ARTICLE IN PRESS

Pervasive and Mobile Computing ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Fast track article

UTravel: Smart mobility with a novel user profiling and recommendation approach

Michele Amoretti [1], Laura Belli [*], Francesco Zanichelli

Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Parma, Parco Area delle Scienze 181a, 43124 Parma, Italy

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Recommendation system
User profiling
Context-awareness

ABSTRACT

The exponentially growing availability of online information calls for personalized search and recommendation. Such systems provide recommendations typically based on user profiles built taking into account user actions. Not yet fully explored, is the domain of context-aware recommendation. In this article, we introduce a novel approach, where user profiling and context-based data filtering both concur to recommendation production. Based on the aforementioned approach, UTravel is a smart mobility application that recommends points of interest (POIs) to end users. After describing the UTravel architecture and implementation, we present the results of an experimental evaluation we carried out involving both simulated and real users.

1. Introduction

The overwhelming growth of the quantity of online information, offered by multiple Internet-connected sources, is an evident phenomenon. Although being a great opportunity, facing such a huge and variegate amount of data can become an obstacle for the user, which needs new decision support tools that must be effective and easy to use. Such a problem has been studied, in recent years, mainly focusing on supporting the *personalized search* of products and services in the Web [1–3]. Personalization means the ability of an information system to automatically filter contents and reshape their presentation, based on user characteristics. Thus, a personalization system can be intended as a computer-based application that monitors the user behavior to create and manage his/her profile, with the purpose to select only relevant products/items (*recommendation*), within a very large repository, or to support other applications and services in adapting to the specific needs of each user.

The main components of a personalization system are usually (1) content description, to be intended as the classification of products or items that the system provides to the user, (2) user profiling, and (3) content selection (filtering), trying to best match the user profile. User profiles are usually derived from collected information about user behaviors and interests. The filtering phase is typically associated to an application, such as an e-commerce site (*e.g.*, Amazon), a search engine (*e.g.*, Google), or a social network (*e.g.*, YouTube), where personalized products or multimedia items must be presented to the user. Google, for example, provides personalized and *contextualized* results, taking into account past history, and trying to understand the purpose behind user queries.

The study of context-awareness has gained momentum also for the pervasive diffusion of smartphones, and mobile devices with high computing, storage and bandwidth capabilities, as well as sensors (such as GPS, accelerometer and

* Corresponding author.
E-mail addresses: michele.amoretti@unipr.it (M. Amoretti), laura.belli1@studenti.unipr.it (L. Belli).
1 Fax: +39 0521 905723.

http://dx.doi.org/10.1016/j.pmcj.2016.08.008
1574-1192/

camera). In particular, the presence of on-board sensors allows the implementation of mobile applications which provide useful information and contents to the user, according to her/his location and current actions.

In this article we present the novel Universal Profiling and Recommendation (UPR) approach, based on context-awareness. In particular, we describe our first UPR-based mobile application, namely *UTravel*, which informs and guides users toward points of interest (POIs). According to the context and to the selected categories of interest, UTravel suggests the best POIs in a user-defined range, with respect to his/her current location. The suitability of POIs is inferred from previously shared user evaluations, according to the principles of *collaborative filtering*. The main purpose of the proposed architecture and application is to illustrate how context-awareness can contribute to improving the recommendation process of POIs for mobile users.

The first methodological contribution of UPR is a hybrid strategy for user profiling, which builds an approximated description of the user, by taking into account not only its individual behavior, but also the behavior of other similar users. Our main reference has been the product recommendation system recently proposed by Park and Chang [4]. With respect to the latter approach, our one clusters users by taking into account not only their demographic profiles, but also their preference profiles. To this purpose, the UTravel architecture integrates evaluation and rating mechanisms that are well known in the field of personalization systems [2,5]. In particular, the UTA* algorithm [6] is used to build preference profiles, allowing the creation of a model of the value system which intervenes in the user decision process. In UTravel, such a model is built by elaborating the multi-criteria evaluations that the user assigns to the POIs proposed by the system. The preference profile and the demographic profile are used to find groups of similar users, with the K-Means clustering algorithm [7]. Such groups are taken into account to define the behavior profile of each user. To this purpose, the system tracks her/his interactions with the mobile applications, including clicks, data saves, evaluations, check-ins and preferences which are expressed by selecting or updating the categories of interests. The second methodological contribution of UPR is a filtering strategy that exploits context data collected by the user mobile device, such as its current location, as complementary information to the behavior profile.

The paper is structured as follows. In Section 2 we discuss some relevant related works in the field of context-awareness, user profiling and recommendation for smart mobility. In Section 3 we illustrate the UPR approach by outlining the UTravel architecture. Section 4 describes the UTravel implementation, which is a platform including services, databases and mobile application (here we refer to the Android version only). In Section 5 we show the results of a medium-scale experimental evaluation of UTravel, involving both simulated and real users. Finally, in Section 6 we conclude the paper, with a summary of the presented research work and further ideas for future development.

2. State of the art

The background of the UPR project includes context-awareness, user profiling, information filtering, as well as smart mobility supported by mobile applications.

2.1. Context-awareness

Depending on the adopted perspective – user-side or application-side [8] – contextual information can be classified into three main categories, namely: user context (such as location, company of other people [9,10], emotional status [11]), device context (*e.g.*, connectivity, network bandwidth) and physical context (*e.g.*, date, time, weather, temperature [12,13]). As the interpretation of context depends on the application domain, it is not possible to provide a unique definition of the concept. In modern recommendation systems, user and physical context play a major role in supporting the information filtering process, for which products, routes, or other items are suggested to the user, by taking into account the circumstances in which the user will make her/his choice [14–17]. In the e-commerce scenario, it is important to know the motivation of the user. Is she/he looking for a product for personal use, or for a gift? In the ubiquitous/mobile computing scenario, instead, it would be important to know the user location, as well as the identity of the persons nearby the user, and the environmental conditions. A mobile user will take a different decision depending on, for example, if she/he is with the family or with friends, how is the weather, and so on.

Within the data mining community, context is sometimes defined as those events that characterize the life stages of a customer and can determine a change in his/her preferences, status, and value for a company [18]. Knowledge of this contextual information helps mining patterns to focus only on the relevant data and selecting only relevant results. The contextual information can be obtained (i) explicitly, *i.e.*, by asking direct questions to the user; (ii) implicitly, *i.e.*, from the data or the environment, such as a change in location; or (iii) by inference, using statistical or data mining methods, such as Naïve Bayes classifiers and Bayesian Networks. Explicit and implicit approaches are mostly used in recommendation systems based on *context-driven querying and search* of repositories of resources [19]. Inference, instead, is used for *contextual preference elicitation and estimation*, a more recent trend in context-aware recommender systems literature [20–23], attempting to model and learn user preferences, *e.g.*, by observing the interactions of this and other users with the systems or by obtaining preference feedback from the user on various previously recommended items. The UTravel system presented in this paper is based on a hybrid approach, where preference feedback is used to build user profiles that support context-driven querying and search.

Please cite this article in press as: M. Amoretti, et al., UTravel: Smart mobility with a novel user profiling and recommendation approach, Pervasive and Mobile Computing (2016), http://dx.doi.org/10.1016/j.pmcj.2016.08.008

Table 1
Main dimensions of user profile.

Name	Description	Effect
Personal data	Basic information (*e.g.*, age, nationality)	Interface
Cognitive style	The way a user processes information	Interaction
Device information	Hardware and network environment	Interface and content
Context	User location and other sensed data	Infer user intention
History	Past user interactions with the application	Infer user behavior / interests
Behavior	User behavior pattern	Content and interaction
Interests	Topics the user is interested in	Content
Intention/Goal	User intentions, goals or purposes	Content and interaction
Interaction experience	User feeling with the system	Interface
Domain knowledge	User knowledge of a particular topic	Infer user intention

Adomavicius et al. identified three different paradigms for the integration of context in recommendation systems [24]. With *contextual pre-filtering*, the information related to the context is used to build an initial set of relevant data, which are processed at a later stage to define the rating of each user–item couple. On the other hand, with *contextual post-filtering*, the context is initially ignored, and user–item couples are rated by means of a traditional recommendation system. Context is used to refine the set of recommended items. The most complex paradigm is *contextual modeling*, where context information is directly used to compute the ratings of the user–item couples.

Specifically concerning the context-aware recommendation of points of interest (POIs), Costa et al. [25] proposed a multi-agent system where each user has a Personal Assistant Agent (PAA), which learns from the users past experiences, in order to improve its recommendations. The PAA assigns a probability value to the relevance of the POI, given the user's context and intentions. Therefore, when the feedback of the true relevance of each recommendation is given by the user to his PAA, the PAA updates its memory. The UPR approach we present in this work is different, as the recommendation system is centralized, and aggregates and exploits global knowledge, in order to improve its recommendations. A decentralized version of UPR – which is matter for our future work – would be based on collaborative PAAs.

2.2. User profiling

According to Gao et al. [2], user profiling involves:

- *Behavior Modeling*, which focuses on the patterns associated to user behavior. To this purpose, user–system interaction data are stored. Then, the analysis of past actions can produce an estimation of future user actions.
- *Interest Modeling*, which is based on the definition of a function for computing the interest degree of the user with respect to new products, venues, etc.
- *Intention Modeling*, which tries to find the final objectives of users, or the reason for which they started interacting with the system. Indeed, Behavior and Interest Modeling can be the basis for Intention Modeling—not vice versa.

The importance of knowing both user interests and motivations has been discussed also in other recent works [26,5]. Under this perspective, the personalization problem is considered as a *decision problem*, which can be solved by means of decision theory techniques, such as *Multiple Criteria Decision Analysis (MCDA)*. The latter is a reasonable approach, because the evaluation process that drives individuals in making decisions usually takes into account several variables or points of view. Thus, MCDA for personalization considers the user as a decision maker, and profiling as the process of creating a model of his/her decision processes.

An almost complete classification of user profile dimensions, by Gao et al. [2], is reported in Table 1.

To collect user information, there are two complementary techniques. On the one hand, it is possible to (transparently) track user actions. On the other hand, the system may (explicitly) request feedback to the user. A widely used approach, in this sense, is to ask the user to rate the proposed/selected items [26,5]. Most frequently, the rating process is based on a unique criterion, representing the global level of interest or appreciation of the user with respect to a proposed/selected item.

Lakiotaki et al. proposed a novel architecture for product recommendation [5], where the profiling process includes three phases: data acquisition, user profile creation, and user clustering. In the first phase, the system collects the ratings given by the users to the purchased items, and to a set of items that are periodically proposed by the system. In detail, a set P of items is proposed to each user u. The latter assigns C numerical values, one for each evaluation parameter. Based on such values, the P items are ranked. Thus, for each user, a $M(P \times C + 1)$ matrix is defined—the first column is for the item identifier, while the other C columns are for the evaluation parameters. From such a matrix, the *weight significance vector* $\mathbf{w}$, with C components representing the value system of the user, is obtained. The numerical value associated to each criterion represents the importance given by the user to such a criterion, in her/his decision processes. The authors adopt the UTA* algorithm to obtain the weight significance vector from the matrix M [6]. UTA* is an iterative algorithm which follows the aggregation–disaggregation approach which is widely used in the context of (financial) management and marketing. The objective of the third phase is to divide users into different categories, based on the weight significance vector. Thus, the users who belong to a same class share a common value system. The adopted clustering algorithm is Global K-Means [27], which is characterized by low computational complexity and fast convergence.

2.3. Filtering

According to Gao et al. [2], filtering approaches can be grouped in the following categories:

1. rule-based filtering
2. content-based filtering
3. collaborative or user-based filtering
4. hybrid methods.

Rule-based filtering requires the specification of rules that associate users to item categories, usually performed by experts of the domain the system refers to.

With content-based filtering, the user profile is directly matched with the descriptions of the items. A utility function $f(u, i)$ evaluating the interest degree of user u with respect to item i is estimated from the $f(u, i_k)$ values, where i_k are the items that u has already rated. For example, a book recommendation system memorizes the books that have been of interest for the users. This knowledge represents the user profile. In future interactions, the system will propose to the user only the books that have high matching degree with those of the user profile. This technique relies on algorithms for computing the similarity among items, which can be defined in multiple ways (being the *cosine similarity* one of the most used).

Collaborative filtering is complementary to content-based filtering. Indeed, recommendations are based on the item preferences of the users that are more similar to u. Formally, the utility function $f(u, i)$ summarizes the interest degree for the item i of the users that are similar to u. Using the same example illustrated above, if the recommendation system would use collaborative filtering, it should find the users S_u that are similar to u, and then recommend only the books that have interested the users in S_u. Gao et al. [2] separated memory-based from model-based collaborative filtering. In the former type of filtering, the utility function is estimated by means of heuristics, while, in the model-based approach, statistical and machine learning techniques are used to obtain a model or a pattern from the collected data. Collaborative filtering is the most used technique, in particular in e-commerce systems (such as Amazon), because it adapts to all types of items, and allows to find the potential user interests.

Hybrid filtering merges the principles of content-based and collaborative filtering, to augment their performance. Example hybrid techniques are:

- to separately implement the content-based and collaborative methods, and join the results;
- to incorporate some features of the content-based method into the collaborative one;
- to incorporate some features of the collaborative method into the content-based one;
- to build a general model which incorporates the features of both methods.

Hybrid filtering systems may be much more efficient—at least for some domains. Park and Chang illustrated an example of hybrid filter applied to a product recommendation system [4]. Such a filter takes into account both user interests and group interests. Unfortunately, groups are defined by taking into account only the demographic profiles, and group interests are inferred by analyzing the items purchased by the members of the groups. Instead, filtering in UPR relies on the hybrid profiling approach, based on the two types of user clusters illustrated in Section 3.1.

The architecture proposed by Lakiotaki et al. [5] includes a filter to propose products to the user, based on her/his weight significance vector and class. There is also a feedback mechanism which is used by the system to correct the possible errors in the definition of the user profile. The correction procedure is activated when the user gives a negative rating to an item that has been recommended by the system. In this case, the product is added to the reference set and the UTA* algorithm is re-executed to update the weight significance vector. Then, the latter is evaluated, in order to decide if the user must be moved to a different class, or if the current one is still the most appropriate.

Recently, Colombo-Mendoza et al. [17] proposed RecomMetz, a mobile recommender system for movie showtimes, characterized by the integration of the time and crowd factors into a context-aware model. Like UPR, RecomMetz has a modular service-oriented architecture. However, there is a fundamental difference between the two architectures. RecomMetz applies location-based, time-based and crowd-based filtering to known items (*i.e.*, movie–movie theater–showtime triples), followed by semantic matching with the preferences provided by the user. UPR, instead, adopts a clustering approach to define (and periodically update) groups of similar users and provides recommendations based on such groups and contextual information.

2.4. Mobile applications for smart mobility

The COMPASS (Context Aware Mobile Personal ASSistant) application [28] is a tourist guide, which provides information and services to the users. The recommendation engine of COMPASS uses multiple strategies to predict how interesting each POI is for the user. Two context factors are considered, namely location and time. Location is used as a hard criterion—COMPASS shows only POIs of the area in which the users are supposed to be. On the other hand, time is used as a soft criterion and combined with the predicted interests, to determine the last time the user visited a POI of the same type and by temporarily decreasing the predictions based on this time period. The relevance of a POI is predicted by measuring the time interval since the last user visit to a POI of the same class, based on the following principle: "yesterday, I ate in a Greek restaurant, so today I will probably want to eat somewhere else".

Admire[2] is an iPhone application that proposes itineraries including culturally relevant locations, as well as events that should be suitable to user interests, preferences and lifestyle. In Admire, the user profile is called cultural profile, and is established through the analysis of the displacements of the user, her/his personal and commercial needs, the preferred location categories, and the related appreciation degree.

Another interesting mobile application is Alfred[3], by CleverSense (which has been recently acquired by Google). Alfred is available for both iPhone and Android platforms. It provides recommendations about restaurants and bars, according to the user location and intentions, and the current time. The user profile is continuously refined, by keeping trace of her/his interactions with the system. Indeed, the user can notify her/his preferences, and rate the preferred locations.

TouristEye[4] is a World tourist guide, available for Android and iPhone. TouristEye, which learns from the user and his friends, provides a trip planner, transport maps, personalized recommendations, trip journal, offline maps, and information for more than 10 000 destinations.

With respect to these applications, the UPR-based UTravel application (described in the following sections) targets a larger set of location categories. Moreover, the POI database is populated by the users themselves: when they move to a location for which no POIs are available, UTravel obtains them from Google Places[5] and FourSquare[6]. Finally, for the reasons already illustrated in Section 1 and elaborated in next sections, the recommendation engine in UTravel constitutes an advancement with respect to the state of the art.

3. UPR-based UTravel architecture

The UPR approach enables highly modular, flexible and extendable architectures. Two are the UPR core processes, devoted to user profiling (more precisely, to the generation of behavior profiles) and item filtering, respectively. Such modules are designed to be general-purpose, in order to be re-used in several scenarios, ranging from smart mobility to cooking. Core modules rely on a database, and expose services which can be called by both Web and mobile applications.

In this and following sections, we specifically refer to the UPR-based UTravel application, which recommends Points of Interest (POIs), *i.e.* locations (shops, pubs, museums, etc.) that match user interests. By means of the UTravel mobile application (app), the user can see his/her position on a map, and a set of surrounding locations with cultural or commercial relevance, according to his/her preferences and to the current context. The app is also a collector of user-generated inputs, which are used to build the user profile and to find clusters of similar users. Moreover, user evaluations of POIs are collected by the core of the system, to enrich POI descriptions, which are initially obtained from Google Places and FourSquare.

Fig. 1 shows the functional scheme of the UPR-based UTravel architecture. The user profiling process starts with the UTravel app, allowing the user to send demographic data, ratings, and behavioral information to the core system. Ratings are used to generate the user preference profile. The demographic and preference profiles enable the generation of user groups, by means of a clustering algorithm. Such user groups, together with collected behavioral information (*i.e.*, user actions like selections of interest categories, POI reference savings, etc.) are used to build behavior profiles. During the item filtering process, behavior profiles are used to decide which POIs must be recommended to each user.

3.1. User profiling

The creation of the user profile is the first step of the context-aware, personalized recommendation process. User profiling in UPR exploits a novel hybrid approach, which combines those proposed by Park and Chang [4], Gao et al. [2] and Lakiotaki et al. [5], and builds an approximated description of the user, by taking into account not only his/her individual behavior, but also the behaviors of other similar users.

With reference to UTravel and POIs, in the following we describe the construction of the demographic and preference user profiles, the clustering process, and finally the construction of the user profile.

3.2. POI classification

To build POI descriptors, *i.e.*, items that represent the product in UTravel, we followed the approach proposed by Park and Chang [4], based on multi-level categories. We also considered the use of ontologies for better POI classification, however we did not integrate them into the system due to performance reasons (as reasoning over non-toy ontologies may introduce

2 http://www.myadmire.it.
3 http://www.alfredmobile.com.
4 https://play.google.com/store/apps/details?id=com.touristeye.
5 http://www.google.com/places.
6 https://foursquare.com.

Please cite this article in press as: M. Amoretti, et al., UTravel: Smart mobility with a novel user profiling and recommendation approach, Pervasive and Mobile Computing (2016), http://dx.doi.org/10.1016/j.pmcj.2016.08.008

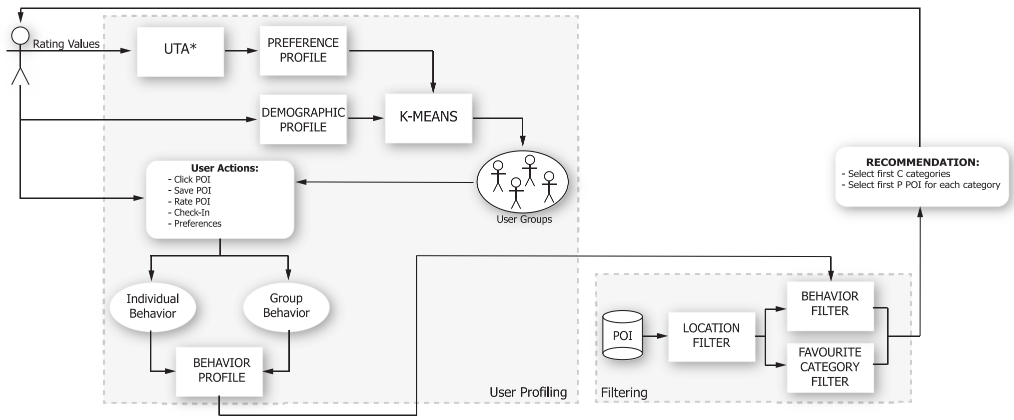

Fig. 1. The UPR-based UTravel architecture.

Table 2
POI categories.

First level category	Second level category
Cultural heritage	Monuments and sculptures, squares, parks and gardens, city walls and Fortifications, bridges, archaeological sites, fountains and obelisks
Cultural buildings	Art galleries and museums, churches and places of worship, archives and Libraries, castles and towers, historic buildings and mansions Theaters and auditoriums
Cultural events	Fairs and markets, exhibits, festival,inaugurations Conferences and conventions, concerts, theatrical performances and Shows
Food services	Restaurants and pizzerias, ethnic restaurants, bar
Entertainment	Pubs, discotheques, concert halls, cinemas, amusement parks, sports, Gambling
Trade	Workshops and food shops, shops, outlets and shopping malls, markets Supermarkets, companies

significant overhead). This choice will be probably confirmed in future research focusing on a decentralized version of the architecture in which all nodes will be running on mobile devices with no centralized database and engines.

We analyzed some existing classifications (in particular, those related to cultural locations[7], which are more difficult to categorize), and then we opted for using two description levels. We included all the categories which may interest tourists and, in general, any user with leisure or commercial interests—other locations are visualized as SOIs, and are not included in the recommendation process. Table 2 illustrates the complete list of first and second level categories adopted by UTravel.

Such a classification has been established with the purpose of finding a tradeoff between user demand, computational costs, and result optimization in the recommendation phase. Indeed, by considering few larger categories, there would be a computational benefit, but the recommendations would be less precise, since it would be more difficult to know the particular user interests. On the other hand, a finer-grain classification of the items would negatively affect the quality of the recommendations, since excessive specialization would hinder or even prevent discovering affinities between user behaviors.

The profile of a POI is a binary set, whose elements are:

$$f(i,j) \quad i = 1, .., F \; j = 1, .., S_i \tag{1}$$

where F is the number of first categorization levels, S_i is the number of second categorization level, and f is 0 or 1, depending on whether the i, jth category does apply or not to the POI. Importantly, categories are not always mutually exclusive, reason why in a POI descriptor there could be several bits with value 1. The POI descriptor is constructed at runtime, to compare it with user profiles. Current UTravel deployment has $F = 6$, while S_i is different for every i. The total number of second categorization levels is 37, therefore each POI descriptor is a string of 37 bits. For example, the descriptor of the Ducal Palace of Colorno (Parma) is shown in Table 3.

The POI descriptor is enriched by additional information, which can be used by the recommendation filters, or to improve the POI presentation to the user. Examples are the geographic coordinates, necessary to filter the POI during the recommendation process.

[7] Our main reference is the classification provided by the Italian Ministry of Culture: http://www.beniculturali.it/mibac/sitoiBAC/LuoghiDellaCultura/.

Please cite this article in press as: M. Amoretti, et al., UTravel: Smart mobility with a novel user profiling and recommendation approach, Pervasive and Mobile Computing (2016), http://dx.doi.org/10.1016/j.pmcj.2016.08.008

Table 3
POI descriptor of the Ducal Palace of Colorno (Parma).

Cultural heritage	Cultural buildings	Cultural events	Food services	Entertainment	Trade
00**1**0000	000**11**0	00000000	000	0000000	000000

3.3. Demographic profile

The demographic profile is composed by a set of personal user data, such as: age, gender, employment. Such data are mandatory, and explicitly requested to the user during the registration phase. They represent the first and most simple form of user profiling, which is used by UPR to place the user in a suitable group, when no other information is available (*e.g.*, in the early stage of the user's interaction with the app).

3.4. Preference profile

The UTravel app allows the users to rate the POIs presented by the system. From these ratings, the preference profile is constructed, by means of a multi-criteria technique similar to the one proposed by Lakiotaki et al. [5]. The criteria that are taken into account for the evaluation of a POI are: quality of services, cost, reachability, waiting time, overall rating. Each criterion can be rated with a value from 0 to 5, with granularity 0.5.

When the number of ratings exceeds a given threshold T, the module which computes the personal weight vector of the user is activated. As previously explained, the weight vector contains the estimates of the importance of each criterion in the decision process of the user. Thus, it allows to know whether the user, when selecting a location to visit, is more influenced by the cost, or by the quality of the services, etc. The weight values, which are normalized in order to sum to 1, are computed by means of the UTA* algorithm [6]. In general, any policy for activating the weight recomputing process can be installed in the system. The default policy activates the process periodically, every T ratings inserted by the user.

With respect to the approach used by Lakiotaki et al. [5], where all the users of the system are asked to evaluate the same set of reference products, UPR users are asked (but not forced) to rate the recommended items (*i.e.*, POIs, in UTravel). Another difference is in the way the ratings to be used with UTA* are ordered. In UPR, such ratings are ordered according to the value of the *overall rating*. In the system proposed by Lakiotaki et al., instead, there is a limited fixed set of items to evaluate, and the system asks the user to explicitly order them by preference.

3.5. User clustering

As previously mentioned, the user profile is computed from two components, an individual one, and another related to the group behavior. The latter component implies that users are grouped into distinct classes, by means of a clustering algorithm. To this purpose, we adopted the K-Means algorithm [7]. Given a set of points in a n-dimensional space, K-Means uses a local search approach to partition the set in K clusters, where K is fixed a priori. K-Means has been preferred to other clustering algorithms because of its simplicity and fast convergence [29].

Regarding input data, users can be separated into two main groups: those who have not provided a sufficient number of ratings, and those who have. The User Clustering module applies K-Means to both groups, using the demographic profiles for the first one, and the preference profiles for the second one. The resulting two sets of centroids and their associations with the users are stored into the backend database. Demographic profiles are sufficient to the system for providing recommendations—that, of course, are less precise than those based on both profile types.

3.6. Behavior profile

With respect to the approach proposed by Park et al. [4], where users are clustered by taking into account only their demographic profile, UPR considers the behavior profile, derived from the characterization of the user activity and the group the user belongs to. In UTravel, specifically, the set of actions which are taken into account are: clicking on a POI to see related details, rating a POI, pressing the check-in button of a POI (which means that the POI has been visited), saving a POI in the list of favorites, adding an interest category to the personal profile. Such actions are stored in the database and processed in order to compute the user behavior profile, by means of the following algorithm.

The interest degree of user A, with respect to a specific POI category (specified by two levels: i and j), is measured by counting the following actions: click, save, rate, check-in, set preference. Then, the weighted relative individual interest (WRII) is computed by means of the following formula:

$$WRII_A^{i,j} = \frac{WAII_A^{i,j}}{\frac{1}{|T|}\sum_{t \in T} WAII_T^{i,j}} \tag{2}$$

Table 4
Parameters of the $WAII_A^{i,j}$ equation.

Parameter	Definition
C_A	Total number of POIs clicked by user A.
$c_A^{i,j}$	POIs of the (i, j)th category, clicked by user A.
S_A	Total number of POIs saved by user A.
$s_A^{i,j}$	POIs of the (i, j)th category, saved by user A.
V_A	Total number of POIs visited by user A.
$v_A^{i,j}$	POIs of the (i, j)th category, visited by user A.
R_A	Total number of POIs rated by user A.
$r_A^{i,j}$	POIs of the (i, j)th category, rated by user A.
P_A	Total number of POIs marked as preferred by user A.
$p_A^{i,j}$	POIs of the (i, j)th category, marked as preferred by user A.

where

$$WAII_A^{i,j} = \alpha_1 \frac{c_A^{i,j}}{C_A} + \alpha_2 \frac{s_A^{i,j}}{S_A} + \alpha_3 \frac{v_A^{i,j}}{V_A} + \alpha_4 \frac{r_A^{i,j}}{R_A} + \alpha_5 \frac{p_A^{i,j}}{P_A} \tag{3}$$

is the weighted absolute individual interest. Its parameters are defined in Table 4.

The weighted relative group interest (WRGI) is computed in a similar fashion, by considering groups of user rather than individuals:

$$WAGI_{G_A}^{i,j} = \alpha_1 \frac{c_{G_A}^{i,j}}{C_{G_A}} + \alpha_2 \frac{s_{G_A}^{i,j}}{S_{G_A}} + \alpha_3 \frac{v_{G_A}^{i,j}}{V_{G_A}} + \alpha_4 \frac{r_{G_A}^{i,j}}{R_{G_A}} + \alpha_5 \frac{p_{G_A}^{i,j}}{P_{G_A}} \tag{4}$$

$$WRGI_{G_A}^{i,j} = \frac{WAGI_{G_A}^{i,j}}{\frac{1}{|P|} \sum_{p \in P} WAGI_P^{i,j}}. \tag{5}$$

Then, the user profile is computed as a weighted average of the WRII and WRGI:

$$WRIGI_A^{i,j} = \beta_I WRII_A^{i,j} + \beta_G WRGI_{G_A}^{i,j} \tag{6}$$

where β_I and β_G are the weights of the individual and group components, respectively, and can be adjusted to improve system performance.

When the user makes a new rating, a check-in or a change in her/his list of interest categories, the module that computes the behavior profile is immediately triggered. Clicks on POIs to read the related details are instead saved in the local database of the mobile application, and sent to the server at next startup.

3.7. Filtering

The filtering problem is defined as follows. Let $\mathcal{U}$ be the set of the users, and $\mathcal{I}$ the set of the POIs that can be recommended. It is possible to define the utility function f which measures the interest degree of user $u \in \mathcal{U}$ with respect to POI $i \in \mathcal{I}$. Being $\mathcal{R}$ the set of possible ratings for i, then f can be expressed as a relation which associates $\mathcal{U}$ and $\mathcal{I}$ with $\mathcal{R}$:

$$f : \mathcal{U} \times \mathcal{I} \rightarrow \mathcal{R}. \tag{7}$$

In general, f is not complete, as the interest degree of the user for all the POIs is unknown. Thus, the objective of the filtering problem is to estimate the interest degree of the user for the POIs he/she has not rated. The POIs with higher estimated ratings will be recommended to the user.

The UPR filtering approach is highly modular. In UTravel, there are three filters which can be combined in different ways, depending on the needs. The *Location Filter* selects POIs from the system database, directly, using the current user location (sent by her/his mobile device) as a context information—input parameters are the user latitude, longitude and interest radius. The latter parameter, which is set by the user, defines the circle outside which no POIs should be recommended. The *Behavior Filter* selects POIs with smaller Euclidean distance from the user behavior profile, using the approach suggested by Park and Chang [4]. In detail, the filter first selects the n_c categories with higher degree of similitude with the user behavior profile. For each category, the filter selects the n_p POIs with higher degree of similitude with the user behavior profile. Thus, no more than $n_c n_p$ POIs are recommended. To reduce the processing time, in the current UTravel prototype, the Behavior Filter is placed *after* the Location Filter. Finally, the *Favorite Category Filter* selects only the POIs in the user interest categories. It is placed after the Location Filter, in parallel with the Behavior Filter.

4. UTravel implementation

In this section we describe some relevant features of the current implementation of the UPR-based UTravel architecture, which includes a set of services, the centralized and the local databases, as well as the mobile application for the end users.

4.1. Services and databases

The core system of the UTravel architecture exposes its functionalities as services, by means of a Web Service implemented in Java language. Service inputs and outputs are strings, formatted with JSON[8] (JavaScript Object Notation). The latter is a lightweight data-interchange format, easy for humans to read and write, and easy for machines to parse and generate. UTravel services serialize and de-serialize JSON messages into Java objects by means of the Gson[9] library, developed by Google.

To store data, the UTravel platform is provided with two databases. The first one, which can be very large, is located on the server side, and stores data sent by users. The second is local to the client application, and is created when the latter is installed on the mobile device of the user. Its purpose is to store personal information, and to act as a buffer, to avoid a continuous transfer of data from client to server.

The server-side database is initially empty and gets filled as long as users interact with the application, by triggering POI retrieval from Google Places and Foursquare. New POIs that do not exist in those databases cannot be defined and saved by users. Such a feature is in development, within a reputation management scheme.

When the first user opens up the UTravel app, after the registration phase, a recommendation request is sent to the server, including user location information. If the system does not find POIs around that specific location, a background process is automatically started on the server, to retrieve and store new POIs within a radius of 20 km, taken from the aforementioned online data sources. Thus, the POI table is automatically filled in. At this point, the components that can be used for producing recommendations are the location filter and the category filter (as the user is allowed to select its favorite categories after registration). In fact, all components of the user behavior profile equal to 0. Therefore, the first recommendation is composed by POIs in user favorite categories and by the first n_p POIs in the n_c categories that are currently available within the system.

4.2. Mobile application

The UTravel mobile application has been implemented for both Android and iOS platforms—its structure is presented in Fig. 2. Importantly, the mobile application has been designed with particular attention to usability.

In our experience, usability is mainly affected by (1) design of the application interface, (2) number of required clicks to achieve the expected result and (3) response time. Regarding point (1), the UTravel app has a simple but effective interface. With respect to point (2), once logged in, the user obtains the recommended POIs with just one click. Moreover, the main views, corresponding to the main functionalities – namely, user profile management and POI management – can be easily reached by means of the menu button. Point (3) is critical and not under the control of developers. To minimize delay inconveniences, we minimized the number of communications between client and server, as well as the size of transmitted data (*e.g.*, JSON has been preferred to XML).

In the following, we refer to the Android implementation—the iOS version only differs in the look & feel.

When the user runs the mobile application for the first time, he/she must register an account. During the registration process, the user must provide a username and a password, as well as the information that is necessary to build his/her demographic profile. The latter includes: gender, birth date, and occupation. After registration, a new screen in shown, which allows to select interest categories (Fig. 3).

After login, the main dialog of the application is shown. The latter consists of a map, centered on the location of the user, with a number of markers indicating the locations that should interest the user (Fig. 4). There are three types of markers, which can be selectively visualized:

- recommended POIs (green marker, with a yellow star)—generated by the system, using the Behavior Filter (illustrated in Section 3.7);
- POIs of the interest categories (green marker)—such locations are not recommended, however they may be of interest for the user, according to his/her preferences (which are taken into account by the Favorite Category Filter, discussed in Section 3);
- SOIs (blue marker, with an S)—Services of Interest for the user (their categories can be selected from a list provided by the application).

Only the markers within the Search Range d, decided by the user (up to 10 km), are shown.

[8] http://www.json.org.

[9] http://code.google.com/p/google-gson/.

Please cite this article in press as: M. Amoretti, et al., UTravel: Smart mobility with a novel user profiling and recommendation approach, Pervasive and Mobile Computing (2016), http://dx.doi.org/10.1016/j.pmcj.2016.08.008

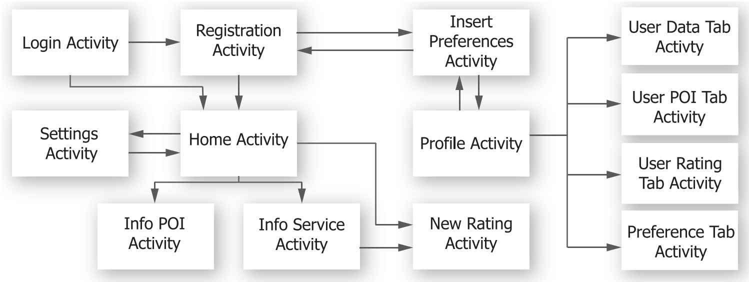

Fig. 2. The structure of the UTravel mobile application.

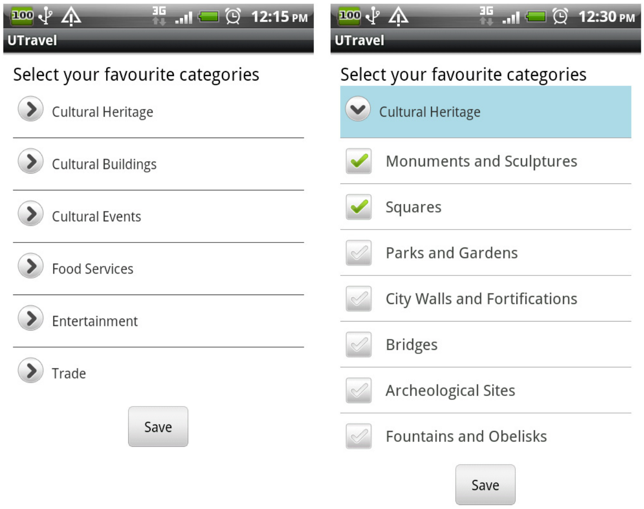

Fig. 3. Selection of interest categories, in the UTravel mobile application.

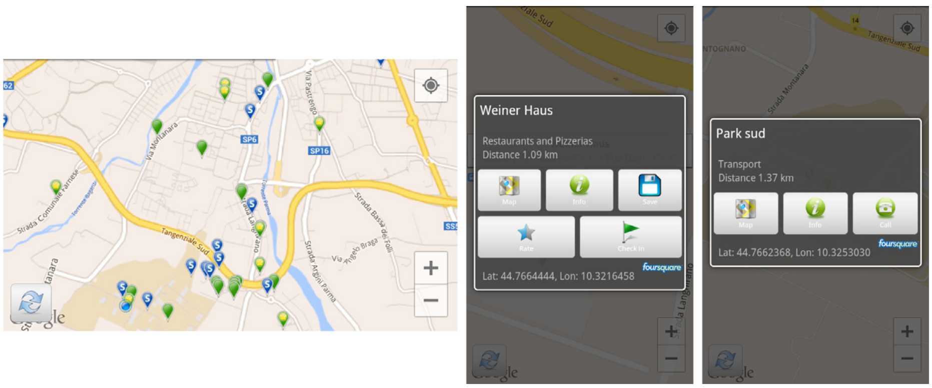

Fig. 4. Main dialog of the UTravel mobile application (left). Dialogs visualized when POIs (center) and SOIs (right) are clicked, in the UTravel mobile application.

By clicking on a marker, a dialog pops up, allowing the user to perform operations which are related to the location associated to the selected marker. As shown in Fig. 4, the available operations are different for POIs and SOIs. Indeed, only POIs can be voted, saved and checked in.

Detailed information about POIs and SOIs can be read by clicking on the info button in the dialog that appears when a marker is clicked. Details are different, depending on the type of location. The name, category and address of the location are shown. A series of buttons indicate the actions that the user can perform, such as visualizing the Web page related to the location, contacting by phone or email, as well as saving, evaluating and checking in. The last three operations, as previously stated, are allowed for POIs only, not for SOIs (which are not taken into account by the recommendation procedure). For some POIs it is also possible to read a textual description. In the bottom part of the view there is the logo of the data source, which may be Google Places, Foursquare or UTravel itself.

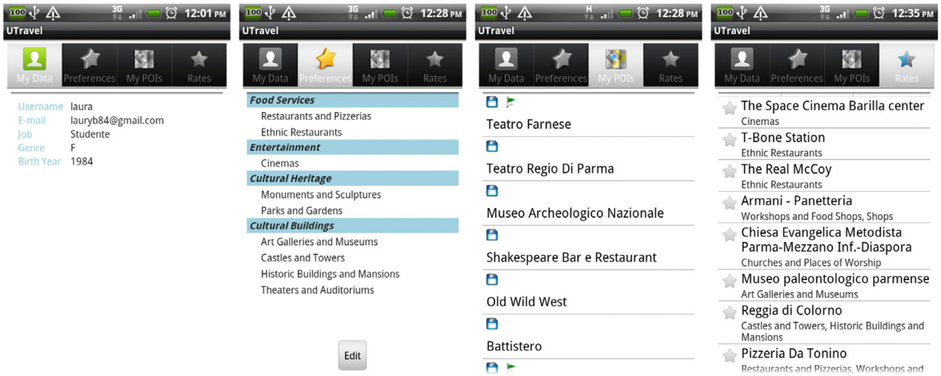

Fig. 5. User details, in the UTravel mobile application.

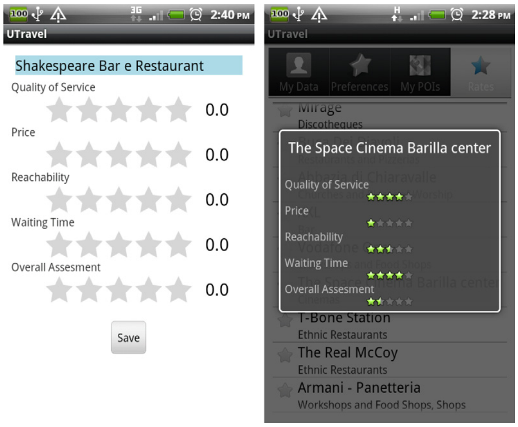

Fig. 6. Dialogs for rating POIs and showing their rates, respectively, in the UTravel mobile application.

The visualization of the info section of a POI is counted as a click, and stored in the local database of the mobile application. This happens also for check-ins and ratings.

The section with user details includes four views:

- My Data, which contains personal user data, inserted during the registration process (Fig. 5, on the left);
- Preferences, which shows the list of current categories of interest of the user, which have two levels of classification (*e.g.*, Entertainment has sub-levels like Pubs, Discotheques, etc.);
- My Places, which lists the POIs that the user has interacted with (different symbols indicate whether they have been saved, or visited);
- Ratings, which lists the POIs that have been rated by the user.

By clicking on saved, visited, or rated POIs, their details are shown in a separate view.

The user can insert an evaluation for a POI, by means of the form illustrated in Fig. 6. By means of rating bars, it is possible to assign a value between 0 and 5 (with granularity 0.5) to each evaluation criterion. Once an evaluation has been saved, it can be seen in the Ratings view in the section with the details of the user (Fig. 5).

The last function offered by the application is the possibility to configure some of the working parameters. There are three groups of configurable parameters:

- *Search Range*, which allows to set the radius of the circle of the recommended POIs, to be centered in the user's current location;
- *Quick Access*, to make the application remember login data (by default it is set to true);
- *Include Services*, which allows to enable/disable the visualization of the markers associated to the SOIs; if enabled, the user is asked to set his/her service categories of interest.

The purpose of UTravel is to make tourist mobility more smart, by suggesting POIs according to user preferences, user behavior and group affinity. Current prototype is the result of our major effort on designing a general-purpose UPR architecture and implementing UTravel-specific profiling and recommendation modules. Additional features to the mobile application are easily achievable, such as enabling automatic "Around me" POI notifications or clicking a point on the map (different from current user location) and receiving information about relevant POIs located around that point. Another

Table 5
Average preference profiles of simulated users.

	Quality of service	Price	Reachability	Waiting time
PQ	**0.27422**	0.24190	0.24199	0.24182
PP	0.24286	**0.27317**	0.24238	0.24158
PR	0.24323	0.24273	**0.27124**	0.24290
PW	0.24271	0.24234	0.24286	**0.27212**

interesting addition would be allowing users to receive suggested paths within a selected area, possibly taking into account performance, cost and environmental impact for the available means of transport. Last but not least, mobility prediction [30] would make the application very attractive and will be considered in future work. In general, all of these extensions would enable UTravel to support to a greater extent smart mobility.

5. Experimental evaluation

With the purpose of evaluating the proposed system, we defined two different testing steps. The first one, carried out with a set of simulated users, aimed to verify the accuracy of user profiles and clustering modules. The second step involved several real users in order to test the system functionalities as a whole.

5.1. Simulated user evaluation

In this first step of evaluation we implemented a Java-based script that simulates actions of users with predefined particular interests and behavior. More in detail, from the behavior perspective, for each user, the script generates a sequence of operations that match one of the following behavior profiles:

- BC: Users mainly interested in Cultural POIs;
- BF: Users mainly interested in Food-related POIs;
- BE: Users interested in Entertainment POIs;
- BT: Users interested in Trade-related POIs.

Moreover, from a preference point of view, the script associates each user to one of the following preference profiles:

- PQ: Users influenced by quality of POI services;
- PP: Users mainly influenced by POI prices;
- PR: Users mainly influenced by POI reachability;
- PW: Users mainly influenced by POI waiting time.

Each simulated user U takes a target preference profile TP and a behavior profile TB as inputs, then executes the following set of actions:

- Select the c_1 categories related with TB as favorite categories. To make simulated users more realistic, U also selects c_2 random favorite categories which are not related with the input profile B; in our simulations, c_1 is set to 3, and c_2 is set to 1.
- Perform n clicks operations, n save operations, n check-in operations and n rating operations. Each group of n operations is composed of n_1 operations on random POIs of categories in c_1, and n_2 operations on random POIs of categories in c_2. In our simulation we set $n = 10$, $n_1 = 9$ and $n_2 = 1$.
- The values of ratings are selected on the basis of the selected preference profile TP.

The script generates a set of 80 different users, 5 for each combination of behavior and preference profiles. The script has been executed 10 times, with different seeds for the random number generator.

After performing all script runs, the resulting profiles and clusters have been evaluated. In Fig. 7, the average behavior profiles – WRIGI indicator, defined in Section 3, Eq. (6) – for each possible group are shown. We can note that there are values higher than zero also for categories which do not match the target behavior profile B of users. This is due mainly for two reasons: (i) simulated users, in order to follow a more realistic behavior, also perform few actions related to POIs in categories different from the target; (ii) the behavior profile is calculated taking into account both user and group behavior, so the profile of a single user is influenced by actions performed by other users in the same group.

Table 5 shows the average preference profiles for simulated users in different groups. We can note that the higher components in each profile correspond to the criteria selected as the target for the group.

Finally, the clustering of simulated users has been analyzed. The K-Means algorithm correctly found the 4 clusters related to 4 different preference profiles. Fig. 8 shows how users have been assigned to clusters.

5.2. Real user evaluation

A controlled test, carried out by means of a Web-based version of the UTravel application, was performed in the area of Parma. We enrolled 100 users, one after the other over two months, to play with the application. Among the 100 enrolled

Please cite this article in press as: M. Amoretti, et al., UTravel: Smart mobility with a novel user profiling and recommendation approach, Pervasive and Mobile Computing (2016), http://dx.doi.org/10.1016/j.pmcj.2016.08.008

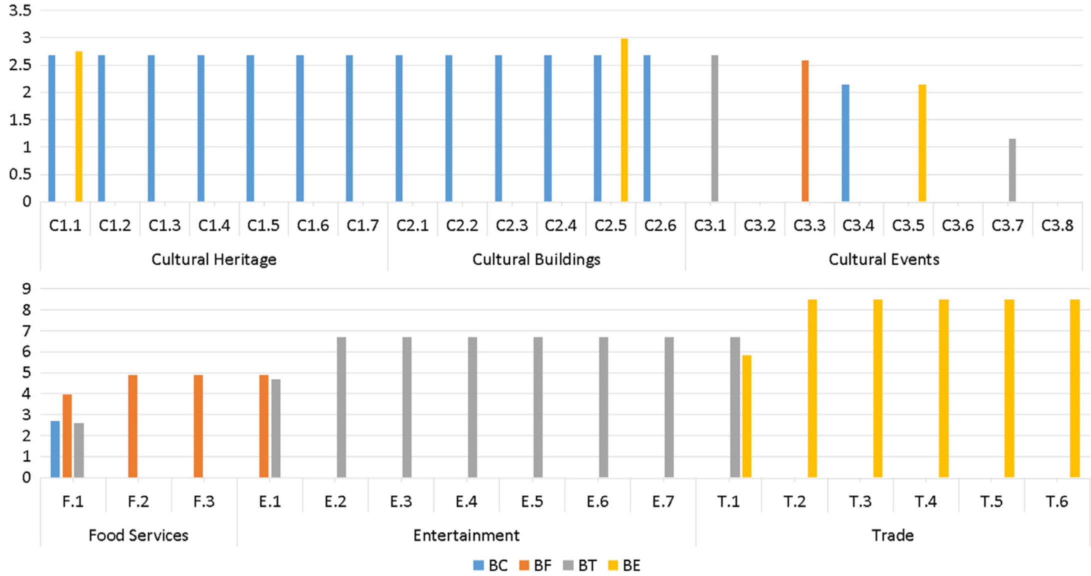

Fig. 7. Average behavior profile (WRIGI) of simulated users for each POI category.

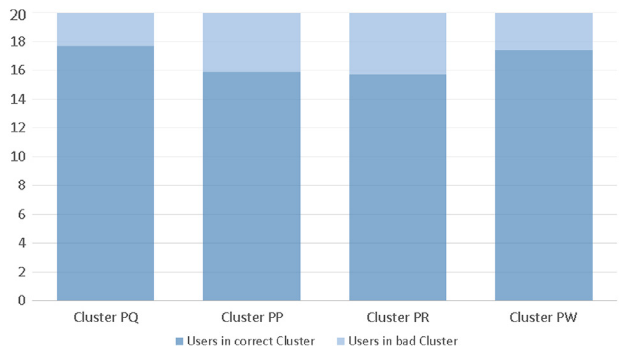

Fig. 8. Clustering of simulated users.

users, 35 were particularly active, as they not only compiled their personal profile and asked for recommendations, but they also rated, checked-in and saved POIs.

Before starting the evaluation, we configured the recommendation system with the following parameters:

- $T = 4$ for the minimum number of ratings a user must provide to be included in the Preference Profile;
- $K = 15$ for the maximum number of clusters, in K-Means;
- $N = 3$ for the number of K-Means runs.

Then, for each joining user, we computed the following variables:

- N_s, total number of recommended POIs the user interacted with (*i.e.* either rated, visited, saved or discarded)—recommended POIs ignored by the user are not taken into account;
- N_r, total number of POIs the user considered as relevant (*i.e.* rated $\geq$ 4, visited or saved);
- N_{rs}, total number of recommended POIs that the user considered as relevant (*i.e.* rated $\geq$ 4, visited or saved).

At the end of the evaluation process, we also obtained the following measures:

- I_r, total number of POIs recommended by the system during the evaluation period;
- I_p, total number of POIs located in the area of Parma.

To measure the recommendation system accuracy, we used the *precision*, *recall*, $F1$ and *coverage %* performance indicators [31]. Precision, defined as

$$P = \frac{N_{rs}}{N_s} \tag{8}$$

Please cite this article in press as: M. Amoretti, et al., UTravel: Smart mobility with a novel user profiling and recommendation approach, Pervasive and Mobile Computing (2016), http://dx.doi.org/10.1016/j.pmcj.2016.08.008

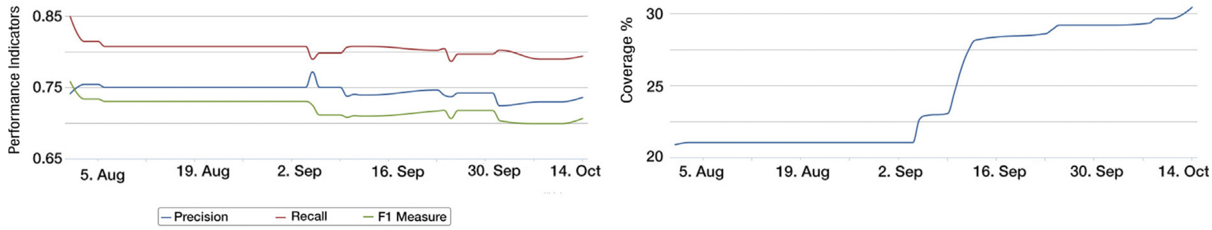

Fig. 9. Evolution of P, R, $F1$ and $C\%$ during the evaluation period.

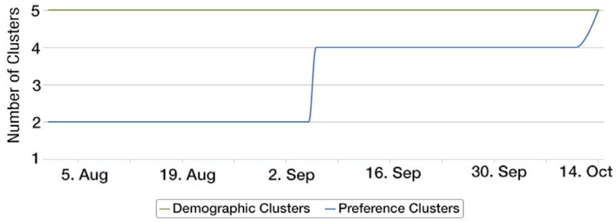

Fig. 10. Evolution of the number of clusters during the evaluation period.

is the fraction of recommended POIs that are relevant to the user profile. Recall

$$R = \frac{N_{rs}}{N_r} \tag{9}$$

is the fraction of relevant POIs that are retrieved. Increasing the number of recommended POIs tends to reduce precision and increase recall. The $F1$ metric

$$F1 = \frac{2PR}{P + R} \tag{10}$$

can be used to balance the trade-off between precision and recall [32]. Finally, coverage %

$$C\% = \frac{I_r}{I_p} \cdot 100 \tag{11}$$

represents the percentage of POIs recommended to users during the evaluation period, over the total number of POIs. In other words, it is a measure of the percentage of items for which a recommendation system can provide recommendations. Thus, a high coverage value indicates that the recommendation system supports the user in selecting among most of the available items.

Fig. 9 illustrates the evolution of the performance indicators (precision, recall and F1) during the evaluation period. We can observe that, after a short transitory, the recommendation system converges to an almost stable state.

As shown in Table 6, the P, R, $F1$ and $C\%$ values obtained at the end of the evaluation period are higher than those of the RS_IGB system proposed by Park et al. [4]. UPR results are also close to those presented by Lakiotaki et al. [5], which were obtained with a large-scale evaluation, with more than 6000 users, although the coverage % was not evaluated.

The temporal evolution of P, R, $F1$ and $C\%$ is reported in Fig. 9. The evolution of the number of clusters is reported in Fig. 10. Demographic clusters are based on user descriptions (personal details, interest categories) provided when registering to UTravel—thus, they include all 100 users. Preference clusters are based on information generated when using the mobile application to rate, check-in, store POIs—as a consequence, such clusters include only the 35 most active users. The reader may observe that, in the first and second weeks of September, there was a considerable coverage increment, and then the number of clusters changed from 2 to 4. The first fact was due to a set of users who interacted with (*i.e.*, checked-in, stored, rated) several POIs, performing an unusual amount of operations. The second fact was a consequence of such behaviors.

Finally, we investigated the usability of the mobile app through a questionnaire submitted to 20 users which had to install and test UTravel on their smartphones in a period of about 3 weeks. At the end of the trial, each participant was required to assign a score (from 0 to 10) to some usability indicators, representing a subset of those described by Ji et al. [33]. The average results are reported in Table 7.

6. Conclusion

In this article we presented the UPR approach, a novel context-aware personalization system currently supporting UTravel, a mobile application which informs and guides the users towards points of interest. We described the UPR-based

Table 6
Precision, Recall, F1 and Coverage % in the UPR approach, compared to RS_IGB [4] and to MR-CF-dim [5].

	P	*R*	*F*1	*C*%
UPR approach	0.73	0.8	0.715	30.42%
RS_IGB	0.340	0.628	0.410	28.108%
MR-CF-dim	0.974	0.882	0.926	–

Table 7
Selected Utravel app usability indicators.

Name	Definition	Mean value (Std. Dev.)
Learnability	The UI must facilitate the process of learning how the app works.	8.75 (1.33)
Structure principle	The UI must be organized purposefully, in meaningful and useful ways that put related things together and separate unrelated things, based on clear, consistent models that are apparent and recognizable to others.	8.9 (1.44)
Visibility	The UI should always keep users informed about what is going on, through appropriate feedback within reasonable time.	8.6 (1.50)
Responsiveness	The system must respond in an appropriate time.	8.5 (1.50)
Effort	The user interface should be designed to minimize effort for using the system.	8.6 (2.16)

UTravel architecture, focusing on two modules – the user profiling one, and the item filtering one – that compose a powerful recommendation system. As demonstrated by the results of the experimental evaluation, system recommendations have a high degree of precision, recall and coverage.

We have already implemented both Android and iPhone versions of the UTravel mobile app, so that we would like to publish them in the appropriate app stores to increase the number of users. To this purpose, it would become necessary to deploy the service-oriented architecture to the Cloud, which would be increasingly expensive. To avoid cloud deployment and reduce maintenance costs, we are taking into consideration to design a decentralized version of UPR (and UTravel), with cooperating mobile applications and no central server. Because of the limited knowledge at each node, such an approach would be less effective than the one we presented in this paper. However, it would be more scalable than the currently used centralized server which might be overloaded by a very large number of users. Last but not least, in a future work we shall describe the other UPR-based applications we are implementing and validating. For example, URun recommends walking/running/cycling itineraries, while UCook recommends recipes.

Acknowledgment

The work of Laura Belli is supported by Multitraccia SC.

References

[1] H.-N. Kim, I. Ha, K.-S. Lee, G.-S. Jo, A. El-Saddik, Collaborative user modeling for enhanced content filtering in recommender systems, Decis. Support Syst. 51 (4) (2011) 772–781.
[2] M. Gao, K. Liu, Z. Wu, Personalisation in web computing and informatics: Theories, techniques, applications, and future research, Inf. Syst. Front. 12 (5) (2010) 607–629.
[3] T.-P. Liang, Y.-F. Yang, D.-N. Chen, Y.-C. Ku, A semantic-expansion approach to personalized knowledge recommendation, Decis. Support Syst. 45 (3) (2008) 401–412.
[4] Y.-J. Park, K.-N. Chang, Individual and group behavior-based customer profile model for personalized product recommendation, Expert Syst. Appl. 36 (2) (2009) 1932–1939.
[5] K. Lakiotaki, N.F. Matsatsinis, A. Tsoukiàs, Multicriteria user modeling in recommender systems, IEEE Intell. Syst. 26 (2) (2011) 64–76.
[6] Y. Siskos, E. Grigoroudis, N.F. Matsatsinis, UTA methods, in: Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, 2005.
[7] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: 5th Symposium on Mathematical Statistics and Probability, UC Press, Berkeley, 1967.
[8] Ö Yilmaz, R.C. Erdur, An intelligent context-aware system, Expert Syst. Appl. 39 (3) (2012) 2907–2918.
[9] B.N. Schilit, M.M. Theimer, Disseminating active map information to mobile hosts, IEEE Netw. 8 (5) (1994) 22–32.
[10] M.J. Barranco, J.M. Noguera, J. Castro, L. Martínez, A context-aware mobile recommender system based on location and trajectory, in: Management Intelligent Systems, Springer, 2012.
[11] A.K. Dey, G.D. Abowd, D. Salber, A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, Hum.-Comput. Interact. 16 (2) (2001) 97–166.
[12] P.J. Brown, J.D. Bovey, X. Chen, Context-aware applications: from the laboratory to the marketplace, IEEE Pers. Commun. (4) (1997) 58–64.
[13] M. Braunhofer, M. Elahi, M. Ge, F. Ricci, STS: Design of weather-aware mobile recommender systems in tourism, in: 1st Workshop on Intelligent User Interfaces (AI*HCI 2013), Turin, Italy, 2013.
[14] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: Recommender Systems Handbook, Springer, 2011.
[15] A. Odic, M. Tkalcic, J.F. Tasic, A. Kosir, Relevant context in a movie recommender system: Users opinion vs. statistical detection, in: 4th Workshop on Context-Aware Recommender Systems, Dublin, Ireland, 2012.
[16] F. Lemos, R. Carmo, W. Viana, R. Andrade, Towards a context-aware photo recommender system, in: 4th Workshop on Context-Aware Recommender Systems, Dublin, Ireland, 2012.
[17] L.O. Colombo-Mendoza, R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández, J.J. Samper-Zapater, RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes, Expert Syst. Appl. 42 (2015) 1202–1222.
[18] M.J. Berry, G. Linoff, Data Mining Techniques: for Marketing, Sales, and Customer Support, John Wiley & Sons, Inc., New York, NY, USA, 1997.

[19] F. Cena, L. Console, C. Gena, A. Goy, G. Levi, S. Modeo, I. Torre, Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide, AI Commun. 19 (4) (2006) 369–384.
[20] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, Incorporating contextual information in recommender systems using a multidimensional approach, ACM Trans. Inf. Syst. (TOIS) 23 (1) (2005) 103–145.
[21] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, Context-aware SVM for context-dependent information recommendation, in: 7th Int.l Conference on Mobile Data Management, Nara, Japan, 2006.
[22] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, A. Pedone, Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems, in: 3rd ACM Conference on Recommender Systems, ACM, 2009, pp. 265–268.
[23] M. Roberts, N. Ducheneaut, B. Begole, K. Partridge, B. Price, V. Bellotti, A. Walendowski, P. Rasmussen, Scalable architecture for context-aware activity-detecting mobile recommendation systems, in: Int.l Symposium on a World of Wireless Mobile and Multimedia Networks, Newport Beach, USA, 2008.
[24] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: 2008 ACM Conference on Recommender Systems, 2008, pp. 335–336.
[25] H. Costa, B. Furtado, D. Pires, L. Macedo, F.A. Cardoso, Context and intention-awareness in POIs recommender systems, in: 4th Workshop on Context-Aware Recommender Systems, Dublin, Ireland, 2012.
[26] G. Adomavicius, N. Manouselis, Y. Kwon, Multi-criteria recommender systems, in: Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners, Springer, 2010.
[27] A. Likasa, N. Vlassisb, J.J. Verbeekb, The global k-means clustering algorithm, Pattern Recognit. 36 (2) (2003) 451–461.
[28] M. van Setten, S. Pokraev, J. Koolwaaij, Context-aware recommendations in the mobile tourist application COMPASS, in: 3rd Int.l Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Eindhoven, The Netherlands, 2004.
[29] L. Bottou, Y. Bengio, Convergence properties of the K-means algorithm, in: Advances in Neural Information Processing Systems, vol. 7, MIT Press, 1995.
[30] A. Noulas, S. Scellato, N. Lathia, C. Mascolo, A random walk around the city: New venue recommendation in location-based social networks, in: 2012 ASE/IEEE Int.l Conference on Privacy, Security, Risk and Trust, Washington DC, USA, 2012.
[31] D. Billsus, M.J. Pazzani, Learning collaborative information filters, in: 15th Int.l Conference on Machine Learning, San Francisco, USA, 1998.
[32] C.J. Van Rijsbergen, Information Retrieval, second ed., Butterworth, 1979.
[33] Y.G. Ji, J.H. Park, C. Lee, M.H. Yun, A usability checklist for the usability evaluation of mobile phone user interface, Int. J. Hum.-Comput. Interact. 20 (3) (2006) 207–231.

Please cite this article in press as: M. Amoretti, et al., UTravel: Smart mobility with a novel user profiling and recommendation approach, Pervasive and Mobile Computing (2016), http://dx.doi.org/10.1016/j.pmcj.2016.08.008