

用户日常频繁行为模式挖掘*

史殿习,李寒,杨若松,莫晓赟,魏菁
(国防科技大学计算机学院,湖南长沙410073)

摘要:针对如何在智能手机上高效准确地进行用户日常频繁行为模式挖掘问题展开研究。提出一个基于智能手机的用户日常频繁行为模式挖掘框架;为了减少用来挖掘的上下文篮子的数量、提高挖掘效率,提出一个动态的滑动窗口算法,进而提出一个将上下文出现的频率和持续时间有机地结合起来的加权模式挖掘算法;在此基础上,基于为期6周21个用户的上下文数据,对所提出的挖掘框架和算法进行实验验证,结果表明,所提出的框架和频繁模式挖掘算法可以高效地在资源有限的智能手机上运行,而且能够挖掘出反映用户生活方式的日常频繁行为模式;从两个纬度对用户日常频繁行为模式进行可视化,以可视化方式对用户在不同地方和不同时段的行为模式进行展现,从而方便用户随时了解其日常行为模式。

关键词:移动数据挖掘;移动感知;行为模式

中图分类号:TP391 **文献标志码:**A **文章编号:**1001-2486(2017)01-074-07

Mining user frequent behavior patterns in daily life

SHI Dianxi, LI Han, YANG Ruosong, MO Xiaoyun, WEI Jing
(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: Research focused on how to mine frequent daily behavior patterns of users on smartphones feasibly and efficiently was started. Firstly, a frequent behavior patterns mining framework based on smartphones was proposed. Secondly, a dynamic sliding window algorithm DSW (dynamic slide window) to decrease the context baskets in quantity and improve mining efficiency was proposed. Furthermore, a frequent patterns mining algorithm WePM (weighted pattern mining) which takes both frequency and duration of context occurrence into consideration was developed. On the basis of the above preparation, the mining framework and algorithm were verified experimentally with the context data from 21 users over 6 weeks. Results indicate that the proposed framework and frequent patterns mining algorithm can feasibly and efficiently run on resources limited smartphones to mine daily behavior patterns, and then to reflect users' lifestyles. Finally, the patterns from two perspectives, namely behavior patterns in different locations and time periods are visualized, which benefits the users to realize their daily behavior patterns at any time.

Key words: mobile data mining; mobile sensing; behavior patterns

随着无线通信技术和传感器技术的快速发展,通过智能手机内嵌的GPS、加速计等传感器以及移动操作系统提供的虚拟传感器,智能手机可以收集大量的原始传感器数据,基于这些数据可以推测出与人们日常生活密切相关的上下文信息^[1],这些信息涵盖了用户日常生活的各个方面,如位置^[2-3]、活动^[4-6]以及健康状况^[7-8],等等。

利用智能手机收集的各种传感器数据可以进一步挖掘用户的行为模式,如移动模式^[9-10]和应用程序使用模式^[11-12]等。通过移动模式可以发现用户频繁出现的位置和轨迹、有意义的位置以及可以预测用户的下一个位置^[13]等。为了全面了解人们在日常生活中的行为模式,近年来,研究人员开始利用智能手机的纵向上下文数据(包括呼叫事件、短信记录、推测的位置、前台应用程序使用事件、简单的活动状态以及手机充电记录等)来进行用户行为模式的挖掘^[14-16]。通常来讲,使用上下文数据进行行为模式挖掘被认为是关联规则挖掘问题,主要有两种方法:一种是收集所有的上下文数据并在服务器上进行模式挖掘^[14-15];另一种是在智能手机上进行模式挖掘^[16]。后者是一个具有挑战性并且更具吸引力的研究方向,因为在智能手机上进行模式挖掘不仅可以让用户随时了解其行为模式,而且可以保护用户的隐私,因为上下文数据不需要在网络上进行传输,从而减少网络带宽的消耗,而且可以避免隐私泄露问题。因此,本文重点研究如何基于

* 收稿日期:2015-09-01
基金项目:国家自然科学基金资助项目(61202117,91118008)
作者简介:史殿习(1966—),男,山东龙口人,教授,博士,博士生导师,E-mail:dxshi@nudt.edu.cn

智能手机,即在智能手机上进行频繁行为模式挖掘问题。

目前,在智能手机上进行行为模式挖掘面临着如下两个方面的挑战:一是如何优化传统数据挖掘算法如 Apriori 算法^[17]等,使得其在手机上高效运行;二是如何挖掘更准确、更贴近用户的真实生活方式的行为模式。针对第一个问题,现有的研究通常采取的做法是预处理所有时段的上下文数据,从而生成上下文篮子,如 MobileMiner^[17]中所有上下文按时间被划分为篮子,篮子的数量对挖掘算法的执行时间有着显著的影响。然而,并不是所有的上下文数据都是必要的,因为事实上,当人们与智能手机交互时,总是做两件事情,即使用应用程序(电子邮件、新闻、照片等)和与其他人交流(打电话或发短信),因此在交互时段的上下文数据比在其他时段的数据更具价值,更能反映用户的日常行为模式。针对第二个问题,目前的研究主要是使用上下文的累计时间来衡量行为模式^[16],事实上,上下文共现频率也是一个与行为模式密切相关的重要因素。例如,如果一个用户每天早上在家里使用手机阅读新闻 10 min,一个星期后就会生成一个 70 min 的数据记录:{早上,在家中,阅读新闻}。如果另一个用户偶尔在一周的某个早晨阅读了 70 min 的新闻,也会生成同样的数据记录。显然前者比后者更可能是一个模式,因为它出现的频率更高。

1 相关工作

数据挖掘旨在从一个大的数据集中发现有用的知识,关联规则和频繁项集挖掘是数据挖掘中的一个重要途径。自 Agrawal 和 Srikant^[17]提出的关联规则挖掘算法即 Apriori 算法以来,该算法被广泛应用于诸多领域,如 web 挖掘^[18]、网络入侵检测^[19]、生物信息学^[20]等。然而,这种挖掘算法通常具有很高的计算复杂度,特别是当数据集规模巨大时,其计算复杂度非常高。目前,在基于智能手机的日常行为模式挖掘方面,大多数研究工作亦都采用离线方式在服务器端进行模式挖掘,如文献[15]基于智能手机收集用户的上下文数据,离线在服务器上进行关联规则挖掘;文献[16]提出了一种上下文获取引擎(Acquisitional Context Engine, ACE),利用该引擎将所收集的上下文数据推送到远程服务器上进行同现模式的挖掘。与本文研究最相关的工作是 MobileMiner^[17], MobileMiner 聚集多种上下文数据并采用 Apriori 算法在智能手机上进行频繁同现模式的挖掘,

MobileMiner 使用固定滑动窗口(Fixed Slide Window, FSW)从上下文数据中提取上下文篮子,其中上下文数据每隔 30 s 被划分到一个篮子,为了减少运行时间,对重复的篮子进行了压缩,并在每个篮子上附加一个权值来表明它重复的次数。与 MobileMiner 不同的是,本文在用户交互时段的上下文数据上使用了一个动态滑动窗口,动态地对窗口进行调整,而且更重要的是,本文为每个上下文篮子添加了一个基于其持续时长的权重,从而将上下文篮子的频率和持续时间有机地结合起来,在此基础上,基于动态滑动窗口,进一步对 Apriori 算法进行扩展,提出了一个加权模式挖掘算法(Weighted Pattern Mining, WePM),该算法通过修改支持度定义,将上下文的持续时间和出现频率共同作为衡量上下文频繁模式的度量,从而挖掘出更合理准确的用户日常频繁行为模式。

2 模式挖掘框架

为了有效地在智能手机上进行用户日常频繁行为模式挖掘,基于 Android 平台设计并实现了一个频繁模式挖掘框架,如图 1 所示。该框架由上下文采集和频繁模式挖掘两部分构成。上下文采集的主要功能是从各种传感器包括物理传感器和虚拟传感器上采集各种原始传感器数据,而后采用不同的方法进行预处理,进而得到离散的上下文数据,采集和预处理工作都是在后台自动进行的。频繁模式挖掘部分的主要功能包括如下几个方面:一是首先使用动态滑动窗口将上下文数据划分成上下文篮子,上下文篮子是一组由不同的同现上下文组成的集合,例如上下文篮子{上午,实验室,电子邮件,WiFi}意味着用户早上在实验室连着 WiFi 写电子邮件。二是使用一个附加权重的模式挖掘算法对上下文篮子进行频繁模

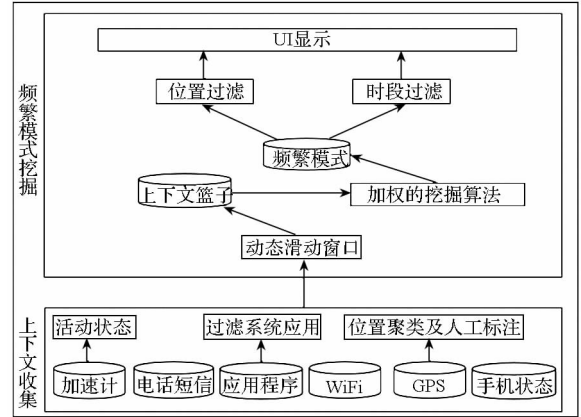


图 1 模式挖掘框架

Fig. 1 Pattern mining framework

式挖掘。最后,使用不同的过滤器对挖掘的频繁模式进行过滤及可视化。

3 上下文数据采集与处理

上下文数据采集是用户日常频繁行为模式挖掘的前提和基础。UbiqLog 是一个基于 Android 系统的生活日志收集框架^[21],该框架可以记录丰富的上下文数据,包括应用程序的使用、电话、短信、GPS、运动以及感知传感器数据等^[22]。本文以 UbiqLog 为基础,根据实际需要对其进行扩展,开发了一个能够满足实际需要又能采集各种上下文数据的数据收集工具 StarLog,并将其集成到模式挖掘框架当中。使用 StarLog 可以持续收集各种传感器数据、应用使用日志以及智能手机的状态记录等,本文将这些数据统称为纵向上下文数据。

使用 StarLog 所采集的数据都是原始的数据,需要进一步地处理形成具有一定含义的上下文数据,进而基于这些有含义的上下文进行进一步的行为模式挖掘。为此,针对加速度传感数据,本文基于阈值的方法来推测用户的简单活动状态,包括静坐、慢速和快速这三种状态。针对应用程序使用日志数据,开发了一个应用程序过滤器,去除与用户操作无关的信息以及系统级应用程序的记录信息,因为这些信息与用户的日常行为无关。位置是最重要的上下文信息之一,由于收集到的 GPS 坐标不具有语义信息,因此推测出相应的具有语义信息的位置是首先要解决的问题。考虑到智能手机资源有限以及避免用户过多地参与,减少其对用户日常生活的影响,本文设计并实现了一个定制的位置识别机制,如图 2 所示。首先,用户可以根据需要设置 GPS 传感器采样频率,收集和记录 GPS 位置信息,每天在恰当的时机询问用户去过的地方的个数 K ,而后采用 K -均值聚类算法对 GPS 进行聚类,并在地图上标注出所有聚类中心。如果一个聚类中心是一个新地点,则允许用户为其标记为像实验室、宿舍等语义标签。之后,新的标记位置被存储在语义位置库中,使用这种方法,用户只需要对每一个新地点标记一次。然后,计算 GPS 记录中的坐标和语义位置库中位置的距离,如果距离小于设置的平均误差的阈值,则该坐标被标记为相应的标签;否则,则标记为标签“在路上”。最后,聚合连续且相同的记录,以产生频繁模式挖掘模块的位置输入。

若用户输入的 K 值不准确,会对聚类结果造成比较大的影响,进而产生与实际位置相差较远

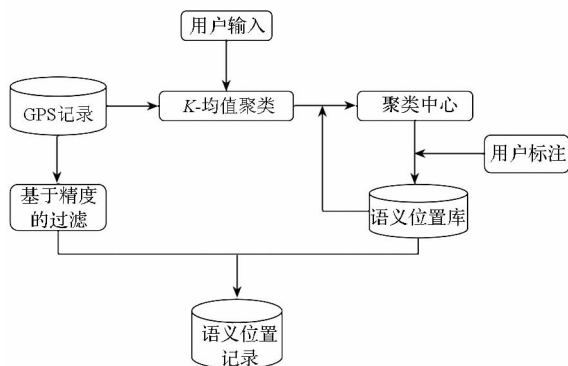


图 2 通过聚类和标注方法从 GPS 坐标中识别位置

Fig. 2 Identifying location from GPS coordinate using clustering and labeling method

的聚类中心,当用户对聚类中心进行标注时,会发现这些没有意义的聚类中心,程序支持用户重新输入 K 值以矫正聚类结果。因此,由于加入了用户的人工判断,错误的 K 值可以得到修正从而得到准确的聚类结果。

4 模式挖掘算法

4.1 动态滑动窗口

由于频繁模式挖掘算法基于上下文篮子进行模式挖掘,因此需要对带时间戳的上下文数据进行处理以提取上下文篮子。针对用户日常与智能手机交互的特点,提出了一种动态滑动窗口算法 (Dynamic Slide Window, DSW) 进行上下文篮子的提取。DSW 算法的输入为应用程序使用记录、通话和短信记录、位置记录、手机模式记录和活动记录等上下文相关数据,输出为上下文篮子列表。在 DSW 算法中,本文更关注于用户和智能手机交互的时段而不是整个使用时间,交互事件包括用户使用应用程序、打电话和发短信等。DSW 算法的核心思想是:对于时间连续的上下文数据,将每个时长不定的交互上下文出现的时段作为一个窗口,动态提取该时间窗口内的上下文到上下文篮子中。因此,DSW 算法首先将应用使用记录和通话、短信记录按时间顺序合并成一个列表,然后遍历此列表,每遇到一个交互记录,就扫描其他上下文记录,包括位置、手机模式和活动以获得相同时间段内相应的上下文。与交互记录同时出现的各种上下文数据被放到一个上下文篮子中。由于上下文记录中包含其开始和结束时间,所以可以计算出相应篮子的持续时间。令 t 表示持续时间,则篮子的权值为对持续时间取对数的原因是:对数函数在持续时间非常大的情况下可以减少对结果的影响。因此,持续时间长但出现频率小的上下

文篮子并不会被识别为频繁模式。

DSW 算法在处理过程中需要解决如下两个难点问题。第一个问题是,在同一个交互时段内其他上下文数据可能不是唯一的。例如,当用户使用微信时,他可能离开实验室走向宿舍,在这种情况下,上下文篮子中的位置包括在实验室和在路上两个值;如果诸如运动状态之类的其他上下文在同一时段也发生了变化,则情况变得更加复杂。考虑到人对时间的感知通常并不是非常精确的,并且通常有一个相对的时间粒度,因而拥有相似时间粒度的行为模式也是可接受的,所以当涉及不同值的上下文时,DSW 选择可能性最大的值。另一个问题是数据丢失问题,由于智能手机通常处于动态复杂的运行环境当中,可能会丢失一些上下文数据,为了保证有足够的上下文数据来生成上下文篮子,DSW 算法使用相应的历史记录补充丢失的上下文数据。

图 3 展示了 DSW 算法可能的输出结果和输出形式。在图 3 中,纵轴代表不同种类的上下文,横轴代表时间。从图 3 中可看到,用户在 9 点到 10 点之间使用了一次微博,上下文篮子 { 9—10, 微博, 实验室, 充电, 静坐, 振动模式, WiFi } 就被提取出来,同时其权值也被计算出来。

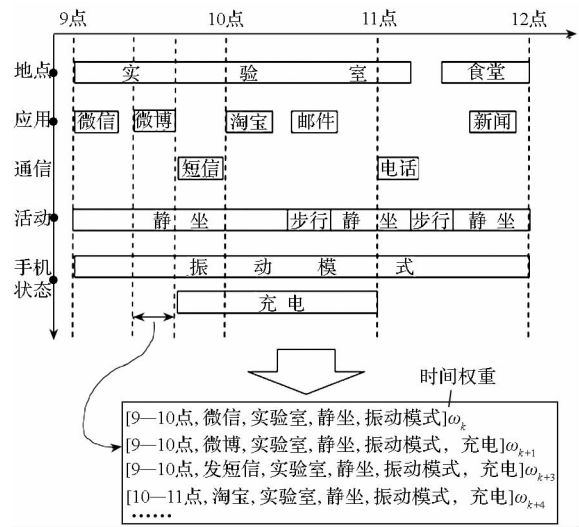


图 3 动态滑动窗口算法的输出结果

Fig. 3 Output of dynamic sliding window algorithm

DSW 算法具有如下两个优点:一是基于用户与智能手机的交互时段来提取上下文篮子,从而可以显著减少上下文篮子的数量;更重要的是,DSW 算法同时考虑了上下文的持续时间和出现频率两个因素,将两者有机地集合起来,从而可以挖掘更有意义的模式。文献[16]中使用 FSW 算法提取上下文篮子,通过压缩重复篮子数量的方法虽然减少了上下文篮子的个数,但是这种方法

的缺点是缺失了上下文的出现频率这一重要因素。

4.2 加权的模式挖掘

Apriori 算法^[17]是最经典的数据挖掘算法。在 Apriori 算法中,支持度是一个关键的参数。形象地讲,令 i_1 和 i_2 为两个项, b_i 为项集,令 $B = \{b_1, b_2, \dots, b_n\}$ 表示所有项集的集合,集合 $\{i_1, i_2\}$ 的支持度定义为其在 B 中发生的概率。如果一个项集的支持度超过了预定的阈值,则它就被定义为一个频繁项集。为了找出所有的频繁项集,通常需要反复扫描集合 B ,以计算出它的每一个子集的支持度。为了降低计算复杂性,Apriori 算法利用了支持度的向下闭包属性,即一个频繁项集的所有子集也是频繁的,所以 Apriori 算法首先生成大小为 k (初始为 1) 的频繁项集,然后开始重复生成大小为 $k+1$ 的所有候选项集,计算所有的候选项集以生成频繁项集 F_{k+1} ,当 F_k 是空集时循环结束。

为了支持从上下文数据中挖掘出用户日常的频繁行为模式,本文基于 Apriori 算法并对其进行扩展,重新定义支持度以将上下文持续时间和出现频率作为频繁的衡量标准,提出了一个加权的频繁模式挖掘算法。在 WePM 算法中,上下文数据是项,上下文篮子是项集,模式是频繁上下文项集。例如,频繁模式 { 18—19, 宿舍, 通话, WiFi } 意味着用户通常在 18 点到 19 点之间在宿舍中连接 WiFi 打电话。WePM 算法的输入为所有上下文数据,输出为频繁上下文项集。首先,WePM 通过 DSW 算法处理所有上下文数据来生成上下文篮子;其次,WePM 利用向下闭包属性来挖掘频繁项集,同时进行上下文篮子的权值计算。为了同时考虑上下文的持续时间和出现频率这两个因素,对支持度重新进行了定义,令 $\{b_1^{\omega_1}, b_2^{\omega_2}, \dots, b_n^{\omega_n}\}$ 表示上下文篮子,项集 X 的支持度定义如下:

$$\text{Support}(X) = \sum_{i=1}^n \text{contain}(b_i^{\omega_i}) \cdot \omega_i / \sum_{i=1}^n \omega_i \quad (1)$$

$$\text{contain}(p, q) = \{1 \text{ if } q \subseteq p \mid 0 \text{ otherwise}\} \quad (2)$$

其中 ω_i 是 DSW 算法中提到的篮子的权值。WePM 与加权时间模式挖掘算法 (Weighted Mining of Temporal patterns, WeMiT)^[16] 的不同在于对上下文频繁行为模式的不同衡量标准,定义中同时考虑了上下文持续时间和出现频率这两个关键因素,使其能够更加准确地反映用户的日常

频繁行为模式。本文在 Android 手机上实现了 WePM 算法,并验证了其在智能手机上可以有效地进行用户频繁模式的挖掘。

5 实验验证

目前,虽然有诸如 Reality Mining^[22], Device Analyzer^[23]和 StudentLife^[24]等公共移动上下文数据集,但是这些数据集并不是针对细粒度的日常频繁行为模式挖掘而收集的,因此无法满足本文的实验需要。

5.1 实验数据采集

招募了 21 名自愿参与的在校研究生进行为期 6 周的数据收集工作,并给予其相应的激励,如为每一个参与者提供 16 G 的 SD 卡等。在 21 名用户中,14 名男性,7 名女性。每个参与者在其智能手机上安装 StarLog,并使用相同的预定义配置。

StarLog 收集以下数据:①用户活动状态(包括静坐、低速、快速),每分钟计算一次;②GPS 记录,每分钟获取两次;③WiFi 记录;④通话记录,包括匿名的电话号码、呼叫类型(呼入、呼出、未接)、开始时间和持续时间;⑤短信记录,包括匿名的电话号码、短信类型和时间戳;⑥应用程序使用记录,包括应用程序名称、开始时间、结束时间和持续时间;⑦手机状态信息,包括手机充电状态、飞行模式、响铃模式和振动模式。数据收集持续了 6 周时间,如图 4 所示,一些用户参与的时间只有几天,而大多数用户参与时间超过 3 周。图 5 展示了每天收集的各种原始上下文数据的比例。每天生成的上下文记录超过 2000 条,其中 81%是活动和位置记录,因为它们的收集频率较高。

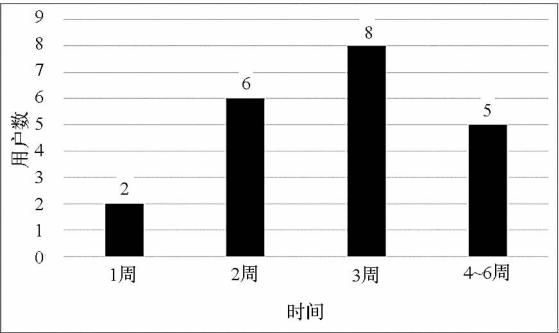


图 4 21 名用户的数据收集时间

Fig. 4 Duration of data collecting among 21 users

5.2 系统性能

实验中,主要使用一部 Moto X 智能手机来对

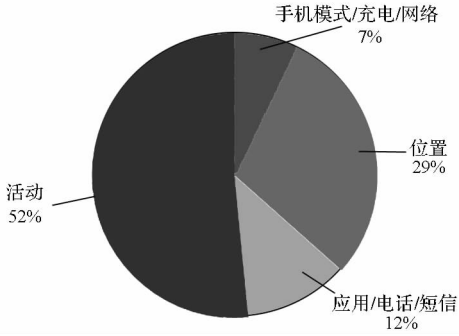


图 5 每天收集的各种上下文数据比例

Fig. 5 Ratio of various context collected everyday

本文所提出的算法性能进行评估,该智能手机配备了四核 2.5 GHz 的处理器和 2 G 内存。在实验中,位置识别是数据处理的第一步。每天晚上参与者被要求输入他们当天去过的地方的数量 K ,然后利用 K -均值聚类算法计算聚类中心。受益于其较低的复杂度,计算时间小于 1 s。因此,聚类结果可以立即显示。如果聚类中心已经被标记过,则会显示出一个绿色标志;如果是一个新的地方,则显示出一个红色标志。用户可以通过在地图上移动它们来调整其位置。然后,用户通过点击红色标记来输入新地点的名字。聚类中心的坐标和相应的语义标签都存储在本地。

为了验证 DSW 算法的有效性,基于所收集的数据,本文对 DSW 算法与 WeMit^[10]中使用的固定滑动窗口算法进行了对比实验,图 6 显示了在 35 d 的数据上,DSW 算法和 FSW 的执行时间,结果表明 DSW 每天只需要约 1 s,而 FSW 平均时间约 3 s。DSW 算法效率更高的原因在于其仅在用户交互时间内提取上下文数据,而 FSW 提取所有时间的上下文数据。同时,DSW 占用约 10 MB 的内存和 24%的 CPU 使用率。因此,采用 DSW 算法几乎不影响用户的体验。

由于支持度的定义不同,而且由于目前尚没有公认的评价挖掘结果准确性的有效方法,因此在实验中,仅测量了 WePM 算法的性能,本文使用了三个用户 35 d 内的数据在一台 Moto X 手机上运行 WePM 算法,对 WePM 算法的性能进行测量。图 7 显示了支持度在 1% 至 6% 范围内的结果。

从图 7 中可以看出,在不同用户的数据上运行 WePM 的时间是近似的,即使当支持度被设置为 1% 时,WePM 算法完成挖掘工作需要的时间也不到 4 min。并且,本文所开发的模式挖掘框架提供了不同的挖掘周期,方便用户可以灵活地进行设置,随时了解其某一段时间内的频繁行为模

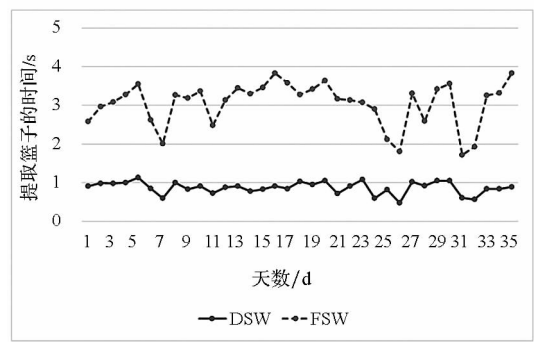


图6 DSW 和 FSW 提取上下文篮子的执行时间
Fig. 6 Executing time of DSW and FSW

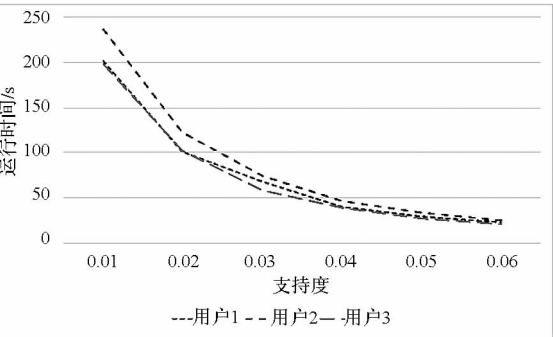


图7 WePM 算法的运行时间
Fig. 7 Executing time of WePM

式。假设挖掘周期设置为 35 d,用户只需要花费不到 4 min 的时间就能挖掘出其 35 d 内的频繁行为模式,因而很容易被用户所接受。

5.3 模式可视化

图 8 显示了用户 A 和 B 在实验室和宿舍的频繁行为模式。图中每一行代表一个模式,这些模式包含了一些如 WiFi 或者手机处于充电状态的上下文数据。从图 8(a)可以看出,用户 A 通常在宿舍连接 WiFi 并使用微信和网易云音乐,并且用户 A 通常处于静坐状态,还经常在宿舍打电话或者给手机充电。从图 8(b)可以看出,用户 A 通常在实验室连接移动网络并使用微信和有道词典;同时还经常给手机充电。用户 A 的活动状态通常是静坐,因为其在实验室时有很多工作要做,很少有时间使用智能手机。图 8(c)显示了用户 B 在实验室的模式,他与用户 A 在同一实验室。可以看到用户 B 在实验室通常连接 WiFi,并使用很多应用,比如微信、墨迹天气、土豆视频和聚美优品等,也经常在实验室给手机充电。尽管用户 A 和用户 B 在同一实验室,但他们的模式却非常不同,用户 B 比用户 A 更趋向于使用多种应用。但是,用户 A 和用户 B 都经常给手机充电并处于静坐状态。



(a) 用户 A 在宿舍的模式 (b) 用户 A 在实验室的模式 (c) 用户 B 在实验室的模式
(a) Pattern of A in dormitory (b) Pattern of A in laboratory (c) Pattern of B in laboratory

图 8 用户 A 和 B 的频繁的模式
Fig. 8 Frequent behavior patterns of A and B

此外,本文过滤了包含时间因素的模式,提供不同粒度的时间,比如一周的某天,按小时粒度划分的时间和像早上或晚上等按更高粒度划分的时间。如图 9 所示,显示了用户 A 按小时粒度划分的行为模式。



图 9 按小时粒度挖掘的行为模式
Fig. 9 Mined behavior pattern at hour granularity

在图 9 中,数字“7—8”表示在上午 7 点到 8 点之间的时间段,可以看到用户 A 通常花费大量的时间在实验室,23 点以后才返回宿舍。当其在宿舍时,便会连接 WiFi。而在实验室是没有 WiFi 的,这和上面的结果很相似。此外,用户 A 通常在中午 11 点到 12 点之间使用微信,这期间他可能去吃午饭或者在放松休息。用户 A 在上午 8 点到 11 点通常会给手机充电,这可能是因为在实验室充电比较方便。

6 结论

本文提出了一个全新的基于智能手机的用户

频繁模式挖掘框架,在该框架中,根据纵向上下文数据的特点,提出了 DSW 方法,相比于 FSW,DSW 可显著降低执行时间,处理每天收集的上下文数据仅需要约 1 s;基于 DSW,将上下文出现频率和持续时间这两个因素有机地结合起来,进一步提出了 WePM。通过实验验证表明,该算法能够在智能手机上高效地进行用户频繁行为模式的挖掘,当支持值设置为 1% 时,WePM 从 35 d 的上下文数据中挖掘频繁模式的时间都在 4 min 以内。最后,以可视化方式展现用户日常的频繁行为模式。本文将进一步在保证用户隐私的情况下,支持用户将其行为模式上传到云端,进而基于众多的个人行为模式来聚合和分析群体的行为模式。

参考文献 (References)

- [1] Khan W Z, Xiang Y, Aalsalem M Y, et al. Mobile phone sensing systems: a survey[J]. IEEE Communications Surveys & Tutorials, 2013, 15(1): 402–427.
- [2] Lu H, Pan W, Lane N D, et al. SoundSense: scalable sound sensing for people-centric applications on mobile phones[C]//Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services, ACM, 2009: 165–178.
- [3] Chon J, Cha H. Lifemap: a smartphone-based context provider for location-based services[J]. IEEE Pervasive Computing, 2011, 10(2): 58–67.
- [4] Miluzzo E, Lane N D, Fodor K, et al. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceMe application[C]//Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, ACM, 2008: 337–350.
- [5] Lane N D, Pengyu L, Zhou L, et al. Connecting personal-scale sensing and networked community behavior to infer human activities[C]//Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2014: 595–606.
- [6] Ouchi K, Doi M. Smartphone-based monitoring system for activities of daily living for elderly people and their relatives etc[C]//Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, ACM, 2013: 103–106.
- [7] Sumida M, Mizumoto T, Yasumoto K. Estimating heart rate variation during walking with smartphone[C]//Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2013: 245–254.
- [8] Gu W, Yang Z, Shanguan L, et al. Intelligent sleep stage mining service with smartphones[C]//Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2014: 649–660.
- [9] Lu H C, Tseng V S, Yu P S. Mining cluster-based temporal mobile sequential patterns in location-based service environments[J]. IEEE Transactions on Knowledge & Data Engineering, 2011, 23(6): 914–927.
- [10] Chen T S, Chou Y S, Chen T C. Mining user movement behavior patterns in a mobile service environment[J]. IEEE Transactions on Systems Man and Cybernetics-Part A Systems and Humans, 2012, 42(1): 87–101.
- [11] Shin C, Hong J H, Dey A K. Understanding and prediction of mobile application usage for smart phones[C]//Proceedings of the ACM Conference on Ubiquitous Computing, ACM, 2012: 173–182.
- [12] Xu Y, Lin M, Lu H, et al. Preference, context and communities: a multi-faceted approach to predicting smartphone app usage patterns[C]//Proceedings of the 17th Annual International Symposium on International Symposium on Wearable Computers, ACM, 2013: 69–76.
- [13] Zhu Y F, Zhang Y B, Shang W X, et al. Trajectory enabled service support platform for mobile users' behavior pattern mining[C]//Proceedings of 6th Annual International Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous, 2009: 1–10.
- [14] Cao H, Bao T, Yang Q, et al. An effective approach for mining mobile user habits[C]//Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM, 2010: 1677–1680.
- [15] Nath S. ACE: exploiting correlation for energy-efficient and continuous context sensing[C]//Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, ACM, 2012: 29–42.
- [16] Srinivasan V, Moghaddam S, Mukherji A, et al. MobileMiner: mining your frequent patterns on your phone[C]//Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2014: 389–400.
- [17] Agrawal R, Srikant R. Fast algorithms for mining association rules[C]//Proceedings of 10th International Conference on Very Large Data Bases, VLDB, 1994, 1215: 487–499.
- [18] Pei J, Han J, Mortazavi-Asl B, et al. Mining access patterns efficiently from web logs[C]//Proceedings of 4th Pacific-Asia Conference, PAKDD, 2000: 396–407.
- [19] Barabási D, Couto J, Jajodia S, et al. ADAM: a testbed for exploring the use of data mining in intrusion detection[J]. ACM SIGMOD Record, 2001, 30(4): 15–24.
- [20] Xiong H, He X, Ding C H Q, et al. Identification of functional modules in protein complexes via hyperclique pattern discovery[C]//Proceedings of Pacific Symposium on Biocomputing, 2005, 10: 221–232.
- [21] Rawassizadeh R, Tomitsch M, Wac K, et al. UbiqLog: a generic mobile phone-based life-log framework[J]. Personal and Ubiquitous Computing, 2013, 17(4): 621–637.
- [22] Eagle N, Pentland A. Reality mining: sensing complex social systems[J]. Personal and Ubiquitous Computing, 2006, 10(4): 255–268.
- [23] Wagner D T, Rice A, Beresford A R. Device analyzer: large-scale mobile data collection[J]. ACM SIGMETRICS Performance Evaluation Review, 2014, 41(4): 53–56.
- [24] Wang R, Chen F, Chen Z, et al. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones[C]//Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2014: 3–14.