

Context-aware media recommendations for smart devices

Abayomi Moradeyo Otebolaku¹ and Maria Teresa Andrade^{1,2}

¹*Telecommunications and Multimedia Unit, INESC TEC, Porto, Portugal*

²*Faculty of Engineering, University of Porto, Porto, Portugal*

Corresponding Author:

Abayomi M. Otebolaku

Email: abayomi.otebolaku@inescporto.pt

Abstract. The emergence of pervasive computing, the rapid advancements in broadband and mobile networks and the incredible appeals of smart devices are driving unprecedented universal access and delivery of online-based media resources. As more and more media services continue to flood the Web, mobile users will continue to waste invaluable time, seeking content of their interest. To deliver relevant media items offering richer experiences to mobile users, media services must be equipped with contextual knowledge of the consumption environment as well as contextual preferences of the users. This article investigates context-aware recommendation techniques for implicit delivery of contextually relevant online media items. The proposed recommendation services work with a contextual user profile and a context recognition framework, using case based reasoning as a methodology to determine user's current contextual preferences, relying on a context recognition framework to identify user's dynamic contextual situations from device's built-in sensors. To evaluate the proposed solution, we developed a case-study context-aware application that provides personalized recommendations adapted to user's current context, namely the activity he/she performs and consumption environment constraints. Experimental evaluations, via the case study application, real-world user data, and online-based movie metadata, demonstrate that context-aware recommendation techniques can provide better efficacy than the traditional approaches. Additionally, evaluations of the underlying context recognition process show that its power consumption is within an acceptable range. The recommendations provided by the case study application were assessed as effective via a user study, which demonstrates that users are pleased with the contextual media recommendations.

Keywords. Smartphones; context-awareness; multimedia; context-aware recommendation; contextual user profile; mobile environment

1. Introduction

The advancements in mobile computing technologies, wireless and mobile networks, and the incredible appeals of smart devices have brought remarkable changes in the way we access online based media items. Using mobile devices, such as smartphones, to access online-based media content, as depicted in Fig. 1, has become easier and ubiquitous. Additionally, with mobile phones now equipped with sophisticated sensors such as video cameras, accelerometers, gyroscopes, and multimedia functionalities, user generated media content has become pervasive (Pessemier et al. 2009). Mobile users can now generate own media content, such as video recordings of events, news, etc., from their devices and upload them to social networks such as YouTube, Facebook, Twitter, etc. Because of this development, much more online-based media contents are published every moment. Consequently, mobile users now have at their fingertips myriad choices of items to select. However, hoping to find relevant media items that meet their contextual situations, users waste invaluable time carefully sifting through countless volumes of media contents.

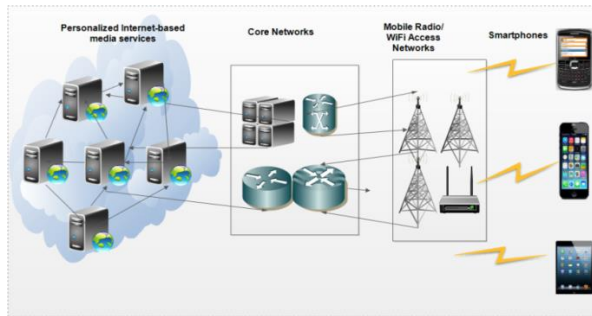


Fig.1 High-level internet-based mobile media delivery architecture

Often, because users' preferences are intuitively subject to contextual situations, users end up with irrelevant media items, which do not match these preferences in the present context. This problem is compounded in mobile environments in which present solutions do not incorporate contextual information into the recommendation process (Pessemier et al. 2009; Yu et al. 2006; Adomavicius and Tuzhilin 2011; Meehan et al. 2013; Chen 2005; Liu et al. 2008; Zhang et al. 2012; Wang et al. 2012; Ostuni et al. 2013). In the last fifteen years or so, contextual information has been used as a general term to describe *the situation of entities (i.e., whether a person, entity or object) that are considered relevant to the interactions between the users and the applications, including users and applications themselves* (Dey, Abowd and Salber, 2001).

Therefore, to assist mobile users make informed choices about relevant media content, many solutions have been proposed (Pessemier et al. 2009; Yu et al. 2006; Adomavicius and Tuzhilin 2011; Meehan et al. 2013; Chen 2005; Liu et al. 2008; Zhang et al. 2012; Wang et al. 2012; Ostuni et al. 2013; Pessemier et al. 2013; Xia et al. 2013). Highly successful traditional approaches based on personalized recommendations only assist users to discover explicitly relevant items. 1) using evaluations of previously consumed content given by users, an approach popularly known as content based filtering (CBF), or by using the evaluations given by users who are similar to the target users, otherwise known as collaborative filtering (CF) (Adomavicius and Tuzhilin 2005). Although these traditional recommendation techniques have been highly successful, they were not developed to handle implicit recommendation of contextually relevant media items. They only anticipate users to make explicit requests before providing recommendations. In addition, they assume that users always give ratings of the consumed media contents. In practice, however, giving ratings of consumed items rarely happens, especially in mobile environments. Additionally, the traditional techniques do not consider contextual information as an important factor that influences both item relevance and user satisfaction when consuming media content (Pessemier et al. 2009; Yu et al. 2006; Adomavicius and Tuzhilin 2011; Meehan et al. 2013; Chen 2005; Liu et al. 2008; Zhang et al. 2012; Wang et al. 2012; Ostuni et al. 2013).

Recently, encouraged by the rapid development in the field of mobile and context-aware computing, as well as the development of various practical context sensing applications, Adomavicius and Tuzhilin (2011) pioneered the incorporation of context information as integral parts of recommendation processes. Many solutions have been subsequently proposed. For example, Wang et al. (2012) used context information to generate music recommendations for mobile users. Ostuni et al. (2013) also explored contextual information to recommend interesting movies to mobile users. Contextual information has also been used in many other domains such as targeted advertisements, tourism, healthcare, etc. (Xia et al. 2013; Adomavicius and Tuzhilin 2005). However, current recommendation systems use contextual information explicitly to recommend media content to users, thereby requiring user's constant interventions. A truly context-aware recommendation system should suggest media content that matches the user's contextual preferences with limited interventions, and without interfering with the user's daily activities (Hong et al. 2009)

In addition, to deliver richer and personalized media experience to users in mobile environments (Resnick et al. 1994; Adomavicius and Tuzhilin 2005), it is important that context-aware recommendation systems understand the relationship between a user's preferences, contextual information (such as his locations and activities), and media services (Yin et al. 2011; Yujie and Licai 2010; Setten et al. 2004). These preconditions have been identified as the most important factors

that influence media consumption choices of mobile users. It is, therefore, important to capture this information dynamically. Additionally, these systems must operate with limited user's interventions, and as such must avoid the disruption of user's activities. (Yin et.al. 2011). More importantly, context-aware recommendation systems must be able to recognize the user's contexts and activities in real time without asking user's for this information(Xia et.al. 2013).

Consider this archetypal scenario, for example. *Sitting at home on a Friday at 8:30 PM, Ana enjoys watching video clips of latest movies using her smartphone. She relies on media recommendation service to provide her favorite choices, especially movies that her friends (who are also subscribers of the media recommendation service) have previewed and preferred. She wants to select at least one interesting movie from the list to watch on Saturday with her friends at the cinema, and she prefers to have such suggestions when she engages in some less demanding activity to have adequate attention to preview the clips of the recommended movies.*

In this scenario, existing solutions can easily provide movie recommendations that Ana would prefer, based on her previous consumptions (if any) or those consumed by her friends. Now, assume that Ana prefers watching a romance movie when she is alone with her partner. If the system suggests to Ana, *Insatiable*, a romance movie, which her friends have previewed and liked, then this recommendation would be a bad recommendation because Ana would have preferred to watch such movie while she is with her partner at home. This type of mistake or preference conflicts could be avoided if the recommendation service was aware of Ana's contextual choices at the time of recommendation.

Traditional recommendation techniques lack the capabilities that can effectively address the requirements, such as those in the above scenario, for implicit and explicit recommendations of rich media content, without conflict of preferences. The goal of this article, therefore, is fourfold. (1) Recognizing user's dynamic contextual information such as location, environment information (noise level, illumination, etc.) and user's daily activities in a mobile environment. (2) Providing a contextual user profile service that relates mobile user's media consumption preferences with contextual information to learn user's consumption behaviors in context. (3) Building contextual recommendation services based on the profile service for suggesting contextually relevant online-based media items. (4) Providing an adaptable presentation of the recommended media items to the users, matching the device and network constraints.

As illustrated in Fig. 2, the stated four goals constitute the main contributions of our work, embodied in the provisioning of a context-aware media recommendation framework for mobile devices (CAMR). CAMR provides features that can automatically provide media content recommendations, based on dynamic contexts inferred from user's smartphone-embedded sensors. It captures and utilizes the dynamic contextual information, especially mobile user activities, and relates the contextual information to user preferences for selecting suitable and relevant online media content, personalizing it for specific users in context. The proposed solution is extensible, and addresses different categories of media content domains such as movie, music, news, etc. Furthermore, we have implemented the system's components as RESTful web services (Java REST 2013), providing a platform-neutral access from any mobile device. We have evaluated its functionalities and its feasibility via an implementation of a mobile context-aware mobile recommendation system based on the proposed solution.

The remainder of this article is structured as follows. In section 2, we present some relevant related solutions that are similar to our proposed solution. Then, the problem definition and solution requirements are presented in section 3. In section 4, we present an overview of our proposed solution. In section 5, the proposed solution framework is presented. In section 6, the implementation and experimental evaluations of the proposed solution are presented. In section 7, we discuss the results of the evaluations, and in section 8, we conclude the paper and present its future direction.

2. Related work

Some recent and popular solutions in context-aware recommendations such as those proposed by Pessemier et al. (2013), Chen (2005), and Yu et al. (2006) focus on using contextual information to explicitly suggest media items. Cinemappy (Ostuni et al. 2013) used the location and movie consumption history of mobile users to recommend relevant movies. Wang et al. (2012) used mobile user's activities such as walking, running, studying, etc. to recommend music, but its weakness is that it considers only a limited contextual information i.e. only user activity. In context-aware recommendation literature, contextual information, especially user's activities and corresponding location information or other contextual information that influence user preferences have not been fully incorporated into existing solutions. Chen (2005) presented a context-aware collaborative approach, where recommendations were generated for users in specific contexts, based on what other users with similar profiles have consumed in the same contexts, by calculating the similarity between these contexts using user-rating data. However, in practice, generating similarity between contexts using rating is practically not feasible in some application domains such as mobile environments, especially if ratings of the consumed items in such contexts are not available. Bessemer et al. (2009) used location, environment information, and the user's mood to recommend user-generated content on a social network. The system, however, uses a tag cloud as contextual information source, and does not provide a practical means to obtain the user's current activity. Setten et.al. (2004) presented a solution that uses contextual information to recommend media content to tourists. Costa et al. (2007) explored the synergy between recommender systems and context-aware computing, aiming at making service provisioning more efficient, personalized, and proactive. Yu et.al. (2006) presented an interesting work using a hybrid recommendation approach and context awareness to recommend and to adapt media content for smartphone users. Nevertheless, these systems only provide explicit recommendations, i.e. they provide recommendations only when users

initiate the recommendation process. They also assume that contextual information is readily available, providing little detail of how dynamic context information can be acquired (Xia et.al. 2013). Pessemier et.al. (2013) also presented an interesting work, which uses daily activities and other contextual information of the user to provide recommendations. However, this work differs from our work because it uses static rules in its contextual user profile model to relate user preferences to contextual information, whereas our contextual user profile model is dynamic and able to use collaborative contextual preferences to obtain current contextual preferences.

Additionally, these cited solutions can only handle domain specific media items such as movie, music, news, etc. However, unlike these solutions, we propose a generic architecture, which incorporates a contextual model that can handle broader contextual information dynamically than just location information to provide recommendations for different categories of media content. In addition to user location, our work incorporates user as well as user device, user's activities, network and environment information, and has the capability to provide both explicit and implicit media recommendation services. The services provided by the architecture such as the recommendation processes, the contextual user profiling service, and context management service, etc. are implementation platform neutral because they are designed as web services, which are exposed as web interfaces, providing a platform-neutral access to any device running any mobile software and on any hardware platform. It also includes an optional adaptation service that can tailor the presentations of the recommended media content to device and network constraints.

3. Problem definition and system requirements

Contextual information is inherently dynamic (Dey and Abowd 2000). Therefore, enabling proactive and contextual recommendations to deliver relevant and richer media items that satisfy the mobile user's preferences in a heterogeneous mobile environment requires careful design considerations (Yujie and Licai 2010). It entails addressing several key issues, such as dynamic context and activity recognition, context representation, contextual user profiling and preference management for quality delivery of relevant media items, and adaptive presentation of the media items. Besides, developing a framework that incorporates these aspects, providing a unifying architecture, is not a trivial issue. In this section, we examine the difficulties that define the design of the proposed framework.

3.1. Context recognition and representation

An item is relevant only if it suits the user's preference in the present context of his active interaction (Yu et al. 2006; Adomavicius and Tuzhilin, 2005; Mobasher, 2010; Yujie and Licai 2010). Therefore, non-contextual recommendations in a mobile environment would lead to irrelevant media items. To explore contextual information for personalized content delivery, accurate recognition of dynamic contextual information is an important precondition, requiring context-sensing capabilities. However, sensory data come in a format that is not suitable for mobile application to adapt. We cannot use these raw context data directly because they contain noise, and in this form, these data cannot be used directly by mobile applications. These conditions necessitate a context recognition model, utilizing machine-learning approach that takes these low-level sensory signals as inputs and produces as outputs, accurate and meaningful high-level contextual information. Additionally, the recognized high-level contextual information must relate to other high-level contextual information, or even sometime with some low-level context data, to infer more semantically meaningful contextual situation of the user. The real-time and dynamic realization of this precondition remains a challenge in the overall process (Yujie and Licai 2010).

3.2. User and media content profiling

Another key requirement to offer truly contextual delivery of media content to mobile user is the user profile. User profile should encode all desirable media content features, customized for the profile owner and in context. The user profile includes, among others, preferences of the user for (1) desired media presentation characteristics, (2) optional user identities such as names, gender and profession, (3) preferred location information, (4) usage history and, (5) other high-level contextual information such as user activities in relation to the user's preferences. Dynamic acquisition of user preference information, user consumption history, and their incorporation with user's dynamic context and activities, to update the user's changing preferences, and to estimate their relevance and importance has also remained a challenge (Yin et al. 2011).

3.3. Content classification and recommendation

Today, the Web contains massive volumes of digital content that users explore daily using their mobile devices. The sheer size and the decentralization of this digital content makes it difficult for mobile users to obtain contents that suit their preferences and situations. Recommendation techniques have been proposed as solutions, which provide the

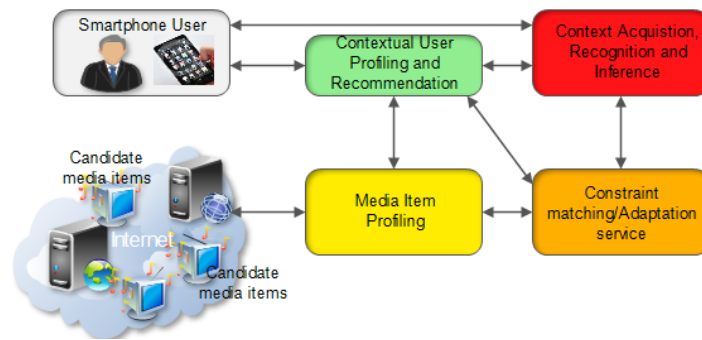


Fig. 2 Proposed system high-level architecture

advantage of guiding users in a personalized way to interesting content in a large space of possible options (Yu et al. 2006; Adomavicius and Tuzhilin 2005). However, the recommended content must be provided according to users's changing contexts, activities, and preferences. Therefore, mobile content recommendation systems must be equipped with this information to assist mobile users get rich media experience.

3.4. Media contents presentation adaptation

The mobility nature of mobile devices and network conditions create a situation whereby device and network resources are unlikely to match the required resources for satisfactory delivery of recommended media content. One of such situations is when available wireless network bandwidth is not adequate for delivering such rich media content, such as video. In this case, the video's spatial and temporal resolutions can be scaled down to match the available bandwidth. In addition, the content could be delivered in another modality, which does not require much bandwidth, such as text or image (Vetro 2004; Tseng et al. 2004; Steiger et al. 2003; Yu et al. 2006).

4. Proposed system overview

Fig. 2 depicts the basic architecture of the proposed solution. It consists of four main services. The first service is the context recognition, which runs on user's devices, monitoring, learning, and predicting user's contexts. This service gathers low-level events from the device's built-in sensors, preprocessing, and inferring user's high-level contexts. The recognized contextual information is sent to the contextual user profile service, which estimates and learns the media consumption preferences of users, and relates them to user's present context and past contexts (if any) as well as the contexts of the system's active users. It also learns and updates user profile in conjunction with the context service, using a case based reasoning process. It stores the user's consumption history and preferences in a contextual profile repository. The third service is the recommendation service, consisting of a context-aware recommendation algorithms and a media content profiling module. The latter obtains content metadata from the Web, crawling, and extracting useful metadata, whereas the former ranks and recommends the media content to users according to their contexts and preferences. The contexts include mobile user activities, locations, time of the day, and day of the week, location noise level and illumination, etc.. The adaptation service determines if the suggested media item is in the appropriate format for user's device supported characteristics and present network conditions, before its presentation to the user. If its format matches, it presents the media items directly to the user. Otherwise, it invokes the media adaptation service to adjust the format of the recommended media item to meet the device characteristics and network conditions (Vetro 2004; Tseng et al. 2004; Steiger et al. 2003).

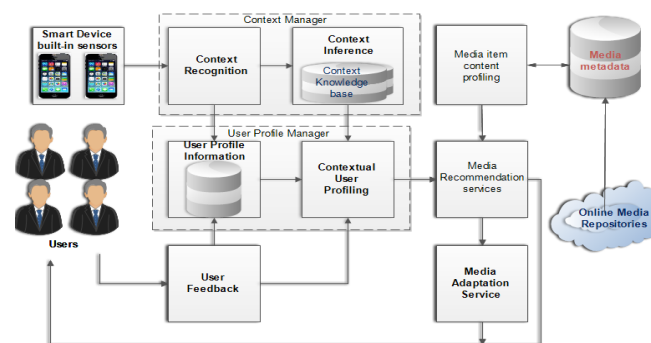


Fig. 3 CAMR architecture

5. Context aware framework for media recommendation

In this section, we present the proposed media recommendation solution for providing contextually relevant media content to mobile users. The proposed system addresses the requirements discussed in section 3. Fig. 3 sketches the architecture of the system, showing its major components. We will elaborate on each of these components in the following sections.

5.1. Context recognition

The ability of a system to identify contextual situations and adapt to them is one of the most important functions of any context-aware system. However, context sensors produce data that are in low-level form, which are not suitable for mobile applications. Context recognition is a process that collects raw data from device sensors and transforms them into information that can be used by mobile applications.

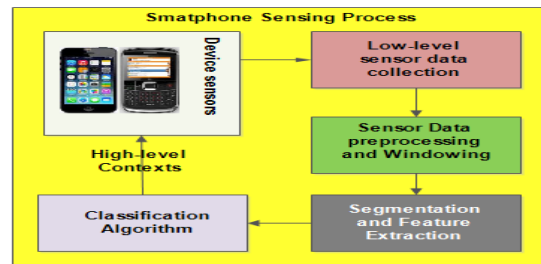


Fig. 4 Context recognition process model

To provide accurate contextual information about media items consumers, CAMR uses context recognition process to identify contextual information such as user activities, user location, and environment situation, e.g. weather, illumination, and the noise level, etc. from the low-level data collected from device's embedded sensors. To realize this, the context recognition service leverages four important processes as shown in Fig. 4, which are described in the next sections.

5.1.1. Sensor data collection and preprocessing

CAMR gathers low-level events from smartphone's embedded sensors, such as accelerometer, GPS, gyroscope, rotation vector, orientation, proximity, microphone, and light sensors. It collects 128 samples of data from each sensor's 3 axes, in continuous 3 seconds, with 64 samples from the previous 3 seconds overlapping the next 3 seconds. In the data preprocessing phase, it removes event outliers (Liu et al. 2008; Kwapisz et al. 2010; Lara and Labrador 2013). This is achieved by removing some samples from the beginning and ending of each example to limit the influence of noise on the sensory data. Additionally, the sensory events are channeled through low and high pass filters to reduce erroneous data. Low pass and high pass filters are the simplest and computationally less expensive filters. Low pass filter is used to de-emphasize high frequency noise, to allow low frequency or slow varying signals. This filter is used to enhance the detection of the direction of changes in the position of the devices. On the other hand, high pass filter is used to de-emphasize static or slowly varying sensory signals, to emphasize higher frequency or its transient components. This filter is used to enhance the recognition of sudden movements (Milette and Stroud 2012).

To realize and to experiment context recognition approach, we adopted Android as our mobile platform because it is now the most popular mobile platform for most popular smartphones. Another reason is that Android platform is open source and it is easier to program. Therefore, we designed and built an Android data collection application to gather representative low-level sensory events to build the context recognition service. The application presents a suitable user visualization interface and six volunteers were involved in the data collection process for an average of five days. We predefined eleven high-level activities: *walking*, *jogging*, *lying*, *running*, *standing*, *driving*, *sitting*, *ascending stairs*, *descending stairs*, *ascending in elevator*, and *descending in elevator*, characterized by location (such as *home*, *shops*, *coffee shops*, *bus station*, *train station*, *office*, *school*, *sport center*) time of the day (*morning*, *afternoon*, *evening*, *night*) and day of the week (*weekdays*, *weekends*, *holidays*), representative of a typical mobile user's common daily activities. Each location is characterized by environment noise level and illumination. Patterns from accelerometer, rotation, orientation, light, and microphone sensors were used to obtain the raw data. The volunteers, before starting each data collection process, pre-selected the activity they were about to perform, while holding the smartphone in their hands. The application can be stopped or allowed to run continuously until the user finishes the selected activity.

5.1.2. Feature extraction

Mobile user's dynamic contexts, such as activities, occur in relatively long period, in seconds or even minutes, when compared with the sampling rate of the device's embedded sensors (Lara and Labrador 2013). This sampling rate usually does not provide sufficient data to describe user activities because they occur faster than user activities. Therefore, activities are usually identified in time window basis rather than on the sampling rate basis (Lara and Labrador 2013;Kwapisz et al.2010). However, comparing a series of time windows to identify activities is almost impossible even if the signals being compared come from the same user performing the same activity context (Lara and Labrador 2013). The feature extraction process addresses this problem by filtering relevant data in windows sizes or lengths, and then obtains quantitative information from each time window. Many approaches have been proposed in literature such as statistical and structural properties of the sensor signals (Lara and Labrador 2013). Structural features such as Fourier transform are quite complex and require more computational resources. Fourier transform based features are computationally costly for an energy starved smart devices and therefore are not feasible in practical applications, especially in mobile environments where device power consumption is a great concern. On the other hand, statistical features are simple and require less computational resources (Lara and Labrador 2013). Thus, CAMR uses simple, labeled statistical features [Range, maximum, minimum, mean and standard deviation], which we have validated in our previous work as effective to discriminate between time windows (Otebolaku and Andrade 2013). These features are extracted into feature vectors that are then used in the context classification process.

5.1.3. Context classification

Extracting time window features from raw sensor signals, without deriving the contextual knowledge from them will produce information not suitable for context-aware applications. Therefore, CAMR uses supervised machine learning algorithms, in particular, Support Vector Machine, Neural Network (NN), Decision Trees, Nearest Neighbors (KNN), and BayesNet to derive high-level context from the statistical features. Details of the modeling and evaluations can be found in(Liu et al. 2008; Kwapisz et al. 2010; Lara and Labrador 2013; Otebolaku and Andrade 2013). These models are incorporated into the recognition service to obtain accurate, independent future activities and contexts of mobile phone users.

5.1.4. Context inference

The context information obtained from the last process is usually inadequate semantically for use in recommendation application. For example, it is important to know what a user is doing, when, and where he is doing it. Nevertheless, without relating this context information with each other, the recognized information might not be useful to the recommendation application. This is one of the weaknesses of the existing work. To address this weakness, CAMR uses a knowledge-based model on top of the context recognition and classification process to relate different atomic context information to obtain contextual information at a higher semantic level in relation to the user preferences. For example, having known that a user sitting at home is located in the living room, if we know that the TV in the living room is turned on (obtained from(infrared(IR) Blaster¹), it is important to relate this information and conclude that the user is watching the TV. The user is watching TV, obtained from sitting, home, living room, TV, etc. can be inferred using the ontology based knowledge inference process of CAMR. The details of this process can be found in (Otebolaku and Andrade 2011). CAMR can also determine such complex context, such as a user is "jogging at the sport arena". This kind of information is crucial to offering rich and personalized media experience to mobile users via context-aware recommendation systems.

5.2. Contextual user profiling service

A user profile describes preferences of as a summary, normally based on the history of the user's actions (Adomavicus and Tuzhilin 2005). CAMR's contextual user profile service (CUPS) summarizes the user's content consumptions into a limited set of categories, where categories are characterized by one or more genre, and a number of properties characterize the genre. Additionally, it incorporates contextual dimensions, associating one or more inferred context to each *category-genre-property* concept. It represents each user profile as a four level tree as shown in Fig.5, with the root of the tree representing user's optional demographic information. The first level nodes correspond to the content category, the second level represents the content genre, and the third level contains the properties of a given category-genre. This level provides the media item's metadata, characterizing at a finer detail, the consumed content and thus, the user preferences. A limited set of properties is used for each genre to obtain a good compromise between sufficient degree of characterization of content (hence, sufficient ability to make distinctions) and a reasonable dimensions of the user profile. The leaf nodes provide information about the contexts where the user preferences have been observed. These leaf nodes have four fields – *type*, *weight*, *intensity*, and *lifetime* – whereas all other nodes have only the type field. In the leaves, the types represent the type of context. The weight represents the preference of the user for a *category-genre-property*, which is obtained by tracking the number of times users have consumed item that matches a given category-genre-property. The newly introduced concepts in the user profile, the intensity and lifetime track user's contextual consumption history to enhance media personalization. The lifetime is the time (e.g. in Hours, days, months, etc.) since the last time a target user consumes item of the *category-genre-property*. Using these weighted parameters, the system is able to ascertain

dynamically, the media content that is relevant to target users, based on their contextual preferences. The *intensity* provides information about the number of times the user has consumed items of that *category-genre-property* and in that specific context. The intensity (the dynamic preference of the user) of those elements, which belongs to the media's term is obtained by summing up the products [weight x lifetime], given in equation (1), of all their occurrences. The intensity value of the retained elements at this level is obtained by visiting their child nodes in a breadth-first traversal. The same process applies to the retained elements of the category level. The intensity of the elements belonging to the genre level is the largest value of their children. Accordingly, these values are obtained by performing a depth-first traversal. This way, the user profile can handle any category of media items such as movie, news, music, etc..

$$p_{ni} = \sum_{i=1}^n (\text{weight} * \text{lifetime}) \quad (1)$$

5.2.1. Contextual user profile definition

As depicted in Fig.5, let U be a set of N mobile users. Each user u_i is defined by m set of optional demographic attributes $A = \{A_1, A_2, A_3, A_m\}$ and a set of preferences $PR = \{Pr_1, Pr_2, Pr_3, Pr_n\}$, where each preference Pr_i of user u_i is further defined by a set of attributes $T = \{T_1, T_2, T_3, \dots, T_p\}$. Additionally, let us define a set of high-level contextual information, $C = \{C_1, C_2, C_3, C_n\}$ associated with each preference Pr_i . In the above definitions, context C has a very complex structure, which reflects the nature of the difficulties to accurately represent contextual information. However, this paper assumes that contextual information is defined by a structure that allows atomic contexts to be related in a way that allows situational information to be inferred. To realize the context representation model, we built a three-layer structure consisting of low-level sensor data and high-level atomic contextual information, which can be related to each other to derive higher-level semantic contextual information. This process has been described in detail in our previous work (Otebolaku and Andrade 2011; Otebolaku and Andrade 2013). For instance, from the above definition, a user u_i where $u_i \in U = \{u_1, u_2, u_3, \dots, u_n\}$ is defined by a set of demographic information A and a set of preferences Pr . Each preference is classified into a category, $Ca = \{Cr_1, Cr_2, Cr_3, \dots, Cr_n\}$ and each category is characterized by a set of genres, $G = \{Gr_1, Gr_2, Gr_3, \dots, Gr_n\}$, and each genre is characterized by a set of attributes properties $Ga = \{Ga_1, Ga_2, Ga_3, \dots, Ga_n\}$. The user preference model is then built in a general form as follows.

$$Y = f(X_1, X_2, X_3, \dots, X_n) \quad (2)$$

Where $X_1, X_2, X_3, \dots, X_n$ are the demographic attributes from A and the preference information from Pr characterized by set G, Ca , and Ga . Y is the dependent variable to be predicted, and function f is the predictive function that is learned via some machine learning algorithm. For example, the system could predict the media item that user u_i would like to consume using this model. However, the model defined inequation(2) does not account for contextual preferences of users. To incorporate contextual information into the model, we modify the above model as follows.

$$Y = f(X_1 C_1, X_2 C_2, X_3 C_3, \dots, X_n C_n) \text{ or } Y = f_{c_n}(X_1, X_2, X_3, \dots, X_n) \quad (3)$$

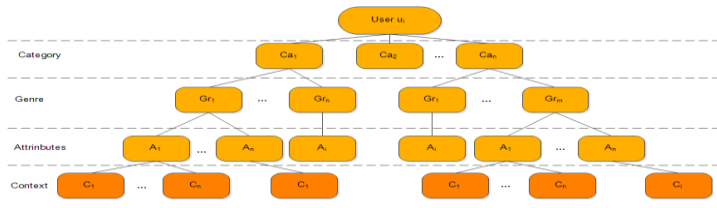


Fig. 5. The generic contextual user profile model

Where c_1, \dots, c_n represents the contextual information, and f_{c_n} is the function responsible for learning users' contextual preferences. This model can be used to build contextual user profile, which can be switched to provide non-contextual or traditional user preferences.

5.2.2. Dynamic tree based contextual user profile model representation

The definition of user profile model in the last section presents a hierarchical structure, which we have modelled as a four-level tree, as illustrated in Fig. 5. The root of the tree is the user u_i with optional demographic information, $A = \{A_1, A_2, A_3, \dots, A_m\}$. The first level of node corresponds to the category, $Ca = \{Cr_1, Cr_2, Cr_3, \dots, Cr_n\}$; the second level represents the genre, $G = \{Gr_1, Gr_2, Gr_3, \dots, Gr_n\}$; the third level contains the properties of a given category-genre, $Ga = \{Ga_1, Ga_2, Ga_3, \dots, Ga_n\}$. The fourth level contains the contexts in which media of category-genre is preferred, $C = \{C_1, C_2, C_3, \dots, C_n\}$. This level provides the media item's context, characterizing the consumed contents and thus, the user preferences. Fig.6 shows a hypothetical contextual profile model that can handle three categories of media content, namely, movies, music, and news with associated genre, properties and contextual information. The information in the profile is used to build the contextual user profile vector V_c . The elements of vector V_c are generated using contextual information to filter the user profile, using contextual pre-filtering method (Adomavicius and Tuzhilin 2011) and case

base reasoning algorithm (Aamodt and Paza 1994) to retrieve and to learn the user's contextual preferences. These processes are presented in the next sections.

5.2.3. Case base reasoning approach for retrieving contextual user preferences

To provide a mobile user with relevant media items that dynamically satisfies his preferences, matching his contextual preferences, is a crucial process that must be accurately determined to obtain accurate user's contextual preferences.

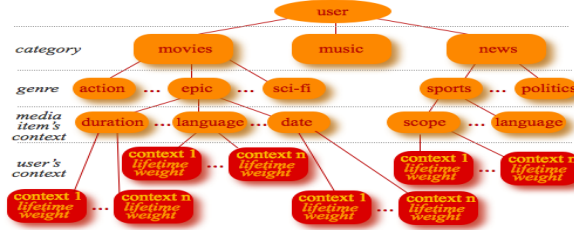


Fig. 6. The contextual user profile model for movie, music and news media

To achieve this, we propose a case base reasoning approach to retrieving and to learning contextual user preferences based on the currently identified user context.

a) Case based reasoning

Case based reasoning or CBR is a machine learning method that solves a new problem, remembering a previous similar situation, and by reusing information and knowledge in that situation (Aamodt and Paza 1994). It relies on the general knowledge of the problem domain or making association along generalized relationships between problem description and conclusion. It uses the specific knowledge of previously experienced, concrete problem situation (cases) to solve a newly identified problem. The new problem is solved by finding a similar past case and reusing it in the new problem situation. One unique and important feature of CBR is its ability for incremental and sustained learning from a new experience, which is retained each time a problem is solved, making it immediately available for future problems (Aamodt and Paza 1994).

In a context-aware media recommendation system, CBR can be used to learn the contextual user profile, using it to retrieve past user's contextual preferences stored in the user profile model, or contextual preferences of other users in the knowledge base, based on the currently recognized user's contextual situation (Bouighaghen et al. 2011). To accomplish this objective, four major processes corresponding to the processes, making up the CBR cycles have been developed, which are: *retrieve contextual preferences*, *reuse contextual preferences*, *revise contextual preference*, and *retain and update contextual preference*. These are cyclic processes as illustrated in Fig. 7. Basically, CBR uses k-nearest neighbor algorithm for retrieving similar cases. In order to compare the existing k-contextual preferences and current contextual preferences, CBR naturally supports similarity function, which is used to calculate the distance between these preferences. In this article, equations (4&5) are used to determine the similarity between k-previous contextual preferences(either of the target user or those of active users).

$$C^{\text{opt}} = \operatorname{argmax} \frac{\sum_{i=1}^n p n_i f(c_i^l, c_i^n)}{\sum_i^w p n_i} \quad (4)$$

b) Identifying and retrieving the current contextual preference

To obtain the present contextual preference by comparing present contextual situation and past contextual situation (with associated preferences), in CBR, this is similar to solving a new problem case by using solutions to past problem cases. Previously identified contextual situations and associated preferences constitute the best case, whereas a newly identified contextual situation is the new case. Therefore, a contextual preference case is described by a tuple <premise, value>. The premise is the description of the case with its characteristics, while value is the result of reasoning based on the premise. Specifically, the premise is a contextual situation (C) of a mobile user when using his smart device, whereas the value part is the user preference (PR) in that situation, which is used in the recommendation process. A contextual situation (a case) is represented as $C_s = \{(C^i, Pr^i)\}$, where C^i and Pr^i are contexts and associated preferences respectively. One of these preferences corresponds to the newly identified contextual situation. To select the most relevant contextual preference for the present contextual situation, for use by the recommendation process, the CBR process compares the present and the past contextual situations (and associated preferences) stored in the contextual user profile model knowledge base. Each preference in the knowledge base is associated with this contextual information as explained in section 5.2.2. For example, intuitively, Ana's contextual preference could be retrieved from the contextual user profile as "Every weekend, Ana prefers romance movies when she is at home with her partner". In this example, following contextual information can be identified: *User = Ana, Content category = Movie, Content Genre = Romance, Location = Home, Company spouse, Time = Evening, Day = Weekend*. The currently recognized contextual situation is obtained

¹An infrared blaster(IR blaster) is a device that emulates an infrared remote control to autonomously control a device that is normally controlled only by remote control's key presses. [http://en.wikipedia.org/wiki/Infrared_blaster]

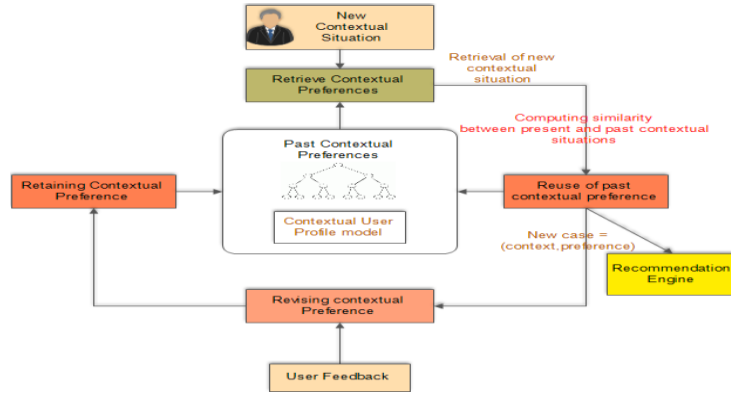


Fig. 7. CBR based user profiles process cycle

Context	Values	Type
Location	Home, Office, Mall, School, Coffee shop, Train station, Bus station	Categorical
Weekday	Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday	Categorical
Weekend	Saturday, Sunday	Categorical
Holiday	Weekend, others	Categorical
Time of the Day	Morning, Afternoon, Evening, Night	Categorical
Weather	Cloudy, Rainy, Snowing, Cloudy, Mostly Cloudy, Partly Cloudy	Categorical
Normal Temperature	Normal daily temperature	Numeric
High temperature	High temperature during the day	Numeric
Low Temperature	Low temperature during the day	Numeric
Activity	Walking, Sitting, Running, Jogging, Driving, Standing, ascending stairs, descending stairs, ascending elevator, and descending elevator.	Categorical
Media Category	News, Music, Movie	Categorical
Genre	Action, Animation, Comedy, Drama, Documentary, Epic, Horror, Politics, Sports, Thriller, etc.	Categorical
Language	English, French, German, Italian, Portuguese, Spanish, etc.	Categorical
Duration	Short (<30 min), Medium (>30 min, <75 min), Long (>75 min)	Numeric
Country	Canada, France, Germany, Italy, Portugal, Brazil, Spain, UK, USA, etc.	Categorical
Date	Old (>5 years), recent (<5 years, >1 years), present (<1 year)	Numeric
Illumination	Dark, very Dark, Moderately dark, Bright, moderately bright, very bright	Numeric
Noise level	Quiet, Moderate, Loud	Numeric

Table 1 example context information in a typical user profile preferences

from the context recognition framework and represented in a vector $C_s = (c_l^*, c_w^*, c_t^*, c_a^*, c_i^*, c_n^*)$, where $c_l^*, c_w^*, c_t^*, c_a^*, c_i^*, c_n^*$ represent user location, days of the week, time of the day, user activity, illumination, noise level of the location, etc. for instance. This is the basic context information, which is used to determine user preferences. For a user vector, which is submitted to the recommendation process, a current contextual situation (case) defined as $CC^* = (C^*, ?)$ is built and sent to the profile manager to determine its value part in the next process.

c) Retrieving the most similar contextual situations and preference

To determine the contextual preference in the present contextual situation case (CC^*), the present case is compared to the past contextual situations represented as vector $PC_s = \{C^l, C^n\}$, we select the contextual case that verifies the distance function in equation (4) similar to Boudghaghen et al. (2011) and Lee & Lee (2007). Where P_{ni} is the weighted intensity, representing the dynamic contextual preference of the user and value associated with the preferential terms in the user profile and $f(C_i^l, C_i^n)$ is the similarity metric for comparing C_i^l and C_i^n , which are the values of the feature i in the contextual situation vectors respectively.

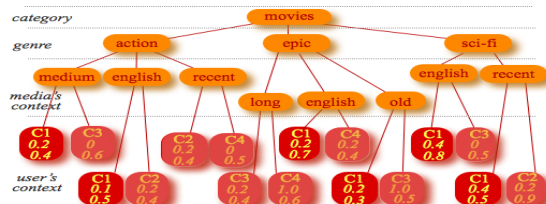


Fig. 8. Example user profile model for movie content category

1) To obtain the similarity between two numeric contextual preference cases, we used the equation (5).

$$f(C_i^l, C_i^n) = \begin{cases} 1-p & \text{if } 0 \leq p \leq 1 \\ 0 & \text{if } p > 1 \end{cases} \quad (5)$$

Where $p = \frac{|c_i^l - c_i^n|}{Max - Min}$, Max is the maximum value of the i th features of all cases in the case base and Min is the minimum value of the i th feature for all cases in the case base.

2) To obtain similarity between two categorical features, a partial matching scheme similar to (Lee & Lee, 2007) was developed using the domain knowledge. Table 2 presents the similarity scores for some select categorical features from the system's knowledge base used in our proposed solution.

d) Reuse the contextual situation to obtain the contextual user profile vector

After obtaining the current contextual preference of the target user, the recognized current context of use is compared with the leaves of the profile tree (context nodes) to generate elements of vector V_C via the process described in (c). Only those nodes whose leaves match the current context are processed. Remember that each element in the final vector is a pair keyword-intensity. Keyword is the textual value of nodes (the type field); intensity is obtained by multiplying the values of the field's weight and lifetime of the respective node. The weight represents the preference of the user while the lifetime represents the time elapse since the last time user consumes the content characterized by that *category-genre-property* node.

$$V_C = [(medium, 0.08), (english, 0.51), (old, 0.06), (recent, 0.2), (action, 0.08), (epic, 0.14), (sci-fi, 0.32), (movies, 0.54)]$$

Fig. 9. Example user vector

Contextual feature	Categorical similarity score							
	New Case	Walking	Running	Jogging	Driving	Lying	Standing	Sitting
Activity	Old Case							
	Walking	1.0	0.2	0.3	0.0	0.0	0.0	0.0
	Running	0.2	1.0	0.5	0.3	0.0	0.2	0.0
	Jogging	0.3	0.5	1.0	0.1	0.0	0.1	0.0
	Driving	0.0	0.3	0.1	1.0	0.0	0.0	0.0
	Lying	0.0	0.0	0.0	0.0	1.0	0.2	0.5
	Standing	0.1	0.0	0.0	0.0	0.2	1.0	0.3
	Sitting	0.1	0.0	0.0	0.0	0.5	0.3	1.0
...
Days		Weekdays	Weekend	Holidays				
	Weekdays	1.0	0.1	0.2				
	Weekend	0.1	1.0	0.9				
	Holidays	0.1	0.9	1.0				
Time		Morning	Afternoon	Evening	Night			
	Morning	1.0	0.3	0.1	0.0			
	Afternoon	0.3	1.0	0.5	0.3			
	Evening	0.1	0.5	1.0	0.5			
	Night	0.0	0.3	0.5	1.0			
...

Table 2 Example Similarity cores or categorical context features

For example, to generate the elements of vector V_C consider a hypothetical user profile of Mr. X represented in Fig. 8. Assuming that the system has inferred that Mr. X is in context C_1 , and that his contextual preference has been obtained, the elements that will be included in the contextualized user profile vector are the ones that have leaves with context value C_1 . The intensity of those elements belonging to the media's term is calculated by summing up the products [weight x lifetime] of all their occurrences (e.g., the node with Value "English" occurs three times for context C_1 ; therefore the intensity of "English" is the sum of the corresponding three intensities: $0.1 \times 0.5 + 0.2 \times 0.7 + 0.4 \times 0.8 = 0.51$). Fig. 9 represents the resulting contextualized user profile vector.

e) Updating the contextual user profile (Revising and retaining contextual user preference)

While generating contextually relevant recommendations, the system must also track the user's consumption behavior, and accordingly update the user profile to improve the system's recommendation accuracy. The user profile can be updated using two approaches, the explicit and the implicit methods (Adomavicus and Tuzhilin 2005). The former grants the users the ability to modify their preferences based on the recommendations they receive. The implicit approach involves preference learning without involving users directly, such as updating the profile when the user has spent a certain amount of time visualizing a given recommended content. The profile is updated whenever users consume any kind of media item. It allows associating importance with the finer detail of content and corresponding contexts of consumption. The model associates weight $\in [0,1]$ that relates the relevance of a media content preference with the contextual situation in which it is consumed, learning the user preference in the contextual situation. Similarly, it registers the contextual information about the consumed content, and those not selected by the users. The proposed system can use this knowledge to make *surprise* and improved recommendations. Thus, its update process is achieved in two ways, depending on the similarity obtained between current user contextual situation and the most similar situations.

- (1) If the $C^* \neq C^{opt}$: Means a new contextual situation (case) has been discovered and this is added accordingly to the contextual user profile. This is a situation where the present contextual case does not match any contextual cases in the knowledge base.
- (2) $C^* = C^{opt}$: The contextual situation is updated in the user profile to reflect the user preference.

A new intensity w_{ij} represents the relevance of content preference pr_j , which belongs to the contextual situation CC^* that user u_i consumes in context C_i . Whenever the context recognition model detects that the user is in a context C_i , for a continuous period of time $[0, T]$, the reasoner finds preference of content m_1, m_2, \dots, m_n , whose contextual situations match the present contextual situation of the user and associates weights $w_i c_i \in [0,1]$ with this m_i preference, which at time T is updated as follows:

$$w_{i+1}c_{i+1} = w_i c_i + \gamma (\alpha - w_i c_i) \quad i = 1, 2, 3, \dots, n. \quad (6)$$

Then for those content $m_1, m_2, m_3, \dots, m_{n-1}$ not consumed by the user, these weights are updated as follows:

$$w_{i+1}b_{i+1} = w_i b_i + \gamma (\alpha - w_i b_i) \quad i = 1, 2, 3, \dots, n \quad (7)$$

γ is a learning parameter whose value is obtained by:

$$\gamma = 1 - \left(\frac{t}{45} \right)^5 \quad (8)$$

$\alpha \in [0,1]$ its value is set to 1 in (6) and 0 in (7)

$w_i c_i$ are intensities of the recommended media items. Factor t in the equation (8) represents the number of days elapsed since the last time the user has consumed an item with the characteristics described by his profile. With equation (8), the parameter γ for each node remains above 0.9 during the first 30 days after it has been visited, rapidly decreasing to zero after that period (non-negative values are automatically converted to zero). For all other nodes, its value is linearly increased daily. This way, nodes that represent items that have not been seen for a long period, will eventually have no impact on the user's current preference's evaluation.

5.2.4. Media content profiling

Candidate media item is classified as relevant by calculating its similarity to the contextualized user profile vector V_C . It is thus necessary to create a media vector, V_M , for each media item. To describe the media items, CAMR relies on the availability of semantic metadata using the ²MPEG-7 MDS semantic tools. Given that in practice most of the media resources available online lack this metadata, the proposed system incorporates alternative methods to obtain the semantic descriptions of the candidate content. One of such alternatives is the public web services APIs, such as Internet Movie Data Base (IMDB) service API, Last.fm API, etc. For each media item, a vector V_M is initially created as an exact replica of V_C . Then, for every element of V_M , the system inspects the ²MPEG-7 metadata for a match. If it finds a match, it retains the intensity value of the matching element in a V_M . Otherwise, it assigns zero to the element.

5.3. Media recommendation service

The recommendation service explores three recommendation algorithms for context-aware media recommendations. The content base (CBF), the collaborative based (CF), and hybrid based recommendation algorithms. The traditional collaborative recommendation generates predictions for users, based on the item previously rated or viewed by other users, and the content-based approach, which uses the consumption history of the user. These two approaches suffer from the so-called overspecialization and new user/item problem respectively, which excludes casual users or those whom the system does not have enough information to generate recommendations [1,16], or suggesting always items similar to those consumed in the past by the user. Hybrid recommenders combine one or more of the conventional recommendation processes to overcome their individual weaknesses to gain better performance [1]. To take advantage of Mobile device's embedded sensors, we have incorporated contextual information into the three algorithms via a contextual user profile model as discussed in section 5.2 The context-aware recommendation processes were designed as web services, which can be invoked to recommend media items. The choice of what process to use depends on the category of media items to be recommended. The category is determined while retrieving user preferences from the

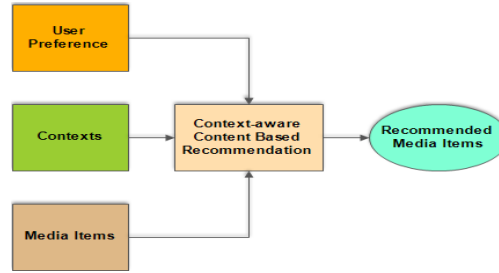


Fig. 10. Context-aware content based recommendation

-
- 1) Sense and classify target user present context
 - 2) Search target user's profile to find if his present contextual case matches any context in his profile satisfying

$$C^{opt} = \operatorname{argmax} \frac{\sum_{i=1}^n p^{n_i} \times \operatorname{sim}(C_i^l, C_i^n)}{\sum_{i=1}^w p^{n_i}}$$

- a. If it matches:
 - i. Retrieve media metadata from the web
 - ii. Generate target user contextual profile vector
 - iii. Generate media content profile vectors
 - iv. Generate similarity matrix for all candidate media contents and target user profile (equation 9)
 - v. Rank the TopN candidate contents for the target User according to the similarity values
- b. If no match is found:
 - i. Retrieve media metadata from the Internet
 - ii. Generate non-contextual user profile vector
 - iii. Generate media content profile vectors repeat a, iv and a v
- c. Present suggested items to target user
Record feedback(implicit/explicit)
- d. Log his present context and update his user profile

Fig. 11 Context-aware content based process

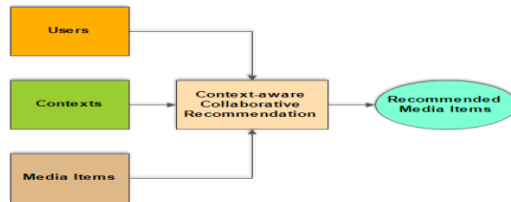


Fig. 12. Context-aware collaborative based recommendation

knowledge base. For example, if the category is news, context-aware content based or context-aware content based collaborative (which we designate as context-aware hybrid) process is used.

5.3.1. Content classification process

The classification and the identification of media items to recommend to the user rely on the contextual user profile model. We have incorporated the contextual user profile into the three traditional recommendation processes, namely, content-based, collaborative filtering, and hybrid processes [see Fig. 10, Fig.11 & Fig. 12]. One common factor in these processes is the contextual user profile, relating user preferences to the contextual information, which in turn is inferred by the context recognition model. We extended these algorithms in a way that allows the system to switch to non-contextual versions (traditional) in situation where context information is not available.

5.3.2. Context-aware content based recommendation

The traditional content based process suggests content to users based on their consumption history and the descriptions of available candidate contents (Adomavicus and Tuzhilin 2005). We have incorporated the contextual user profile model with the content based algorithm for recommending multimedia items to users based on the contexts in which they have consumed content in the past and the descriptions of those consumed contents. Fig. 10 shows the architecture of the context-aware media content based recommendation and Fig. 11 shows the simplified recommendation steps. It uses the classic cosine correlation (equation 9) to measure the similarity between the contextual user profile vector and the media item profile vectors, i.e. V_M .

-
- 1) Sense and classify target user present context
 - 2) Search active user profiles to find if target user's present contextual case matches any contextual in his profile satisfying
- $$C^{opt} = \operatorname{argmax} \frac{\sum_{i=1}^n p_{n_i} \times \operatorname{sim}(C_i^l, C_T^n)}{\sum_{i=1}^n p_{n_i}}$$
- a. If it matches:
 - i. Retrieve intensity values from all active user profiles
 - ii. Retrieve intensity values from the target user profile
 - iii. Retrieve media metadata from the Web
 - iv. Generate the similarity matrix for every active users and target user (using equation 10)
 - v. Rank the TopN active users according to their similarity to target user
 - vi. For every content consumed or similar to those consumed in this context Generate a prediction for target user and produce the topN according to the prediction (equation 11)
 - b. If no match is found:
 - i. Retrieve media metadata from the Internet
 - ii. Generate non-contextual user profile vector
 - iii. Generate media content profile vector repeat a (iv) and a (v)
 - c. Present suggested items to target user
 - d. Record feedback (implicit/explicit)
 - e. Log his present context and update his user profile
-

Fig. 13: Context aware Collaborative Process

5.3.2.1. Context-aware collaborative recommendation

Conventional collaborative recommendation suggests content to users, based on the opinions and preferences of other users (Adomavicus and Tuzhilin 2005). In the proposed solution, this process suggests content to the target user using the contexts in which active users who are similar to the target user have consumed content. Fig.12 shows the context aware collaborative recommendation architecture and Fig. 13 shows the recommendation steps. The similarity between each user profile and the target user profile is calculated using Pearson correlation equation (10). In equation (10), p stands for the intensity, u denotes candidate user and n denotes its neighbor.

$$\operatorname{Sim}(\vec{v}_c, \vec{v}_m) = \frac{V_c \cdot V_m}{V_c \times V_m} = \frac{\sum_{i=1}^n c_i \times \sum_{i=1}^n m_i}{\sqrt{\sum_{i=1}^n c_i^2} \times \sqrt{\sum_{i=1}^n m_i^2}} \quad (9)$$

$$\operatorname{Sim}(p_u, p_n) = \frac{\sum_{i \in CP_{u,n}} (p_{u_i} - \bar{p}_u)(p_{n_i} - \bar{p}_n)}{\sqrt{\sum_{i \in CP_{u,n}} (p_{u_i} - \bar{p}_u)^2} \times \sqrt{\sum_{i \in CP_{u,n}} (p_{n_i} - \bar{p}_n)^2}} \quad (10)$$

$$C_i = \bar{C} + \frac{\sum_{i \in \text{neighbors}(u)} (p_{n_i} - \bar{p}_n) \operatorname{Sim}(p_u, p_n)}{\sum_{i \in \text{neighbors}(u)} |\operatorname{Sim}(p_u, p_n)|} \quad (11)$$

In (9) $\vec{v}_c = [c_1, c_2, \dots, c_n]$; $\vec{v}_m = [m_1, m_2, \dots, m_n]$ And \vec{v}_c And \vec{v}_m are the contextual user profile and media item vectors respectively.

$CP_{u,n}$ denotes the set of matching *category-genre-property* with the same context as the target user's context. \bar{p}_u and \bar{p}_n are the mean intensities of the target user and active users who are similar to the target user.

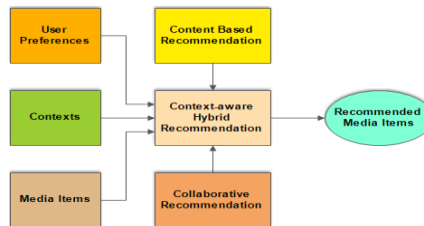


Fig. 14. Context-aware hybrid recommendation

-
- 1) Sense and classify target user present context
 - 2) Search target user's profile to find if current context matches any context in his profile
 - a. If $C^{opt} = \text{argmax} \frac{\sum_{i=1}^n p_{ni} \times \text{sim}(C_i^l, C_i^n)}{\sum_i^w p_{ni}}$:
 - i. Generate his contextual profile vector
 - ii. Generate contextual profile vector for every other user in the profile repository
 - iii. Generate all users similar to the target user(using the above contextual profile vector and Pearson correlation(equation 10))
 - iv. Retrieve media metadata from the Internet
 - v. Generate media content profile vectors
 - vi. For every similar user in(iii),generate its similarity value with every media content obtained using equation (9)
 - vii. Rank the candidate contents for each similar user formula
 - viii. Generate a preference prediction for target user for every media item using modified formula(11)
 - ix. Rank the TopN candidate content from (vii) for the target User
 - b. If no match is found:
 - i. Generate non-contextual user profile vector
 - ii. Generate non-contextual user profile vector for every user in the profile repository
 - iii. Generate all friends of the active user(using the non-contextual user profile vectors)
 - iv Repeat a(iv-x) above
 - c. Present suggested items to target user
 - d. record feedback(implicit/explicit)
 - e. Log the present context and update his user profile

Fig. 15 Context-aware content based collaborative process

5.3.2.2. Context-aware content-based collaborative recommendation

The traditional collaborative recommendations generate predictions for users, based on the item previously rated or viewed by other users, whereas content-based approach uses the consumption history of users for recommendations. These two approaches suffer from the so-called ramp-up problem, which excludes casual users or new items use from recommendations (Burke 2002). Hybrid recommenders combine one or more of the conventional recommendation processes to overcome their individual weaknesses to gain better performance. However, hybrid recommendation inherits the new user problem of both CF and CBF. Thus, to exploit the advantage of the hybrid technique, and to address the new user problem of CF and CBF, we proposed a context-aware hybrid technique whose architecture is depicted in Fig. 14, to integrate the context-aware collaborative filtering and context aware content based filtering (CBF) to address this problem. Fig. 15 shows the simplified recommendation steps. We used contexts in which other users have had media consumption previously to find users that are similar to the target user, by comparing the active users' context history and the target user's present context. This way, it addresses the new user/item problem of collaborative process and the overspecialization problem of the content based process. The context-aware hybrid recommendation can be realized in three phases. In the first phase, it identifies every user (neighbor) that is similar to the target user by searching through each user's profile tree, looking for context that matches the target user's recognized context. For every user profile with a match, the intensity value p_{ni} of the *category-genre-property* nodes in the user profiles is retrieved into a vector. The vector is then used to calculate the similarities, using equation (5), between all users. After this calculation, it then selects the *top n* most similar users, called neighbors or friends of the target user who have consumed content in the same or similar contexts to the target user's current context. In the second phase, it ranks the candidate contents for each neighbor by obtaining vectors V_C and V_M corresponding to contextual user profile of every neighbor and candidate media profile vectors respectively. By applying the cosine equation (9), we calculate the distance between these factors. In the third phase, it generates prediction value for each of the contents for the target user, using Resnick (Resnick et al. 1994) prediction equation (11). In this formula, I_i is the intensity to be predicted for each candidate content i is generated from the equation (10) for the target user and p_{ni} is the intensity(dynamic preference) of the neighbors who have consumed content i in the same contexts as the target user.

\bar{C} is the average intensity of all terms in target user's profile, whereas \bar{p}_n is the average intensity of all terms in neighbor i profile. The $\text{Sim}(p_u, p_n)$ is the similarity measure between profiles of the target user p_u and neighbor p_n , calculated using equation (11). The Resnick prediction discounts the contribution of every neighbor's prediction according to its degree of similarity with the target (Resnick et al. 1994). Finally, recommendation list is built by ordering the candidate media items in descending order of magnitude of the computed prediction values. Fig.15 summarizes the entire hybrid content classification process.

5.4. Media Item presentation adaptation

The proposed system includes an optional presentation adaptation service, which is responsible for adapting media item contents based on summarization and transcoding techniques. The summarization technique summarizes content such as audio or video into short ones, according to the battery power remaining, and network bandwidth. On the other hands, media transcoding transforms the content from one media type to another based on the network conditions such as

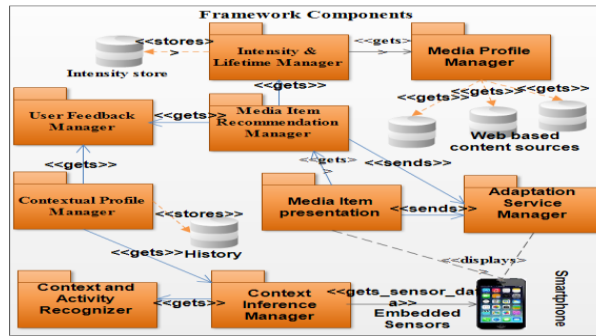


Fig. 16 Context-aware media recommendation component model

bandwidth and according to the device capabilities such as screen size or according to the type of media it is capable of playing. The presentation adaptation service incorporates *Context-Aware Ontologies for Multimedia Applications (MULTICAO)*, which was developed by Andrade et.al. (2012) to provide adaptation services. MULTICAO (<http://telecom.inescporto.pt/owl/multicaio/>) was developed using Web Ontology Language (OWL) DL with adequate provision for adaptation decisions relying on various rules based reasoning mechanisms that support the automatic inference over adaptation decision knowledge base. MULTICAO provides Adaptation Decision Engine (ADE) responsible for adaptation reasoning and decision-making services. It also provides a set of Adaptation Engine operation services responsible for executing adaptation of media content, such as modification of the content into different modalities such as text, video, or images depending on the device and network constraints (Andrade et. al. 2012).

6. Implementation and experimental evaluation

The Ana's scenario given in section 1 summarizes a case study of the processes described in this article, and has guided the implementation of the proof-of-concept that was used to evaluate the proposed solution.

6.1. Implementation

To experiment and to evaluate the proposed system, we developed a proof-of-concept system guided by the component architecture illustrated in Fig. 16. The architecture consists of eight services, representing the operations offered by the system. However, the media adaptation service is not presented in the present article, its details can be found in (Andrade et.al. 2012). The client service has been developed on top of the Android platform, version 2.3.6 Gingerbread, running on the Galaxy S I9000 smartphone and Samsung Tab GT P2600 running Android 4.1 Jelly Bean. The client takes advantage of a number of sensing APIs available in the Android platform to collect contextual data in a predetermined intervals and sending these data to the server. The Android client, thus, serves as a context service provider. It equally serves as the recommendation client service that initiates the interaction with the recommendation services, and presents the recommended content to the user. The components of Fig. 16 including the media recommendation and the contextual user profile have been implemented using Java technologies, integrating Java EE 7 (EJB, JPA) and RESTful Web Services (JavaREST 2013) and deployed on the Oracle WebLogic server (JavaEE 2013). MySQL 5.6 database serves as the backend, hosting user profiles and user context history. The contextual user profile information service registers the user's actions and performs all necessary processing such as the user profiling, and update. The recommendation generation is performed by media recommendation service, at the same time working in tandem with the context service and recommendation client service on the Android device. Fig. 17 shows the client-server implementation architecture of the case study, a context-aware media recommendation. Fig. 18 and Fig. 19 present some screenshots of the mobile client. Fig. 19 (a) shows the user profile management interface, where a mobile user can optionally manage her profile. Fig. 19 (b) is the context browser where contextual information can be managed on the device. It shows the high-level context indicating, for example, that the user is at location (*Home*), which is obtained using GPS or Wi-Fi, point of interest (POI) and proximity alert. This context information is fed into the contextual user-profiling model, which is stored in a context history repository. The process for deriving high-level context information is based on context recognition using context classification algorithms described in section 5. The visualization in Fig. 18 also shows that the user is indoors, which has been obtained using the device built-in GPS. The system intermittently checks to see if the GPS can receive signals from the GPS satellites. If no signal is received from the GPS satellite, we assume that the user is indoors. Fig. 19 (a) & Fig 19(b) present the recommendation interface showing recommended movie content. The Android client service provides a friendly interface to the user to browse and consume multimedia content through Web streaming in case of video content.

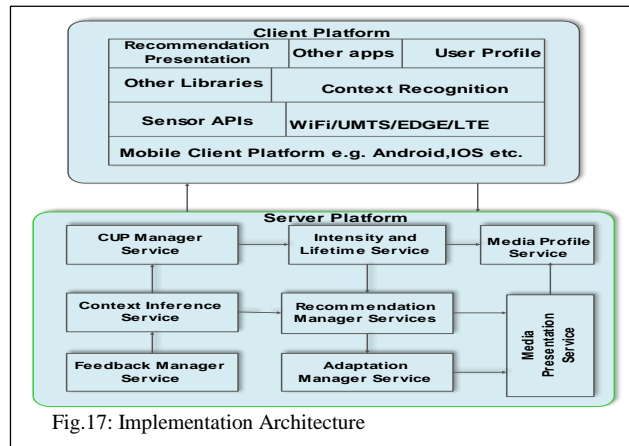


Fig.17: Implementation Architecture

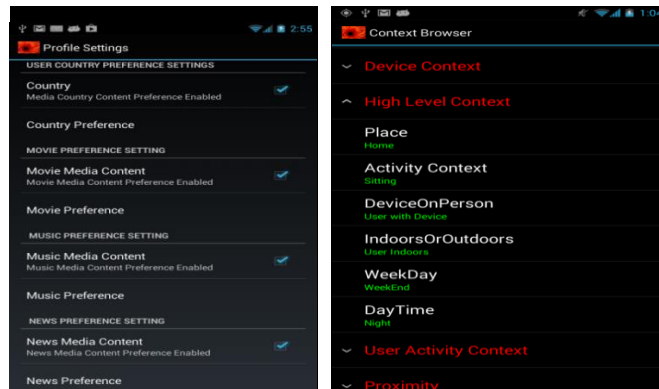


Fig. 18(a&b). User Profile and Context Management Visualization

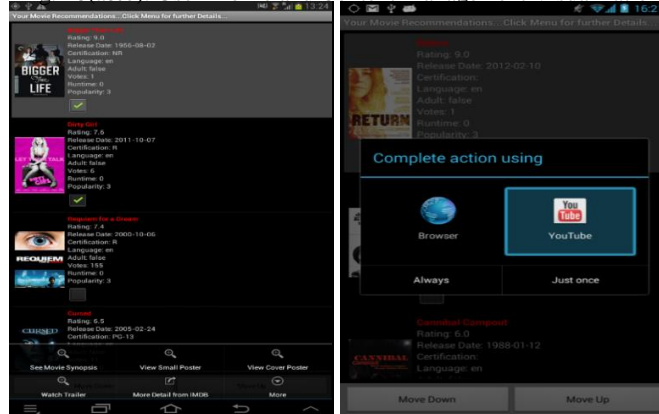


Fig. 19 (a&b). Evaluation interface and recommendation interface

6.2. Experimental Evaluation

Evaluating context-aware recommendation in mobile environments is a challenging task (Vallet et al. 2007). In the absence of a standard evaluation framework, we evaluated the proposed solution using the implemented the framework and the context-aware mobile recommendation built on top of it, and a user study based on movie recommendations. The main objectives of the empirical evaluation are: (1) to evaluate the context recognition accuracy of the underlying context recognition framework when running five machine-learning algorithms. (2) To evaluate the power consumption of the proposed system running on mobile phones. (3) To evaluate the accuracy of the proposed CBR based user profiling technique to select an appropriate contextual user preference. (4) To evaluate the prediction efficacies of the context-aware recommendation processes and compare the results with those of traditional recommendation processes. (5) To evaluate its ability and accuracy to recommend the contents to groups of users (friends) who are in a similar contextual situation. (6) To evaluate the usefulness and the effectiveness of the proposed solution via a user study, using a test panel consisting of 20 individuals. In the following sections, we present the experimental procedures for the above mentioned evaluation objectives, and the dataset, and then present the results of the experiments.

6.2.1. Context Recognition Accuracy Evaluation

In this experiment, we evaluated the recognition accuracy of the context recognition capability of the proposed system.

Experimental Data: The data collection process for the construction of the context recognition has been described in section 5.1. However, to evaluate the performance of the recognition models, we divided the dataset into two. First, activity only dataset and second, the activity with location dataset. We want to study the performance of the models, recognizing only activity, and recognizing activity with the location of users. In addition, we want to emphasize that we used data from different users to evaluate the models. This method involves using test data of the users whose data have not been used to build the recognition model. In other words, it involves training on one set of user data and testing on another unrelated set of user data. This is necessary to ensure user independence of the models (Kwapisz et al. 2010). Thus, for this type of evaluation, data from all users are used for training except a user whose data are used as test data set. This is then repeated for each user. The user independence evaluation is called leave-one-subject-out validation (Lester et.al.2006). It allows the recognition model to recognize the contexts of users whose data were not used to build the context recognition models. We computed the accuracy using precision metric, which measures how well the models are able to recognize the context information. The recognition time measures the time it takes the models to arrive at the identified high-level context information from low-level raw sensor data. Table 2 shows the accuracy and time for activity + location and activity only respectively.

Classifier	Activity Context		Activity +Location	
	Accuracy	Times(s)	Accuracy	Times(s)
SVM	70.88	10.16	57.01	3.9
MLP	97.99	96.47	84.76	64.89
KNN	98.47	0.45	90.99	0.39
C4.5	97.28	0.65	83.94	0.66
BayesNet	94.49	0.4	74.42	0.27

Table 3 context recognition accuracy

6.2.2. Power consumption evaluation:

Since the proposed solution regularly monitors sensor events to recognize user contexts and activities, battery drain becomes a very critical issue for the smartphone users. GPS, WiFi or cellular network, accelerometer, orientation, rotations, and other sensors have a significant impact on the battery lifetime. Therefore, to evaluate the impact of the context recognition service on the battery lifetime, the context recognition service was deployed on a Samsung Galaxy S I9000 smartphone as the only active application. We then evaluated the system in eight situations. These situations and the percentages of power consumed are shown in Table 4. The table shows that the smartphone screen consumes a lot of energy in the active state. However, our context recognition service does not need the device screen to be in an active state to operate. Therefore, the context recognition service has been implemented as a background service. This is an important energy optimization consideration in our implementation. In addition, the context recognition service does not have to continuously collect data from all sensors at every second because, for example, we can use a motion sensor such as accelerometers or orientation sensor to accurately recognize a user's activity. We have established this in our previous article where we evaluated the accuracies of the recognition model when each motion sensor is used separately or in combination for context recognition (Otebolaku & Andrade 2013). Therefore, based on this finding and assumption, we implemented another power optimization. The recognition service at every 30 and 60 seconds switches on and off some sensors to conserve power. With screen off, and with all sensors on, the power consumption percentage dropped from 40.66% to 20.84%. In addition, with the periodic switching of sensors, we were able to reduce the power consumption by almost 12% for 30s and about 13% for 60s. Similarly, we optimized the power consumption by switching on and off the GPS sensor, and using either WiFi or Cell-ID to sense user location. We observed that leaving the GPS in active mode results in a power consumption surge. It is therefore important to note, from this result, that the proposed system will only use GPS as location sensors only in situations where the user is outdoors, but when indoors, we use Wi-Fi or Cell-ID in order conserve power.

6.2.3. Context-aware multimedia content classification evaluation

Experimental Data: Evaluating contextual recommendation systems is a difficult and expensive task (Vallet et al. 2007). Issues of privacy, safety, etc. are involved, and it requires a significant amount of feedbacks from users (Vallet et al. 2007). Many users are not willing to compromise this information. Nevertheless, to test the feasibility of our system, we have conducted experiments, using anonymous survey data solicited from online users mainly from our faculty. Specific media content terms have been included. In the survey, users were asked to associate media content terms with location, time, and activity characterizing a situation where they prefer to consume content with such terms. Using these data, we built the contextual user profiles, based on our contextual user profile model. We created over 200 different user profiles each having 3 entries in the category level (Movie, Music and News), 19 different entries in the genre level (the entry in the category level is the same for all users – movies) and various properties at the property level. High-level contexts, as presented in Table 1, have been associated with these entries. To obtain content dataset, we obtained online movie dataset by crawling over 4500 movie metadata records from the Movie Database (themoviedb.org), further enhanced with additional metadata retrieved from the IMDB (<http://www.imdb.com/>).

	Application	Percentage (%) of Power Consumed
	All sensors+ Screen	40.66
	All sensors - Screen	20.84
2	Periodic 30 s + GPS+WiFi+all	8.90
3	Periodic 60 s + GPS+WiFi+all	7.35
4	GPS + WiFi+Accelerometer	5.60
5	GPS+WiFi+Orientation+other	10.78
6	GPS+WiFi+rotation+other	10.40
7	WiFi +other sensors-GPS	7.55
8	GPS +other sensors -WiFi	8.5%

Table 4 Power consumption evaluations of the system

This metadata set contains 23 different movie genres and each record contains an average of three different genre labels. Language, cast, country, duration, and release date characterize the genres. These terms, thus were processed to constitute the media item's context in our user profile model.

Experimental procedure: To evaluate the quality of the recommendations, given that the 200 users were anonymous and thus most of them were not available to provide continuous on-device feedback, we devised an approach to allow marking recommended items as *relevant* or as *irrelevant*. This allowed us to simulate the acceptance or rejection of suggested content by the users as shown in Fig. 19 (a). In the evaluation process, an item is marked as relevant provided at least 2/3 of the terms that appear in its metadata record, also appear in the user profile with a weight larger than the average intensity of all terms in the user profile. For example, a suggested movie item with a metadata record presenting three terms is marked as relevant if two of those three terms appear in the user profile with intensities larger than the average intensity of all terms. Otherwise, it is marked as irrelevant. We adopted this approach because we observed in our preliminary experiments that the classification of candidate items is influenced by the number of terms contained in their metadata records, particularly in the *genre-property* nodes.

1) CBR user profiling process accuracy evaluation

To evaluate the impact of CBR based user profiling technique in identifying similarity between contextual preference cases, we simulated the recommendation process, while varying the values of α_i in equation (6) in the range [0,1]. The metrics used to obtain the precision values are:

Automatically and Correctly Identified Similar Contextual Preference Cases (ACISCPC) and the Total Number of Similar Contextual Preference Cases (TNSCPC). Therefore, the precision is calculated as follows:

$$\text{Precision} = \frac{\text{ACISCPC}}{\text{TNSCPC}} \quad (12)$$

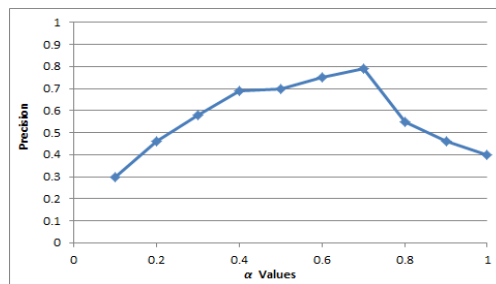


Fig. 20 Impact of contextual preference similarity on recommendation precision

Fig. 20 shows the curve of the precisions against the values of α , depicting the impact of α on the precision of the recommendation process. The best precision was obtained at $\alpha = 0.7$ and with the highest precision of 0.79. At $\alpha > 0.7$, the precision begins to fall, which shows that those values have a negative impact on the accuracy of the recommendation precision. This simulation helps to peg the value of α at 0.7 in all subsequent evaluations of the recommendation process quality.

2) The second evaluation involves user profiles with specific content associated to a category-genre-property-context node in the profile as explained in section 5.2. This is the evaluation of the contextual recommendation efficacies. This involves users who have consumed the contents at specific contextual situations. In practice, this translates to using a large variance for setting bigger weights for preferences assigned to genres and properties in specific contexts, whereas others are assigned smaller weights or even zero since the user does not prefer content with such genres in that specific

context. To evaluate its performance, we calculated the precision, which is the ratio of the amount of recommended content in which the user was interested in a given context to the amount of content recommended.

3) Additionally, unlike the evaluation in (2), we evaluated situation where content is only associated with the category-genre-property, i.e. the non-contextual version of the processes. We wanted to compare the result of both situations to observe if context indeed can enhance recommendation quality, i.e., if context can assist recommendation systems to deliver more relevant items. Fig.21 & Fig. 22 are the precision plots comparing the contextual (in 2) and non-contextual content based, collaborative filtering algorithms respectively, representing 5 target users with $topN$ recommendations (Deshphande and Karypis 2004), $n = 5, 10, 15, 20, 25$. Fig.23 is the precision plot for the hybrid recommendations.

4) Performance context based group recommendation

To understand the ability of the proposed system to recommend media items to users who consume similar items in similar context, we evaluated the precision of the recommendation process. The first step in the group recommendation process is to use cases of users who have consumed similar items in similar contexts. Using the 200 user profiles in the profile repository, these individuals were divided into five groups, each group consisting 40 members on the basis of the similarity generated. We then issue recommendations based on the context-aware algorithm and the traditional algorithms to a user representative of each group. The recommendation is repeated for $n = 5, 10, 15, 20$, and 25, which represent n items with highest predicted preferences. Fig. 24 is the precision plot showing the recommendation quality for each group.

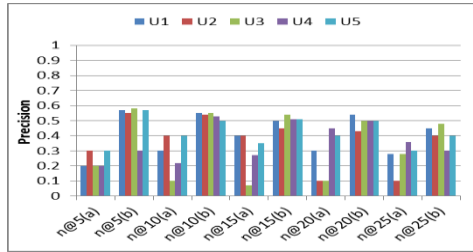


Fig.21 (a) Content-based and (b) context-aware content based recommendation precisions

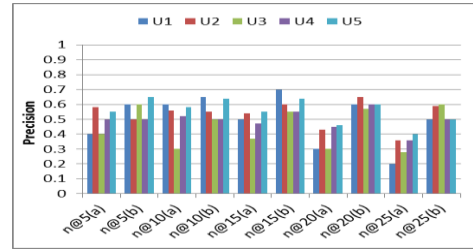


Fig.22 (a) Collaborative based and (b) Context-aware collaborative-based recommendation precisions

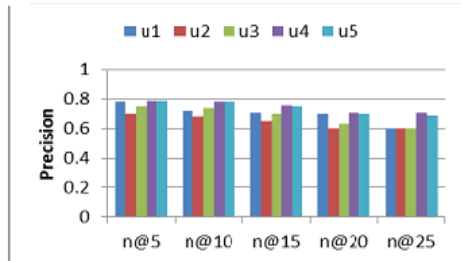


Fig. 23 Context-aware content based collaborative recommendation

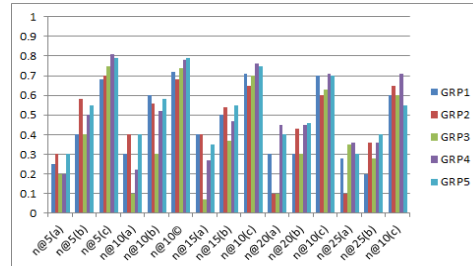


Fig. 24 Group based recommendation precision

5) User Satisfaction

We evaluated user satisfaction of the system to ascertain its usefulness and effectiveness in real life scenario. However, only context-aware hybrid recommendation was evaluated in this situation. In the experiment, 20 individuals who were representatives of the system's target users constituted the test panel, whereas 200 user profiles constituted the active contextual profiles in the knowledge base. Using the online-based movie metadata that we crawled, we generated 10 recommendations for each test user, using the case study context-aware mobile recommendation application. We asked the test users to evaluate the degree of satisfaction of the recommendations on a 5-point scale, where "5", the maximum point, means totally *satisfied* and "1", the minimum point, means *totally not satisfied*. We have analyzed the feedback provided by these users. At every 10-time point, we collected all the scores and then calculated the average to obtain the overall satisfaction from the user's perspective. The overall satisfaction obtained was 70%. Some of the users were excited about the kind of recommendations they received. For example, a test user exclaimed that there is a particular movie she would "*see over and over again*", which the application recommended for her. However, some users expressed dissatisfaction with the recommendations, whereas others were indifferent, questioning the need for such an application since they could have easily selected any content they needed, using the content stored in their mobile phones.

7. Result and discussion

The experimentation in the last section shows promising results. We evaluated the traditional algorithms to compare their performances with context-based algorithms. Generally, and as expected, the context-based algorithms have better

recommendation efficacies than the traditional algorithms [Fig. 21-24]. The traditional algorithms show situations where recommendations without contexts would produce poor recommendation efficacies and vice versa. The reason is that in the filtering process, user preferences are not filtered according to the contexts where users have expressed such preferences. Therefore, more irrelevant contents have been included in the recommendations. For example, in Fig. 21 and Fig.22, precisions e.g. $n@15$ (a) for target user 3 are as low as 0.07 and 0.37, whereas context-based algorithms consistently produced better precision results.

The same precisions $n@15$ increased to 0.54 and 0.55 respectively. The reason is that the system uses context information to filter out irrelevant contents, thereby producing better recommendations. For example, if user 3 prefers watching erotic movies while at home alone and if she likes to watch action movies with her friend at the cinema, it is then normal for the system to filter out preferences that relate to watching action movies when the location context is “home” or relating to her watching an erotic movie when the location is “cinema” and the companion is “friends”. The hybrid recommendation approach that combines contextual content and collaborative algorithms produced the overall best results, the same precision in those algorithms increased to 0.7, whilst achieving highest precision of 0.79 @n5. In addition, the contextual hybrid recommendation does not suffer from new item or new user problems because it is the user’s contexts, and not ratings, that drive the recommendations.

One important result is that the contextual hybrid algorithm produced better performance than context-aware content-based or context-aware collaborative algorithm. This confirmed the social and personalization capability of contextual information on user’s content consumption preference. Additionally, the satisfaction obtained from users shows that the context-aware hybrid recommendation is feasible in practice. For example, unlike the work reported by Yu et al. (2006) Pessemier et.al. (2013), we have been able to establish the performance of the proposed system in terms of prediction precisions for the three algorithms implemented as web services. The precisions demonstrate the superior efficacies of context-aware recommendations to traditional recommendations.

The evaluation of the context recognition shows promising results. The context recognition time is encouraging with BayeNet and Random forest (RF) being the fastest of the five models. In terms recognition quality, RF has the highest accuracy, though the accuracies of other models are better than average.

In terms of power consumption, the context recognition service performed fairly well, though the performance must be improved via additional optimization. Nevertheless, we established that running the recognition service continuously with all sensors in active state would drain the battery quicker than if it was left to switch off and on some sensors at intervals. For example, we established that continuous monitoring of the GPS is a key power consumer. Therefore, we have redesigned the context recognition to monitor Wi-Fi or Cell-ID as the primary source of location information, but to use GPS only when user is outdoors.

8. Conclusion

In this article we investigated how user’s contexts and activities can be obtained dynamically from user’s smartphones sensors and how this recognized context information can be used to enhance recommendation efficacies via a contextual user profile model. We have incorporated the contextual user profile model, which uses case based reasoning process and a context recognition process, with traditional CF and CBF. We developed a mobile movie recommendation system on top of the proposed system as a proof-of-concept, which was used to evaluate the proposed solution.

The results showed how contextual information can influence the recommendation efficacies, helping to filter out preferences that are not relevant to the target user in the present context during filtering process. One distinctive feature of the proposed system is its ability to combine dynamic context recognition and recommendation algorithms, integrated with a common contextual user profile model. The contextual user profile model, using case base reasoning to process user’s contextual preferences, has a wide scope, and can be extended to cater for context-aware filtering processes. The context service relies on the context recognition model to glean accurate context information from mobile device built-in sensor events, using machine-learning technique. We take advantage of the smartphone built-in sensors to enhance seamless and automatic recognition of the user context information in real-time without using additional wired devices. The recommendation system can filter out irrelevant recommendations using user’s current context, unlike the traditional algorithms that use both relevant and irrelevant preferences for content classification. The initial user satisfaction evaluation demonstrates that the proposed solution can provide users with contextually relevant recommendations.

In the future, we intend to carry out extensive usability testing of the system. Also, in terms of power consumption, we need to implement optimization mechanism to reduce the present power consumption of the proposed system. The present version of the proposed solution uses case base reasoning to infer user’s current contextual preferences. We will explore, in the future, other elegant approaches for dynamic determination of user’s contextual preferences, especially Multi Attribute Decision Making (MADM) based methods, which have been proven as efficient and less computationally expensive methods in dynamic environments (TalebiFard & Leung, 2011). We would explore and compare the performance of a MADM based method with the performance of the present CBR based approach implemented in the current version of our system.

Finally, the present version of the evaluations explored only online movie data. In the future, we plan to use other online multimedia content such as music and news to evaluate our proposed system.

Acknowledgments This work was partially financed by the Portuguese Foundation for Science and Technology (FCT) with grant SFRH / BD / 69517 / 2010.

References

- Aamodt E, Plaza E(1994) Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. In: AI Communications 7(1):39-52.
- Adomavicius G, Tuzhilin A (2011) Context-aware recommender systems (2011) In: Ricci F, Rokach L, Shapira B, Kantor P (eds) Recommender systems handbook. Springer, Berlin, pp 217-256.
- Adomavicius G, Tuzhilin A(2005) Towards the next Generation of Recommender Systems: A survey of the State-of-the-art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 17 (6), 200, 734-749.and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on , vol., no., pp.129,134, 21-25 March 2011 doi: 10.1109/PERCOMW.2011.5766854
- Andrade MT, Dogan S, Carreras A, Barbosa V, Arachchi HK, Delgado J, Kondo AM(2012) Advanced Delivery of Sensitive Multimedia Content for better serving User Expectations in Virtual Collaboration Applications". Multimedia Tools Applications. 58(3): pp. 633-661.
- Benitez AB, Zhong D, Chang SF, Smith JR(2001) MPEG-7 MDS Content Description Tools and Applications. W. Skarbek (Ed.): Computer Analysis of Images and Patterns, Lecture Notes in Computer Science 2124, Springer, (2001).
- Bobadilla J, Ortega F, Hernando A, Gutierrez A(2013) Recommender systems survey. Knowledge-Based Systems, 46 (0), pp. 109–132.
- Bouidghaghen O, Tamine L, Boughanem M (2011) Context-Aware User's Interests for Personalizing Mobile Search. 12th IEEE International, DOI 10.1109/MDM.2011.51.
- Burke R(2002) Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, p.331-370 [doi>10.1023/A: 1021240730564.
- Chen A (2005) Context-Aware Collaborative Filtering System: Predicting the User's Preference in the Ubiquitous Computing Environment. In International Workshop on Location- and Context-Awareness, pp. 244–253. Oberpfaffenhofen,Germany.Conference on Mobile Data Management (MDM), pp.129,134, 6-9 June 2011.
- Costa A, Guizzardi R, Filho J (2007), CORES: Context-aware, Ontology-Based Recommender System for Service Recommendation. In Proceedings of the 19th international conference on advanced information systems engineering (CAiSE'07), Trondheim, Norway, pp 11-15.
- Deshpande M, Karypis G (2004) Item-based top-N recommendation algorithms. ACM Transactions on Information Systems 22(1), 143–177.
- Dey A K and Abowd G D(2000)The Context Toolkit: Aiding the Development of Context-Aware Applications, Proc. Workshop Software Eng. for Wearable and Pervasive Computing, ACM Press, New York, pp. 434-441.
- Dey A, Abowd G and Salber D (2001) A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Hum-Comp Interact 16(2–4):97–166
- Hong et al. (2009) Hong J E-H, Suh J, Kiim, Kim S(2009) Context-aware system for proactive personalized service based on context history Expert Systems with Applications, 36(2009) pp. 7448–7457.
- Java EE: <http://www.oracle.com/technetwork/java/javaee/overview/index.html>[Accessed in October,2013].
- Java REST: Java RESTful Web Services.<http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html> [Accessed in October,2013].
- Kwapisz J, Weiss G, Moor S (2010) Activity Recognition using Cell Phone Accelerometers. ACM SIGKDD Explorations Newsletter, 12 (2), 74-82.
- Lara OD, Labrador MA(2013) A Survey on Human Activity Recognition using Wearable Sensors, Communications Surveys & Tutorials. IEEE Vol. 15, no. 3, pp. 1192,1209, Third Quarter 2013, doi: 10.1109/SURV.2012.110112.00192.
- Lester J, Choudhury T, Borriello G.(2006) A practical approach to recognizing physical activities. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) PERVASIVE 2006. LNCS, vol. 3968, Springer, Heidelberg.
- Liu D, Meng X, Chen JL (2008) A Framework for Context-Aware Service Recommendation. Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on , Vol. 3, no., pp. 2131,2134, 17-20.
- Meehan K, Lunney T, Curran K, McCaughey A (2013) Context-aware intelligent recommendation system for tourism. Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on, vol., no., pp. 328,331, 18-22 March 2013.
- Millette G and Stroud A (2012) Professional Android Sensor Programming, 1st Edition, John Willey and Sons Inc.
- Mobasher B (2010) Contextual user modeling for Recommendation (2010), Keynote at the 2nd Workshop on Context-Aware Recommender Systems.
- Ostuni CV, Gentile G, Noia TD, Mirizzini R, Remito D, Sciascio ED (2013) Mobile Movie Recommendation with Linked Data. LNCS Vol.8127, pp. 400-415.
- Otebolaku AM, Andrade MT (2011), Context Representation for Context-Aware Mobile Multimedia Recommendation. In Proceedings of the 15th IASTED International Conference on Internet and Multimedia Systems and Applications, Washington, USA.
- Otebolaku AM, Andrade MT(2013) Recognizing High-Level Contexts from Smartphone Built-In Sensors for Mobile Media Content Recommendation. Mobile Data Management (MDM), 2013 IEEE 14th International Conference on, Vol. 2, no., pp. 142,147, 3-6 June 2013.
- Pessemer TD, Dooms S, Martens L (2013) Context-aware recommendation through context and activity recognition in a mobile environment. Multimed Tools Appl. Do: 10.1007/s11042-013-1582-x.
- Pessemer TD, Deryckere T, and Martens L (2009) Context-Aware Recommendations for User-Generated Content on a Social Network Site. In Proceedings of the EuroITV '09 Conference, pp 133–136, New York, USA.

- Resnick P, Neophytos P, Mitesh S, Bergstrom P, Riedl J (1994) GroupLens: An open architecture for collaborative filtering of netnews. In Proceedings of ACM CSCW'94 Conference on Computer Supported Cooperative Work, Sharing Information and Creating Meaning, pages, 175–186.
- Setten V, Pokraev M, Koolwaaij S(2004) Context-Aware Recommendations in the Mobile Tourist Application. Nejdil, W.& De Bra, P. (Eds), LNCS 3137, Springer-Verlag.
- Steiger O, Ebrahmi T, Marimon D (2003), MPEG-based Personalized Content Delivery. Proceedings of the 2003 International Conference on Image Processing(ICIP'03), Vol. 3, pp 45-48, Barcelona, Spain.
- TalebiFard, P., & Leung, V. C. (2011, April). A dynamic context-aware access network selection for handover in heterogeneous network environments. In Computer Communications Workshops (INFOCOM WKSHP), 2011 IEEE Conference on (pp. 385-390). IEEE.
- Tseng BL, Lin C,Smith JR (2004) Using MPEG-7 and MPEG-21 for Personalizing Video. IEEE Multimedia, Vol.11, Issue 1, pp. 42-52
- Vallet D, Fernandez M , Castells P, Mylonas P, Avrithis Y(2007) Personalized information retrieval in context. 3rd Int.Workshop Modeling Retrieval Context 21st Nat. Conf. Artif. Intell., 2007.
- Vetro A (2004) MPEG-21 Digital Item Adaptation: Enabling Universal Multimedia Access. IEEE Multimedia Vol.11, No1, pp. 84- 87
- Wang X, Rosenblum D, Wang Y(2012) Context-aware mobile music recommendation for daily activities. Proceedings of the 20th ACM international conference on Multimedia, October 29-November 02, 2012, Nara,Japan.
- Wang XH, Zhang D, GU T, Pung HK (2004) Ontology Based Context Modeling and Reasoning using OWL. In Proceedings of CoMoRea, the Second IEEE International Conference on Pervasive Computing and Communications (PerCom 2004), Orlando, Florida USA, March 2004.
- Xia F, Asabere NY, Ahmed AM, Li J, Kong X(2013), Mobile Multimedia Recommendation in Smart Communities: A Survey. IEEE , vol.1, no., pp.606-624.
- Yin W, Zhu X, Wen Chen C W (2011) Contemporary ubiquitous media services: Content recommendation and adaptation. IEEE Internation Conference on Pervasive Computing Workshop,March 2013, pp 129-134.
- Yu Z, Zhou X, Zhang D, Chin CY, Wang X, and J. Men J (2006) Supporting Context-Aware Media Recommendations for Smart Phone.IEEE Pervasive Computing, Vol.5, No.3, pp.68-75.
- Yujie Z, Licai W (2010) Some challenges for context-aware recommender systems. Computer Science and Education (ICCSE), 2010 5th International Conference on computer science and education. Pp.24-27, Aug. 2010
- Zhang W, Lau R, Tao X(2012) Mining Contextual Knowledge for Context-Aware Recommender Systems. e-Business Engineering (ICEBE), 2012 IEEE Ninth International Conference on, vol., no., pp.356,360, 9-11 Sept. 2012.