The Case for Context-collaborative Filtering in Pervasive Service Environments

Matthias Böhmer and Gernot Bauer
Münster University of Applied Sciences
Lab for Software Engineering, Germany
{matthias.boehmer,gernot.bauer}@fh-muenster.de

ABSTRACT

Our environment soon will be pervaded with a high density of digital services for usage on mobile phones. Context-awareness plays a crucial role for the success and acceptance of mobile services. In this paper we describe and discuss a new approach for filtering pervasive services. It is based on context reasoning, collaborative filtering, and a mixed-initiative method. The proposed context-collaborative filtering recommends services to a user which have been used by other people in the same context.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering, relevance feedback

General Terms

Human Factors.

1. INTRODUCTION

Emerging technologies in mobile and ubiquitous computing support the growth of a pervasive service environment of high density. Many approaches propose methods and realize systems for embedding digital services into our everyday environment [4, 5, 1]. There are boundless use cases: information delivery (e.g. news, weather), mobile guiding (e.g. for cities or museums), communication (e.g. presence, messaging), advertising (e.g. coupons), domotics (e.g. health care), or even user-generated content. Such pervasive services will supplement todays device inherent applications. Soon we will be faced with the problem of choice overload for mobile services in our daily life. Recommender systems deal with this problem. They suggest only elected resources to their users aiming to satisfy the user's current demand.

Hitherto, context-awareness has a limited view on single users and their actual environment: the user's environment is sensed and matched against available services. Methods for reasoning a high-level context (e.g. an activity like cycling, driving a car or being at work) from low-level context features (e.g. location, acceleration, temperature, audio, or movement) are available. However, correlations between contexts of different users have not been taken into account yet. People in similar situations act in similar ways and have similar needs. The contextual information is a stimulus for the action. Therefore, the actions of a user can be related to his current context.

The relevance of a specific service for a certain context can not be modeled a priori, mainly due to the unpredictable adhoc behavior of humans. Nevertheless, if a large number of people uses specific services particularly in certain contexts, the relevance can be inferred by observation. Thereby, the question for the relevance of a specific service in a certain context can be answered by the specialists - i.e. the user himself.

The contribution of this paper is a collaborative approach to improve the context-aware filtering of pervasive services. To that aim the service usage is correlated with context. We study how a user's behavior and system interaction can be related to his current context. And how the knowledge about this relation can be used for improving context-aware recommender systems for service selection. Section 2 motivates our approach and discusses different scenarios. Section 3 describes a design for context-related collaborative filtering. Section 4 closes with concluding remarks and future work.

2. SCENARIOS FOR MOBILE SERVICES

Figure 1 shows four different cases (C1-C4) how mobile devices can provide services. In one direction, the matrix differentiates between device-inherent and pervasive services. Device-inherent services are fixed on the device. Once installed, the user takes them along everywhere. Pervasive services, in contrast, are embedded in the ether and pushed to the device when they fit to the user's context. In the other direction, the matrix distinguishes whether the services are filtered by a user-centric or a collaborative approach.

The first case (C1) covers conventional services for mobile devices. The manufacturer or the user installs applications manually. Those are fixed until they are uninstalled. The selection of applications is done by the user according to his personal requirements. Usually, he adds useful and interesting applications and removes useless ones. This works fine for core services (e.g. "I need SMS, but no browser"). However, the problem of choice overload arises as soon as the number of available applications increases.

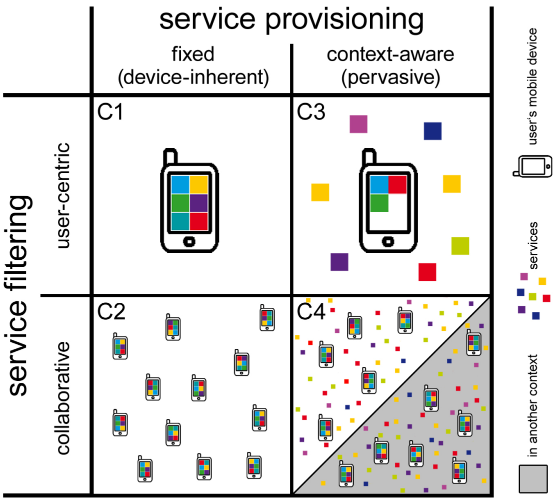

Figure 1: Provisioning and filtering of services.

Therefore in the second case (C2), the user is supported by a collaborative filter for choosing appropriate applications. Some services are suggested to simplify the user's selection. A similar function already appears on upcoming application stores, where the most bought or downloaded, and best rated applications are suggested. Indeed, this supports the user to distinguish useful and useless applications. An attribute which can not be estimated in advance.

The evolution of C1 and C2 already took place. As novelty, case three (C3) introduces the pervasiveness and context-awareness of services. Those are – from a user's point of view – not installed on the device. They appear instantly on the device, when they are relevant in the user's context. This may result in icons fading in and out of the device's main menu. In a simplified case, where context is reduced to location, the icon appearance might depend on where the user is. For example, a user might be identified to be in the context *taking a sight-seeing tour* (e.g. by recognition of a move from one sight to another with certain resting times). According to his context, special services are provided to the user, e.g. information about the sights, historical facts and figures on the city in general, city guides, and digital maps. For this setup, there has to be a metric or a rule for deciding which services fit to the user-centric context a priori.

The last case (C4) uses a collaborative filter for extending the previous introduced context-awareness. A person in the context *being at a party* (e.g. inferred from the mean spatial distance of buddies, number of people nearby, time, and day of week) might favor services taking and showing pictures and videos, remote music control, and instant messaging. In another context, the same person may have a completely different demand for digital services. During *traveling-by-train* news services or train schedules or *call-a-cab* will be more relevant. In this case, a collaborative filter will suggest those services to the user, which have been used by other people in the same context. For example, if a lot of people at a party use the remote control services for making a request for their favorite song, this service might also be useful for other people at the the party. And when many people use *call-a-cab* when they are *traveling-by-train*, this is a relevant service for people traveling by train.

Such a semantic relatedness between services and contexts

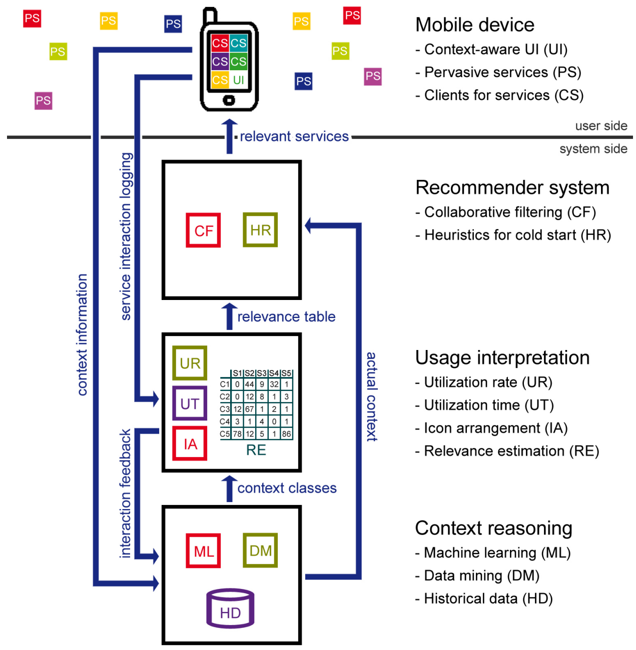

Figure 2: Design of a context-collaborative filter.

cannot be modeled a priori. Instead, it can be inferred from observation of large groups of users in the same context over a significant period of time. The next section describes a system design for the realization of this approach.

3. SYSTEM DESIGN

Our approach aims at supporting users in pervasive service environments by reducing the set of available services to the set of contextual relevant ones. The user is able to consume these services on his adaptive mobile device, as figure 2 shows. Therefore, the output of the proposed context-collaborative filtering system is the set of relevant services. The input of the system is information on context, and logged data about the user's service interaction. Current pervasive computing environments support the implicit collection of such knowledge.

The next sections describes a draft of three core components from bottom up: the context reasoning, the usage interpreter, and the recommender system.

3.1 Context Reasoning

Context is any information that characterizes the situation of a person [3]. On todays mobile devices, there are usually a lot of sensors giving such information, e.g. time, speed, light, temperature, acceleration, geographic position, and azimut. Such low level context features can be measured directly. Zimmermann et al. [7] suggest relations as additional context category. Raw data can be enriched with such knowledge for deriving more context information, e.g. mean distance of buddies, density nearby people, traveled distance in last hour, or current distance from home. Further [7] also describes activity as context feature, which – for example – can be realized by users defining tags for their current activity (e.g. "train, traveling, journey").

Especially for physical context features, but also for more vague concepts like collaborative tagging, metrics are available. Methods from machine learning and data mining can be used to reason about contexts. For example, the contexts *being-at-work*, *being-at-home* or *being-on-the-move* can be distinguished by low-level data [6]. Based on a large pool of historical data, patterns or clusters can be found and disjunct classes can be built. At this point, no semantic interpretation of the context classes is possible.

Based on an actual set of low-level context features it is possible to infer the likelihood of a user currently being in a certain context class. Ideally, the result would be a non-ambiguous boolean, but we assume that it will be fuzzy in most cases.

3.2 Interpreting Service Usage

With the knowledge of the user's actual context, we can observe his behavior accordingly. By interpreting the actions of the user, we can infer knowledge about the relevance of a service in a certain context. In a *waiting-for-the-train* context the application for the train schedule will be used presumably much more often than a service for bank transactions. As a rule of the thumb we can assume, that in certain contexts irrelevant services are used not at all, and relevant ones are used much more often. In contrast to incomputable properties like the relevance of a service and the satisfaction of needs, this property can simply be quantified by counting.

Another metric can be derived from utilization time. For example, if the train schedule is used during the whole ride, it is likely more relevant than the flight schedule. Although both are only used one time, the usage of the latter is shorter. Because the user will draw his attention only to situational useful applications, the service utilization time correlates with its relevance. The measured time should be normalized by the complexity of the service's user interface. For instance, it takes more time to extract a piece of information from a long text than from an image, but the entropy is the same. This approach can also prevent spam, because annoying applications will be closed instantaneously.

We are also thinking of drawing back the arrangement of menu icons to the relevance of the corresponding services. We observed people putting icons of more relevant applications to a more prominent position, e.g. to the first page of the menu on the iPhone.

Besides all this positive feedback making a service more relevant, there is also a benefit from negative feedback, which gives a hint for irrelevant services in a context. For example, this can be concluded from delete actions. Also a temporal decay of the relevance of services seems promising. Services become irrelevant, when they are not used anymore.

The interpretation of service usage results in a table, that contains the relevance measure of a service for each context class. All figures are aggregated for a service in a context, e.g. by summing up the usage counters.

3.3 Recommender System

The recommender system will suggest services to the user which have been useful for other people in the same context. This behavior is similar to how people behave in real life.

Having figured out the user's context, a simple lookup in the table gives the weighted relevance of the services. In case of the fuzzy context reasoning, the likelihood of the contexts needs to be considered in the relevance weights.

According to their weights, the services are pushed to the user's device from the most relevant to the least. Due to the physical limitation of screen size there has to be a reduction to the most relevant services. Also the user should not be overstrained with too much functionality – a cognitive constraint.

A cold start problem appears, when a user is in a context where no service usage was recorded before. In this case, simple heuristics and other filters can be used, e.g. location- or content-based. Users should also be to able search and browse for services not recommended yet. This problem further occurs when a new service is available in the system. When a similarity measurement for the services is available, the relevance weights could be bootstrapped from similar services.

3.4 Usage-based Context Reasoning

Figure 2 also shows a coupling of the user-given feedback to the context reasoning. The inclusion of the service usage data can extend the context inference. For example, the context *traveling-by-train* might be independent from the weather condition (people go by train on sunshine and rain). But if people use the *rent-a-bike* service in this context only when the sun is shining, and *call-a-cab* only when it is raining, it will be beneficial for the users, when the system is able to subdivide the context, record the service usage statistics accordingly and give corresponding recommendations. By reasoning context not only on raw sensor data, but also on the peoples' service usage, the recommending becomes more valuable for the user.

4. CONCLUSION AND FUTURE WORK

In this paper we presented an approach for collaborative filtering based on the correlation of service usage and high-level contexts. We extended a classical collaborative filtering approach. The patterns of use and implicitly given ratings of services are related to context instead of to users.

Future work will concentrate on the implementation of the proposed system and evaluation of different methods and algorithms. The context-collaborative filtering will be a centerpiece of a platform we are currently implementing. Here, users are already able to become authors of mobile services [1, 2]. The proposed system aims on enabling a context-aware consumption of those.

5. REFERENCES

[1] M. Böhmer, G. Bauer, and W. Wicht. Lbs 2.0 - enabling user-driven provision and context-aware utilisation of location-based services. In *Proc. UBICOMM '08*, pages 374–375, 2008.

[2] M. Böhmer, G. Bauer, and W. Wicht. Hiding the complexity of lbs. 2009.

[3] A. K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, February 2001.

[4] R. Hull, B. Clayton, and T. Melamed. Rapid authoring of mediascapes. In *Proc. UbiComp '04*, pages 125–142, 2004.

[5] J. Krösche and S. Boll. The xpoi concept. In *Proc. LOCA '05*, pages 113–119. 2005.

[6] H. Verkasalo. Contextual patterns in mobile service usage. *Personal and Ubiquitous Computing*, 13(5), 2009.

[7] A. Zimmermann, A. Lorenz, and R. Oppermann. An operational definition of context. pages 558–571. 2007.