A New Perspective in Pervasive Advertising

Lorenzo Carrara[1] and Giorgio Orsi[2]

[1]Cubica S.r.l
lorenzo.carrara@cubica.eu

[2]Oxford University
giorgio.orsi@ox.ac.uk

Abstract. In commerce, advertising a potential customer with relevant offers is beneficial for both the advertiser, who increases the chances of acquiring a new customer, and the customer himself, who is stimulated to purchase without being overwhelmed by irrelevant ads. To achieve this goal, the advertiser must know the situation, the interests and the activities of his potential customers; in one word, their context. However, the exposure or the collection of these personal data has a clear repercussion on their privacy and is often constrained by law. PervADs is a user-centred pervasive advertisement-distribution platform which maximizes the relevance of the ads sent to potential customers while preserving their privacy. The relevance is determined by semantically matching the profile and the context of the customer against those carried by the ads, while privacy is guaranteed by a client-side matching algorithm where user's personal data are never accessed by the advertisers or transmitted to a channel. PervADs provides small/medium businesses with an autonomous and inexpensive infrastructure for contextualized advertising, without the need to traditional mass advertising channels such as radio, TV and newspapers which often have prohibitive costs.

1 Introduction

The web and digital technologies revolutionized advertising by providing new and richer channels for reaching potential customers and by enabling new forms of interaction with them. Electronic and interactive advertising impacted also radio and TV that are traditionally mass and one-way media. Digital Audio Broadcasting (DAB), Digital Terrestrial TV (DTTV), and even the old-fashioned Teletext have been the means to provide interactive advertisements that try to emulate the two-way interaction of web advertising. At the same time, the advancements in mobile computing made possible to pervasively access other devices and online information sources through portable devices such as smartphones and tablets while maintaining a satisfactory user experience. One of the consequences of these developments has been the opportunity to advertise users in mobility during their everyday life activities through the web or through advertisements placed in mobile applications[1]. Mobile and pervasive advertising

[1] see e.g., the Android AppBrain application at http://www.appbrain.com/.

is expected to generate billions of dollars of revenues in the near future [10] but this new scenario heightened some of the most fundamental issues about advertising such as the *reach* i.e., the size of the audience to whom we will communicate, the *target rating*, i.e., how many times the advertisement reaches the targeted audience and, above all, user's *privacy*.

As a consequence of the enormous amount of advertisers, simultaneous advertising campaigns and the diversity of available channels, the random and massive distribution of advertisements constantly floods the users with rarely interesting commercial offers. This has a negative impact on advertising since the excess of irrelevant ads leads to disaffection towards the advertised product, the manufacturer, the product category and, in general, the advertising channel. A key tool used to maximise the relevance of an ads is therefore *user profiling*; collecting data about the current situation, interests and intentions of the user enables a profile-base crafting of the offers for products and services. In other words, in order to send relevant ads, the habits and tastes of the user must be known.

In the case of traditional media (e.g., radio, billboarding and TV), the context of the user can only be defined in a partial and inaccurate way. In billboard advertising, for example, posters can be placed in places attended by people who are probably interested in the subject of the advertisement. In a similar way, with radio and TV advertisements, the commercials can be proposed during a show whose audience is close to the target audience of the ads. These approaches have some evident weaknesses: (i) an ads rarely reaches the specific target while annoying the other individuals and (ii) there is no way to know whether the target is interested in the offer in the specific moment the ads reaches him (e.g., the commercial advertises food but the target just had lunch).

The above issues have also a considerable economical impact on the advertisers. An obvious quality metric for an advertising campaign is the return on investment (ROI), that is higher if the ads precisely reach the target audience. Billboards and paper advertisements have enormous costs of design, production and publishing and so is also for TV commercials. A report from MediaWeek shown that a single 30 seconds commercial on a US television costs in average 100K$ and doubles if the commercial is shown during popular TV shows to reach peaks of 2M$ during special events such as the Superbowl. This costs are prohibitive for the majority of the companies and especially for small and medium businesses (e.g., shops). As an answer to these costs, such business realities downgrade the advertising campaign i.e., radio instead of TV commercials and leaflets instead of billboards. However, the downgrading option considerably lowers the effectiveness of the advertising campaign while remaining sensibly expensive.

A remarkable step toward the democratization of advertising and the maximization of the target rating points has been the introduction of web advertising. Despite the wariness that many users show towards online advertising (including the authors of this paper), various studies confirm its overall effectiveness when compared to traditional media [21]. This is mainly due to the fact that online advertisements do not necessarily interrupt the user's activities (e.g., browsing or watching a movie) like TV commercials and is therefore more tolerated. In addition, the Web allows a more sophisticated and interactive advertising that exploits personalization and contextualization techniques. For example, search engines usually enhance their result-pages with coherent advertising offers depending on the user's search queries. Obviously, the more detailed the representation of the user's context is, the more effective these approaches become. This basic principle made the ownership of user's personal data one of the most pressing priorities for the communication industry and arose non trivial legal and ethical issues that affect also pervasive advertising. The aggressive behaviour of some advertising strategies recently forced the European Union to forbid certain practices like online data collection and user localisation through the EU Directive 2002/58/EC also known as E-privacy Directive. However, since its introduction, this directive created a loss of revenues of 65% for European online advertising [12], a situation that created problems not only to advertising channels but also to OEMs.

The ownership of data about users is one of the most valuable assets for realities such as social networks and web companies. Facebook and Google are amongst the most known companies that generate profit by distributing advertisements on their platforms. In order to access their services, users are required to authenticate themselves and accept data processing policies that allow the use of personal data for various purposes, sometimes also commercial ones. These policies have a great impact on privacy since, while interacting with these services, the user discloses (often unconsciously) personal information which can be used to progressively refine their profiles, which in turn is used to select relevant third-party advertisements. In many occasions, the users of these services rebelled against the provider's will to relax their privacy policies; however, a partial give up on privacy remains an intrinsically necessary condition for the survival of these companies, since they hold the role of mediators between the advertiser and the final customer.

On mobile devices the problems are not different and can become even more serious if we think of the possibility of locating the exact position of the users through services such as Facebook places[2] and Google Latitude[3]. Finding a compromise between the need for advertising of the companies and the quality and the number of ads received by an individual is the goal of *PervADs*.

PervADs stands for *Pervasive Advertisement* and is a novel paradigm for pervasive, context-aware and user-centred advertising that enables one-to-one interaction between small/medium shops and potential customers who, due to their current context, can be interested in purchasing their products. PervADs transfers the control of the advertising process to the user and accesses voluntarily-provided personal data to filter digital advertisements locally on the users device without disclosing personal data. On the other hand, PervADs provides small/medium businesses with an autonomous and inexpensive advertising infrastructure that replaces traditional paper-based advertising such as leaflets and

[2] `http://www.facebook.com/places/`.
[3] `http://www.google.com/latitude/`.

billboards. PervADs removes the need for intermediate third-party advertising companies (e.g., TBWA) and the dependency from a service provider (e.g., Virgin Media) since the interaction occurs on a local area network and not on the web.

The approach. In PervADs, advertising is carried out by distributing to the users digital advertisements (i.e., the PervADs) describing a certain number of commercial offers about products, services or combinations thereof. A PervAD consists essentially of two elements:

- *Advertisement description.* Each PervAD contains a formal and machine-readable description of the offer in terms of an ontology shared by advertiser and user applications. These data include a categorization of the offered product or service as well as the type of offer (e.g., discount or special price) and the details about the advertiser. The user interacts only with the embedded media such as e-coupons, images and audio/video documents.
- *Advertisement context.* Each PervAD targets a specific customer profile, in a given situation and for a certain period of time; in other words, in a given context. This information is also part of the PervAD and allows to: (i) reach only individuals within the targeted audience that are potentially interested to the product. (ii) reaching individuals outside the targeted audience that, due to their current situation, might be interested in an occasional purchase of the product (*transumerism* [27]).

A typical PervADs scenario is sketched in Figure 1 and involves mainly two actors: the *advertiser* and the *user*. The former is equipped with a PervADs transmitter i.e., a customized WiFi router which is used to craft the PervADs and to transmit them on a WLAN. On the other hand, the user is equipped with a PervADs receiver, i.e., an application installed on his mobile device which can receive and filter the PervADs on the basis of the user's context.

A typical PervADs interaction works as follows. The PervADs receiver scans the available WiFi networks in search of a PervAD transmitter. When it finds one, it downloads the contexts associated to the advertisements and, once it has compared them with that of the user, it downloads and displays the relevant offers while discarding the others. The entire process is completely anonymous; personal data are never transmitted over the channel and the PervADs transmitter receives only a notification if the advertisement has been evaluated as relevant by the receiver based on the user's context.

The infrastructure is based on standard local area network technologies (WiFi or WiMax) without the need for accessing the Web. The Communication between the user and the advertiser is free-of-charge since users can access the hotspot with a WiFi-enabled phone with no cost for data traffic.

The same infrastructure can also be used to convey non commercial information such as news, cultural events and emergency information.

Three main use-cases characterize the interaction of a user with the PervADs system:

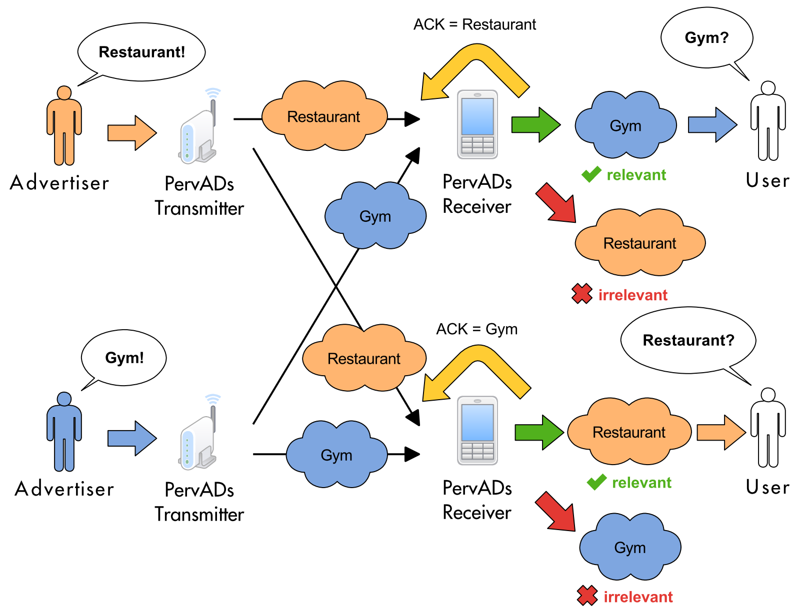

Fig. 1. Advertisement distribution with PervADs

- *Self profiling.* The user constructs his profile that the PervADs receiver will use to compute the relevance of the PervADs. These data are organized in two distinct sets: (i) *user-profile data*, not bound to a specific user task, i.e., describing time- and situation-independent facts such as age, gender, dietary preferences or religion. The profile is usually defined the first time the user configures the application. (ii) *Search queries*, describing contingent needs and asynchronously activated by the user. As an example, the user might configure a query to scan for lunch-offers in restaurants located within 500 metres from the user current position at lunch time.
- *PervADs gathering.* PervADs can be collected in two ways: by (i) *synchronous scanning*, i.e., the user actively asks for a PervADs refresh, possibly specifying one or more search queries. The new PervADs are received and stored on the user device that filters and presents to the user only the relevant offers. A second paradigm is (ii) *asynchronous scanning*, i.e., the user activates a passive refresh of the PervADs. The application passively updates the PervADs, filters them on the basis of the contextual data and notifies the user when new relevant advertisements are received.
- *PervADs interaction.* When the user is notified about a new PervAD he can access the human-readable content of the advertisement, i.e., the coupon. These coupons are attached to the PervAD itself and enable the user to access a more detailed description of each offer or discard it. The coupons can contain multimedia documents describing the offer (e.g. images, audio,

video or textual resources), structured details about the offer or the offering business, and links to external resources (e.g., the advertiser's website or a booking form for a table in a restaurant).

Advertisers interact with the administration application installed on their PervADs transmitters. These devices are in fact specialized WiFi routers, which control a PervADs wireless local network. The installed application provides a simple web server, to serve the PervADs requests from the receivers, and a backend UI, where the advertiser can define the PervADs. The main tasks of the advertisers are:

– *PervADs creation and management.* The advertiser defines the offers of the ads and its context describing the target user profile.
– *Performance analysis*: Each PervAD can be monitored in order determine its effectiveness and success. Performance analysis can be accomplished by the standard means used in web advertising. For example, the receiver can notify the transmitter whenever a pervAD is evaluated as relevant by the user application and when it is actually visualized by the user. In this way, the advertiser can determine whether the target profile he created for the ads is correct or the offer is appealing to his potential customers.

Contributions. In this paper we propose: (i) an extensible formalism for the description of semantic and pervasive advertisements, (ii) a publish/subscribe architecture for user-centred, pervasive advertisement, and (iii) a filtering mechanism for pervasive advertisements based on context matching and reasoning.

Roadmap. In the next section we overview the literature on pervasive advertisement and its related fields, positioning PervADs accordingly. Section 3 describes in detail the structure of a PervAD, while in Section 4 we go into the details of PervADs filtering, discussing how ontologies contribute to this task. The implementation of the system is described in Section 5, while preliminary experimental results are presented and discussed in Section 6. The paper concludes with Section 7 by giving some insights on future extensions of this work and by drawing few concluding remarks.

2 Related Work

Computational advertising is a research area concerned with problems and solutions in automated advertising. The main problems in this area include: (i) *contextual advertising*, i.e., placing advertisements on the basis of the interests and the activities of the user, (ii) *sponsored search*, i.e., delivering advertisements on the basis of user queries on a search engine, (iii) *behavioral targeting*, i.e., exploiting data collected from the online behaviour of the user to target the advertisements and, (iv) *recommender systems*, i.e., recommending products or services by comparing the user profiles and behaviours.

Context awareness is a key factor in pervasive and mobile advertisement since it affects the effectiveness of an advertising campaign. In most of the current approaches [25, 1, 22, 11, 16, 9, 23], the context of the user is represented as a combination of his location and profile represented as key-value pairs or as an organised taxonomy of keywords. As a consequence, users and advertisements are matched using syntax-based techniques mainly coming from information retrieval, such as the vector-space model and the cosine similarity. Various studies in computational advertising demonstrate how a purely syntactic matching often yields a poor performance in targeting the advertisements since plain keywords carry poor semantics. In PervADs plain keywords are used only at the very end of the advertisement selection process to distinguish between equally interesting advertisements.

Some systems tried to solve this problem by *inferring* the context from the activity of the users [13, 20, 6] through the analysis of SMSs, MMSs, emails and browsed web pages, or by exploiting social knowledge [8], e.g., by counting how many times similar users click on an advertisement. However, this requires a discrete amount of resources in order to analyse the vast amount of data generated by the users on their devices. PervADs determines the behaviour of the users using the evolution of his context and provides computational guarantees (i.e., polynomiality) for this analysis.

Another direction is to adopt semantic resources [14, 17, 5, 29, 19, 28, 24, 26] (e.g., DBPedia, WordNet, ontologies, gazetteers) possibly coupled with probabilistic models [30] to improve the targeting of the advertisements. These approaches have shown to be more reliable than simple vector-based metrics and confirm that a semantic representation of context, such as the one of PervADs, gives the best results in terms of accuracy and reliability in advertisement relevance computation, w.r.t. purely syntactical models.

Besides contextual matching, the main feature of PervADs is that it guarantees the privacy of the user. There is a general agreement in European regulators and in the literature over the need for enforcing strict privacy policies in context-aware pervasive computing environments [21, 15, 2, 6]. In accordance to the principles outlined in [21], PervADs never sends the context of the user to the advertisers or to any mediator. In addition, the matching between the context of the user and that of the advertisement is performed locally on the device; the client application only resorts to external services for geocoding purposes. This prevents advertisers from profiling users, while allowing the measurement of advertisements performance. Regarding privacy aspects, PervADs is the only system that completely shifts the control of the advertisement process in the hands of the users, excluding not only the advertisers but any intermediate entity.

3 PervADs

A PervAD is an electronic advertisement organised in two sections: (i) a machine-readable description of the advertisement (called *descriptor*), and (ii) a human-readable resource such as a graphic coupon, a video or a web page (called *flyer*). The descriptor provides data about one or more commercial offers plus context data defining the target of the advertisement. The advertisement description is defined using OWL[4] whose extensible vocabulary and formalised semantics are particularly suited for our modelling needs. On the other hand, the context of the advertisement has been modelled using the Context-Dimension Tree (CDT) [4], a very expressive context model originally used for personalisation and data tailoring in relational databases [3]. In this work we adopt one of the extensions of the CDT called Context-Dimension Ontology (CDO) that has proven to be an effective tool in personalised search [18].

3.1 Advertisement-description vocabulary

A semantic and machine-processable description of the advertisement allows to abstract from the presentation details of the offer and to focus on its actual content. The advertisement-description vocabulary provides concepts and relations used to model the different types of commercial offers (see Figure 2).

Each PervAD is modelled as an instance of the class `PervAD` and is associated to: (i) one or more context-definitions, (ii) the organisation transmitting the advertiser, and (iii) the actual commercial offers, specialised into special prices and discounts. The ontological representation of the PervAD makes easy to extend the vocabulary with new classes and relations using the extensible features of OWL. The offers can be associated with an arbitrary number of media (e.g., coupons, videos and pictures) through the `attachedMedia` property. If the offer has a limited duration, the limits of validity can be expressed using the `validFrom` and `validUntil` properties. Each `Offer` cites at least one `OfferedItem` by means of the `offers` property that is then refined into `Product` and `Service`.

Advertisers are modelled by means of the `Organization` class which carries a number of properties such as the `name`, `address`, `email` and `website`. These data will be exploited by the user application in order to ease the interaction between the users and the organizations, e.g., by providing shortcuts to a phone number or to show their location on a map.

Clearly, only a finite number of PervADs can be shown to the user on his device. In order to break ties when one of more PervADs match the user context with the same matching score, the advertiser can annotate his PervADs with tags. During the selection of the relevant PervADs, tags are used to discriminate between advertisements on the basis of their similarity with user search keywords. The similarity between user keywords and tags is computed by representing both sets as vectors of terms and by computing the cosine similarity.

[4] Web Ontology Language: `http://www.w3.org/TR/owl2-overview/`

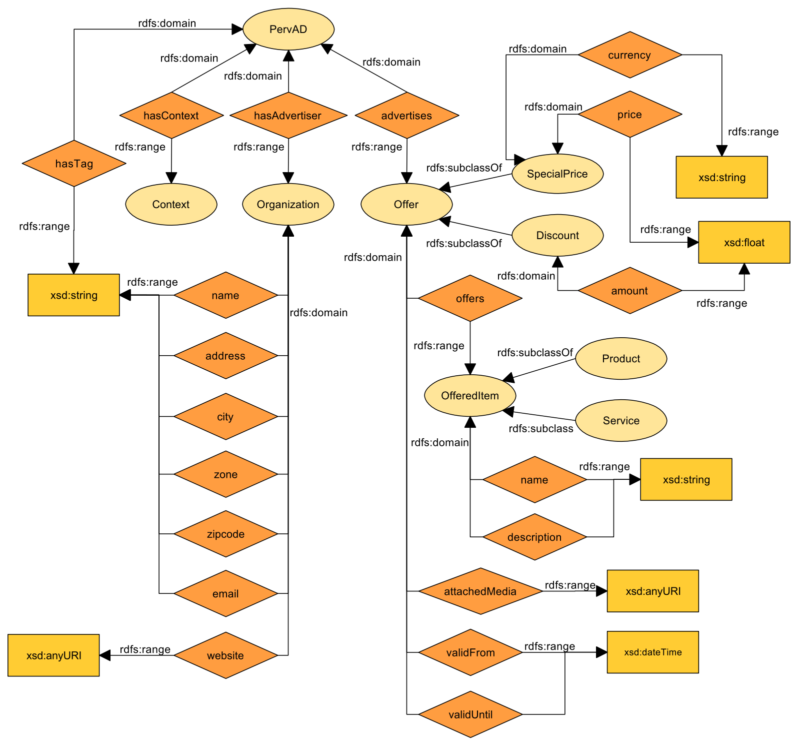

Fig. 2. Advertisement-description vocabulary

3.2 Context modelling

The contextual data in a PervAD are organised according to the *Context-Dimension Ontology* (CDO) that provides an extensible vocabulary to model hierarchical and multi-dimensional context-models.

The basic elements in a CDO are the context dimensions, e.g., `time`, `space`, `interest-topic`, that are then assigned to values, e.g., `now`, `within 5 miles`, `food`. The CDO allows hierarchical dimension assignments e.g., an assignment `region=chinese` can be modelled as a special case of an assignment `interest-topic=food`. Another useful feature is the possibility to decorate the values with parameters describing specific facets. Consider, as an example, the assignment `location=nearby`, specifying that the current context considers locations nearby the user location. In this case, the value `nearby` can be parametrized with a variable `$radius` specifying the radius we are referring to, e.g., `within 50 metres`.

Definition 1. *A context model $\mathcal{M}$ is defined as:*

$$\mathcal{M} = \langle D, V_D, P, V_P, d\text{-}value(\cdot), p\text{-}value(\cdot), parent, r\rangle$$

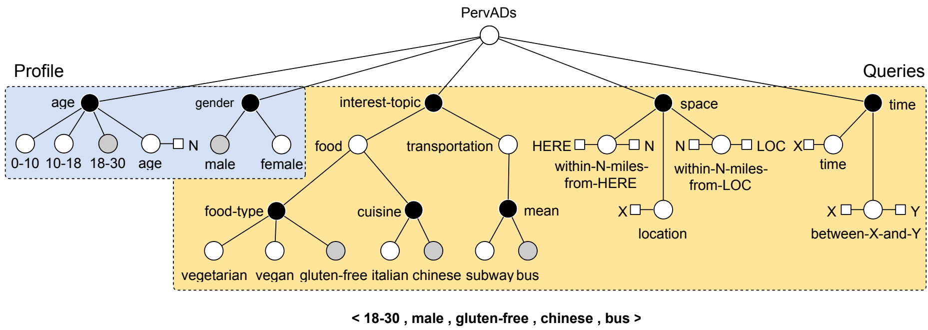

Fig. 3. The pervADs context model (excerpt)

where: (i) D is the set of dimensions of the context model; (ii) P is the set of parameters; (iii) V_D is the set of dimension values; (iv) V_P is the set of parameter values; (v) d-value$(\cdot)$ is a relation that maps each dimension value $v_d \in V_D \setminus \{r\}$ to a single dimension $d \in D$, (vi) p-value$(\cdot)$ is a relation that maps each parameter $p \in P$ to a single value $v_d \in V_D$. (vii) parent$(\cdot)$ is a relation that maps each dimension $d \in D$ to a single value $v_d \in V_D$. (viii) r is a special value representing the entire information space.

In addition, the model requires that the structure obtained by considering the elements in $D \cup V \cup P$ as nodes and the pairs (d, v_d) such $v_d = parent(d)$ as edges is a tree rooted in r.

A graphical representation of the PervADs CDO is shown in Figure 3 where dimensions and values are represented as black and white nodes respectively while parameters are represented as square nodes. In this example the root of the tree is the `PervADs` node, representing the context of our application. A dimension assignment is represented as an edge between a white node and a black node while a sub-dimension is represented as an edge between a white and a black node. Here we represented five dimensions: `age`, `gender`, `interest-topic`, `space` and `time` with the corresponding values. The age and gender dimensions are part of what is usually considered the user-profile, while space and time are more common geo-localization dimensions. However, as we know, the notion of context goes beyond profile data and extends to any information that can be used to specify the current situation of the user and of the application including transient properties such as the current tastes of the user. The pivot dimension for PervADs is the `Interest-Topic` that in our model consists of more than 3000 possible assignments from the Yellow Pages[5] categories organized in a hierarchy of about 150 sub-dimensions.

Definition 2. *An instance for a context model $\mathcal{M}_C$ is a pair $\langle \mathcal{I}_C, \mathcal{I}_P \rangle$ where $\mathcal{I}_C \subset$ d-value$(\cdot)$ and $\mathcal{I}_P$ is a relation that maps a parameter $p \in P$ to a parameter value $v_p \in V_P$ where:*

[5] http://www.yell.com/

– *if* $(v, d) \in \mathcal{I}_C$ *and* $(d, v') \in parent(\cdot)$ *and* $(d', v') \in d\text{-}value$ *then* $(d', v') \in \mathcal{I}_C$.
– *if* $(v, d) \in \mathcal{I}_C$ *and there exists* $v' \in V_D$ *such that* $(V', d) \in d\text{-}value(\cdot)$ *then* $(V', d) \notin \mathcal{I}_C$.
– *if* $(v_p, P) \in \mathcal{I}_P$ *and there exists* $v \in V_D$ *such that* $v = p\text{-}value(p)$ *then there exists* $d \in D$ *such that* $(v, d) \in \mathcal{I}_C$.

Roughly, a context instance $\mathcal{I}_C$ is a set of dimension assignments organized in a tree hierarchy. The first condition ensures that whenever a sub-dimension is assigned to a value all its parent dimensions are associated to values according to the context-model. The second condition ensures that each dimension is assigned to a single value while the third condition ensures that whenever a parameter for a value is instantiated, then there exists an assignment of a dimension to that value in the context-instance.

A possible context instance can be graphically represented as a selection of white nodes (with associated parameters) such that, for each dimension, only one white node belongs to the set. Consider again Figure 3; the darkened nodes identify the context of 18-30 years-old man interested in gluten-free chinese restaurants reachable by bus. Notice that time and space dimensions have not been assigned; in PervADs unassigned dimensions are interpreted with a don't care semantics and are therefore ignored during the filtering phase.

4 PervADs Filtering

PervADs filtering is the core feature of PervADs as it allows to filter the advertisements on the basis of the current user situation and interest. the filtering is *user-centric* and *privacy-preserving* and is based on *context matching*. The algorithm matches the context of the user with that of the PervADs directly on the user's device thus preventing context data from being transmitted over the network. This in turns prevents advertisers or service providers from collecting valuable personal data. In addition, the matching algorithm exploits *context-validation*, preventing advertisers from associating general contexts to their PervADs that could match any user context.

The matching algorithm takes as input two context instances and produces as output a similarity measure called *context similarity*. The algorithm proceeds by first computing the similarity between each assignment (v, d) of values to the leaf dimensions in the user's context and all the assignments in the PervAD's context. This similarity is called *dimension similarity* (denoted by $sim((v, d), \mathcal{I}_P))$ and is computed as follows:

– if $(v, d) \in \mathcal{I}_U$ and $(v, d) \notin \mathcal{I}_P$ then $sim((v, d), \mathcal{I}_P) = 0$.
– if $(v, d) \in \mathcal{I}_U \cap \mathcal{I}_P$ then $sim((v, d), \mathcal{I}_P) = \frac{1}{1+\delta}$, where δ be the number of assignments $(v', d') \in \mathcal{I}_P$ such that d' has the dimension d has ancestor.

Roughly, if the PervADs context instance does not contain the assignment (v, d) of the user's context instance, then we assume no similarity (i.e., $sim = 0$). Otherwise, we consider the number of assignments to the sub-dimensions of d.

If the PervAD's context instance contains only the assignment (v, d) with no specializations, we get the maximum similarity (i.e., $sim = 1$) otherwise we reduce its similarity by a factor proportional to the number of the assignments for the descendant dimensions of d. This mechanism discourages the advertiser from specifying too-general contexts and encourages the user to be as precise as possible in the definition of its context.

The context similarity $sim(\mathcal{I}_U, \mathcal{I}_P)$ is an aggregated representation of the dimension similarities and is computed as follows. For all $(v, d) \in \mathcal{I}_U$:

- if $(v', d) \in \mathcal{I}_P$, with $v' \neq v$, then $sim(\mathcal{I}_U, \mathcal{I}_P) = 0$.
- otherwise $sim(\mathcal{I}_U, \mathcal{I}_P) = \frac{sim(\mathcal{I}_U, \mathcal{I}_P) + sim((v,d), \mathcal{I}_P)}{2}$.

Roughly, if the same dimension is assigned to different values in the user's and in the PervAD's context, then the contexts are incomparable and the similarity is zero. Otherwise, the similarity is the average similarity of the dimension similarities.

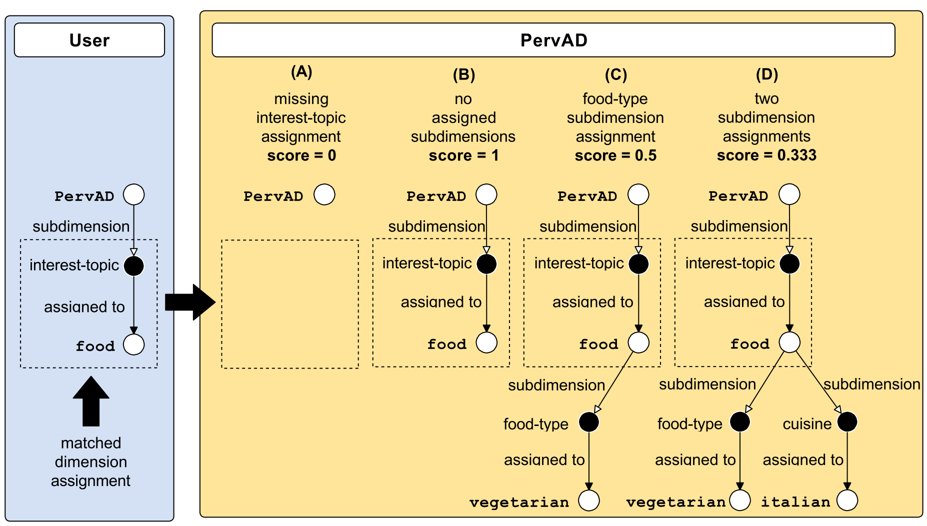

Fig. 4. Context-distance computation

Figure 4 shows some examples of context-distance computation. In situation (a) the context of the PervAD does not define an assignment for the `interest-topic` dimension, so the distance similarity for `interest-topic` (and for the entire context instance) is zero. In the second case (b) the PervAD's context defines an assignment to the same value of the user's context and does not assign any value to its sub-dimensions, thus, its dimension-similarity is 1. Cases (c) and (d) add to case (b) one and two assignments to sub-dimensions respectively. The dimension similarity in these cases is 1 for (c) and 2 for (D) resulting in a context similarity of 0.5 and 0.33 respectively.

In PervADs reasoning is performed on context instances in order to (i) validate them and (ii) infer implicit knowledge. Validation is the process of determining whether a given context instance is consistent w.r.t. the context model while inference is used to compute implicit assignments of values to dimensions. Figure 5 shows an example of context inferencing where the source context instance contains an assignment of the `food` value to the `interest-topic` dimension. The target context, instead, contains only an assignment of the `vegan` value to the `food-type` dimension but the assignment `(food,interest-topic)` is automatically inferred due to the constraints of the context-model.

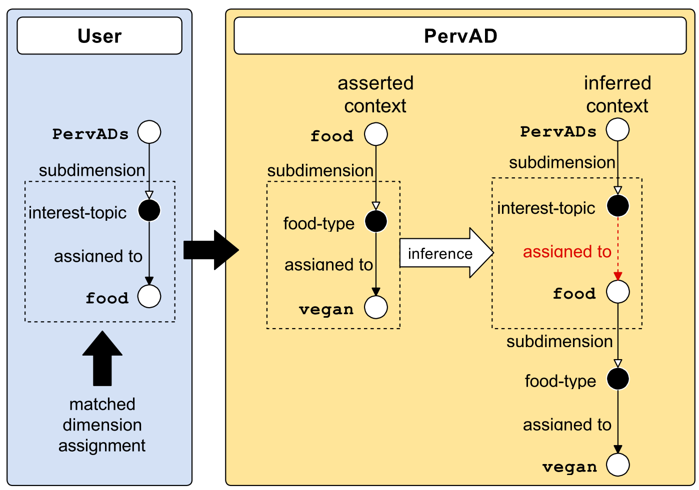

Fig. 5. Context reasoning

In pervADs context-reasoning is reduced to ontological reasoning by translating the context-model and the context-instances in ontological format and by leveraging on existing inference procedures. In particular the expressive power of the context-model falls into that of DL-Lite ontologies [7], therefore, if we fix the context model (such in PervADs) reasoning over the context-instances can be done highly efficiently (i.e., in LOGSPACE complexity) and is not more expensive than executing SQL-queries over a relational database. If we consider variable context-models the complexity of reasoning jumps to NP-complete.

5 System and Implementation

PervADs basically operates in a publish/subscribe fashion. Figure 6 provides an overview of the system's architecture, the different modules and their interaction.

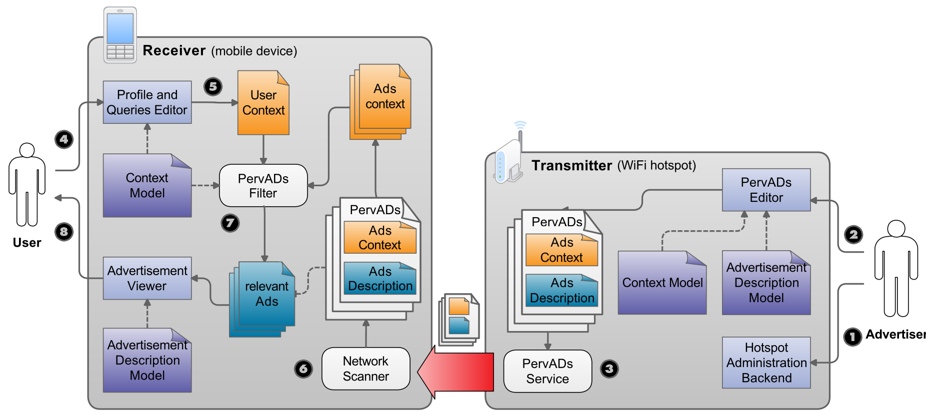

Fig. 6. PervADs architecture.

The PervADs receiver comes as an extension to OpenWRT[6], a customizable linux-based operating system for embedded devices, installed on a Linksys WRT54GL WiFi router. The PervADs extensions augment the router's administration interface with PervADs management functions, and provides a web server that allows clients to download PervADs. The PervADs server application is a LuCI[7] extension written in the Lua[8] language. The LuCI framework provides a web-based administration interface which allows a fine-grained management of every device feature, from simple networking configuration to the setup and management of a RADIUS server or PBX station. The web server support is provided by the uHTTPd package, a tiny single threaded HTTP server with TLS, CGI and Lua support.

The advertiser interacts with two modules: the *Hotspot Administration Backend* (1) and the *PervADs Editor* (2). The former is used for standard networking management while PervADs are created with the *PervADs Editor*. This module provides a user interface for the definitions of the description and the context of the advertisement that are then packaged together into the final PervAD by the *PervADs Editor*. This module is also responsible for PervADs management e.g., modification, deletion and monitoring of PervADs.

The *PervADs Service* (3) is a transparent layer that provides an external interface for clients. We implemented this module as a minimal web server that listens for client requests, and sends stored PervADs through HTTP protocol. In future releases, this module will provide advanced features such as peer-to-peer exchange of PervADs between servers.

[6] `http://www.openwrt.org`.
[7] `http://luci.subsignal.org/`.
[8] `http://www.lua.org/`.

The PervADs client has been developed as an application for the Google Android[9] platform and uses the AndroJena[10] library to model and manage the ontological models on the mobile device. Androjena is an open-source porting of the OpenJena[11] library for Android-based devices and includes SPARQL and inference support. Figure 7 shows our prototype user interface.

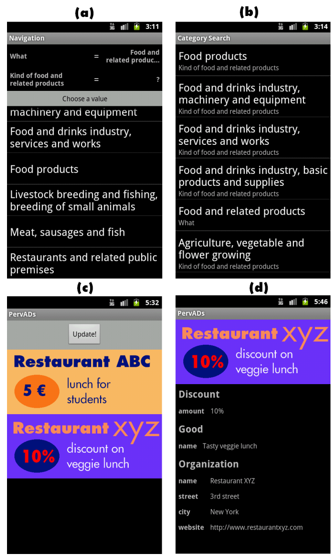

Fig. 7. The PervADs prototype GUI.

The user has to configure the PervADs application by filling his profile and by defining the search queries with the *Profile and Queries Editor* (4). Since both profile and queries are sets of dimension assignments, they are defined using the same interface, only at different times: profile is defined when the user initially configures the application, while queries are created, deleted, activated and deactivated during the whole application life-cycle and based on the evolution of the user needs. The product of the previous activities is the *User's Context* (5).

Now that user's context is in place, he can actively query for relevant advertisements from nearby hotspots or waiting to be notified about available offers. The *Network Scanner* module scans reachable WiFi networks looking for PervADs transmitters. Whenever it finds a PervADs network, it connects to the *PervADs Service* of the server and downloads the available PervADs (6). These PervADs are stored on the user's device for later user inspection.

[9] `http://www.android.com/`.
[10] `http://code.google.com/p/androjena/`.
[11] `http://www.openjena.org`.

Caching mechanisms for the PervADs and attached media are in place, e.g., to avoid multiple reception of the same PervAD. Whenever the *Network Scanner* finds and stores a new PervAD, it checks if it has already been seen before. This is done by checking two metadata that the *PervADs Editor* automatically bundles with each PervAD: a UID and a timestamp. The first is a global unique identifier, generated when the PervAD is first created; the second is an UNIX timestamp that marks the time of the last modification made to that PervAD. Using these metadata, the *Network Scanner* is able to recognise whether the newly downloaded PervAD is the same as another stored PervAD, and whether it is more recent.

The *PervADs Filter* module contains the core functionality of the whole system, as its purpose is to decide which PervADs are relevant and which are not. The filtering engine takes as input the user and the PervAD context and outputs the matching degree between them (7). The user will be then notified about the presence of relevant advertisements only when the matching degree between a query and a PervAD is above the acceptance threshold (8).

6 Experiments

The adoption of PervADs is strictly connected to the efficiency of the receiver. There are two main factors that affect the performance of the Receiver: (i) the efficiency of the scanning process, which must be fast enough to connect to the hotspots and download the PervADs while the user is moving and (ii) the efficiency of the matching process, which can be computationally expensive as it involves logical reasoning. Our experimental activity has been centred on measuring the speed of these two processes, in order to evaluate the behaviour of the system under critical conditions such as fast-moving users and large context models. All our tests have been performed using a Samsung Galaxy S i9000 smartphone, which features a 1GHz processor and has a 64 MBytes maximum heap size for each application, and a Linksys WRT54GL WiFi router, which is equipped with 4 MBytes of RAM and 16 MBytes of Flash memory.

6.1 PervADs Discovery and Download

When an update event is triggered by the PervADs Receiver, either synchronous or asynchronous, the application scans the nearby networks and, for each discovered PervADs hotspot, performs the following steps: 1. scan of the available hotspots, 2. connection to the hotspot's network, 3. download of the available PervADs and context-inference, and 4. retrieval and download of the human-readable media.

Figure 8 shows the time taken by each phase relatively to the signal strength of the hotspot's network. We chose to use the signal strength as a measure instead of the distance because the range of a WiFi router is affected, among the other factors, by its RF transmission power, the indoor/outdoor positioning and obstacles.

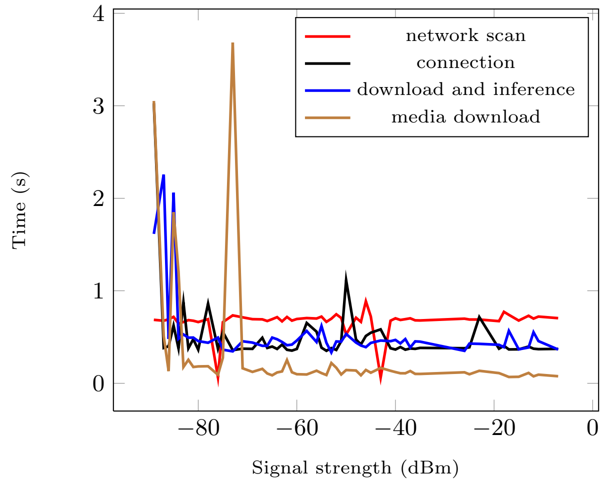

Fig. 8. Discovery and download speed

Our results show that the network-scanning time remains constant as it does not depend on the signal strength of the reachable networks, while the processing time is slightly affected by signal strength. This is mainly due to the intermittent connection and error correction mechanisms. In average, the total discovery and download time is approximately of 2 seconds, which we believe is a reasonable delay in a realistic usage scenario. Clearly, if the downloaded PervADs defines more than one offer, the coupon download time must be scaled accordingly.

6.2 Matching

The filtering phase must be fast enough to notify the user about relevant offers in a timely fashion, with a low memory footprint and without requiring impractical CPU-time. In this phase reasoning dominates all the other components of the matching and its efficiency depends on the size of the context model and its instances.

In PervADs, reasoning is performed using the Jena backward-chaining rule-based inference engine. The number of inference rules is proportional to the number of dimensions and parameters in the context model. In our tests we measured the time taken by the matching phase w.r.t. the number of dimensions in the model and the depth of the dimension hierarchy.

We used six different context models as datasets. All the models define a single root dimension (`interest-topic`) that is the root of a tree of subdimensions, based on taxonomies of categories from UNSPSC®[12] or the Yellow Pages. The

[12] United Nations Standard Products and Services Code®, see `http://www.unspsc.org`.

depth of the tree and the number of subdimensions are different in each model, as detailed in Table 1.

Table 1. Benchmark context models

Model	**Nodes**	**Dimensions**	**Values**	**Depth**
YP-1	22	**1**	21	1
UNSPSC-1	60	**3**	57	1
YP-2	221	**22**	199	2
UNSPSC-2	467	**58**	409	2
YP-3	3361	**200**	3161	3
UNSPSC-3	2849	**410**	2439	3

The matching process proceeds in two phases. First, the context model is loaded and the reasoner is initialized with a custom set of inference rules for the context model (we call this the *setup* phase); then, the contexts contained in each PervAD are matched against the user's context. The setup phase occurs only once for every matching session.

Each context usually contains between 1 and 10 dimension assignments; however, the `interest-topic` dimension is the most expensive to match. This is the case since the interest topic is modelled as a large subtree of values and subdimensions that describe the various types of products and services the user might be interested in. If the user's context contains an assignment for `interest-topic` or another dimension near the root of its subtree, and the PervAD's context contains an assignment for one of the leaf dimensions in the subree, the reasoner must infer all the implicit assignments of the parent dimensions in order to be able to tell whether the two contexts match.

For these reasons, in our benchmark we measured the average speed of the setup phase and the average matching speed when comparing two dimensions in the `interest-topic` subtree, in the worst case where the two dimensions are as far as possible in the tree.

Figure 9 shows the results of the benchmark. As expected, both the setup and matching speeds grow exponentially with the number of dimensions in the context model; however, the matching process shows acceptable performances even with the *UNSPSC-3* model, which defines a quite large number of dimensions and values. The current PervADs context model uses the *YP-3* model for the `interest-topic` dimension, yielding an average matching time of about 1 minute per PervAD in the worst case.

Our matching algorithm can be further improved by speeding up the reasoning engine (cutting down the number of inference rules, pre-generating the ruleset and/or by optimizing the reasoner's implementation) and by compacting the representation of the context model and context instances; these and other enhancements will be considered in future versions.

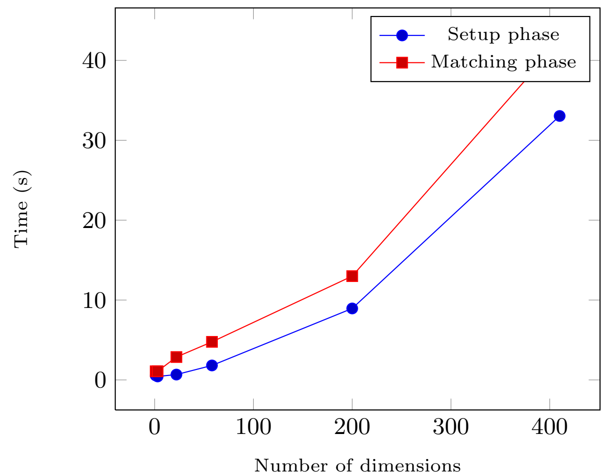

Fig. 9. Matching speed

7 Conclusions

PervADs enables users to control the amount and the pertinence of the advertisements they receive. At the same time, it provides companies with a powerful advertising mechanism capable of reaching potential customers with a very high precision. In the future, we plan to introduce the possibility for users to propagate PervADs to other users in proximity and to study the impact of this form of social advertisement in the real-estate scenario. Our research agenda includes also the definition and the evaluation of different business models for PervADs, as well as patenting possibilities.

References

1. L. Aalto, N. Göthlin, J. Korhonen, and T. Ojala. Bluetooth and WAP push-based location-aware mobile advertising system. In *Proc. of MobiSys*, pages 49–58, 2004.
2. L. Barkhuus and A. Dey. Location-based services for mobile telephony: a study of users' privacy concerns. In *Proc. of Interact*, pages 709–712, 2003.
3. C. Bolchini, C. Curino, G. Orsi, E. Quintarelli, R. Rossato, F. Schreiber, and L. Tanca. And what can context do for data? *Commun. ACM*, 52(11):136–140, 2009.
4. C. Bolchini, C. Curino, E. Quintarelli, F. Schreiber, and L. Tanca. Context information for knowledge reshaping. *Int. J. Web Eng. Technol.*, 5(1):88–103, 2009.
5. A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *Proc. of SIGIR*, pages 559–566, 2007.
6. R. Bulander, M. Decker, G. Schiefer, and B. Kolmel. Comparison of different approaches for mobile advertising. In *Proc. of WMCS*, pages 174–182, 2005.

7. D. Calvanese, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-*Lite* family. *J. of Automated Reasoning*, 2007.
8. S. Dave. Computational advertising: leveraging user interaction & contextual factors for improved ad retrieval & ranking. In *Proc. of WWW*, 2011.
9. J. de Castro and H. Shimakawa. Mobile advertisement system utilizing user's contextual information. In *Proc. of MDM*, pages 91–95, 2006.
10. W. Federation. Wireless news, analysis: are mobile ads the next gold rush?, 2007.
11. J. Gao and A. Ji. Smartmobile-AD: An intelligent mobile advertising system. In *Proc. of GPC Workshops*, pages 164 –171, 2008.
12. A. Goldfarb and C. Tucker. Privacy regulation and online advertising. *Management Science*, 57(1):57–71, 2011.
13. H. Haddadi, P. Hui, and I. Brown. MobiAd: private and scalable mobile advertising. In *Proc. of MobiArch*, pages 33–38, 2010.
14. Y. Jang, T. Lee, K. Kim, W. Lee, D. Ann, and S. Chung. Keyword management system based on ontology for contextual advertising. In *Proc. of ALPIT*, pages 440–445, 2007.
15. J. Kang and G. Borriello. Ubiquitous computing using wireless broadcast. In *Proc. of WMCSA*, pages 72–81, 2004.
16. S. Kurkovsky and K. Harihar. Using ubiquitous computing in interactive mobile marketing. *Personal Ubiquitous Comput.*, 10(4):227–240, 2006.
17. R. Mirizzi, A. Ragone, T. D. Noia, and E. D. Sciascio. Semantic tags generation and retrieval for online advertising. In *Proc. of CIKM*, pages 1089–1098, 2010.
18. G. Orsi, E. Zimeo, and L. Tanca. Context aware, keyword based selection of natural language query patterns. In *Proc. of the EDBT*, pages 189–200, 2011.
19. S. Papadopoulos, F. Menemenis, I. Kompatsiaris, and B. Bratu. Lexical graphs for improved contextual ad recommendation. In *Proc. of ECIR*, pages 216–227, 2009.
20. A. Penev and R. Wong. Framework for timely and accurate ads on mobile devices. In *proc. of CIKM*, pages 1067–1076, 2009.
21. A. Ranganathan and R. Campbell. Advertising in a pervasive computing environment. In *Proc. of WMC*, pages 10–14, 2002.
22. O. Rashid, P. Coulton, and R. Edwards. Providing location based information/advertising for existing mobile phone users. *Personal Ubiquitous Comput.*, 12(1):3–10, 2008.
23. O. Ratsimor, T. Finin, A. Joshi, and Y. Yesha. eNcentive: a framework for intelligent marketing in mobile peer-to-peer environments. In *Proc of ICEC*, pages 87–94, 2003.
24. B. Ribeiro-Neto, M. Cristo, P. Golgher, and E. S. de Moura. Impedance coupling in content-targeted advertising. In *Proc. of SIGIR*, pages 496–503, 2005.
25. J. Sánchez, J. Cano, C. Calafate, and P. Manzoni. Bluemall: a bluetooth-based advertisement system for commercial areas. In *Proc. of PM2HW2N*, pages 17–22, 2008.
26. A. Sieg, B. Mobasher, and R. Burke. Learning ontology-based user profiles: A semantic approach to personalized web search, 2007.
27. Trendwatching.com. Transumerism, 2003.
28. D. Tsatsou, F. Menemenis, I. Kompatsiaris, and P. Davis. A semantic framework for personalized ad recommendation based on advanced textual analysis. In *Proc. of RecSys*, pages 217–220, 2009.
29. X. Wu and A. Bolivar. Keyword extraction for contextual advertisement. In *Proc. of WWW*, pages 1195–1196, 2008.

30. X. Wu, J. Yan, N. Liu, S. Yan, Y. Chen, and Z. Chen. Probabilistic latent semantic user segmentation for behavioral targeted advertising. In *Proc. ADKDD*, pages 10–17, 2009.