Social Context Discovery from Temporal App Use Patterns

Panagiotis Papapetrou
Stockholm University
Forum 100, Isafjordsgatan 39
Stockholm, 164 40 Sweden
panagiotis@dsv.su.se

George Roussos
Birkbeck College
University of London
Malet Street
London, WC1E 7HX UK
g.roussos@bbk.ac.uk

UbiComp '14, September 13 - 17 2014, Seattle, WA, USA

ACM 978-1-4503-3047-3/14/09. . . $15.00.
http://dx.doi.org/10.1145/2638728.2641699

Abstract
A key ingredient of mobile computing is automated adaptation of system behaviour to match user context. In this paper we investigate how temporal patterns of app use can reveal the social context of the user, in the sense of their specific social role during a period of interaction. Individual users typically have multiple distinct identities associated with different social roles such as professional and family members. We are specifically interested in exploring whether we can employ Device Analyzer data to construct distinct profiles for each of these roles. We introduce a temporal sequence clustering technique that successfully identifies periods associated with such distinct social contexts.

Author Keywords
Mobile Analytics, Temporal Sequencing Mobile Analytics, Temporal Sequencing.

ACM Classification Keywords
H.5.m [Information interfaces and presentation]: Miscellaneous.

General Terms
Device Analyzer, Ubicomp Programming Challenge Device Analyzer, Ubicomp Programming Challenge.

Introduction

Context-awareness is widely recognised as a desirable property for mobile computing and in recent years considerable effort has been invested by the research community in developing modelling and reasoning techniques that allow applications to adapt to changing physical, computation and activity-related context information [2]. Less attention has been given to social context that is context related to the current social situation of the user, likely due to the complexity associated with this task [3]. Indeed, individual users of mobile computing have multiple identities expressed in particular contexts, and associated with distinct or overlapping relationships, roles and communities, and that attach to different contexts.

Social context is arguably the most powerful type of context to employ in applications and in this paper we propose a methodology which addresses specifically the need to identify periods of time that correspond to particular contexts associated with different social roles. Using the Device Analyser (DA) data set [9] we explore this hypothesis in the context of app use so as to identify specific elements of user and device context that are influential in this activity. We initiate this exploration by looking at temporal patterns of app activation and use in the sample DA dataset, which we consider as the first step towards a comprehensive effort to explore the whole range of device data.

Rationale

The availability of a general framework for understanding the different facets of personal identity in the context of mobile computing provides distinct benefits to context adaptation [6]. This reflects the observation that competing demands of individuality and community, and the many forces bear on a person's sense of self [8]. In past work we identified three characteristics that stand out in adequate treatments of identity:

- The locality principle says that identities are situated in particular contexts, relationships, roles and communities, and that we may have different or overlapping identities attaching to different contexts.
- The reciprocity principle says that both sides in a relationship need to know what is going on so that they can check and correct each others perceptions.
- The principle of understanding says that identity serves in two-way relationships as a basis for mutual understanding.

The locality principle has been identified in various treatments of the self [4], and has significant implications for mobile computing. It implies, for example, that a global or universal identity makes little sense. We cannot expect consumers of mobile services to be comfortable with a single identity profile in relation to a universe of activities and services that entail all aspects of their life.

Rather, we can expect a strong preference to maintain different identities attaching to different functions, roles and communities, and to have control over these. This would explain the overall negative reaction of the participants of the focus groups to ubiquitous retail since users of the system were characterised singularly as consumers. We believe that since a system that extends to all types of activities including professional and family, refusing to acknowledge the different identities, the system did not address locality concerns. For this reason

it can be perceived as being designed to benefit the business only without taking into account the users needs.

The locality principle also highlights the need to balance two forces in mobile computing: first, the need to respect the consumers localized and multiple identities, and second, the significant advantages of open, collaborative and ubiquitous mobile business. Although it may be convenient to share consumer data with trading partners this action is liable to destroy trust. It is not that the details in question involve anything profoundly secret or private to the consumer, but rather that the localized identity developed via significant personal investment is forcefully removed by an external entity and used beyond the locality that has been developed.

Data Processing

We apply a two-stage pre-processing step to the raw Device Analyzer data sets:

1. Extract complete records associated with app use.
2. Convert reduced logs into interval-based event sequences.

Stage 1 involves the application of a selection filter implemented as an AWK script, applied to each individual source file separately. As a result form each raw DA file we compose a reduced app use log.

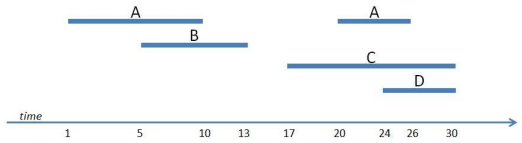

Figure 1: Example of an e-sequence of 5 temporal intervals.

Stage 2 involves the processing of the reduced logs into interval-based event sequences (e-sequences), with the following form:

```
<seq id> <class id> {<event intervals>}
```

where *sequence ID* is a unique identifier provided by the pre-processing function sequentially and *event intervals* is a set of events characterised by a start and an end time as follows:

```
<event ID> <start> <end>
```

where event ID is the anonymised app handle as retrieved by the raw DA log files, and the start and end time delimit the bounds of the event and are calculated in the manner detailed below. The result of this processing step is that in each sequence we incorporate one or several events occurring over a time interval. Such sequence is commonly known as *e-sequence* [5]. A visual example of an e-sequence is given in Figure 1, and it corresponds to the following e-sequence:

```
{<A,1,10> <B,5,13> <C,17,30> <A,20,26> <D,24,30>}
```

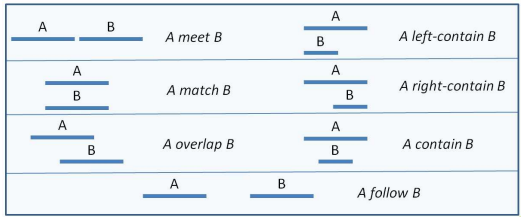

Figure 2: The seven temporal relations between two event-intervals that are considered in this paper.

It becomes apparent that in an e-sequence there exist temporal relations between the event intervals. Based on Allen's model for temporal interval relations [1], given two

event intervals A and B, we consider the seven temporal relations shown in Figure 2.

Moreover, each e-sequence is also labeled with a class identifier that denotes related time period in the day that it occurs following the convention:

```
M1   1: 05:00 - 08:59
M2   2: 09:00 - 12:59
A1   3: 13:00 - 16:59
A2   4: 17:00 - 20:59
EV   5: 21:00 - 00:59
NT   6: 01:00 - 04:59
```

To calculate the start and end times for each record, we follow this process: we begin by identifying which event(s) are on, that is which apps are active, within each hourly slot and we terminate the sequence at the following hourly slot when the event transitions to off state.

For example, all active apps between 05:00 and 11:00 belong to a single sequence with sequence ID 5432. For each app the starting hourly slot is recorded as well as the ending hourly slot so that we encode the fact that app `gfdsa` is active between 05:24 and 07:02:

```
1  M1  A    5   7.
```

It is of course possible to delimit sequences more or less frequently than at an hourly basis and the we investigate the effects of this choice further.

The output of this two-stage pre-processing process is then streamed into a clustering algorithm that employs a novel approach for the computation of distances between sequences such as the ones described above [5]. Using this metric we can directly perform agglomerative clustering with the Ward distance, which allows the observation of similarities between sequences characterising specific periods during the day.

Sequence Clustering

We used `IBSM`, shorthand for Interval-Based Sequence Matching, a novel method proposed by Kotsifakos et al. [5] for assessing the similarity of two e-sequences. IBSM performs e-sequence matching by mapping each e-sequence to its corresponding *re-sized event table* representation. This representation is obtained in two phases:

- **event table construction:** each e-sequence is converted to an event table, where each row corresponds to a event label (in our case an app) and each column is a time slot. Each cell in the event table records whether an event is active at each time slot (indicated by 1) or not (indicated by 0). If more than one instance of the same event are active at a certain time slot, then the value of the event table in that slot equals the number of active events.
- **event table re-size:** the event table is then simply re-sized by performing bi-linear interpolation on the columns so as to ensure that each table has the same number of columns.

Finally, IBSM computes the Euclidean distance of these representations (the two re-sized event tables). As shown in Kotsifakos et al. [5] IBSM performs e-sequence matching in time linear to the maximum length of the involved e-sequences. Additional heuristics are introduced to speed up this computation, such as alphabet reduction (removal of the very sparsely active event labels) and

sampling (uniformly random selection of columns of the event table).

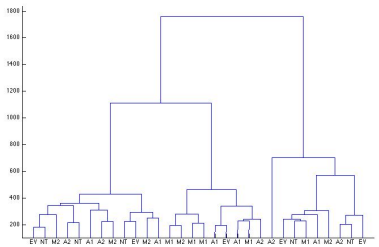

Figure 3: The deondrogram produced by agglomerative clustering under the Ward distance for Dataset 1. We observe three dominant clusters.

Using the above technique the similarity scores between the e-sequences are computed and then fed to the agglomerative clustering algorithm. Examples of the produced dendrograms are shown in Figures 3 and 4 for two of the datasets using hourly slots for defining the event intervals. The horizontal axis represents individual clusters identified using the labelling convention associated to the daily period during which they occur as discussed above. The horizontal axis shows the calculated similarity between connected clusters so that shorter vertical lines represent smaller calculated distances and thus a closer match. This procedure was applied to all datasets considered to calculate clusters for both hourly and half-hourly slots.

Results

In this study we considered individual user datasets selected from the DA repository. After the application of the processing steps described in the previous Section, we investigated the structure of the clusters identified using an interactive exploration methodology to analyse the obtained visualizations following the approach proposed by Seo and Sneiderman [7]. We discovered that in all cases there are three dominant clusters.

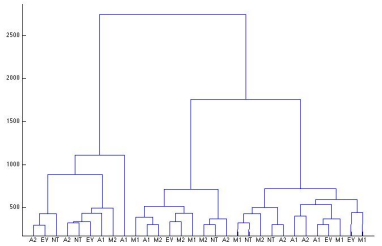

Figure 4: The deondrogram produced by agglomerative clustering under the Ward distance for Dataset 2. Again we observe three dominant clusters.

This phenomenon was present for both types of event interval generation (using hourly and half-hourly slots). We further scrutinised the clusters by studying their *purity*, which measures the degree of overlap between the event intervals contained in the three clusters. More precisely, we first compute the degree of *impurity* of a clustering, which corresponds to the fraction of clusters each event label (i.e., app) participates in on average. Hence, purity equals $(1-\text{impurity})$, so that purity of 1 means that all event labels participate in only one cluster. According to our working hypothesis discusses previously, we expect purity to be close to but less than one is all useful cases, since at each distinct time period there is one dominant role that dictates the principle pattern of app use scewed due to overlaps reflecting patterns of secondary social context. In all cases studies conducted we considered the three most distinctive clusters.

Our findings are shown in Table 1, where we can see that the proposed method can achieve clusterings of high purity, and hence distinctive apps within each cluster. We can also observe that as we increase the level of granularity (slots) we obtain even higher purity as fewer role shifts occur during the specified period.

	Hourly slot	Half-hourly slot
Dataset 1	0.87	0.90
Dataset 2	0.89	0.91
Dataset 3	0.86	0.89
Dataset 4	0.92	0.94
Dataset 5	0.93	0.94

Table 1: Purity of clustering when we use hourly and half-hourly slots for generating the event intervals. In all cases (five datasets) we considered the three most distinctive clusters.

Conclusions

We investigated the importance and applicability of temporal patterns in app use for identifying social contexts of users. DA records are first converted to sequences of labeled temporal intervals on which we apply the IBSM algorithm to compute their similarity which is subsequently employed to identify clusters via agglomerative clustering. We discover that the identified clusterings have high purity, which suggests distinct profiles of app usage. This matches our hypothesis that temporal app use profiles can reveal the distinct social roles of individual users. Although clearly requiring further investigation, the presence of three dominant clusters in all cases is suggestive of three principal modes likely to reflect patterns of professional, family and leisure activity correspondingly, a hypothesis corroborated by the specific temporal labelling associated with each cluster.

We are currently processing the complete DA dataset following the methodology presented in this paper with a view to further explore the clustering structure identified, In particular we examine specific app patterns within the clusters and investigate the effect of varying levels of granularity for the event interval construction.

References

[1] Allen, J., and Ferguson, G. Actions and events in interval temporal logic. *Journal of Logic and Computation 4*, 5 (1994), 531 – 579.

[2] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Comput. 6*, 2 (2010), 161 – 180.

[3] Chen, G., and Kotz, D. A survey of context-aware mobile computing research. *Dartmouth College Technical Report 2000-381* (2000).

[4] Goffman, E. *The Presentation of Self in Everyday Life*. Doubleday, New York, 1956.

[5] Kotsifakos, A., Papapetrou, P., and Athitsos, V. IBSM: Interval-based sequence matching. In *SIAM Int. Conf. Data Mining*, SIAM (2013), 596–604.

[6] Roussos, G., Peterson, D., and Patel, U. Mobile identity management: An enacted view. *Int. Jour. E-Commerce 8*, 1 (2003), 81–100.

[7] Seo, J., and Shneiderman, B. Interactively exploring hierarchical clustering results. *Computer 35*, 7 (2002), 80–86.

[8] Taylor, C. *Sources of the Self: The Making of the Modern Identity*. Harvard University Press, Cambridge, MA, 1989.

[9] Wagner, D., Rice, A., and Beresford, A. Device analyzer: Understanding smartphone usage. In *10th Int Conf Mobile and Ubiquitous Systems* (Tokyo, Japan, December 2013).