Privacy-preserving recommendations in context-aware mobile environments

Abstract

Purpose – Mobile recommender systems aim to solve the information overload problem by recommending products or services to users of web services on mobile devices, such as smartphones or tablets, at any given point in time and in any possible location. They utilize recommendation methods, such as collaborative filtering or content-based filtering and use a considerable amount of contextual information in order to provide relevant recommendations. However due to privacy concerns users are not willing to provide the required personal information that would allow their views to be recorded and make these systems usable.

Design/methodology/approach – This work is focused on user privacy by providing a method for context privacy-preservation and privacy protection at user interface level. Thus, a set of algorithms that are part of the method have been designed with privacy protection in mind, which is done by using realistic dummy parameter creation. To demonstrate the applicability of the method, a relevant context-aware dataset has been used to run performance and usability tests.

Findings – The proposed method has been experimentally evaluated using performance and usability evaluation tests and is shown that with a small decrease in terms of performance user privacy can be protected.

Originality/value – This is a novel research paper that proposes a method for protecting the privacy of mobile recommender systems users when context parameters are used.

Keywords Mobile recommender systems, Context-awareness, Privacy, Dummy-based, User interface

Paper type Research paper

1. Introduction

The evolution of the concept of e-democracy and its related electronic government services has led to information overload and privacy issues (Drogkaris et al., 2013). Moreover, the information overload problem encountered in numerous online systems / services such as e-commerce and e-government, among others, has given rise to the use of recommender systems (Lu et al., 2015). Such systems have swiftly become necessary to the wider public but at the same time have contributed heavily to an increase in privacy concerns amongst service users. Recommender systems are algorithms and computer software that are designed to provide suggestions for products or services that could be of interest to a user of a website or an online application (Bobadilla et al., 2013; Konstan and Riedl, 2012). They are considered to be a subset of information retrieval systems whose job is to provide personalized recommendations to users and solve the information overload problem found in various online environments. Recommender systems are valuable to users that do not have the experience or the time to cope with the process of decision making while using the web, particularly where a choice of products or services is available.

Recent advances in the field of mobile computing and the rapid evolution of mobile devices such as smartphones and tablets, has led to the need for advances in the field of mobile recommender systems (Cao et al., 2014; Ahluwalia et al., 2014; Polatidis and Georgiadis, 2015; Ricci, 2010; del Carmen Rodríguez-Hernández and Ilarri, 2016). The access to a mobile recommender system at any given point in time and location is called ubiquity, thus an alternative term used to describe such systems is that of ubiquitous recommender systems (Mettouris and Papadopoulos 2014). Additionally, the use of location data and the use of other contextual information, such as the time, weather information, physical conditions, social and others, have become common in mobile recommender systems (Mettouris and Papadopoulos, 2014).

These new uses of data have contributed to the creation of more personalized recommendations in mobile environments. It should be noted though that it is not clear whether a specific research domain of mobile recommender systems exists and that only specific goals are set for mobile recommendations, where a mobile application or mobile website is designed and developed for a specific scenario (Jannach et al., 2010; Polatidis et al., 2015; Ricci, 2010). In this context different application domains exist, such as those for mobile commerce and tourism related services (Jannach et al., 2010; Ricci, 2010). Applications designed for any mobile recommendation domain share some common characteristics such as (Jannach et al., 2010): All run on a mobile device, such as a smartphone or a tablet, all provide some form of recommendation, all utilize some form of context and all rely on a wireless connection that could probably be slow.

Mobile context-aware recommender systems heavily rely on context to provide accurate and personalized recommendations in mobile environments. However typical privacy protection techniques such as the use of pseudonyms or the use of anonymity cannot be applied properly due to the fact that recommender systems rely on the use of personal contextual data. In such a context –Kido et al. (2005) proposed an approach to anonymous communication for location-based services that is based on use of dummies. Similar methods that have been used for privacy protection in location-based services include query enlargement techniques, progressive retrieval techniques and transformation based techniques (Jensen et al., 2009). These are different protection methods that can be adjusted to the context privacy problem found in mobile recommender systems. Privacy is an important part of context-aware mobile recommender systems that has not been properly exploited yet and, to the best of our knowledge, this is the first effort found in the literature to do that. Furthermore, the constant growth of available wireless technologies gives the ability to users to be connected to the Internet from virtually any place and at any given time. Thus, privacy concerns become higher when users want to submit a rating or retrieve recommendations and are located in a public place (or, in a different situation, located somewhere private but with friends or family near them). It must be noted that many users are both busy and unsatisfactorily proficient technically, to watch out for themselves. Consequently, thinking in advance about privacy can help both designers and users (Camp, 2015). Figure 1 shows a typical mobile recommender system. In this scenario, the context is acquired first from mobile device sensors and/or third parties (providing users' profiles and ratings) and then a recommendation method is used to provide recommendations.

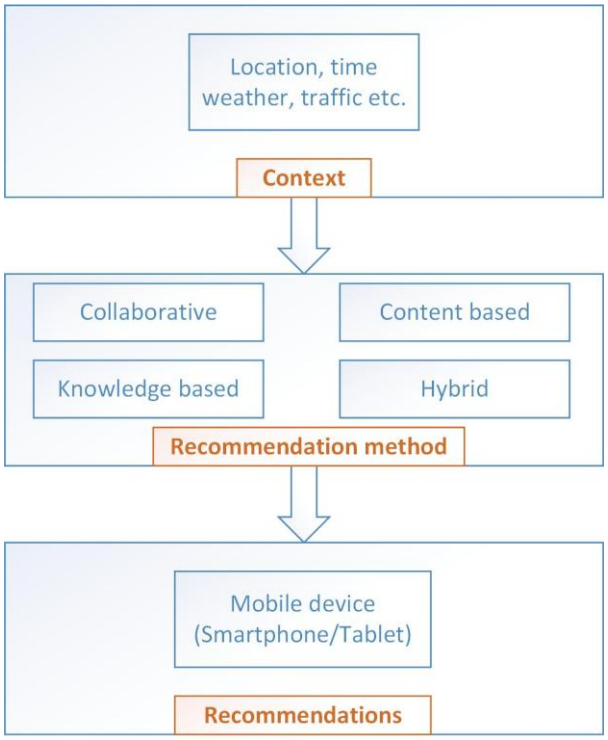


Figure 1. A typical mobile recommender system

In order to protect the user privacy at the context level the following contributions have been made:

- We have developed a method that aims to protect the user privacy in mobile context-aware recommender systems.
- It introduces an approach for privacy-preserving context-aware mobile recommendations that is based on realistic dummy context parameter creation.
- Developed a privacy-friendly user interface for mobile context-aware recommender systems.
- Experimentally evaluated the method, showing that at the expense of a small performance decrease user privacy can be fully protected.

The rest of the paper is organized as follows: Section 2 overviews the factors affecting mobile recommender systems, Section 3 delivers a motivating scenario, Section 4 is the related work, Section 5 describes the proposed methods, Section 6 explains the experimental evaluation and Section 7 contains the conclusions and future work part.

2. Factors Affecting Mobile Recommender Systems

A number of factors exist that can affect mobile recommender systems and their ability to provide accurate personalized recommendations. These include the recommendation method, the context and privacy concerns.

2.1 Recommendation method

Recommender systems rely in some form of recommendation method to suggest the appropriate products or services to the user. The most important recommendation methods include (Bobadilla et al., 2013):

Collaborative filtering which is a method that recommends items to users that other users with similar ratings have liked them in the past. This works by asking each user to submit ratings for products or services and then searches between the ratings for similar users and provides the recommendations (Shi et al., 2014; Bobadilla et al., 2013). Content-based filtering which is a method that uses a set of keywords supplied by the user that can be matched in the item's description (Bobadilla et al., 2013; Konstan and Riedl, 2012). Finally, hybrid is a method that uses a combination of two or more recommendations methods (Bobadilla et al., 2013; Konstan and Riedl, 2012).

2.2 Context

Context is utilized by mobile recommender systems to provide more accurate and personalized recommendations. It is a type of data that is necessary to users that move constantly and their status changes. Different types of context can be employed in mobile scenarios and include, among others, location, time, weather and social presence. Contextual information is important for location-based recommendations (Adomavicius and Tuzhilin, 2011; Ricci, 2010; Liu et al., 2013). Information can be collected either explicitly, by asking the user to provide data, or implicitly by collecting data from the mobile device and related sensors, such as the Global Positioning System (Liu et al., 2013).

The context can be applied by using three different ways (Adomavicius and Tuzhlin, 2011): First, Contextual pre-filtering is a method where the contextual data is used to filter out irrelevant data from the dataset and then apply the recommendation method. Also, Contextual post-filtering is a method where the recommendation method takes places and then the irrelevant data are filtered out. Finally, Contextual modeling is a method where the recommendation method is designed in a way that the context is utilized within.

2.3 Privacy

Mobile recommender systems offer the benefit of providing personalized recommendations to users of a context that constantly changes. On the other hand, the ways that user data might be processed direct users towards a negative attitude, when it comes to supplying personal contextual information (Liu et al., 2013; Mettouris and Papadopoulos, 2014). Privacy protection techniques have relied mainly on location-based services (Scipioni, 2011) and do not take into consideration the whole concept of context. Privacy is an important factor that can be addressed properly using the right methods and makes it possible for the user to supply the required contextual information, thus making both the system usable and receive highly accurate personalized recommendations.

3. Motivating Scenario

This section describes a motivating scenario that illustrates the necessity for a privacy-preserving context-aware mobile recommendation architecture. The scenario shows the main benefits that a user can gain if a mobile recommender system is used and that a privacy-preserving system is necessary to assist the user in the process of gaining those benefits. We follow an example from a fictional user to describe the motivating scenario and we also assume the existence of a mobile application, MobiRec (Mobile Recommender), which can be installed in mobile devices, such as smartphones or tablets. This application recommends movies of interests to the user, considering past common ratings between users and available context parameters.

3.1 Example scenario

Bob is at home at 7:30pm. It is Saturday and the weather is rainy. Bob is relaxing with some of his friends while they are deciding to watch a movie. Bob then chooses to use MobiRec to assist him with finding a relevant movie to watch with his friends. He opens the applications and selects the option of receiving recommendations of movies to his screen. Automatically, MobiRec enriches the input with available

contextual information such as the hour of day, the location, company and weather. The mobile recommender then communicates with the central database where the ratings of users about movies are stored, passes the contextual information to the server so the most relevant context-aware recommendations are provided. At this moment the server is aware about private user information, which somehow need to be protected from unauthorized use. Thus, MobiRec uses the method described in section 5 to create a set of dummy parameters that are passed to the server among the real parameters. MobiRec also has the option for extra privacy under its settings by using the privacy-aware interface, which in this cases utilizes the context to see if the user is alone or with company in order to provide a warning saying that other people might have a look at the recommendations and if with the press of a button the list of the recommendations is released to the screen.

Then, at the end of the movie Bob is asked by MobiRec to actually rate the movie that he just watched, so better recommendations can be provided in the future. The appropriate user interface pop-ups in the screen where Bob can select a numerical value. However, if the extra privacy is selected under the parameters of the mobile recommender application then a different rating submission policy applies. Privacy can be threatened from nearby people staring at the screen of the mobile device. To avoid any breach of privacy the entered value of the ratings is manipulated with the method described in section 5. Now, Bob can submit a rating freely with him only knowing the real value passed to the server and while his friends are watching.

4. Related Work

A large number of different works exist that demonstrate the importance of privacy issues in different parts of the recommendations process (from the client part to the server part) in various domains. Mobile recommender systems are used in a variety of different domains, such as the one presented in Anacleto et al. (2014), where a mobile application is used to provide personalized sightseeing tours to its users. Another mobile application that is designed to provide context-aware tourism information to its user has been provided in Noguera et al. (2012). Colombo-Mendoza et al. (2015) proposed a context-aware, knowledge-based, mobile recommendation system for movie show times. Privacy is indeed an essential part of mobile recommender systems (del Carmen Rodríguez-Hernández & Ilarri, 2016; Polatidis et al., 2015; Liu et al., 2013). However, there is a gap between recommender systems and mobile computing. For example most of the related work is about the protection of personal user data, such as the ratings. Furthermore, protecting the user location is considered an important aspect in mobile computing services. Thus, in our proposed method we aim to protect user privacy when contextual parameters are used in order to provide personalized recommendations in mobile environments. Moreover, our method also protects the privacy at the user interface level. Most of the privacy-protection methods for location-based services perform well for protecting the location of a user when only non-personalized services are requested, our approach delivers recommendations in mobile environments without losing any accuracy and at the same time preserving the privacy of the user at the context and interface level. An example of preserving privacy in collaborative filtering is the use of distribution techniques that use an obfuscation scheme and a randomized dissemination protocol (Boutet et al., 2015). Also, the use of ratings perturbations is a well-known approach that is used in collaborative filtering for personal data protection (Polat and Du, 2005). Additionally, other privacy protection approaches exist in recommender systems such as Aïmeur et al. (2008) where the use of a semi-trusted third party is proposed. The data are split between the server and the semi-trusted third party, thus no single entity can derive sensitive information and the system can work only if these two separate parties collaborate.

Moreover, approaches in location based services have been developed to protect the location of the user but no other context parameters and this is done for non-personalized services. For example, in Kido et al. (2005) a technique that is used for location privacy based on dummies is described. Similarly, in Lu et al. (2008) an approach to location privacy is proposed and generates dummy locations based on a virtual grid or a circle. The approach requires only a lightweight front-end that can work tightly in a client-server

model. Furthermore, in Kato et al. (2012) a dummy generation algorithm is proposed in order to protect location privacy. In this approach various restrictions are taking place and assume that users do not stop regularly. In Niu et al. (2014) two dummy based solutions are proposed to achieve *k*-anonymity for privacy-aware users. Also, in Tran et al. (2010) a binomial mix-based solution is proposed which aims to protect privacy by using a centralized dummy generation mechanism that exploits the activities of each user to perform better in overall. In Jensen et al. (2009) a number of different techniques are described for location privacy and include the use of query enlargement, which enlarge the position of the user to a larger set of positions and then send it to the service provider. Additionally, the use of *k*-anonymous approaches is irrelevant in our case since these types of privacy-protection methods are dependent on the distribution of other users of the system. Besides the above, Palapa et al. (2012) is an approach to privacy preservation using adaptive and context-aware user interactions, although it is used in smart environments it is still important. Furthermore, the use of progressive retrieval techniques has been described by the same authors where the client iteratively retrieves results from the service provider without revealing its exact location. Also, the use of transformation based techniques is described in Kido et al. (2005) that use cryptographic transformation, hence the service provider is not able to identify the exact location and only the client has the decryption functionality to derive the actual results.

In addition, the privacy at the user interface level is a very important aspect in mobile environments. Privacy is certainly an essential part at the Human-Computer Interaction (HCI) level and in more particular at the user interface (Ackerman & Mainwaring, 2005; Iachello & Hong, 2007). Users of mobile recommender systems suffer from privacy concerns at different levels, including the user interface among others. A shoulder adversary from humans is a factor that needs to be taken into consideration. Kwon et al. (2014) proposed an approach for privacy protection from human adversaries at the interface level. This work has influenced our approach. Moreover, another stimulus came by Gamecho et al. (2015): authors propose a method for the automatic generation of accessible user interfaces (although this is not a privacy aware method).

Existing approaches although sufficient in their domain, are only concentrated in one area (such as rating protection or location protection). Our motivation is to provide a unified method for privacy protection both at the context level and at the user interface level. In addition, our focus is to serve mobile users. Thus, we associate the mobile user interface functionality with the context variables. As a consequence, our proposed method is focused on privacy protection of users utilizing mobile context-aware recommender systems, by shielding not only the location but also other context parameters and by protecting the mobile user by human adversaries. The detailed explanation of our method can be found in the following section.

5. Proposed Method

Privacy is becoming increasingly important in mobile computing environments, including the field of recommender systems in such domains. However, efforts have been made towards the location-based services problem, which is only one of the many parameters of context that can be found in context-aware mobile recommender systems. A method that protects the user privacy when context parameters are used for mobile recommendations is proposed. Mobile recommender systems are based on a regular recommendation algorithm such as collaborative filtering, content based filtering or a hybrid approach. Furthermore, a context filtering method needs to be applied in order to sort the recommendations and propose the ones that are more relevant according to the contextual parameters. In our method we use collaborative filtering with contextual post filtering with an overview of the recommendation process taking place at the server and shown in figure 2.

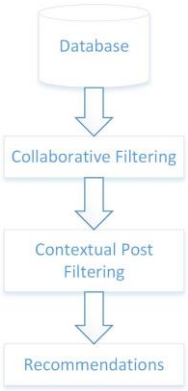


Figure 2. Recommendation process service provider overview

5.1 Privacy at the context level

To protect the privacy of users requesting context aware recommendations we need to explain the architecture of the system and how it works efficiently.

Mobile device: A mobile device could be a smartphone, tablet or another device that is portable and capable of utilizing location through the Global Positioning System or through a wireless network.

Secure communication: It is assumed that a secure communication link is available at all times between the client (mobile device) and the server (service provider).

Service provider: The service provider, or server, provides personalized recommendations to registered users of mobile devices.

A hybrid client-server recommendation approach is proposed in order to protect the privacy of the user and provide recommendations within a reasonable time. A mobile user submits a request to the server for recommendations. The ratings of all users are stored at the server, therefore collaborative filtering takes place at the server side. Furthermore, when the user makes a request both real and dummy contextual parameters are being sent, such as location, day type, weather, mood, physical and/or others to the service provider. For the recommendation approach to work algorithm 1 is called at the mobile device which shows a request submitted from a user to the server. The request includes a dummy creation for every context parameter. The next step is for the service provider to reply using algorithm 2, which is the recommendation process that takes place at the server in typical client-server architectures. Consequently, the real and fake recommendations are being sent back to the mobile device in order to be sorted out and the real recommendations shown to the user. Figure 3 is the interaction between a user of a mobile device requesting recommendations and the service provider. The steps are as follows:

1. Initially, the client-side algorithm 1 checks whether a previous generation of dummy data has taken place (within a specified time frame). Then, either it uses these dummy values again, or it randomly generates new dummy values.
2. The next step for the mobile device is to make a request for recommendations to the server, accompanied by a set of real and dummy context parameters' values.
3. The recommendation process takes place at the server, including collaborative filtering and contextual post filtering.
4. The server sends the real and fake recommendations to the mobile device.
5. The mobile device deletes the fake recommendations (the ones based on dummy context values) and presents the real recommendations to the user.

- Note: After the (client-side) creation of the real and dummy context values and just before these are sent to the server (along with the recommendation request), they should be assigned some form of identification: the mobile client should be able to distinguish the real recommendations (those related with real context values) from the fake ones.

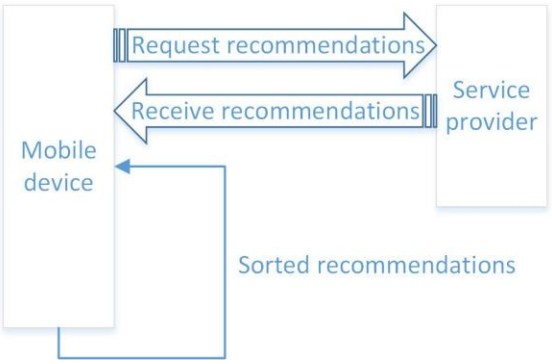


Figure 3. Client-Server interaction

Algorithm 1: Recommendation request (Mobile client)

Input: User id, Context parameters
Output: Recommendations */* List of recommendations */*

Retrieve Location, Context_Parameters[n]
Retrieve Previous_request Location, Context_Paremeters[n]
Retrieve Current_time _time_of_previous_request
If Current_time **Within Time Interval** with previous_time_request
/ Time intervals could be for example 1 set to morning, 2 set to noon, 3 set to evening Oor any other value set */*
Then
    Use Previous_request Dummy_Location, Dummy_Context_Parameters[n]
    / Dummy_Location and Dummy_Context_Parameters[n] variables contain the dummy values that will be passed to the server. These variables are populated during a previous execution of this algorithm */*
Else
    Generate Dummy_Location, Dummy_Context_Parameters[n]
    / When no dummy values are available.*
    *One fake context parameter for each real context parameter is generated */*
Request /* to the service provider with parameters */*
    User id, Location, Dummy_Location, Context_Parameters[n], Dummy_Context_Parameters[n]
/ The service provider receives the request, produces the recommendation as shown in algorithm 2 and provides them back to the client */*
Receive Recommendations */* real and fake from the service provider */*
For (int i=0; i<Recommendations.size; i++)
    If
        i.hasParameter (Dummy_Location)
        Delete i;

```
        Else
                For (int j=1; j<=n; j++)
                        If
                        i.hasParameter (Dummy_Context_Parameters[j]
                        Delete i;
                        End If
                End For
        End If
End For
Return Recommendations
```

Algorithm 2: Recommendation process (Service provider)

Input: User id, Context Parameters
Output: Recommendations

```
/* Starts with collaborative filtering */
Load User ratings
Load Similarity measure /* Pearson correlation similarity */
Provide Recommendations
/* Contextual post filtering follows */
For (int i=0; i<Recommendations.size; i++)
  If
– i.hasParameter != (Location || Dummy_Location)
  Delete i;
  Else
  For (int j=1; j<=n; j++)
        If
i.hasParameter != (Context_Parameters[j] || Dummy_Context_Parameters[j])
        /* Context_Parameters contains the real context parameters received as input from
        algorithm 1. Dummy_Context_Parameters contains the dummy context parameters
         received as input from algorithm 1. */
        Delete i;
        End If
  End For
  End If
End For
Return Recommendations
```

5.2 Privacy at the user interface level

Additionally, we consider privacy to be an important aspect at the user interface level of mobile recommender systems. Thus, we propose the use of two different user interfaces, a regular and privacy-friendly. The interfaces swap according to the context parameters available. Algorithm 3 decides, according to relevant context parameters, if a privacy-friendly user interface will be used. If algorithm 3 returns a privacy-friendly interface, then algorithm 4 needs to run in order to derive the rating value from that interface. In the case that a regular interface is selected then algorithm 4 is not relevant and won't have to be executed.

Algorithm 3: Privacy Decision
/ This algorithm makes a decision if a privacy-friendly interface is necessary, according to the context, or a regular interface */*
Input: location, social
Output: Interface /* either the regular or the privacy-friendly one */

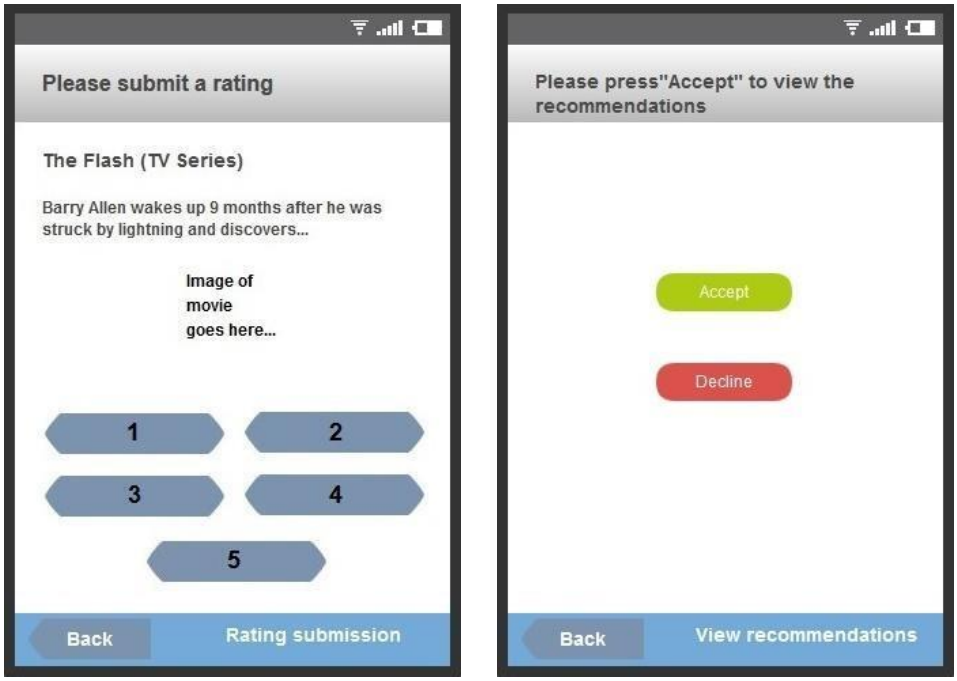
```
If location == home && social == alone
        Then Interface == Regular interface
Else
        Interface == Privacy-friendly interface
Return Interface
```

Algorithm 4: Rating Decision
/ This algorithm derives the exact numerical value from the privacy-friendly rating interface */*
Input: Button_number, Button_number.part /* Loads button number eg. 1, 2, 3, 4, 5 and the part of the button eg. left or right */
Output: Rating

```
Switch (Button_number)
        Case 1: If Button_number.part == left    Then Rating == 1
                  Else    /*    Button_number.part == right    */
                        Rating == 2
                Break;
        Case 2: If Button_number.part == left    Then Rating == 2
                  Else    /*    Button_number.part == right    */
                        Rating == 3
                Break;
        Case 3: If Button_number.part == left    Then Rating == 3
                  Else    /*    Button_number.part == right    */
                        Rating == 4
                Break;
        Case 4: If Button_number.part == left    Then Rating == 4
                  Else    /*    Button_number.part == right    */
                        Rating == 5
                Break;
        Case 5: If Button_number.part == left    Then Rating == 5
                  Else    /*    Button_number.part == right    */
                        Rating == 1
                Break;
Return Rating
```

5.3 Prototype Implementation

We have developed a privacy-friendly prototype as shown in figure 4. Figure 4 (a) is the privacy-friendly rating user interface. Figure 4 (b) is a typical warning given to the user if he is in a public location or with someone else before the recommendation list is released.



(a) (b)

Figure 4. A privacy-friendly rating interface and a privacy-friendly warning interface

6. Experimental Evaluation

For the experimental evaluation a Pentium i3 2.13 GHz with 4GBs of RAM, running windows 8.1 was used. All the algorithms have been implemented in Java and used Collaborative filtering with contextual post filtering. Moreover, a mobile smartphone running android 5.0 was used.

6.1 Real dataset

For the evaluation part we have used the LDOS-CoMoDa context aware dataset (Košir et al., 2011). This is a real dataset that apart from the usual user-rating scale from 1-5 for movies it also contains 12 contextual variables, which are described in table 1. Furthermore, table 2 is the statistical description of the dataset.

Context Parameter	Values	Description of values
Time	1 to 4	1=morning, 2=afternoon, 3=evening, 4=night
Daytype	1 to 3	1=working, 2=weekend, 3=holiday
Season	1 to 4	1=spring, 2=summer, 3=autumn, 4=winter
Location	1 to 3	1=home, 2=public, 3=friend's house
Weather	1 to 5	1=sunny, 2=rainy, 3=stormy, 4=snowy, 5=cloudy

Social	1 to 7	1=alone, 2=partner, 3=friends, 4=colleagues, 5=parents, 6=public, 7=family
endEmo	1 to 7	1=sad, 2=happy, 3=scared, 4=surprised, 5=angry, 6=disgusted, 7=neutral
dominantEmo	1 to 7	1=sad, 2=happy, 3=scared, 4=surprised, 5=angry, 6=disgusted, 7=neutral
Mood	1 to 3	1=positive, 2=neutral, 3=negative
Physical	1 to 2	1=healthy, 2=ill
Decision	1 to 2	1=By user, 2=By other
Interaction	1 to 2	1=first, 2=number of int, after first

Table 1. Description of Context Variables of LDOS-CoMoDa dataset

Description	**Value**
Users	95
Items	961
Ratings	1665
Average age of users	27
Countries	6
Cities	18
Maximum submitted ratings from one user	220
Minimum submitted ratings from one user	1

Table 2. Statistical description of LDOS-CoMoDa dataset

6.2 Context privacy performance evaluation

User Bob is located at his home, which according to the description of the context parameters of the dataset is set to number 1. Moreover, the time of the day is set to number 3 because it is evening time. The other available contextual parameters are social that is set to 1 (alone) and mood which is set to 1 (positive). Now, Bob wants to use his mobile application to recommend him a movie to watch, while he is at home. The following steps take place.

1. Bob starts the mobile application and makes the request.

2. The mobile application automatically selects the current location and the algorithm randomly assigns another location. In this case locations 1 and 2 have been selected.

3. The time is not necessary to be protected. Therefore, the value remains to 3.

4. The social parameter is set to 1 and 3.

5. The mood parameter is set to 1 and 2.

All the information is being sent to the service provider, which then provides movie recommendations according to ratings supplied by the users of the systems and with the use of collaborative filtering and by taking into consideration all the above contextual parameters described in steps 2 to 5. Figure 5 shows the performance comparison when the service provider uses one contextual parameter for each type of context and when the second, dummy, parameter is introduced for every type of context. The number of the requested recommendations is set to 5, 10 and 20. We assumed that user with id no 23 in the dataset is Bob and that's how the experiment took place. It should be noted though that the collaborative filtering method returned 14 relevant results with our provided settings and when the system requested 20. In all cases all the context parameters applied after the recommendations returned from the collaborative filtering algorithm.

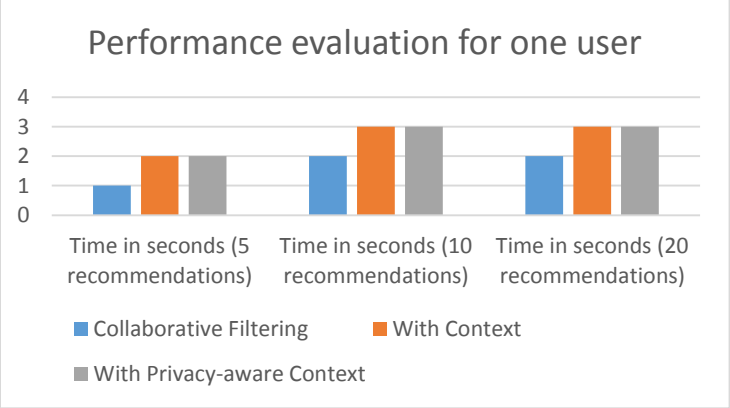


Figure 5. Performance evaluation for one user

Figure 6 shows the performance results when five users concurrently request for five recommendations each, whereas figure 7 shows the performance results for ten users requesting five recommendations each.

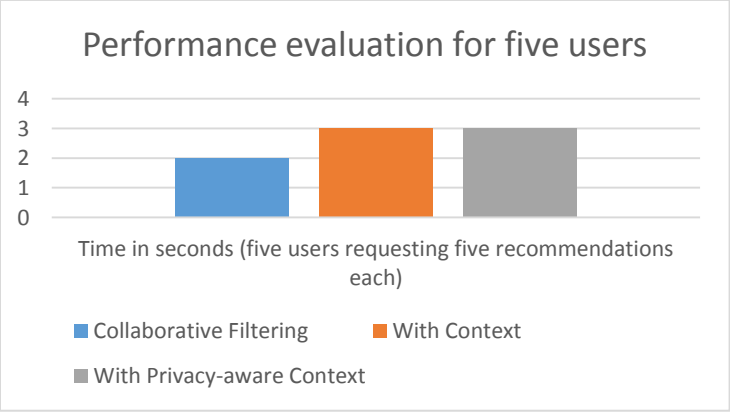


Figure 6. Performance evaluation for five users

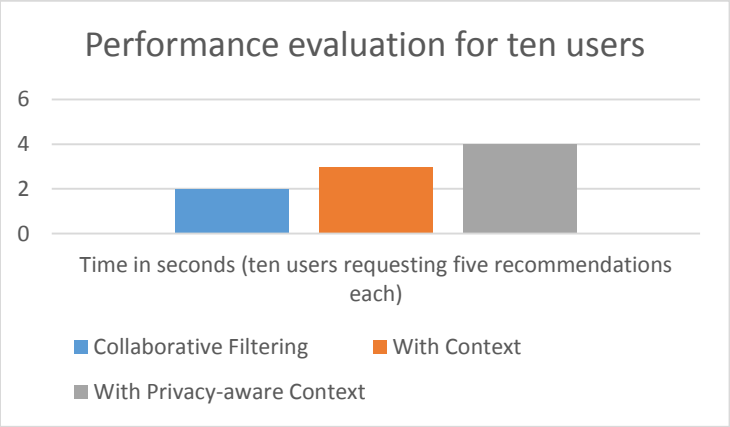


Figure 7. Performance evaluation for ten users

Furthermore, a performance regarding the transfer time is necessary and is shown in table 3. Assuming that a number of data needs to be transferred from the service provider to the mobile client a wireless channel needs to be used. Supposing that two images for a recommended item are necessary and they are of 100kb each and regular text which we have set to 100Kb. Also note that rendering time and presentation in the mobile device was not included in these tests. Only transfer times between a computer and a mobile device using a wireless network are included. Moreover, these times may vary depending on the number of concurrent requests to the server and any overheads included.

Number of Recommendations	Wireless Speed	Size in Megabytes	Transfer Time in Seconds
5	11 Mbits	1.5	4
10	11 Mbits	3	8
5	11 Mbits	1.5 (+10% overhead)	5
10	11 Mbits	3 (+10% overhead)	8

Table 3. Transfer Time between the computer and the mobile device

6.3 User interface evaluation

Initially, the system usability scale has been used to evaluate the user interface of the prototype. The number of participants was 15. Furthermore, we have used the System Usability Scale (SUS) (Brooke 1996) which is one of the most widely used approaches in user interface evaluation (Charfi et al., 2015; Braunhofer et al., 2014). Moreover, the average SUS score computed in 500 studies is 68 and thus this number it may be considered as an acceptable baseline (Braunhofer, 2014). The SUS scale detailed explanation can be found in appendix A.

The experimental results of the SUS were based on a user study of 15 participants aged from 20 to 50 years, with usable knowledge of information technology. The users were asked to rate using the privacy-friendly interface five different times each one, and then answer the questions.

- The average SUS score obtained is 72, which exceeds the previously mentioned threshold of '68'.

Additionally, performance evaluation results involved the average entry time in seconds for a regular rating interface from 1 to 5 and the privacy-friendly interface introduced here. –More specifically we measured how long it took in seconds for the same 15 users to submit a rating in both cases. We believe that this measurement is important since a user needs a certain amount of time to think and interpret that a selection button provides a different rating value from the one that is supposed to give in the privacy-friendly interface. Figure 8 shows the results.

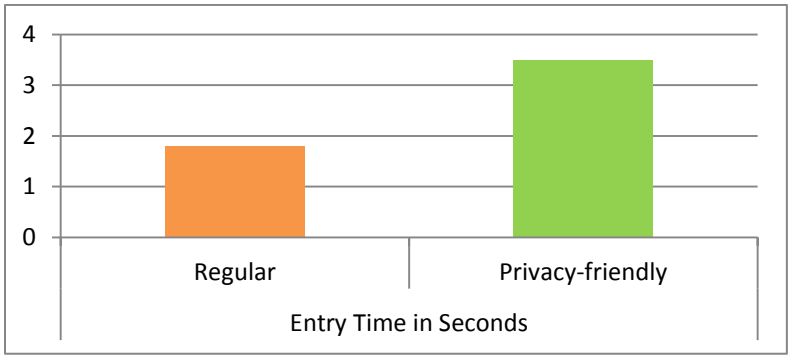


Figure 8. Entry Time in Seconds

7. Conclusions and future work

Various online services, including m-commerce and m-government among others, use technologies such as mobile recommender systems that aim to solve the information overload problem of users utilizing such a software system in their mobile device. Privacy though is an open issue and most privacy protection techniques have been based either on personal data, such as user ratings, protection or use third party systems for keeping private the exchange of information. Various privacy-protection methods are evolving, capable to preserve privacy at different parts of the recommendations process. If we consider privacy as one of the most important aspects in mobile recommender systems, we have to both protect privacy at the context and at the user interface level. Users are moving constantly among other people and their privacy should be respected both from other people and service providers. We proposed a practical and effective method that aims to protect privacy of contextual parameters and user interface found in context-aware mobile recommender systems, without the use of a third party. The proposed method automatically generates a set of realistic dummies that are sent from the client to the server, with the real values hidden between the dummy ones. Then the system, with a small utility cost, provides a set of recommendations to the client without any privacy risks. On completion the client disregards the dummy recommendations, eliminating any surplus and/or confusing data.

Such a system may have wide applicability as it can be used in various real life scenarios. , In the case of a user using a mobile recommender application for tourists seeking information on local points of interest, the user may protect her contextual information (such as location, social status etc.) when these are being sent to a central server. Another indicative example is that of a user watching a video in a public place. She may utilize the privacy-preserving user interface in her mobile device to protect her ratings and preferences against indiscreet looks by onlookers or passersby.

In this work, we experiment on a privacy approach to protect user privacy at the context and interface levels and as a result, the proposed method has been experimentally evaluated using a real dataset and with real users, with the results being satisfactory. Although our approach makes use of simple algorithms (both at the client-server level and at the user interface level), a certain level of innovation can be found in the association of the context variables with the user interface functionality. Furthermore, the simplicity of the algorithms employed results in faster delivery of the recommendations and this is highly desirable in mobile environments.

As a future work we plan to investigate the possibility to bridge the current gap between mobile computing and recommender systems and deliver a complete framework that could protect ~~the~~ user privacy at different levels of the recommendation process. Furthermore, we wish to investigate the possibility of making the algorithms more intelligent and evaluating their performance by engaging real users as opposed to using datasets.

References

Ackerman, M., & Mainwaring, S. (2005), "Privacy issues and human-computer interaction. Computer", Vol. 27 No. 5, pp. 19-26.

Adomavicius, G., & Tuzhilin, A. (2011), "Context-aware recommender systems", In Recommender systems handbook, Springer US, pp. 217-253.

Ahluwalia, P., Varshney, U., Koong, K. S., & Wei, J. (2014), "Ubiquitous, mobile, pervasive and wireless information systems: current research and future directions", International Journal of Mobile Communications, Vol. 12 No. 2, pp. 103-141.

Aïmeur, E., Brassard, G., Fernandez, J. M., & Onana, F. S. M. (2008), "Alambic: a privacy-preserving recommender system for electronic commerce". International Journal of Information Security, Vol. 7 No. 5, pp. 307-334.

Anacleto, R., Figueiredo, L., Almeida, A., & Novais, P. (2014), "Mobile application to provide personalized sightseeing tours", Journal of Network and Computer Applications, Vol. 41, pp. 56-64.

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013), "Recommender systems survey", *Knowledge-Based Systems*, Vol. 46, pp. 109-132.

Boutet, A., Frey, D., Guerraoui, R., Jégou, A., & Kermarrec, A. M. (2015), "Privacy-preserving distributed collaborative filtering" In *Computing. dx.doi.org/10.1007/s00607-015-0451-z*

Braunhofer, M., Elahi, M., & Ricci, F. (2014), "Usability assessment of a context-aware and personality-based mobile recommender system", In E-Commerce and Web Technologies, Springer International Publishing, pp. 77-88.

Brooke, J. (1996), "SUS-A quick and dirty usability scale", Usability evaluation in industry, Vol. 189 No. 194, pp. 4-7.

Camp, L.J. (2015). "Respecting people and respecting privacy", Communications of the ACM, Vol. 58 No. 7, pp. 27-28.

Cao, Y., Lu, Y., Gupta, S., & Yang, S. (2014), "The effects of differences between e–commerce and m–commerce on the consumers' usage transfer from online to mobile channel", International Journal of Mobile Communications, Vol. 13 No. 1, pp. 51-70.

Charfi, S., Ezzedine, H., & Kolski, C. (2015), "RITA: a useR Interface evaluaTion framework", Journal of Universal Computer Science, Vol. 21 No. 4, pp. 526-560.

Colombo-Mendoza, L. O., Valencia-García, R., Rodríguez-González, A., Alor-Hernández, G., & Samper-Zapater, J. J. (2015). RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes. Expert Systems with Applications, 42(3), 1202-1222.

del Carmen Rodríguez-Hernández, M., & Ilarri, S. (2016), "Pull-based recommendations in mobile environments". Computer Standards & Interfaces, Vol. 44, pp. 185-204.

Drogkaris, P., Gritzalis, S., & Lambrinoudakis, C. (2013), "Employing privacy policies and preferences in modern e–government environments" *International Journal of Electronic Governance*, Vol. 6 No. 2, pp. 101-116.

Gamecho, B., Minón, R., Aizpurua, A., Cearreta, I., Arrue, M., Garay-Vitoria, N., & Abascal, J. (2015), "Automatic Generation of Tailored Accessible User Interfaces for Ubiquitous Services", IEEE Transactions on Human-Machine Systems. DOI:10.1109/THMS.2014.2384452

Iachello, G., & Hong, J. (2007), "End-user privacy in human-computer interaction", Foundations and Trends in Human-Computer Interaction, Vol. 1 No. 1, pp. 1-137.

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010), *Recommender systems: an introduction*. Cambridge University Press.

Jensen, C. S., Lu, H., & Yiu, M. L. (2009), "Location privacy techniques in client-server architectures" In Privacy in location-based applications, Springer Berlin Heidelberg, pp. 31-58.

Kato, R., Iwata, M., Hara, T., Suzuki, A., Xie, X., Arase, Y., & Nishio, S. (2012), "A dummy-based anonymization method based on user trajectory with pauses", In Proceedings of the 20th International Conference on Advances in Geographic Information Systems, ACM, pp. 249-258.

Kido, H., Yanagisawa, Y., & Satoh, T. (2005), "An anonymous communication technique using dummies for location-based services", In *Pervasive Services, 2005. ICPS'05. Proceedings. International Conference on*, IEEE, pp. 88-97.

Konstan, J. A., & Riedl, J. (2012), "Recommender systems: from algorithms to user experience" User Modeling and User-Adapted Interaction, Vol. 22, No. 1-2, pp. 101-123.

Košir, A., Odic, A., Kunaver, M., Tkalcic, M., & Tasic, J. F. (2011), "Database for contextual personalization". *Elektrotehniški vestnik*, Vol. 78 No. 5, pp. 270-274.

Kwon, T., Shin, S., & Na, S. (2014), "Covert attentional shoulder surfing: Human adversaries are more powerful than expected", Systems, Man, and Cybernetics: Systems, IEEE Transactions on, Vol. 44 No. 6, pp. 716-727.

Liu, Q., Ma, H., Chen, E., & Xiong, H. (2013), "A survey of context-aware mobile recommendations" *International Journal of Information Technology & Decision Making*, Vol. 12 No. 1, pp. 139-172.

Lu, H., Jensen, C. S., & Yiu, M. L. (2008), "Pad: privacy-area aware, dummy-based location privacy in mobile services" In Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access, ACM, pp. 16-23.

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015), "Recommender system application developments: A survey", Decision Support Systems, Vol. 74, pp. 12-32.

Mettouris, C., & Papadopoulos, G. A. (2014), "Ubiquitous recommender systems". *Computing*, Vol. 96 No. 3, pp. 223-257.

Niu, B., Zhang, Z., Li, X., & Li, H. (2014), "Privacy-area aware dummy generation algorithms for location-based services". In Communications (ICC), 2014 IEEE International Conference on, IEEE, pp. 957-962.

Noguera, J. M., Barranco, M. J., Segura, R. J., & Martínez, L. (2012), "A mobile 3D-GIS hybrid recommender system for tourism", Information Sciences, Vol. 215, pp. 37-52.

Pallapa, G., Francesco, M. D., & Das, S. K. (2012, June). Adaptive and context-aware privacy preservation schemes exploiting user interactions in pervasive environments. In World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a (pp. 1-6). IEEE.

Polat, H., & Du, W. (2005), "Privacy-preserving collaborative filtering", *International Journal of Electronic Commerce*, Vol. 9 No. 4, pp. 9-35.

Polatidis, N., & Georgiadis, C. K. (2015), "A ubiquitous recommender system based on collaborative filtering and social networking data", International Journal of Intelligent Engineering Informatics, Vol. 3 No. 2-3, pp. 186-204.

Polatidis, N., Georgiadis, C. K., Pimenidis, E., & Stiakakis, E. (2015), "A method for privacy-preserving context-aware mobile recommendations" In E-Democracy–Citizen Rights in the World of the New Computing Paradigms, Springer International Publishing, pp. 62-74.

Ricci, F. (2010), "Mobile recommender systems". *Information Technology & Tourism*, Vol. 12 No. 3, pp. 205-231.

Scipioni, M. P. (2011). Towards privacy-aware location-based recommender systems. IFIP Summer School 2011.

Shi, Y., Larson, M., & Hanjalic, A. (2014), "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges", *ACM Computing Surveys (CSUR)*, Vol. 47 No. 1, 3.

Tran, M. T., Echizen, I., & Duong, A. D. (2010). "Binomial-mix-based location anonymizer system with global dummy generation to preserve user location privacy in location-based services", In Availability, Reliability, and Security, 2010. ARES'10 International Conference on, IEEE, pp. 580-585.

Appendix A

The System Usability Scale (SUS)

In SUS, the people who participated in the evaluation are asked to answer the following ten questions by choosing one of the five proposed ratings, that range between strongly agree to strongly disagree:

I think that I would like to use this system frequently.

I found the system unnecessarily complex.

I thought the system was easy to use.

I think that I would need the support of a technical person to be able to use this system.

I found the various functions in this system were well integrated.

I thought there was too much inconsistency in this system.

I would imagine that most people would learn to use this system very quickly.

I found the system very cumbersome to use.

I felt very confident using the system.

I needed to learn a lot of things before I could get going with this system.

The SUS uses the following rating format:

Strongly Disagree **1**	2	3	3	**Strongly Agree** **5**

The scoring of SUS is then calculated by using the following rules:

For odd questions, such as Q1, Q3, Q5, Q7 and Q9, subtract one from the response received from the user.

For even questions, such as Q2, Q4, Q6, Q8 and Q10, subtract the response received from the user from 5.

The two above steps scale all the values from 0 to 4.

Then we multiply the sum of the scores by 2.5 to obtain a score between 0 and 100.