Investigating how users engage with a pedometer app

Parvin Asadzadeh

University of Glasgow

parvin.asadzadehbirjandi@glasgow.ac.uk

John Rooksby

University of Glasgow

john.rooksby@glasgow.ac.uk

Matthew Chalmers

University of Glasgow

matthew.chalmers@glasgow.ac.uk

Abstract

Mobile application usage data has been investigated by many researchers to explore reasoning about users' contexts and their routines. A large number of early studies in this area provide relatively simple analyses, and some more recent works look more deeply at the patterns of logged events. This paper explains a new work on the analysis of interaction logs collected from a pedometer-based mobile app to extract different usage patterns of the app.

Author Keywords

Personal informatics; mobile applications; interaction logs; app usage patterns.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction

Understanding how users interact with a smartphone application is central to its informed redesign [1]. While well-designed user studies and surveys reveal specific user behaviours, analysis of app-level interaction logs extracts usage patterns in larger user populations. Many researchers have explored how people use their smartphones [2-6]. Further, many studies collect users interactions with specific mobile apps to perform different user studies and app evaluations [7-10].

4,483
TODAY'S STEPS

5,369
THIS WEEK'S
DAILY AVERAGE

6,000
THIS WEEK'S
DAILY GOAL
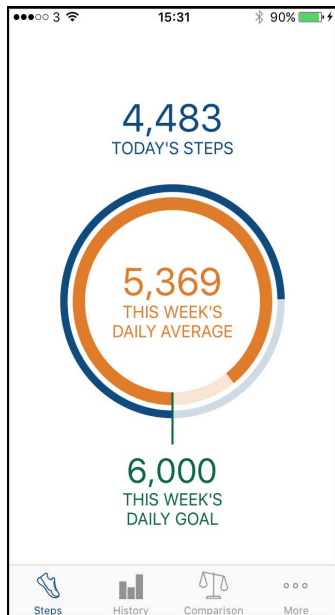
Steps   History   Comparison   More

Figure 1(a): Quped Steps Screen: users can view their progress in real-time on this screen. It shows the user's step-counts on current day and daily average of user's step-counts in current week (week starting with Monday).

Gouveia et al. [7], in particular, analyse the interaction logs from an activity tracker to explore the use of the app. However, their analysis is limited to descriptive statistics of the app usage data.

In this paper, we introduce a statistical method to extract usage patterns of individual apps. We apply the method on interaction data logged from Quped, a pedometer app that runs on iPhones 5s and above. Quped is currently available through Apple's app store. We explain the analysis of our initial dataset that consists of the logs from 36 users that have used Quped for at least four weeks.

Quped App

Quped (Figure 1a-c) is a pedometer app that allows users to keep track of their daily step-counts and provides them with a personalised weekly goal.

Features
Quped 1.0 consists of three main screens. Their features are explained in Figures 1a-c.

Data Logging
Users were informed that their data would be collected and analysed for research purposes. We logged data from consenting users that are over 18. The logged data is anonymous. Data logs are mainly interaction logs. They include usage data surrounding the user's interaction with the app, including information about when the app is opened and closed, what screens are visited and all button clicks within the app. Moreover, we log how many steps users take each day and what goals are set for them.

Quped Study

Quped was released on Apple's App Store late February 2016. So far, 66 users have voluntarily started the app and consented to data collection. Up until now, 44 (67%) of users used Quped longer than a week, 36 (54%) longer than four weeks and 22 (33%) longer than eight weeks.

To run an initial experiment, we create a dataset by including the interaction logs generated by 36 users that have used the app for at least four weeks. These are users for whom the time gap between their installation date and their last app launch is more than four weeks. Our dataset contains the logs from their first four weeks use of app.

Extracting usage patterns from Quped data
At the lowest level our data consists of time-stamped interaction logs, which are captured whenever a user accesses a user interface element. We aggregate all logs in our dataset into usage sessions. A session begins when a user opens the app and ends when the app is closed.

To extract usage patterns, we cluster users at two phases. At a high level, we structure all sessions into three large groups and label them with their group-type, as discussed in the *Sessions Analysis* section. Then, for each user we create a session-stream that shows the type of sessions that the user generates and the time-gaps between them. We apply hierarchical clustering on created session-streams to identify different usage patterns across all users.
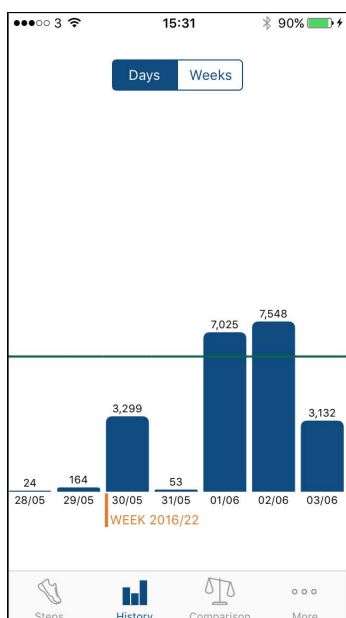
Figure 1(b): Quped History Screen: users are able to go through their past daily and weekly steps, which is adjustable by a segmented control on top of the screen.
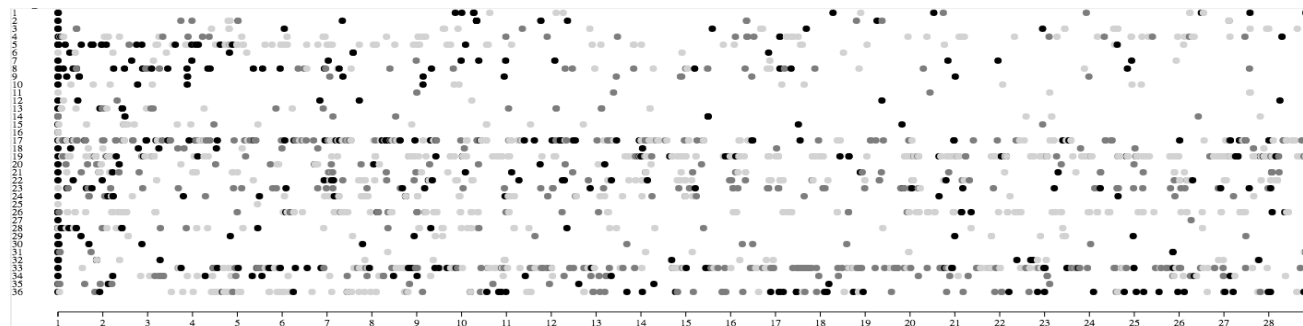

Figure 2: Plot shows users sessions during 28 days. Each row is a user. A circle indicates a session. Sessions of types A, B and C are shown with light grey, dark grey and black, respectively.

Sessions Analysis

The 36 experimental users generated 2392 individual usage sessions (median usage sessions per user of 25, IQR: 15-64) over their first 28 days of app use. User sessions were mostly brief, with 50% of them not longer than 10 sec and 86% not longer than 60 sec. The median session duration is 9.9 sec.

We structure the sessions into the following three types via clustering. 55% of sessions are of type (A). These are the sessions where users open the app to check their step-counts on the first screen with no further interaction. The median session duration in this group is 5.84 sec. Sessions of type (B) are the sessions in which the user also checks her history of step-counts. They constitute 26% of the sessions and their median length is 15.68 sec. The remaining 19% of the sessions are of type (C) and are the session in which the user explores the app by comparing her steps with other

users and/or looking into app settings. The median length of this type of sessions is 32.84 sec.

Users' Session-streams Analysis

Each user generates a sequence of sessions starting from the installation date. For each user, we generate a session-stream similar to how clickstreams are formed in [11]. This presentation captures both the types of sessions generated by users and the magnitude of time gaps between them, as explained below.

For example, $A(t_1)A(t_2)B(t_3)C$ is the session-stream for a user with four usage sessions of type A, A, B, and C. Time gap t_i is the time interval between the ith and $(i+1)^{th}$ sessions. Similar to [11], we replace time gaps with discrete time buckets, represented by $T_i \{1…m\}$, where m is the number of time buckets. In our experiment we use nine time buckets of T_1: [<a day]; T_2: [1 day-2 days]; T_3: [2 days-3 days]; … ; T_8: [7 days-10 days] and ; T_9: [>10 days].
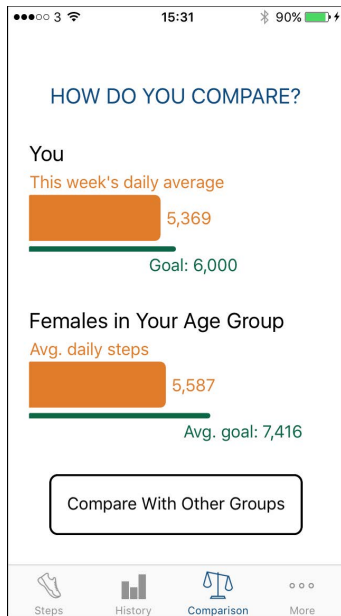
Figure 1(c): Quped Comparison Screen: users are able to compare their steps with others in different demographic groups. We ask about user demographic data only if they want to use this feature of the app. This is to avoid putting off the users when they start using the app.

We use an agglomerative hierarchical clustering algorithm (as implemented in the *clusterfck* javascript library [12]) to cluster session-streams and to identify common usage patterns. Using this algorithm, each user (represented by her session-stream) is in a separate initial cluster, and pairs of clusters are merged based on how similar they are. The result is a hierarchical tree plot.

To measure the similarity between any two session-streams, we convert them into count-arrays. The count-array of a session-stream holds the occurrence count of every session type A,B,C and time bucket T_1-T_9 in that session-stream. For example, the session-stream of $A.T_8.A.T_2.A.T_1.C.T_1.B$ will be converted to the count-array of $[3,1,1,2,1,0,0,0,0,0,1,0]$. Euclidean distance of the normalised count-arrays is used to measure the similarity between any two session-streams (i.e. users).
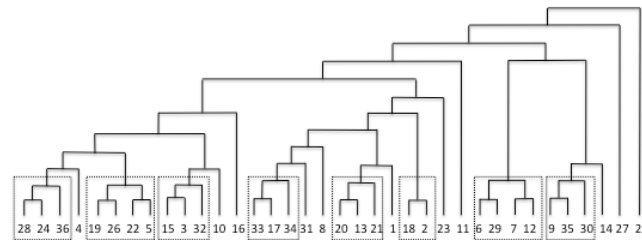

Figure 3: Plot shows clustering of 36 users (session-streams). At the bottom of the tree are users 1-36. They are merged into clusters as we traverse up.

The sessions generated by all 36 users are shown as colored circles in Figure 2. Figure 3 shows the result tree plot of clustering the session-streams of these users. This presentation shows which users are merged

at each tree level. It allows us to understand what possible clusters are and how they are formed. Users with higher similarities are merged in the lower levels of the tree. Figure 3 highlights the clusters formed after two levels of merging. Here we provide an analysis of the three of these clusters and explain them considering their associated count-arrays. Our analysis shows that different usage patterns (within formed clusters) are identifiable. However, the clusters will be more distinct when the method is applied on larger datasets.

Users Clusters Discussion
The second cluster from left consists of users 19,26,22,5. These are the users that a high fraction (68%-85%) of their sessions are of type A and they use Quped on most days (over 92% of app openings are within one day of each other). Similarly, users 33, 17, 34 use Quped on most days (87%-100% of app openings within one day of each other). However, only 14 to 18% of their sessions are of type A. They explore Quped more deeply than previous cluster with most sessions (over 82%) being of type B and C. On right side of the tree, users 6,29,7,12 form another small cluster. These users use Quped more sparsely, with only 36-50% of app launches within one day of each other. Their session types are also more distributed with 33-67% of type A sessions. Users 25 and 27 at the far right end of the tree are two sample users that are not merged into larger clusters at this level. These users used Quped only twice during the 28 days interval (see Figure 2).

In our dataset, there is no correlation between the users step-counts/goals and how they engage with the app (their session-streams). Here we discuss the step-

counts collection and goal setting in Quped and we provide a general analysis of collected step-counts data. Then we discuss how step-counts collection and goal setting is improved in a newer version of Quped.

Step-counts Collection and Goal Setting
We logged daily step-counts of users along with their goals. Once installed, Quped retrieves and stores the step-counts of the past six days (available in internal memory of the built-in motion co-processors), which is always available if motion and fitness tracking is enabled by users. Quped also provides a weekly goal for each user based on their previous week's step-counts and goal. Users are first provided with a goal on the first Monday after the app installation. Thus, users have a goal set for them between 1 to 6 days after installation depending on the date on which they install the app. On each Monday afterwards, their goal is increased if they reach their previous week's goal. Afterwards, users daily step-counts are fetched whenever they open the app (up to 6 days past) or when a background fetch (BF) is invoked (os-dependent). Therefore, depending on how often users open the app and how often BF is invoked, there might be some missing daily step-counts in our dataset.

On average, the 36 users in this study had their first goal on the 4th day after installation. We could collect step-counts of users for all 28 days for 20 (55%) of users. For the remaining users, median of lost daily steps is 5.5 days. On average, users reached their daily goals on 34% of the days (for which the data is available). User goals range from 1000 to 14900 steps. In average, the difference between daily goal and step-count of users are from −0.88 to 1.70 of their goal.

We plan to incorporate step-counts and goals of users to inform clustering on a larger dataset. To do so, step-counts/goal collection has been improved in Quped 1.1 (available on Apple app store recently) in two main ways. First, step-counts on the days that are past more than six days (which may not be available to retrieve from built-in motion co-processors) are accessed through HealthKit, if user authorises. Secondly, users are able to set their weekly goal if they wish. This is to make sure that goals are always encouraging for users and adapted to their conditions.

Conclusion and Future Work

In this paper, we describe the concept of session-streams that are streams of sessions and time gaps between them that shows how and when users have interacted with an app during a specific period of time. We summarise the results of applying a hierarchical clustering algorithm on session-streams of 36 users of our developed app, Quped, over 28 days. Quped is still actively in use by many users and we plan to attract more users via public app stores. Our initial findings validate our method in distinguishing different app usage patterns. Future work involves analysis of larger datasets to inform redesign of Quped for different groups of users. Further, we plan to incorporate step-counts and goals of users to inform clustering once a larger dataset is created.

Acknowledgements

References

1. Oana Andrei, Muffy Calder, Matthew Chalmers, Alistair Morrison, and Mattias Rost. 2016. Probabilistic formal analysis of app usage to inform

redesign. *In International Conference on Integrated Formal Methods* (iFM'16), 115-129.

2. Niels van Berkel, Chu Luo, Theodoros Anagnostopoulos, Denzil Ferreira, Jorge Goncalves, Simo Hosio, and Vassilis Kostakos. 2016. A systematic assessment of smartphone usage gaps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 4711-4721. http://dx.doi.org/10.1145/2858036.2858348

3. Qiang Xu, Jeffrey Erman, Alexandre Gerber, Z. Morley Mao, Jeffrey Pang, and Shobha Venkataraman. 2011. Identifying diverse usage behaviors of smartphone apps. *In Proceedings ACM SIGCOMM Conference on Internet Measurement* (IMC'11)*, 329-344.* http://dx.doi.org/10.1145/2068816.2068847.

4. Matthias Böhmer, Christian Lander and Antonio Krüger. 2013. What's in the apps for context? Extending a sensor for studying app usage to informing context-awareness. *In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (UbiComp '13), 1423-1426. http://dx.doi.org/10.1145/2494091.2496038.

5. Alina Hang, Alexander De Luca, Jonas Hartmann, and Heinrich Hussmann. 2013. On app, where art thou? On app launching habits of smartphone users. *In Proceedings of the Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI'13)*, 392-395.* http://dx.doi.org/10.1145/2493190.2493219.

6. Karen Church, Denzil Ferreira, Nikola Banovic and Kent Lyons. 2015. Understanding the challenges of mobile phone usage data. *In Proceedings of the Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI'15), 504-514. http://dx.doi.org/10.1145/2785830.2785891.

7. Rúben Gouveia, Evangelos Karapanos and Marc Hassenzahl. 2015. How do we engage with activity trackers? A longitudinal study of Habito. *In Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing* (UbiComp '15), 1305-1316. http://dx.doi.org/10.1145/2750858.2804290.

8. Lenin Ravindranath, Jitendra Padhye, Sharad Agarwal, Ratul Mahajan, Ian Obermiller, and Shahin Shayandeh. 2012. AppInsight: mobile app performance monitoring in the wild. *USENIX Symposium on Operating Systems Design and Implementation* (OSDI'12), 107-120.

9. Niels Henze, Enrico Rukzio and Susanne Boll. 2011. 100,000,000 taps: analysis and improvement of touch performance in the large. *In Proceedings of the Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI'11)*, 133-142.

10. Donald McMillan, Alistair Morrison, Owain Brown, Malcolm Hall, and Matthew Chalmers. 2010. Further into the wild: running worldwide trials of mobile systems. *In Proceedings of the Conference on Pervasive Computing* (Pervasive'10), 210-227.

11. Gang Wang, Xinyi Zhang, Shiliang Tang, Haitao Zheng, and Ben Y. Zhao. 2016. Unsupervised clickstream clustering for user behavior analysis. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 225-236. http://dx.doi.org/10.1145/2858036.2858107.

12. Clusterfck: A JavaScript agglomerative hierarchical clustering library. Retrieved on 15 May 2016 from https://harthur.github.io/clusterfck/.