EFFICIENT SUPPORT OF CONTEXT-DRIVEN INTERFACES FOR MOBILE DEVICES

Vassilios Stefanis[1,2], Athanasios Plessas[1,2], Andreas Komninos[3] and John Garofalakis[1,2]
[1]Department of Computer Engineering and Informatics, University of Patras, Patras, Greece
[2]Computer Technology Institute "Diophantus", Rion, Patras, Greece
[3]Mobile and Ubiquitous Computing Group, Glasgow Caledonian University, Cowcaddens Road, Glasgow, UK
{ stefanis, plessas, garofala}@ceid.upatras.gr, andreas.komninos@gcu.ac.uk

Abstract – **We examine the performance of the FastMap algorithm when applied on mobile devices for context dimensionality reduction. Our results show that this algorithm is extremely fast and is appropriate for mobile context datasets that need to be processed in real time. This fact supports our vision for context-driven interfaces generated by applying the method of mobile context dimensionality reduction that will support the richness of data stored in personal devices**.

Context awareness; Dimensionality reduction; FastMap performance

I. INTRODUCTION

Technological progress in mobile devices, in terms of storage space and ways of generating digital data, allow mobile device users the rapid creation and storing of large volumes of personal data. Much of this data is organized in structured repositories (e.g. contact list, photo gallery, message inbox), affording the user a means of learnable, procedurised retrieval. Even though multiple types of personal information are meaningful and helpful to the user in daily situations, in most devices the structured repositories remain mostly "walled gardens", requiring the user to sequentially visit each of them in order to assemble the information pieces she needs in one coherent mentally held collection, relevant to their current activity. Naturally, the cognitive load on the user increases with the number of information items that have to be retrieved from the repositories. But still, even considering each "walled garden" individually, it quickly becomes apparent that current retrieval methods are largely inefficient, in view of the ever-expanding storage space and richness of personal information stored in mobile devices. For example consider work on visualizing and retrieval from large mobile photo galleries [1] or contact lists. What can a user do with all this data? Surely users want the data, but is it really useful in its raw form as a singular item and can it be used to help them achieve their goals?

Mobile devices such as smartphones, are still primarily information access devices and communication devices. Much of the activity on mobile devices, especially mobile phones, is in support of Human-to-Human interaction. Consequently, many of the actions that someone might perform on their mobile device involve the act of looking up a contact, that perhaps carries at that time some importance to the user, in order to initiate some form of communication (call, sms, email, comment on facebook etc.). Not forgetting that communication is by definition the exchange of information and information items are an important part of it. Thus, mobile devices should support not just the retrieval of contacts, but also of information that is somehow relevant to these contacts. It is critical here to underline that our work extends beyond the notion of contact importance – in fact, we believe that importance as a multidimensional vector can provide a complex query "key" on the large, rich dataset in a user's mobile device for any type of personal information item (e.g. a photograph, or a set of SMS messages or stored Word documents) so as to answer the question of who to communicate with and what to communicate. As presented in [2] we have proposed an extensible, flexible approach for the definition of importance in a k-dimensional context space, which can be applied to any mobile information retrieval problem. However, we have chosen to start with the concept of contacts as a problem domain, largely due to its importance in everyday interaction with mobile devices. Our idea was to apply a fast dimensionality reduction algorithm to a dataset of personal information items (contacts in our case) in order to visualize these items in 2 or 3 dimensions, produce meaningful clusters of related information and match items' importance with current mobile social context. In this paper we are presenting a functionality and performance evaluation of the FastMap algorithm for dimensionality reduction when executed in mobile devices.

II. Background

Our motivation stems from the observation of user behaviour in accessing non-contextualised information from single repositories, namely the mobile contact list. In our recent work [3], we demonstrated some interesting user behaviours, which we believe, highlight the issue of non-contextualisation. Firstly, we observed that our 18 young users (<25 yrs. old) from varied backgrounds tend to keep fairly large repositories of contacts (m=92.47 sd=56.93), which are bound to get bigger as they grow older. In these collections, a large percentage of contacts (av=39%, sd=17%) has either not been used in over 6 months or not at all. Viewing the collections as a whole (n=754 contacts), over 47% of all contacts had not been used in over 6 months or at any other point, while just 15% were identified as frequently used contacts, the remaining 38% having been used between 1 and 6 months previous to our study period. We also found a highly significant correlation between the size of the contact list and the number of unused contacts (r=0.81). In a carefully designed experiment, we presented our users with a contextualised UI which placed "important" contacts in an alphabetic list, followed by an alphabetic list of all other contacts and asked them to perform several retrieval tasks on this augmented UI and on a classic (Nokia) UI. Importance in this experiment was set as a single-value vector of frequency of use, with 6 months or more being a threshold for considering a contact as not important. We found that users provided very positive subjective feedback to their experience using this approach (86% found it easier to use than the traditional UI and 64% would like to have this feature on their next device). This was backed up by significant performance metrics. On average users required less button presses to successfully "find" a contact (mean reduction=1.96, sd=3.56) and significantly less time with the augmented UI (m=4,424ms, sd=1,872ms) than the traditional UI (m=5,204ms, sd=2,829ms).

While these findings assumed a naïve classification of importance (just frequency of use), they demonstrate quite effectively how contextualisation on just a single dimension can have a very positive effect on the retrievability of personal mobile information. In the past, other researchers have demonstrated either issues with the usability of contact and call lists [4] or improvement in usability through the introduction of user activity, proximity & state context [5], social context [6] and temporal context [7]. Further work by Ankolekar et al. [8] discusses how a combination of contextual cues might offer usability advantages but leaves the categorisation of contacts to users and does not present any tangible research into one of the most oft-used applications of mobile devices. The volume of research in contact list use remains low and the technology behind contact list UIs in today's devices is mainly based on alphabetic lists. Finally, Rana et al. [9] discuss how an algorithm for inferring social priority based on a variety of social communication context sources (SMS, Facebook, Twitter, Email, Telephony, Blogs) can be used alongside other physical context (e.g. location, time) to recommend content from these social communication sources. Currently, only the Android OS provides a feature of presenting a contact list by use frequency, though again this is a simplistic view of importance, considering just one type of context. Some Android applications take advantage of this feature by presenting alternative contact list views (such as quickdial lists or tiles of icons or photos representing *top-n* contacts in terms of frequency of use), but frequency of use is merely scratching the surface of context richness that can be automatically built by a device and which can extend to correlating people, activities and information items.

III. The need for a K-dimensional approach to context mapping

Mobile devices collect a significant amount of data and information about the user's context. Such information includes location (absolute or relative), the current time, whether the user of the device is on move and their speed, the orientation of the device, the user's current task (e.g. on the phone, messaging), whether the vibration or the silent mode are enabled etc. The user considers her mobile device a "trusted device". She usually has the device close to her, sometimes operating 24 hours per day. Devices contain a lot of personal information related to the user's social environment [10]. These are often generated automatically by the device (e.g. a smartphone's phone list saves the calls that have been made, the time of the day for each call and the duration of each call). Moreover, mobile devices store user-generated content (e.g. SMS/MMS and audio files, browser's history, calendar with user's events etc). Therefore, a mobile device could also be aware of the social environment of the user (social context). In addition, mobile applications can take advantage of social data from online social networks in order to enrich a contact with more information [11]. The combination of social and mobile context results in a dynamically defined social context, termed the mobile social context [12]. Therefore, a truly context aware mobile information access application has to consider social context as part of its context representation.

It is therefore quite apparent that true contextualisation is much more complex than in our previous experiment's assumption, and that it requires k-dimensional space in order to be defined. As shown in equations (1), (2) and (3) we use a vector model to represent the context of an information item i in a k-dimensional space as a k-tuple, where d_n is the context atom value for dimension n. Its importance can be considered as the sum of the weighted distances between the vector atoms describing current context properties and an item's context atoms ($\Delta \vec{C}(i)$).

$$\vec{C}(i) = (d_1, d_2, ..., d_n) \quad (1)$$

$$I_i = \sum_{i=1}^{n} \left(w_{d_n} \times \left(1 - \Delta \vec{C}(i) \right) \right) \quad (2)$$

$$\Delta \vec{C}(i) = \left| \vec{C}(n) - \vec{C}(i) \right| \quad (3)$$

In literature, context has been represented in the form of vectors, e.g. [13] while other researchers have adopted an ontological approach.Vector based approaches carry the disadvantage of computational complexity in similarity searches (each dimension needs to be compared separately) and that weighed vectors require an empirical (hence error-prone or narrowly applicable) estimation of the weights. On the other hand, ontologies tend to be inflexible and too strict to be applicable to wider ranges of problems, requiring careful, manual approaches to their construction.

IV. MAPPING CONTEXT IN K-DIMENSIONAL SPACE

In our view, the vector approach can offer a more flexible solution to context acquisition and representation, hence our work relates to overcoming its computational complexity and vector weighting issues. In [14] we proposed four criteria for the determination of a m-PIM (mobile Personal Information Management) item importance (namely contacts). From these criteria, we can derive the following context dimensions for the contact list problem domain, on which a contact can be mapped:

- Frequency (e.g. Frequency of use in last n months)
- Recency (e.g. days since last use)
- Location (e.g. Geographic areas from which at least n% of uses are made)
- Time of Day (e.g. Time segment of day in which at least n% of uses are made)
- Task (e.g. Boolean measure of existence of a scheduled task involving a contact within a certain window of time or temporal distance between now and such a task)
- Personal preference (e.g. explicit user rating of importance for a given contact)

In order to estimate a "match" signifying importance between these types of contextual information and the user's current context, a typical approach would be to measure the distance between current context and contextual data (e.g. time of day now vs. usual time of day of contact use) and combine this with static context (e.g. explicit importance rating). The derived metrics would need to be weighted and the sum of these weighted metrics could then be used to infer "importance" for a single contact under any context (equations (1), (2) and (3)). This approach though is not without challenges: Firstly, one must determine appropriate weights for each context type. Subsequently, it is easy to realize that this would be a futile attempt, as the weights of each context type are naturally dynamic and can vary under different use contexts (see scenario in fig. 1).

"*Take the example of John, a contact that the user calls every day and Jane another contact that is only called once a year. John can be considered generally important. However if the user is running late for a meeting with Jane that will take place in 10 minutes, then Jane becomes undeniably more important than anyone else, and her importance should rise and decay naturally as time flows around the scheduled event.*"

Figure 1. The Importance Scenario.

The example scenario introduces the problem of context-derived specific importance (vs. general importance) and shows that a decent approximation to the calculation of this term is very difficult, due to the infinite variability of context itself and the complexity of its dimensions. The complexity of our simple scenario rises as we consider the correlation of "contacts" with activities and additionally try to match information items on the device, or retrievable by it, with these activities and contacts. A possible solution however could come from the field of Databases and IR, where dimensionality reduction is a technique often used to automatically extract important features from complex data and reduce the complexity of searches in multidimensional spaces.

V. DIMENSIONALITY REDUCTION (DR) IN M-PIM ACCESS

DR, as the name suggests, is an algorithmic technique for reducing the dimensionality of data, applied in several computer science fields such as databases, information retrieval, data mining etc. Real-world data usually has a high dimensionality, a fact that affects data processing performance (the so called "curse of dimensionality" that suggests exponential dependence of an algorithm on the dimension of the input). The idea is to transform data from a high-dimensional space to a low-dimensional space, preserving some critical relationships among elements of the data set. In mathematical terms, given a p-dimensional object $x=(x_1,\ldots,x_p)^T$, find a lower dimensional representation of it, $s=(s_1,\ldots,s_k)^T$ with $k\leq p$, that captures the content in the original data, according to some criterion.

Our research idea is to perform DR to context augmented personal information items, such as entries in a contact list, an idea that has not yet been proposed and applied in scientific literature as far as we know. Since, as already presented, context augmented personal information items can be represented as multidimensional vectors, we find it highly appealing to try to extract a small number of features that could accurately represent the original items and their relationships, so as to enable quick and accurate similarity searches for related personal information items. Furthermore, after reducing the dimensions of the items, it might be desirable to map them to a 1-d, 2-d or 3-d space, as often done in high-dimensional data projections, since visualization tends to reveal existing groups of objects.

There is a wide range of algorithms with diverse characteristics that achieve dimensionality reduction, following different approaches. An interesting method that could be appropriate for the case of mobile phones due to its simplicity and computational efficiency is the FastMap algorithm [15]. FastMap is a fast algorithm that maps high-dimensional objects into lower-dimensional spaces, while preserving well distances between objects and the structure of the data set, as a result preserving also dis-similarities between objects. Experiments presented in [15] show that the algorithm performs well for visualization and clustering. The algorithm functions as shown in fig. 2 and its complexity is $O(Nk^2)$, where k are the dimensions of the target space. A further advantage of this approach is that context vector "atoms" need to be defined by application developers just once – the recursive and atom-agnostic nature of the algorithm allow it to work for any context description vector. In the following paragraph we are presenting our findings concerning the performance of the Fastmap Algorithm in mobile devices.

FastMap algorithm:

```
1.Find two objects that are far
   away.
2.Project all points on the line
   the two objects define, to get
   the first coordinate.
3.Project all objects on a
   hyperplane perpendicular to
   the line the two objects
   define.
4.Repeat k-1 times
```

Figure 2. The FastMap Algorithm.

VI. PERFORMANCE OF FASTMAP IN MOBILE DEVICES

Our experiments were performed in order to show that the FastMap algorithm has an appropriate execution time for real time calculations on mobile devices. We implemented the algorithm using the Android SDK API Level 8 (Android 2.2.x) based on the C implementation of [15]. The experiments were executed with synthetic datasets as we didn't have in our disposal real data from mobile users yet. For this reason we created seven different datasets using the Random.org number generator (as shown in Table 1).

TABLE 1. Data sets used in the experiments

Set	Objects	Dimensions	Total atoms
1	500	5	2,500
2	1,000	5	5,000
3	2,000	5	10,000
4	500	8	4,000
5	1,000	8	8,000
6	2,000	8	16,000
7	2,000	10	20,000

The random objects represent Personal Information items (such as contacts). Each object is represented by a vector of mobile social context information (dimensions as described by Faloutsos and Lin). For our experiments we used a variety of mobile phones running the Android OS, from low end to mid range and high end devices. Table 2 presents the features of these smartphones.

For each device (D1 – D6) the FastMap algorithm was executed for all sets (Set 1 – Set 7) for k=2 and k=3, where k the dimensionality of the output target space. During each run, the algorithm was successively executed 10 times and the application reported the average execution times for all data sets. The reason for this was so as to obtain a more realistic execution time average, given the consideration that Android smartphones tend to run many background services that could adversely influence execution time measurements if only single runs were made. We also took care to measure execution time excluding time taken to render graphics and text on the screen, as well as I/O times required to read the datasets into memory (as these are stored in the device filesystem). Tables 3 and 4 contain the average execution times in milliseconds for each experiment.

TABLE 2. Android Smartphones used in our experiments

#	Model	OS Version	CPU	RAM
D1	HTC Magic	2.3.7	Qualcomm MSM7200A ARM 11 - 528 MHz	288 MB
D2	Huawei U8150 IDEOS	2.2	Qualcomm MSM7225 - 528 MHz	256 MB
D3	Samsung Galaxy 3	2.2	Samsung S5P6422 667 MHz	256 MB
D4	Samsung Nexus S	2.3.6	Cortex-A8 - 1 GHz	512 MB
D5	Sony Ericsson Xperia Arc S	2.3.4	Qualcomm MSM8255T Scorpion - 1.4 GHz	512 MB
D6	Sony Ericsson Xperia mini pro	2.3.4	Qualcomm MSM8255 Scorpion – 1 GHz	512 MB

TABLE 3. Average execution times in msec for k=2

	Set1	Set2	Set3	Set4	Set5	Set6	Set7
D1	54.7	116.1	171.9	72.1	113.5	218.7	265.7
D2	79.2	161.3	325.9	45.1	205	348	405.2
D3	44.6	83.3	141.9	45.1	95.6	212	238.4
D4	4.9	15.4	25.1	6.5	18	31.5	37.2
D5	5.1	8	20	7.7	11.2	19.6	23
D6	10.5	13.4	24.8	4.1	13	26.6	29

The results from the above experiments are very encouraging concerning the performance of the FastMap algorithm when executed on mobile devices. As one can see the worst execution time (k=3, Set 7, D2) is below 0.4 seconds for a low end device and for a dataset which is much larger than an average expected in a mobile device. We note also from the visualisation of the execution time results, the following interesting facts. Firstly, all devices seem to exhibit similar trend lines of behaviour for the same dataset, with results that are consistent with the variations in the processor speed and memory capacity. We also note that D1 and D2 have the same processing capability while D1 has 32Mb of RAM more and runs a later version of the OS. At the same time, D2 and D3 have the same OS and RAM but D3 has a 139MHz advantage. From the execution time curves we can see that memory probably has a more important role than CPU speed or OS version. This is corroborated by the performance of the higher-end devices, where memory capacity is identical and the performance is almost identical (in absolute figures) despite the 40% greater speed of D5 over D4 an D6.

TABLE 4. Average execution times in msec for k=3

	Set1	Set2	Set3	Set4	Set5	Set6	Set7
D1	54.6	119.5	254.3	73.9	171	328	346.4
D2	68.2	110	235.9	81.6	153.2	289.4	376.4
D3	65.8	125.2	241.9	82.4	146.1	337.1	365.4
D4	7.4	21	40	16	25	49.3	53.4
D5	6.8	16.8	23.4	8.5	9.2	24.4	26.5
D6	5.9	15.7	31.1	12.1	18.9	36.1	40

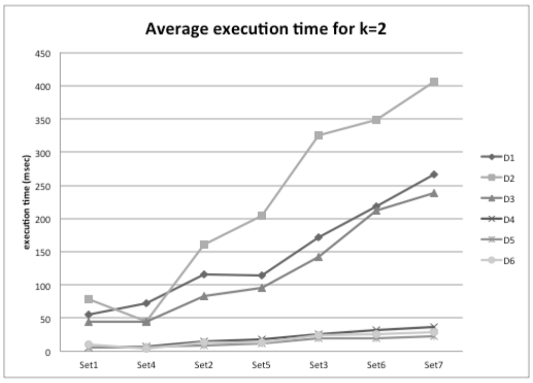

Figure 3. Average execution times for k=2

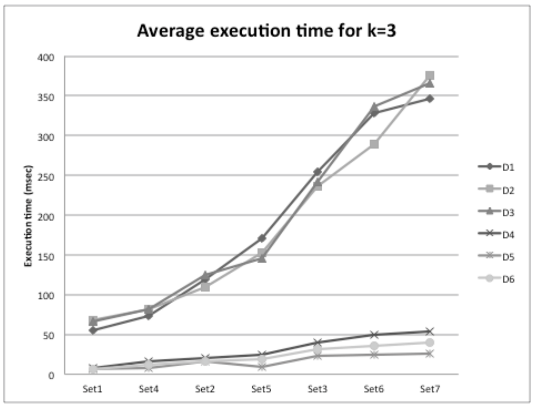

Figure 4. Average execution times for k=3

VII. Conclusions

We believe that our technique as described above can be very useful in order to build rich singular, two or three-dimensional information retrieval interfaces that will support data from multiple information repositories and present them in context, as demonstrated in the mock-ups presented in this section (fig. 5 & 6).

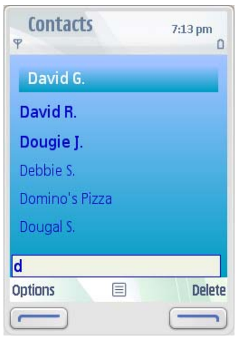

Figure 5. Hybrid one-dimensional mapping of importance (important items are highlighted in bold but ordered alphabetically to maintain user familiarity with existing UIs).

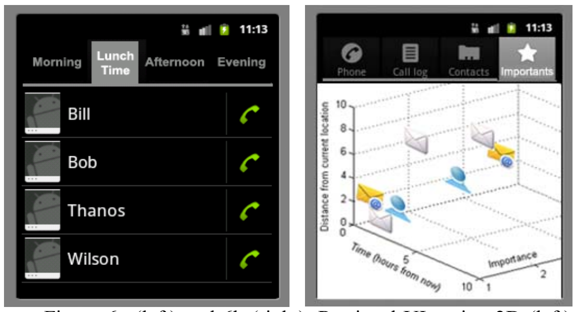

Figure 6a (left) and 6b (right). Retrieval UIs using 2D (left) and 3D (right) projections of item (contacts, e-mails, photos etc.) importance combined with retained, unprojected dimensions (time, distance from current location).

In fig. 5 a one-dimensional hybrid interface is presented, where contacts are sorted alphabetically, but for each letter the important contacts are on top of the list and the remaining follow in alphabetical order. In fig. 6 we show some concept renderings of 2-d and 3-d retrieval interfaces. In the case of 2 dimensions, the dimension of time can be preserved and as a result the user is able to retrieve the most important contacts during each time period. Finally, in the case of 3 dimensions an example of how this technique extends beyond the domain of contact lists is presented. The respective figure (6b) shows how several personal information items (contacts, e-mails, SMSs etc.) could be projected in a 3-dimensional space (with possible dimensions presented on the axes of time, importance and distance from current location).

From our experiments FastMap seems to have a performance complexity that is suitable for applying dimensionality reduction on mobile devices datasets. Our future work includes the evaluation of the algorithm with real mobile data in order to assess the quality of visualization and clustering. Moreover, we intend to experiment and compare FastMap quality and performance against non-linear dimensionality reduction algorithms. Non linear dimensionality algorithms tend to preserve in a better way the features of the objects in the input data set but their complexity is in most cases quadratic.

REFERENCES

[1] Hsu, S.-H., Cubauld, P., Jumpertz, S. 2009. Phorigami: A Photo Browser Based on Meta-categorization and Origami Visualization. *Human-Computer Interaction: Novel Interaction Methods and Techniques Lecture Notes in Computer Science*, Volume 5611/2009, 801-810, Springer.

[2] Komninos, A., Plessas, A., Stefanis, V., Garofalakis, J. 2011. Context Dimensionality Reduction for Mobile Personal Information Access. In *Proc. of KDIR 2011,* Paris.

[3] Bergman O., Komninos A., Liarokapis D., Clarke J., 2011. "You never call: Demoting unused contacts on mobile phones using DMTR", *Personal & Ubiquitous Computing*, Online First, DOI 10.1007/s00779-011-0411-3.

[4] Klockar, T., Carr, D., Hedman, A., Johansson, T., Bengtsson, F., 2003 Usability of Mobile Phones. In Proc. Of *HFT '03*, Berlin, Germany.

[5] Oulasvirta, A., Raento, M., Tiitta, S., 2005. ContextContacts: re-designing SmartPhone's contact book to support mobile awareness and collaboration. In *Proc. of MobileHCI'05*, Salzburg, Austria.

[6] Gaur, S., 2008. Mobile Phone Contact. *In: Ylä-Jääski A, Takkinen L, Technical Reports in Computer Science and Engineering*.

[7] Jung, Y., Anttila, A., Blom, J., 2008. Designing for the evolution of mobile contacts application. In *Proc. of MobileHCI'08*, Amsterdam, Netherlands.

[8] Ankolekar, A., Szabo, G., Luon, Y., Huberman, B.A., Wilkinson, D., Wu, F., 2009. Friendlee: a mobile application for your social life. In *Proc. of MobileHCI '09*, Bonn, Germany.

[9] Rana, J., Kristiansson, J., Synnes, K. 2010 Enriching and Simplifying Communication by Social Prioritisation, In *Proc. of ASONAM 2010*, Odense, Denmark.

[10] Toninelli, A., Khushraj, D., Lassila, O., Montanari, R., 2008. Towards Socially Aware Mobile Phones, In *Proceedings of SDoW 2008,* Karlsruhe, Germany.

[11] Bentley, F., Kames, J., Ahmed, R., Zivin, R., Schwendimann, L., 2010. Contacts 3.0: bringing together research and design teams to reinvent the phonebook. *In Proc. of CHI '10*, USA

[12] Gilbert, P., Cuervo, E., Cox, L., 2009. Experimenting in Mobile Social Contexts Using JellyNets. In *Proc. of HotMobile 2009*, Santa Cruz, USA.

[13] Du, W., Wang, L., 2008. Context-aware application programming for mobile devices. In *Proc. of C3S2E Conference*, Canada.

[14] Komninos, A., Liarokapis, D., 2009 The Use of Mobile Contact List applications and a Context-Oriented Framework to Support their Design. In *MobileHCI'09*, Bonn, Germany.

[15] Faloutsos, C., Lin, K. I., 1995. Fastmap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. In Proc. of ACM SIGMOD '95, USA.