Achieving Targeted Mobile Advertisements While Respecting Privacy

Elia Palme[1,2], Basil Hess[2], and Juliana Sutanto[2]

[1] Newscron, Lugano, Switzerland
[2] ETH Zürich, Management Information Systems
Zürich, Switzerland
{epalme,bhess,jsutanto}@ethz.ch

Abstract. Broadcasted mobile advertisements are increasingly being replaced by targeted mobile advertisements through consumer profiling. However privacy is a growing concern among consumers who may eventually prevent the advertising companies from profiling them. This paper proposes an agent-based targeting algorithm that is able to guarantee full consumer privacy while achieving mobile targeted advertising. We implemented a grocery discount-discovery application for iPhone that makes use of the new approach. We show that on modern hardware like on the iPhone, it's feasible to run a client-based and privacy-preserving targeting algorithm with minimal additional computational overhead compared to a random advertising approach. We evaluated the targeting method by conducting a large-scale field-experiment with 903 participants. Results show that the computational overhead on user devices is well tolerated, compared to the control group with randomized advertising the targeting group showed a significantly increased application usage of 18%.

Keywords: mobile pervasive advertisement, privacy, targeted one-to-one marketing, mobile agent.

1 Introduction

Advertisements of the future are likely to be mobile and highly targeted [5]. Mobile advertising refers to advertising using mobile devices such as tablet pcs, smart phones or cellular phones [7]. The International Telecommunication Union (ITU) estimated 5.3 billion cellular subscribers at the end of 2010, including 940 million subscribers to 3G services. The high penetration of mobile devices enables merchants to reach a large number of potential customers. Gartner estimates 70 billion mobile app. downloads for 2014, compared to 17.7 billion in 2011[1].

Besides the term "mobile", another popular term in the advertising industry is "targeting". Researchers have consistently shown that targeted advertising messages are more effective than the non-targeted ones [1]. Furthermore targeted advertisements can minimize consumers' annoyance level, and mobile device is

[1] http://www.gartner.com/it/page.jsp?id=1826214

D. Uhler, K. Mehta, and J.L. Wong (Eds.): MobiCase 2012, LNICST 110, pp. 245–263, 2013.

a good platform to achieve this aim. A mobile device "follows" its owner anytime anywhere. Because of the awareness of its owner's context and preferences, mobile devices are a useful medium for implementing targeted marketing.

In this paper, we present a concrete solution to achieve targeted mobile-advertising while respecting the consumer need for privacy. Specifically, we propose an unsupervised, targeted mobile advertisement framework. We profile individual consumers and use their profile information to select the most captivating advertisement messages for each consumer while at the same time ensuring their privacy. In our framework, we develop a mobile advertising-agent that reaches to the customers and estimates their interests in the advertisement messages of their mobile device.

Customer privacy is protected through a distributed mobile framework in which a customer profile is locally stored in her mobile device, and the advertisement targeting is performed by her respective mobile device. The level of privacy protection that our method achieves is the following: Users first have to explicitly authorize local storage of their profile information. If authorized, we consider the internal storage of a mobile device to be trusted by its owner, so that local storage of private profile information on the phone doesn't affect privacy. The privacy-policy however guarantees that none of this information is ever transferred through the device's external communication facilities (e.g. WiFi, 3G) to any second party.

In the next sections, we review what other researchers have accomplished in this area, and introduce our mobile advertising-agent. This is followed by a detailed explanation of our proposed framework, and the results of our field experiment.

2 Related Work

While much work has been done on introducing the targeted advertising method to mobile devices, no study has so far proposed a technical solution to overcome the trade-off between targeted advertising and privacy concerns, and no large-scale field study on targeting methods has been done. Most of the studies on targeting are conceptual, presenting architecture design without details on how targeting is actually achieved. Our study on the other hand demonstrates that the flexibility provided by the mobile agent-architecture can be efficiently applied to the problem while transferring an acceptable computational overhead to the customer device.

We classify previous works based on three dimensions: 1) their advertising targeting method, i.e., how customer information is used to compute the matching between the advertisements and the customer, 2) their privacy handling, i.e., how the system protects customer privacy, and 3) how the system performance and effectiveness are tested.

2.1 Advertising Targeting Methods

Location filtering is the most basic method to select the most appropriate advertisements for the customers.

Location-Based Targeting. [2][6][25][9] uses one single piece of information to perform the targeting by comparing the customer's location with the ads target location.

Rule-Based Targeting. [3] will only display the advertisements if the customer's profile satisfies the ad's targeting rules, e.g., a gender rule.

Category-Based Targeting. [19][9][11][4][24][12] groups ads into categories. By analyzing customer profiles, it identifies which category is potentially interesting for the customer.

Ontology-Based Targeting. [16] defines hierarchical relationships among entities (user data and ad description) and tries to find the ad that has the highest number of related entities to the user profile.

Agent-Based Targeting. [14] is the most flexible method, because it allows to implement arbitrary targeting methods. Mobile agents carry interpretable code which allows them to implement any of the previously discussed strategies. For instance, an agent could simply check for the customer's location (Location-based targeting) or compute more elaborated targeting by identifying if the ad category fits with the customer profile. In our method an agent is used to compute an integer value that represents the degree to which an advertisement matches the customer's profile. More details on how mobile agents are used are in section 3.1. Mobile agents have the singularity of excuting the targeting code on the customer's devices; the evaluations will provide valuable insights on the agents computation overhead and the customer's acceptance of it.

2.2 Privacy Handling

To mitigate consumer's privacy concern, several distributed mechanisms have been proposed. These mechanisms try to reduce the consumer's concern either by providing high level of transparency on the collected data or by protecting the consumer's identity. We classified previous works on privacy handling in the following way:

Permission-Based Mechanism. [2] asks for consumer permission to collect specific information; it consequently provides high transparency on the amount and type of the collected data. In 1997 the World Wide Web Consortium (W3C) launched the platform for the Privacy Preferences Project (P3P). The main goal of P3P is to provide an automated mechanism to inform users on their private information treatment. Langheinrich [13] developed a Privacy Awareness System (PawS) based on P3P providing high level of awareness and "data collection" service opt-out option.

Table 1. Comparison of our study with related work on mobile advertising

	Targeting Method	**Privacy Handling**	**Validity Test**
Ranganathan and Campbell (2002) [19]	Category-based	Local Profiling	None
Aalto et al. (2004) [2]	Location-based	Permission-based	35 participants
Mahmoud et. al. (2007) [14]	Agent-based	None	None
Barnes et. al. (2008) [6]	Location-based	None	None
Gao and Ji [9]	Category-based, Location-based	None	None
Yang et. al. [25]	Location-based	None	137 students
Narayanaswami et. al. [16]	Ontology-based	Anonymity	None
Shannon et. al. (2009) [21]	Unknown	Aggregation	None
Aggarwal et. al. [3]	Rule-based	Local Profiling	None
Guha et. al. [11]	Category-based	Local Profiling	None
Alt et. al. [4]	Category-based	Anonymity	None
Toubiana et. al. (2010) [24]	Category-based	Local Profiling	None
Haddadi et. al. (2010/11) [12]	Category-based	Local Profiling	None
This paper	**Agent-based**	**Local Profiling**	**903 participants**

Aggregation Mechanism. [21] groups customers into clusters based on their demographic profile (e.g. a cluster of young people) and then targets the ad based on the cluster information. The privacy protection of this method is limited because it involves collecting consumer data.

Anonymity Mechanism. [16][4] protects consumer's privacy by anonymizing the data collected. However by collecting a large enough amount of anonymized data, the consumer's identity could be reverse engineered.

Local Profiling Privacy Enhancement. [19][3] [11][24][12] mechanism protects customer identity by avoiding the transmission of any data. Usually this method is coupled with other security mechanism (e.g. sand-boxed environment, anonymous pings or delay tolerant networks [12]) to guarantee that no information can be stolen or recovered while being able to bill for the ads impressions. This study adopts the local profiling method.

2.3 System Performance and Effectiveness Test

Most of the previous studies are conceptual in nature without actual testing of their proposed methods. Aalto [2] and Yang [25] validated their studies with a small amount of around 100 participants. In contrast, we propose a system that has been fully implemented and validated through a field experiment (with around 1000 participants) to test the performance and effectiveness of the proposed method. Especially we demonstrate that mobile agents are efficiently used to distribute the targeting computation on the customer's devices. Agents have a minor impact on the customer devices performance and are therefore tolerated by users. The following sections provide insights on our system architecture, implementation, and evaluation.

3 Privacy-Preserving Targeted Advertising Framework

In this section, we present our privacy-preserving targeted advertising framework that dispatches, targets, and distributes advertisements in a privacy-preserving way. Three components compose the framework: the mobile advertising-agent (Section 3.1), mobile advertising-agents' client (Section 3.2), and mobile advertising-agents' server (Section 3.3).

Our framework is based on the following main principles: (1) Privacy-enhanced mobile agency for targeting advertisements, (2) a central advertising-agents server, and (3) mobile devices with capabilities to process agents (adver-tising-agents client). The mobile agents are privacy-enhanced because they don't leak any user profile data to any second party. Each agent corresponds to one advertisement, plus the targeting mechanism associated to it, whose goal is to find matching customers. Finally the user's mobile device decides if the agent with its advertisement and targeting method matches the user's profile.

As the matching between the carried advertisement and the consumer profile is performed on the client's mobile device, the number of consumers that can be handled is highly scalable and the server's infrastructure is very light. The process of creating an agent and displaying the advertisement is illustrated in Figure 1.

3.1 Mobile Advertising-Agent

The mobile agent comprises two separate and distinct concepts, i.e., mobility and agency [18]. Mobility denotes the ability to move and visit multiple hosts. Agency is the characteristic of an agent, who is hired by a principal (in our case an advertiser) to sell its product to interested customers (in our case the mobile device user). Upon reaching a customer's mobile device, a mobile advertising-agent checks if the customer is in the target group of the carried advertisement.

The mobile advertising-agent is a small piece of binary data, which carries a digital advertisement and a match estimator, as shown in Figure 2. The match estimator is an interpretable code used to select only those people that are within the target of the carried advertisement. Principally the mobile agent concept [17] comprises the ability of agents to autonomously traverse multiple hosts. However, to mitigate security risks [8], we restrain this feature and allow the migration process only through the central server. This allows us to control the migration flow and to trigger security checks. Each agent has to be cryptographically signed by the principal of the agent (i.e. the advertiser). Before delivering an agent, the

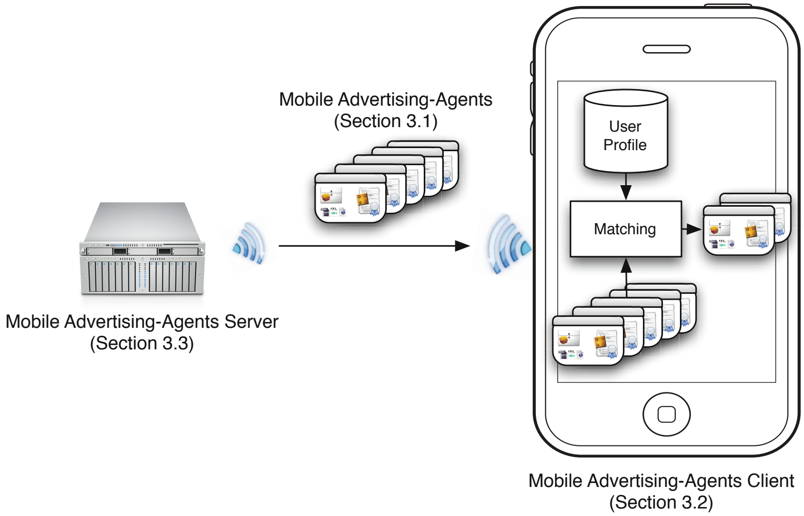

Mobile Advertising-Agents
(Section 3.1)
Mobile Advertising-Agents Server
(Section 3.3)
User
Profile
Matching
Mobile Advertising-Agents Client
(Section 3.2)

Fig. 1. The privacy-preserving targeted advertising framework, consisting of the server part (left), the mobile advertising-agents (middle), and the client part in a smartphone device (right)

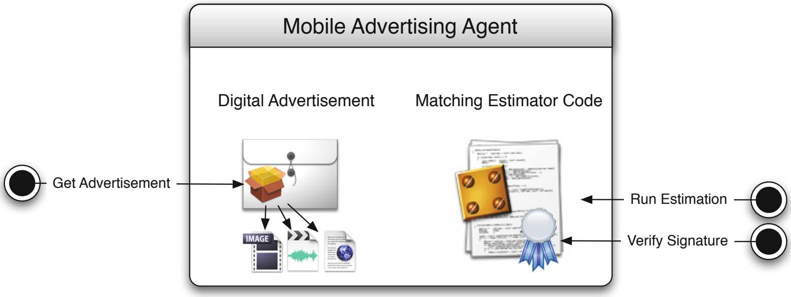


Fig. 2. Schema of the Mobile Advertising-Agent. An agent comprises of a digital advertisement and a matching estimator code, which are cryptographically signed.

server verifies the cryptographic signature of the agent to be able to guarantee that only authorized advertisements are circulated. The server then singes the agent with its own signature, which is eventually verified by the client upon reception of the agent.

Match Estimator. The match estimator is a piece of software code that is carried by the agent. The code is originally generated by an advertiser, with the goal to determine if a user belongs to his target group or not. As part of the agent, the match estimator is transferred from the central server to the customer's mobile device, where it is eventually executed. Match estimator can be any type of executable code. We prefer to use JavaScript [22] because many mobile platforms are offering APIs to run the scripts in a sandboxed environment. In our implementation iOS offers to run JavaScript in a sandboxed WebKit-instance that prevents all Internet connections, and protects the user profile from being modified.

The goal of match estimator is to determine the customer's affinity with the advertisement target. The outputs of the match estimator code are two integer values: *affinity coefficient* and *uncertainty value*. The higher the affinity coefficient, the closer the customer is to the advertisement's optimal target. The higher the uncertainty, the less accurate the computed affinity coefficient is. The uncertainty value depends on the availability and importance of the information needed to compute the affinity coefficient, which means that the value will be higher if the user didn't disclose personal data used by the match estimator. In order to compute the affinity coefficient, match estimator has to access the local information repository on the client side.

The local information repository is unique for each client and following our privacy policy, it is not sharable or accessible outside the client. In the information repository, personal information such as age, gender, willingness to buy budget products, or life style diet, is stored. Before interpreting the match, a repository is assembled at the client side. Assembling the repository means extracting personal information from whatever storage support is offered by the smartphone (e.g. in iPhone, the information can be stored and retrieved through the core data framework) and generate a dictionary that is readable by the match estimator code. The generated dictionary should be read only and accessible by the match estimator code as a global variable in the same sandbox environment where the match estimator is interpreted.

Finally the "evaluate" method of the match estimator code is called, the affinity coefficient and uncertainty value is computed and returned. The decision of displaying or dropping the advertisement is mainly based on the returned affinity coefficient. If the uncertainty value is too high, the client will postpone this decision and re-run the match estimator as soon as new information is available on the local repository.

We illustrate the principles of the match estimator with a short example: A beer producer wants to advertise a new sweet beer especially conceived for females. In order to make the advertising companying as much effective as possible the beer producer has an interest to only advertise its product to major age female customers. He therefore needs to purposely configure a mobile advertising agent by setting the correct match estimator code. The match estimator code takes the local customer profile (the repository where all private user information are stored) as input and returns the affinity and incertitude value as output. In this case the affinity is at maximum if the customer is a major age female. On the other side the affinity is at minimum if the customer is too young. It can happen that on the local repository some required information is missing, for example the customer's age in not present. In this case the match estimator code cannot compute the affinity value or can only partially compute it. If all information required to compute the affinity are available the uncertainty value is zero. But in the case of the sweet beer if the customer age is missing the incertitude is at maximum because knowing if the customer is major age is a required information. If the uncertainty is too big the advertisement is put on hold till the required information is available. Besides this simple example, the interpretable code allows to achieve much more complex targeting.

Digital Advertisements. As illustrated in Figure 2, each mobile advertising-agent carries a digital advertisement. Modern advertisements rely on several types of media types like text, images or videos, and it should be possible to support all these cases. An important restriction is that no content must be loaded from remote servers, all contents have to be embedded in the advertisement itself. This implies two things: consumer privacy is protected, and the amount of transferred data is controlled. Privacy is protected because without explicit advertisement download, it is not possible to track who opens which advertisement. If all media contents are embedded in the mobile advertising-agent, we control the total data throughput to prevent consumers from downloading massive amounts of data. Finally, having all contents contained and stored in the digital advertisements allows displaying them even if Internet connectivity is lost at a later point in time.

In our implementation we choose to encode the advertisement as a HTML document to give us the liberty in defining the advertisement's graphical design. A digital advertisement is basically a webpage that can be composed by rich text and images. By choosing HTML to represent the advertisement, we "uniform" the encodings to some extent. Since our match estimators are JavaScript-based and the advertisement is HTML-encoded, we can manage both of them with the same web browser engine instance and reduce the client's memory consumption. When no external content is downloaded, external references in HTML would usually not be available. To overcome this limitation, we use an artifice provided by the data URI scheme to include in-line data in the HTML documents. The URI scheme is defined in RFC 2397 [15] and adopted by most browsers.

3.2 Mobile Advertising-Agents' Client

The mobile advertising-agents' client is the software part running on the smartphones (the client side of our framework architecture). The client is a place where: 1) mobile advertising-agents are hosted and their match estimator code is interpreted, 2) consumer's profile is stored, and 3) advertisement is displayed. The agents' client is integrated inside an application (e.g., a gaming or utility application) that would display the targeted advertisements.

Advertisements Selector. The client is responsible for fetching new mobile advertising-agents from the main server. All new agents are transmitted to the client that will then store them in the local database (see Figure 3). Newly transmitted advertising-agents are marked as unrated. All unrated agents' matching

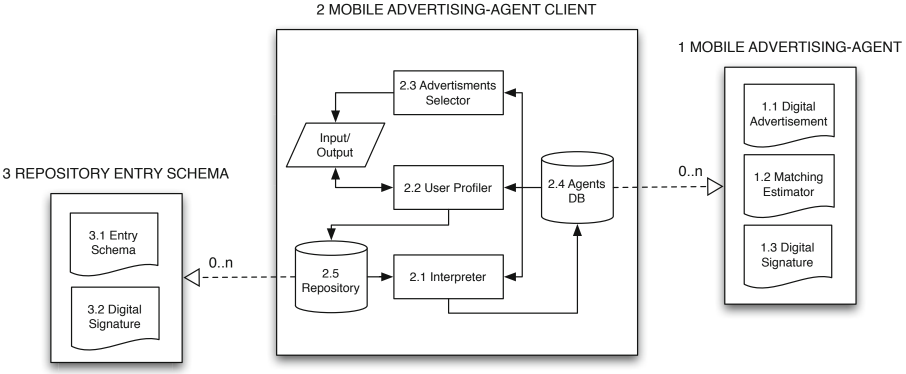


Fig. 3. A schematic overview of the Mobile Advertising-Agents' Client, showing the relationship between agents (1), client (2), and the client's local repository (3)

estimators are interpreted to establish their affinity coefficient and uncertainty value. After all match estimator codes of all new agents are computed, the client will proceed to an initial screening, i.e., only those agents that have a uncertainty value below a certain threshold will be considered. Those agents having an appropriate uncertainty value are than eligible for advertisement selection. The advertisement selection algorithm depends on the kind of advertising strategy the application wants to implement. For example, if the mobile advertising-agent client is embedded in a game application, it may show a small banner for a certain period of time. In this case, the advertisement selection algorithm could be: Take the advertisement with the highest affinity coefficient and display it for a certain amount of time, then take the advertisement with the second highest affinity coefficient, and so on.

User Profiler. The local profile is a repository containing personal information entities such as sex, religion, diet, etc. Entities have a schema that defines their nature. They are composed of: a name, a question, and possible answer. For example, the "sex" entity contains a question: "What is your gender?", and a possible answer of either "male" or "female". Entities are defined on the main server and downloaded by the client when needed (see Figure 5).

The client is responsible to query the consumer in order to populate the local profile repository. Whenever a mobile advertising-agent requires an entity that is not present in the local repository, the client will download the entity definition from the central server, after which the client will attempt to optimize the information retrieval. The more helpful the information is in computing affinity coefficients, the more urgent it is to acquire it. The agents' uncertainty values enable the client to extrapolate with a relatively high probability the most discriminating yet missing entity's answer. An unavailable entity's answer that is required by multiple mobile advertising-agents with high uncertainty values has higher discrimination power. The higher the discrimination power of an unanswered entity is, the more urgent is the retrieval of its answer.

Missing Information Handler. As previously mentioned, the matching estimator code handles missing information by increasing the uncertainty value. The uncertainty value returned by match estimator depends on the importance of the missing information needed to compute affinity coefficient. Missing information may have higher or lower impact in computing the match between the user and the advertisement depending on the importance of the information and how this information is used by the match estimator code. Thus, only the match estimator knows precisely how big the impact of missing information is, and only match estimator code can compute the uncertainty value. Match estimators of all non-expired agents with uncertainty values higher than 0 are periodically recomputed by the client. Every time the uncertainty value and affinity coefficient of an advertising-agent is recomputed, it gets a new opportunity to be selected and its advertisement has a chance to be displayed.

3.3 Mobile Advertising-Agents' Server

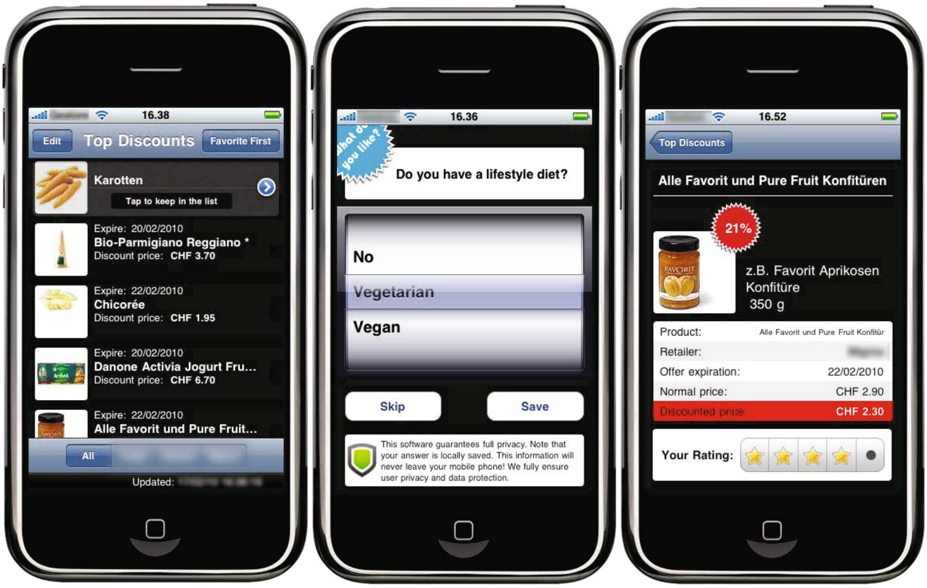
The server side is very light and does not require much computation since most of the logic is sent to the client and executed remotely. However it is the server's duty to verify the authenticity of the agents that will be delivered to the clients. To achieve this, advertisers have to cryptographically sign the agents. The server then verifies this signature. If verified, it adds its own signature and is ready to deliver the agent. Furthermore the server is responsible for limiting the client's workload. The server should limit the number of agents sent to the client, check for estimators codes that are too big to be executed, and make sure that a reasonable amount of information is stored on the client. We will further discuss this point in the discussion section.

In our implementation, the server is designed as a state-of-the-art RESTful web service provider. Fetching newly available mobile advertising-agent is a simple procedure. After client provides the identity (ID) of the last downloaded agent, the web service will assemble a list of non-expired agents with an ID higher than the recently downloaded one. For performance reason, we stream a list of agents instead of allowing one-to-one agent recuperation.

4 Implementation and Evaluation

The framework presented in this paper was implemented and tested on a large-scale field experiment. Based on our framework, we implemented a grocery discounts-discovery application for iPhone and made it available for free installation in the App Store of our country. The aims were to show the feasibility to execute targeting algorithms on modern smartphones like the iPhone, and to show the effectiveness of our algorithm in comparison with a random advertising approach.

To assemble the local information repository, we designed an ad hoc user interface, which is a full screen dialog window that asks profiling questions such as gender and lifestyle diet (see Figure 4 for a query example). Users have the



Expire: 20/02/2010
Bio-Parmigiano Reggiano *
Discount price: CHF 3.70

Expire: 22/02/2010
Chicorée
Discount price: CHF 1.95

Expire: 20/02/2010
Danone Activia Jogurt Fru...
Discount price: CHF 6.70

Expire: 22/02/2010
Alle Favorit und Pure Fruit...

Updated: [redacted]

Do you have a lifestyle diet?

This software guarantees full privacy. Note that your answer is locally saved. This information will never leave your mobile phone! We fully ensure user privacy and data protection.

Alle Favorit und Pure Fruit Konfitüren

z.B. Favorit Aprikosen Konfitüre 350 g

Product:	Alle Favorit und Pure Fruit Konfitür
Retailer:	[redacted]
Offer expiration:	22/02/2010
Normal price:	CHF 2.90
Discounted price	CHF 2.30

Fig. 4. The grocery discounts-discovery application. The main screen shows current discounts (left). Users can enter personal data, and are guaranteed that it will never be leaked from the phone (middle). Upon entering personal data, the user receives targeted discounts (right), and has the option to rate the discount in a five star scale.

possibility to either select the correct answer and save it, or skip the question. When a question is skipped three times, it is never asked again. A maximum of two profiling questions were asked per session[2].

To increase the usefulness of the application, we implemented a filtering and sorting function. The filtering functions allow users to filter discounts based on the grocery store, i.e., only the discounts in a specific grocery store are shown. The sorting function allows users to sort the discounts based on the ratings given by the users themselves (see Figure 4). The ratings are given in a one to five star scale.

To capture user interaction with the application, we implemented a logging system that locally saved important events such as application started, discount offer deleted, etc. The logging system also saved the events' timestamp for later transfer to our server.

Users were tracked using the iPhone unique identifier (UDID); a 40 bytes token that uniquely identifies each device. When users launched the application, the logs collected in the previous session together with the UDID were sent to the server for analysis. Note that the logging system was added for the field experiment purpose only. As it is not part of the conceptual framework it doesn't compromise user privacy there. We only logged the total number of events, i.e.,

[2] A session starts from when the user launches the application until he/she closes it.

the number of advertisements displayed, the number of advertisements deleted, the number of questions asked and skipped, etc.

4.1 Preliminary Conceptual Test

We ran a two-week pilot test with 10 participants, recruited among the undergraduate students of ETH Zürich. The pilot test was designed to check our concept and validate the functions of our prototype. In all, 509 mobile agents were transmitted, 275 discounts information were displayed, 45 discounts information were deleted, and 156 out of the 281 discounts information that were rated by the consumers had 3 stars or higher on a scale of 5. The 10 participants were surveyed after completing the test phase and all affirmed that none of the presented discounts information was off-target (e.g. a vegetarian did not receive a discount for meat). The participants also gave some comments on the usability of the application: 'it is useful to keep track of ads", 'it's easy to find out where the best discounts are, so you can choose at home where to shop', 'best offer in a glance', 'since the interface was nicely done it was never a problem for me to answer the profiling questions'.

4.2 Field Experiment

After the successful pilot test, we ran a field experiment. In collaboration with a local firm, we distributed real grocery discounts information through our application. Each week we distributed an average of 150 new discounts information of the three largest local grocery stores. We ran the experiment for 3 months with 903 participants. To evaluate our targeting mechanism, we incorporated a control group in the field experiment. The control group was equipped with a version of the application that randomly selected the discounts information without any targeting feature. We randomly assigned the consumers to the test or control group. It is important to note here that once a consumer was assigned to a particular group, he/she could not change it. The advertisement selection algorithm and iPhone identifier were stored in a DB. Even if they uninstalled and reinstalled the application, they would still get the same advertisement selection algorithm. 52% of the users (470) were assigned to the test group (with targeting algorithm) and 48% (433) to the control group (with the random selection algorithm).

Performance Analysis. The first analysis focuses on the performance of the proposed framework to demonstrate its robustness and feasibility of a concrete implementation. Our measurement units are sessions. We consider each session as the time from when the application is launched until it is closed. Sessions were retrieved from 470 users in the test group, i.e., those running the targeting algorithm. We selected 725 sessions for the analysis based on the consideration that at least one agent was downloaded, and the evaluation and selection phases were successfully completed. This is to make sure that we excluded those who closed the app immediately after launching it. We also conducted an anomaly filtering, i.e., all sessions having unusual time duration for agents' downloads due to unexpected networks, hardware, or software behavior were excluded. The descriptive statistics of the 725 sessions are shown in the table below.

Table 2. Descriptive statistics of the sessions. User hardware: iPhone 3G/3GS.

	Min.	**Max.**	**Mean**	**Std. Dev.**
# of Agents Downloaded	1.00	50.00	40.13	15.85
Avg. Download Time (in sec.)	0.00	37.00	5.24	5.17
# of Carried Ads Displayed	1.00	71.00	11.03	6.96
Processing Time (in sec.)	0.00	48.00	3.03	8.55

The average number of the carried advertisements displayed per session is 11.03 and the average time to process the downloaded agent is 3.03 seconds. Hence approximately 0.27 seconds are needed to execute the match estimator code of each agent and decide if the agent's carried advertisement should be displayed to the consumer. The average time to download an agent is 0.13 seconds. By analyzing the collected log entries, we estimate that the total time to process an agent (download + code interpretation + selection) is around 0.4 seconds.

Usage Analysis. Both test and control groups used their respective application mostly during the weekdays. As shown in Figure 5, there is no significant difference in the time-of-the-day when the test and control groups launched their respective application. However when we compared the number of times the test group launched our targeted advertising application (c.f. the control group) with an independent sample t-test analysis, we found a significant difference in the launching frequency of the test and control groups (sig. 0.006). The test group who had the targeted advertising application launched the application 18.2% more as compared to the control group. On average the application was launched 11.7 times by the test group and 9.9 times by the control group.

Besides examining the application launch, we also examined four common events in the test and control groups, i.e., open the details of discount information (eventID 7), rate the discount information (eventID 10), and sort the discount information based on the ratings (eventID 200) or retailers (eventID 201). We found that besides clicking on the advertisements to see more details, users in both groups extensively used the app feature that allows them to sort the advertisements based on the retailers (see Figure 5). Most probably this is because they are loyal to a specific retailer, and were using this feature when they were inside the shop. We also noted that application launches (EventID 1) reach peeks at 9h, 12h, 17h and 21h, which indicates that users prefer to browse for offers during breaks and in their free time, and most preferably in the evening.

The effectiveness of our targeting algorithm is proportional to the amount of user information in the local repository. The more often the application was used, the more information was collected in the local repository, and thus the better

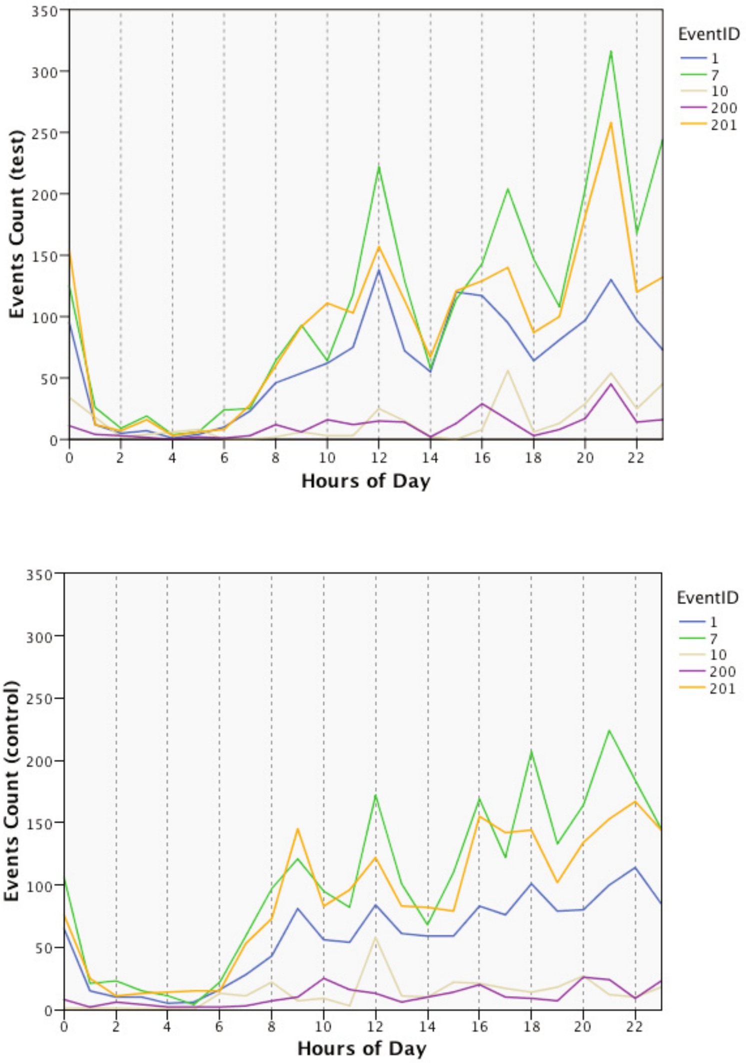


Fig. 5. Hourly user activity of the test (above) and control group (below). Events IDs: 1='app launched', 7='read discount', 10='rate discount', 200='sort discounts by rating', 201='sort discounts by retailer'.

the performance of our targeting algorithm should be. To gauge satisfaction with the displayed advertisements, we compared the average ratings given to the discounts information in the test and control groups. As shown in Figure 6, the test group had greater satisfaction with the displayed advertisements as they gave significantly higher ratings to the advertisements. The longer they used the targeted application, the higher the ratings they gave to the displayed advertisements. This confirms the effectiveness of our targeting algorithm.

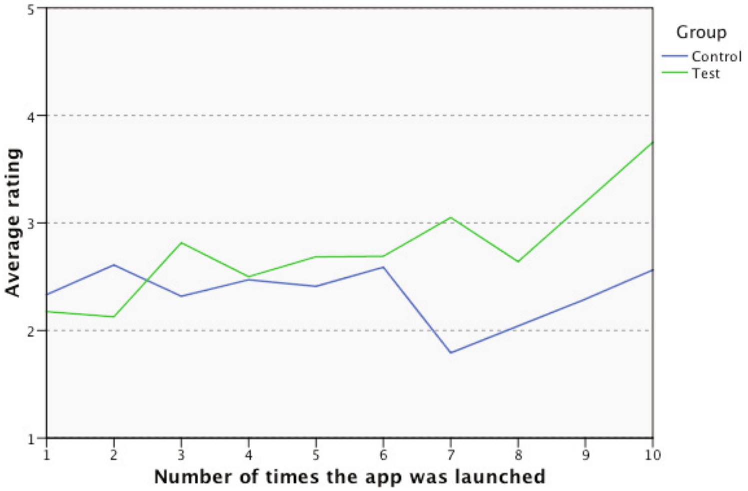


Fig. 6. Average ratings of the displayed advertisements, depending on the number app usages, for test and control group

5 Privacy and Threats

Having claimed that our targeting framework is privacy-preserving, in this section we argument on threats on user privacy, and how they are mitigated.

In our setting we define privacy the following way: no attacker should be able to derive any user profile data. Conceptually, our framework fulfills this property, because targeting is done on the user device itself rather than on a untrusted remote server, and that no user profile data is transmitted to any second party.

To break the privacy property, the goal of an attacker would be the following two things. 1): accessing user profile data on the user device and transmit it to the attacker or 2): deriving user profile data implicitly from user behavior.

Let's first consider the first point: One possibility to access the user data would be to install a malicious program on the user device that reads out the desired values and that transmits them to the attacker's server. Here we rely on the smartphone OS security policy (in our case iOS) and our configuration that disables read access from other apps to our app's data. Apple further tightly reviews any app before it can be published to the app store, which further diminishes this threat. Another potential threat comes from the JavaScript targeting codes that run on the user device and that have access to profile data. Here we rely on the security of the sandboxed Webkit instance that runs the JavaScript. If configured accordingly, any data transmission in the sandboxed environment to the external is blocked. Therefore, even if malicious JavaScripts were able to reach the client they would be unable to communicate any stolen information. In practice the sandboxed environment may experience vulnerabilities that allow to circumvent the security policies. Browser security and especially sandboxing is however widely researched and implemented nowadays (e.g., [20]), so that we decide to rely on the Webkit implementation in our case.

The second point is mitigated because in our discount discovery application, no click data is transmitted to any server. In contrast to the presented case study, marketers however often require the advertising publishers to provide statistical feedbacks on the advertising campaign (E.g. impressions, clicks). In case of the presence of a statistical feedback loop, the customer privacy could theoretically be compromised. In theory a malicious server could estimate the customer profile by analyzing the client feedbacks. To mitigate probing attacks, security mechanisms for anonymizing requests such as anonymous ping, delay tolerant networks [12] and onion routing [10] can be combined to the presented framework. Passive eavesdropping attacks would so be circumvented. Hence successful attacks would have to actively alter the advert server's normal behavior. For example, if the server only sends the advertisement agents to one mobile client, he could track his clicks even if the clicks are anonymized.

Summarized, our implementation fulfills the proposed privacy policy. In a future implementation that enables to track clicks, the mechanisms described in this section should be implemented.

6 Discussion and Further Work

While offering numerous benefits, our proposed framework may encounter resource management problems when the match estimator codes are too complex or the repository is too big such that they may overload the mobile advertising-agents' client. In our field experiment, we overcame this potential issue by limiting the number of agents sent per session (max. 50 agents per session), limiting the code size of the agents (max. 1KB per agent), and meticulously defining which information should be collected and stored in the consumers' repository. While they may not be ideal measures, these three measures could prevent the resource management problem from happening.

A second limitation of our framework is that it may not be able to address special cases such as wrong match estimator codes and codes that never terminate (infinite loops). By now our assumption is that the server delivers agents with valid code. Nevertheless, our field experiment showed that the proposed framework has well-performed and such special cases could be prevented. In our field experiment, all agents' match estimator codes were computationally generated by assembling multiple previously verified pieces of codes.

The above limitations present opportunities for future research. Future research can also benefit from this study as it offers detailed guidelines on how to implement and field test a proposed framework. Such field test is currently lacking in most of the previous studies proposing targeted advertising methods.

In the field experiment we assessed the effectiveness of the targeting algorithm by using the framework in a grocery discounts-discovery context, we examined the application performance and usage, and compared consumer satisfaction with the targeted advertisements vis-à-vis randomly displayed advertisements.

Future research could further compare our mobile-advertising agent framework with other existing frameworks for targeted advertising. An indication for improved user-satisfaction with privacy-preserving targeting vis-à-vis non-privacy-preserving targeting is given in [23].

By addressing the security of the proposed framework two important aspect need to be mentioned: First, it is crucial that the targeting code runs in an environment that prevents any Internet connection and any profile modifications. We here rely on a sandboxed environment (i.e. Apple WebKit) that fulfills these requirements. So even malicious advertising agents are unable to transmit or alter any private information. Second, advertisers generally like to track the clicks of their advertising campaigns. In our current implementation this is not possible as this would leak information on who clicked on the ads. A way to overcome this limitation is to anonymize the click information. This can be achieved with onion routing [10], anonymous pings or delay tolerant networks [12]. Given that many threats on privacy arise because advertisers wish to track user clicks for payment reasons (pay-per-click), we finally highlight the need for further research on privacy-preserving ad-payment schemes.

7 Conclusion

In this paper, we presented a novel framework of agent-based targeted advertisement that allows for advertisement targeting while preserving users' privacy. We presented its concrete implementation with a grocery discounts-discovery application for iPhone, and evaluated it in a large-scale field experiment with 903 real users. Based on the field experiment, we analyzed among other things the application performance and user satisfaction. We showed that the total time needed to fetch and display an agent-based advertisement in our testing environment (iPhone 3G/3GS) is a mere 0.4 seconds. To assess the effectiveness of our targeting algorithm, we compared the average ratings given to the targeted advertisements vis-à-vis randomly displayed advertisements. We discovered that users were generally more satisfied with the targeted advertisements as reflected in the significantly higher ratings given to the targeted advertisements. Moreover, we discovered that the longer they used the targeted application, the higher the ratings they gave to the displayed advertisements. This paper demonstrates that mobile agents can be effectively used to achieve targeted advertisement. The computational overhead imposed by the mobile agents is minimal and therefore tolerated by end users. Furthermore, we demonstrated how mobile agents can be used to preserve user privacy. We presented a validated framework that can be used as base for future development of privacy-safe and context-aware mobile target content delivery.

References

1. Aaker, J., Brumbaugh, A., Grier, S.: Nontarget markets and viewer distinctiveness: The impact of target marketing on advertising attitudes. Journal of Consumer Psychology 9(3), 127–140 (2000)

2. Aalto, L., Göthlin, N., Korhonen, J., Ojala, T.: Bluetooth and wap push based location-aware mobile advertising system. In: 2nd International Conference on Mobile Systems, Applications and Services, pp. 49–58 (2004)
3. Aggarwal, P., Krishnaswamy, D., Daley, R.S., Lundqvist, P.: User profile generation architecture for mibile content - message targeting. U.S. Patent Application No.: 0319329A1 (20090319329) (December 2009)
4. Alt, F., Balz, M., Kristes, S., Shirazi, A., Mennenöh, J., Schmidt, A., Schröder, H., Goedicke, M.: Adaptive user profiles in pervasive advertising environments. In: Ambient Intelligence, pp. 276–286 (2009)
5. Altmeyer, C.: Smartphones, social networks to boost mobile advertising (June 2009), `http://www.reuters.com/article/idUSTRE55S2FY20090629`
6. Barnes, J., Distler, P., McMullen, M., Sharma, S.: Architecture for mobile advertising with location. US Patent Application No.:0227467A1 (March 2007)
7. Bulander, R., Decker, M., Schiefer, G., Kolmel, B.: Comparison of different approaches for mobile advertising. In: 2nd IEEE International Workshop on Mobile Commerce and Services, pp. 174–182 (July 2005)
8. Farmer, W., Guttman, J., Swarup, V.: Security for mobile agents: Issues and requirements. In: 19th National Information Systems Security Conference, vol. 2, pp. 591–597 (1996)
9. Gao, J.Z., Ji, A.: Smartmobile-ad: An intelligent mobile advertising system. In: 3rd International Conference on Grid and Pervasive Computing Workshops, pp. 164–171 (2008)
10. Goldschlag, D., Reed, M., Syverson, P.: Onion routing. Communications of the ACM 42(2), 39–41 (1999)
11. Guha, S., Reznichenko, A., Tang, K., Haddadi, H., Francis, P.: Serving ads from localhost for performance, privacy, and profit. In: 8th Workshop on Hot Topics in Networks (2009)
12. Haddadi, H., Hui, P., Brown, I.: Mobiad: private and scalable mobile advertising. In: 5th ACM International Workshop on Mobility in the Evolving Internet Architecture, pp. 33–38 (2010)
13. Langheinrich, M.: Personal privacy in ubiquitous computing. Diss., ETH Zürich, Nr. 16100 (2005)
14. Mahmoud, Q., Al-Masri, E., Wang, Z.: Design and implementation of a smart system for personalization and accurate selection of mobile services. Requirements Engineering 12(4), 221–230 (2007)
15. Masinter, L.: The "data" url scheme. IETF (August 1998), `http://tools.ietf.org/html/rfc2397`
16. Narayanaswami, C., Coffman, D., Lee, M., Moon, Y., Han, J., Jang, H., McFaddin, S., Paik, Y., Kim, J., Lee, J., et al.: Pervasive symbiotic advertising. In: 9th Workshop on Mobile Computing Systems and Applications, pp. 25–26 (February 2008)
17. Perdikeas, M., Chatzipapadopoulos, F., Venieris, I., Marino, G.: Mobile agent standards and available platforms. Computer Networks 31(19), 1999–2016 (1999)
18. Pham, V., Karmouch, A.: Mobile software agents: an overview. IEEE Communications Magazine 36(7), 26–37 (1998)
19. Ranganathan, A., Campbell, R.: Advertising in a pervasive computing environment. In: 2nd International Workshop on Mobile Commerce, pp. 10–14 (2002)
20. Reis, C., Barth, A., Pizano, C.: Browser security: lessons from google chrome. Communications of the ACM 52(8), 45–49 (2009)
21. Shannon, R., Stabeler, M., Quigley, A., Nixon, P.: Profiling and targeting opportunities in pervasive advertising. In: 1st Workshop on Pervasive Advertising at Pervasive (2009)

22. Specification, E.: Standard ecma-262. ECMA Standardizing Information and Communication Systems (1999)
23. Sutanto, J., Palme, E., Tan, C.H., Phang, C.W.: Addressing the personalization-privacy paradox: An empirical assessment from a field experiment on smartphone users (unpublished working paper, 2012)
24. Toubiana, V., Narayanan, A., Boneh, D., Nissenbaum, H., Barocas, S.: Adnostic: Privacy preserving targeted advertising. In: 17th Annual Network & Distributed System Security Symposium (2010)
25. Yang, W.S., Cheng, H.C., Dia, J.B.: A location-aware recommender system for mobile shopping environments. Expert Systems with Applications 34(1), 437–445 (2008)