索词汇库, JWI 是 MIT 面向 Wordnet 开发的 Java 库, 封装的 API 支持

对数据库词汇进行检索操作。

目标树构建完成后, 因为在不同上下文环境下用户对节点的选择概率不同, 需要利用上下文数据 *context_data* 及历史记录数据 *history_record* 为目标树节点添加权重生成个性化的目标分解树, 最后只选择权重较高的子目标为用户进行推荐. 将格式化的数据记录 *history_record* 记为 R_1 , 在 R_1 筛选其中与 *context_data* 匹配的记录集合 R_2 , 然后遍历 R_2 中的每条记录, 统计每个关键字的频次 n 以及事件计数总和 *counts*, 每个节点 N 的权重值为 $n / counts$. 如上所述, 计算节点权重 W 的函数表示为:

$$W(N) = f(freq(N), history_record, context_data)$$

个性化的目标分解树构建完毕, 然后使用深度优先搜索算法将目标定位到某一节点, 读取子节点信息并对子节点权重数值排序, 提取排名前两位的子节点信息推荐给用户, 该过程一直迭代进行直到定位到叶子节点(具体目标), 分解过程完毕.

3.3 服务绑定

在实际应用场景中, 系统需要为用户展示多个服务的信息; 例如在上述场景中用户选择中餐, 可能会再选择新闻浏览, 因此整个流程需要调用餐饮信息服务和新闻信息服务, 但是满足单个功能需求的服务不止一个, 有很多服务满足相同的功能但是具有不同的 QoS 参数, 如何给每个事件绑定合适的服务从而实现最佳的服务组合呢? 本小节主要解决这一问题.

为了方便对服务信息进行操作, 首先我们对服务集成中间件提供的服务 SIDL 文件解析, 得到如下格式的服务信息.

表 2 服务信息格式

服务信息格式				
ID	name	Tag	SIDL_URI	params

获取统一的服务信息后, 我们再利用服务组合算法选择服务. 目前基于 QoS 的 Web 服务组合算法主要是采用启发式算法设计的选择算法, 如遗传算法, 粒子群优化算法、蚁群优化算法及一些融合算法等^[15]. 并且服务动态选择全局最优问题可以转化成带有约束条件的多目标服务组合优化问题, 因此我们选择使用支持多目标智能优化的遗传算法解决该问题^[4].

在门户系统的实现过程中, 我们关注服务的四个 QoS 参数: 信誉 *Rep*, 可靠性 *Rea*, 执行时间 *T*, 执行费用 *C*; 首先需要使用约束条件对服务信息集合进行筛

选, 约束条件是信誉和可靠性大于一定的阈值 ($Rep(p) \geq Rep_0, Rea(p) \geq Rea_0$), 另外为了缩减解集合, 我们对服务的执行费用和执行时间也做了相应的限制: ($T \geq 0s \& T < 60s$) & ($C \geq 0Y \& C < 100Y$); 根据上述约束条件对服务集合进行筛选后, 然后提取执行时间 *T* 和执行费用 *C* 两个参数作为“数据对”(T,C)组成变量集合, S_m, S_n 分别代表不同类别的服务集合:

$$S_m = \{(T_{m1}, C_{m1}), (T_{m2}, C_{m2}), \dots, (T_{mi}, C_{mi})\}$$

$$S_n = \{(T_{n1}, C_{n1}), (T_{n2}, C_{n2}), \dots, (T_{nj}, C_{nj})\}$$

我们希望最后选择的服务执行时间和执行费用最低, 同时为了后续利用目标函数倒数计算适应度, 我们将目标函数指定为 $MinF(P) = (T(P), C(P)) = S_{mi} * S_{nj} + 1$. 我们使用整数定长编码方式对个体进行编码, 另外因为算法中需要计算基因的适应度, 并且筛选适应度大的基因进行遗传, 所以我们使用目标函数的倒数度量适应度的大小. 同时交叉运算采用单点交叉、群体随机配对、随机设置交叉点的方法, 变异运算则采用基本位变异的方法实现. 算法输出适应度最高的个体, 即最佳服务组合. 算法实现如下所示:

算法 用于服务组合的算法实现

```

输入: 服务参数集合, 变量范围
输出: 适应度最高的服务组合
Genotype[] population;
Void initialize() { //初始化操作
    population.fitness = 0;
    newpopulation.fitness = 0;
}
Void caculateFitness() { //个体适应度计算
    population.rfitness = population.fitness/sum;
}
Void evaluate() { //个体评价
    population.fitness = 1/functionvalue;
}
Void crossover() { //交叉运算
    swap(population[odd].geloc, population[even].geloc);
}
Void mutate() { //变异运算
    population.geloc = randval(geloc.MIN, geloc.MAX);
}
initialize();
    
```

```
while(generation < MAXGENS){
    generation++;
    caculateFitness ();
    evaluate();
    crossover();
    mutate();
}
Genotype output = keep_the_best(); //输出适应度最大的个体
```

4 系统展示

下面以一个实际场景为例，展示系统的移动端帮助用户实现目标的过程。

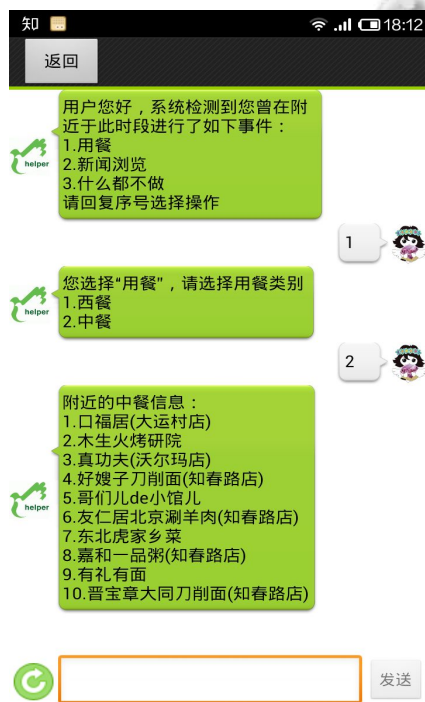


图 2 系统移动端展示一

系统的实现环境：移动端可支持的最低版本是 Android 2.3.3(API-10)；服务器端使用 Tomcat7.0, JDK1.7 开发实现；数据库使用 MySQL Server 5.6。如图 2 所示，用户打开系统后会收到目标推荐信息，用户回复“用餐”后，系统帮助用户细化餐饮类别；用户再选择“中餐”然后服务器调用餐饮服务返回附近的中餐餐馆信息。图 3 展示的是在相同上下文环境下，用户选择“新闻浏览”后的展示结果。

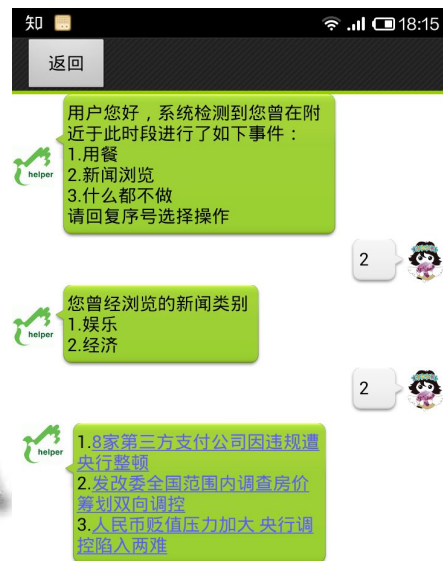


图 3 系统移动端展示二

5 系统展示

随着移动终端的普及和用户信息消费模式的变化，用户使用移动终端的个性化需求日益增加，为了满足用户的个性化需求，本文围绕智能化服务组合的概念提出了以用户为中心的移动信息服务门户系统，该系统可以帮助用户快速精准地定位目标，并组合服务帮助用户达成目标；在系统实现的过程中设计实现了上下文敏感的关联规则挖掘算法以及个性化的目标分解方法。文章展示了系统的设计以及核心组件功能和协作关系，并通过已实现的应用场景对系统做出展示。我们下一步的工作目标是：(1)开发实现更多的应用场景；(2)提高系统的性能。

参考文献

- 1 第 33 次中国互联网发展状况统计报告 http://www.cnnic.net.cn/hlwfzyj/hlwzxbg/hlwtjbg/201403/t20140305_46240.htm
- 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 孟祥武,胡勋.移动推荐系统及其应用.软件学报,2010,24(1): 91-108.
- 4 刘书雷,刘云翔,张帆,唐桂芬,景宁.一种服务聚合中 QoS 全局最优服务动态选择算法.软件学报,2007,18(3):646-656.
- 5 Zeng LZ, Benatallah B. QoS-aware middleware for web services composition. IEEE Trans. on Software Engineering,

- 2004, 30(5): 311–327.
- 6 百度地图位置服务 <http://developer.baidu.com/map/webservice-placeapi.htm>
- 7 邵小哲,黄涛.面向移动应用的服务集成中间件设计与实现[硕士学位论文].合肥:中国科学技术大学,2013.
- 8 陈敏,顾亦然.个性化推荐系统研究[硕士学位论文].南京:南京邮电大学,2012.
- 9 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.
- 10 刘建国,周涛,汪秉宏.个性化推荐系统研究进展.自然科学进展,2009,19(1):1–15.
- 11 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.
- 12 WordNet <http://wordnet.princeton.edu/>
- 13 张俐,李晶皎,胡明涵,姚天顺.中文 WordNet 的研究及实现, 2003,24(4):327–329.
- 14 JWI <http://projects.csail.mit.edu/jwi/>
- 15 李金忠,夏洁武,唐卫东,曾劲涛,王翔,吴兰英.基于 QoS 的 Web 服务选择算法综述.计算机应用研究,2010,27(10): 3622–3627.
- 16 李虎林,姚青.提取个性化目标的需求分析方法.计算机工程与设计,2009,30(9):2182–2186.