Pers Ubiquit Comput
DOI 10.1007/s00779-012-0632-0

ORIGINAL ARTICLE

Crowdsourcing privacy preferences in context-aware applications

Eran Toch

Received: 29 February 2012 / Accepted: 22 October 2012

Abstract Developers of context-aware applications are faced with a tough challenge: powerful privacy controls are essential to maintain user trust, but they are also hard to use and not adequate in all situations. To address this tradeoff, we present Super-Ego, a crowdsourcing framework for privacy management of location information in ubiquitous environment. We study how crowdsourcing can be used to predict the user's privacy preferences for different location on the basis of the general user population. The crowdsourcing methods are evaluated in a 2-week user study in which we tracked the locations of 30 subjects and asked them to provide privacy preferences for the locations they had visited. Our results show that by employing simple methods for semantic analysis of locations and by profiling the user's privacy inclination, our methods can accurately predict the privacy preferences for 80 % of the user's locations. By employing semi-automatic decision strategies, which ask the user to decide regarding the privacy of some of the locations, the accuracy rate raises to 90 %.

Keywords Context awareness · Privacy · Crowdsourcing · Usability

1 Introduction

Ensuring users' privacy is becoming a major challenge in context-aware applications. As mobile applications increasingly rely on automatic context sensing to simplify and personalize services to users, users may find it difficult to trust the process in which services collect and use their context information. Users need to know that their information is collected and used in a way which is consistent with their expectations. Otherwise, their information might be taken out of its intended context, and used in ways which may harm their privacy. This is becoming a major challenge for designers of context-based systems due to two trends. The first trend is the huge advancement of mobile technology that simplifies the way application collect diverse context information, including exact physical location, proximity to other users, interaction with other users, calendar information and so forth. The second trend is the wide adoption of social networks, which increases the possible use of context information. Context information can now be reported to friends, family, co-workers and other social relations, complicating privacy risks and making them tangible to users.

Current theories highlight the inherent challenges in protecting privacy in context-aware applications, and in particular in systems that are used for sharing information between users. Helen Nissenbaum's contextual integrity theory explains why transmitting information from the original context to a new context erodes the trust in the system [19]. For instance, information about the user's location, which can be safely shared with work colleagues during work hours, can harm the trust in the system if the information is shared with the same people during the night. Users' sense of privacy is correlated with the control they have over their information [20]. However, controlling how information is collected and shared has its cost: control is directly contrasted against the user-burden involved in setting and adapting privacy settings [23]. Context-aware applications strive to act independently, but this can lead to a compromised sense of privacy if systems use information in contexts that the user did not anticipate and cannot control.

E. Toch (✉)
Faculty of Engineering, Tel Aviv University, Tel Aviv, Israel
e-mail: erant@post.tau.ac.il

Published online: 28 December 2012

In this paper, we ask the following question: how can crowdsourcing be used to improve privacy management of context-aware applications? Crowdsourcing is used extensively in ubiquitous computing to gather information from the public: from finding out restaurant rating to tracking traffic conditions. In this work, we extend the notion of crowdsourcing to gathering privacy preferences from individual users, and we evaluate how this information can be used to augment privacy management. Let us walk through a typical scenario: a location-based dating service, which is installed on the user's smartphone, similarly to applications such as Grindr, Skout or Girls-Around-Me. The dating application requires the location of the user when other users wish to see whether there are possible romantic partners nearby . The dating application would request the location from a privacy crowdsourcing framework. The framework would aggregate the privacy preferences decisions of the general user population and predict whether a particular user would be interested in sharing the current location. Based on this prediction, the privacy framework would either provide the current location to the application, deny it, or let the user decides if no confident decision can be made automatically. For instance, if the user is currently sitting in a coffee shop, a place with high probability of being shared [25], then the framework would tend release it. On the other hand, if the user is currently at home, the framework would delegate the decision to the user.

To examine the feasibility of crowdsourcing in privacy management for context-aware computing, we intend to define and evaluate a model that can be used to develop applications that rely on crowdsourcing. To this end, we identify two major research questions:

1. How can crowdsourcing be used for predicting privacy preferences of an individual user?
2. How can crowdsourcing produce trustworthy and accurate privacy decisions?

Crowdsourcing can benefit privacy management if we can identify meaningful patterns in the privacy preferences [13]. We have multiple evidence that this is indeed the case in privacy preferences related to location and context information: empirical research shows that users consistently discriminate between location context disclosure, believing that some locations are more private than others [4, 5, 16, 17] and that privacy patterns are shared between large number of users [1, 25].

To evaluate our method, we present "Super-Ego", a framework for crowdsourcing privacy preferences in location context. The framework controls the flow of context information from the mobile phone to context-aware applications. As in Sigmund Freud's structural model, where the Super-Ego plays the critical and moralizing role in our mental life, our framework plays a similar role in the lifecycle of context information in context-aware applications. Similarly, it is a channel for relating the individual user to the preferences of the society. When a context-aware application requires a location context information, it requests the location from Super-Ego, which uses a crowd-opinion model and a mixture of automatic and manual decision making strategies. The first version of Super-Ego was presented in [24] as a general framework for filtering sensitive information in mobile operating system environment. In this paper, we develop and evaluate methods for privacy decision making based on crowdsourcing. We embed our methods in Super-Ego, exemplifying how crowdsourcing can be used as part as privacy mechanisms in mobile operating systems.

The crowdsourcing model is empirically evaluated using a dataset of privacy preferences for location disclosure. The dataset includes preferences by 30 participants, given in a period of two weeks. Our results show that crowd knowledge is useful for predicting individual privacy preferences and that two mechanisms are crucial for generating quality predictions: (a) semantic analysis of the location content (i.e., learning from crowd opinions on location with similar activity patterns); (b) analysis of individual privacy approaches and biases. Furthermore, our results show that semi-automatic strategies exhibit an optimal balance between accuracy and automation. Finally, in our proposed architecture, all privacy preferences are stored in a centralized server, an architecture that can potentially harm the privacy of users. Section 6.2 discusses possible protections for the crowd information, using anonymization techniques and architectural solutions to protect the privacy of users that contribute the preference data.

To summarize, the contributions of the papers are threefold: (a) proposing a novel architecture for crowdsourcing of privacy management in context-aware computing; (b) developing and empirically evaluating models for privacy preferences prediction, including semantic analysis of places and analysis of individual privacy biases; (c) analyzing the properties of automatic and semi-automatic decision strategies with respect to accuracy, automation and overall efficiency.

2 Related work

Privacy in context-aware systems is a growing concern to consumers, system designers and regulators. Consumer uproar regarding the mishandling of location data on iPhone and Android phones illustrates the sensitivity of users' location [2]. In current mobile operating systems, such as the iOS 4 and Android 2.3, controlling location context is extremely limited, basically allowing users to

allow or disallow an application to access the location API. In this "all or nothing" approaches to location context utilization, users cannot differentiate between private locations, which they do now wish to disclose, and locations they do wish to disclose. However, mounting empirical evidence shows that users have detailed preferences regarding disclosure of specific locations. The willingness of users to share their location depends on the specific identity of the person receiving the location [4], the activity of the user in the location [16], the time and place [5] and the properties of the location, for example, the variety of people that visit the location [25]. Therefore, we believe that frameworks for managing location context information should be able to provide users with fine-grained and usable control over individual locations.

Context-aware applications use context information about the state of people, places, and objects, to adapt the applications' behavior [9]. As information about the state of the users is inherently private, the tension between privacy and context awareness is an ever challenging research question [15]. This tension has been addressed by using technologies such as rule-based access control mechanism [5, 7, 14, 23], by automatically learning user's ongoing privacy behavior and suggesting modifications to their privacy behavior [8], by limiting the level of detail given about a location [11] and by nudging users toward more secure privacy behaviors using salient information [3].

Our work complements these research efforts by exploring crowdsourcing as a way to predict privacy preferences in situations when an explicit user decision is unavailable or to hard to obtain. In crowdsourcing, the collective intelligence of a crowd is harnessed to solve a difficult problem that cannot be solved directly using computation or human efforts. In many ubiquitous computing applications, users benefit from a general model of their physical environment. The ability of many people to contribute implicitly or explicitly to generate such a model is increasingly used in ubiquitous computing applications. As crowdsourcing does not necessarily depends on the user's own input, it can be used in situations where the user's privacy preferences are missing, not yet expressed or too complicated to be expressed. For example, crowdsourcing can augment methods for suggesting privacy rules for users, such as the method suggested by Cranshaw et al. [8], by exploring privacy preferences from large number of users. Furthermore, crowdsourcing can be used to analyze and to assess user-controllable privacy management, comparing the user's policies to the general population, discovering patterns, guidelines and anomalies.

Crowdsourcing is used in ubiquitous computing for creating knowledge models of physical surroundings. Crowdsourcing is used for gathering geographical knowledge [13], assembling geo-wikis [21], locating objects (e.g., DARPA's red balloons[1]), detecting traffic congestions[2], reporting crisis information on a map [22], providing venue ranking and recommendations [28, 29] and allowing citizens to report local problems[3]. As similar places have similar patterns of privacy preferences [25], privacy preferences can be viewed as an intrinsic property of a geographical location. As preferences vary from user to user, the main challenge faced by this research is to explore how crowdsourcing can be useful in our context. The challenge of variability is twofold: to analyze preferences in spite of variability within the crowd and to produce meaningful information even if the user's privacy preferences are different than the crowd's. This challenge is similar to the one that venue ranking algorithms face when recommending a place to a new user [28], and we employ similar approaches to handle preference variation.

3 Crowdsourcing framework

In this section, we introduce Super-Ego, our privacy preferences framework. We describe the architecture of Super-Ego, show how crowdsourcing is embedded within the framework and explain how it is used to make decisions regarding the user's privacy preferences. We explain two key concepts of ubiquitous privacy crowdsourcing: quality control of crowd-sourced privacy predictions and user engagement in handling decisions.

3.1 Architecture

Super-Ego is based on a simple architecture that enables privacy-sensitive context awareness. Current mobile operating systems provide an API (Application Programming Interface) that is used by mobile application to access context information and other operating system resources. The Super-Ego framework is positioned between the original operating system API and the mobile application. In our approach, mobile applications access context using the framework, which decides whether to grant access to the context information. As Fig. 1 depicts, when a mobile application requests a location context information from the operating system API, the request is forwarded to the Super-Ego framework. If the request is granted, then the mobile application received the location coordinates (or any additional context). Otherwise, the mobile application does not receive the location and should be able to handle the rejection in a user-friendly way. Decisions are made by

[1] http://archive.darpa.mil/networkchallenge/.

[2] Waze, http://www.waze.com.

[3] FixMyStreet. http://Fixmystreet.org.uk.

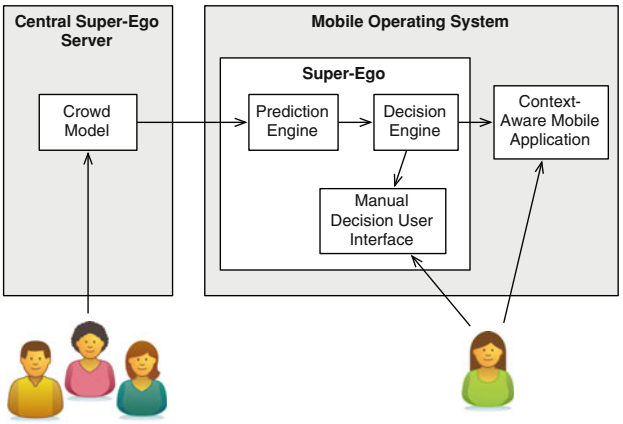

Fig. 1 The architecture of the Super-Ego framework, containing two main components: a centralized crowd model and a local client, embedded within the mobile operating system (e.g., Android and iOS)

the user, or are made automatically if enough information is available to make an accurate decision.

The decision making process includes a mixture of automatic and manual decision strategies. If the decision is received manually, then the content of the decision is stored in a **crowd model**, which is hosted on a centralized server. The crowd model stores all prior disclosure decisions by all users of Super-Ego, as privacy preferences relating to a specific geographical location. Each Super-Ego client securely sends the privacy preferences given by all users of the systems to a centralized server, where preferences are stored in reference to the respective location they were recorded in. The crowd model is used by the prediction engine to predict the user's privacy preferences for a given location. The decision engine computes the response to the location requests based on the prediction. The response regarding a particular location request can be one of three outcomes: *disclose*—to accept the request, *deny*—to reject the request, or *manual*—to let the user decide. If the response is disclosed, then the location is returned to the mobile application. If the response is to deny the request, then the function throws an exception that the calling application needs to handle. If the response is manual, then the user is presented with a user interface that asks whether to release the location to the current application. We assume that all location requests are done through the operating system's APIs and therefore that the application can be identified. From this point, Super-Ego would follow the user's decision and document that decision for further use.

In our architecture, Super-Ego is embedded within the operating system and is used by application as the sole method for accessing the user's location information. Currently, Super-Ego is implemented on Android 2.3 mobile operating system. The library wraps most of the native location API and provides access through a set of methods that first call the decision engine.

3.2 Crowdsourcing methods

In this section, we define a series of methods that make use of the crowd knowledge for privacy management. In this section, we define these methods in a methodological way, formulating the fundamental research questions of this paper. After the crowd preferences are collected, there are two further steps in the privacy management process: predicting the user's privacy preferences and making the privacy management decision. Predicting privacy preferences raises several questions regarding the way crowd opinions can be summarized and analyzed. We unfold several methods of privacy preferences prediction analysis, each based on a different analysis of the crowd opinions, and each encapsulates a hypothesis we evaluate in this paper. Specifically, we look at two methods for analyzing crowd opinions: place-based prediction and semantic-based prediction.

In place-based prediction, the crowd's privacy preferences are aggregated according to the geographical context of the preferences. For example, if users are comfortable sharing their location when they are in the faculty of engineering, then a new user might have similar preferences when visiting the faculty. Place-based prediction builds on that assumption, predicting the preferences for a given place according to the preferences by existing users. Naturally, this type of aggregation model is limited, as it is useful only if the user had visited the exact same physical place as the crowd. It requires users to visit, and provide preferences, to similar places. Our empirical results confirm the intuition that the method would not work for places which are not visited by a large number of users, such as users' homes.

In semantic-based prediction, the privacy preferences are predicted based on analyzing the semantic meaning of a given place. A place is semantically different than another if it is geographically distant and if the set of activities taking place in the places are different. In semantic-based prediction, the framework aggregates the crowd preferences from places with similar semantic categories. For example, to predict whether a user would be willing to share the location of her workplace, the framework would analyze the extent to which other users are willing to share the location of their workplaces. We hypothesize that there will be a correlation between the semantics of a place and the patterns of its privacy preferences. The semantic-based prediction method can bridge the main limitation of the place-based prediction, as it can be used to draw predictions for places that the crowd have not actually visited.

To analyze the semantics of a place, we expand a technique by Do et al. [10], which is based on a combination of user tagging and statistical analysis. In our implementation, we ask the participants to annotate a sample of locations using a closed set of categories. We then correlate all the user's locations with the tagged locations by comparing the number of visits and their time-of-day. To exemplify the robustness of semantic analysis, we asked the participants to annotate the places with a set of the three most basic semantic places: "home", "work" and "other". Home is where the user spend most of the night time, work is where the user spend most of the day time, and the "other" place represents any place which is not either work or home.

The two methods for crowdsourcing management provide us with the first research question that this research asks: *how effective are these methods for predicting privacy preferences?* In Sect. 5, we answer this question empirically. Our second research question is rooted in the usability of crowdsourcing for predicting individual privacy preferences. Individual users differ significantly from each other regarding their privacy approaches [27]. Therefore, we can expect variations between crowd preferences and each individual user, a phenomenon that our results clearly show. Therefore, our second research question regards the use of general statistical models for predicting an individual property, and specifically: *how can general crowd model be personalized to predict individual privacy preferences*?

The next step in crowdsourcing privacy management is to use preference predictions to decide whether to release a location to the requesting application. In order to do that, we define a set of decision strategies that take a prediction generated by the prediction engine and decide whether to release the location. In this work, we focus on simple strategies that can be configured using a simple set of two parameters. A strategy is configured by its threshold for disclosure and by the level of manual control it requires from the user. The use of strategies in a framework that allows both manual and automatic decision making. It raises a question which is critical to the understanding of assimilating automatic privacy management in a privacy-sensitive context-awareness framework: *what is the impact of manual decision delegation on the automation, accuracy and overall performance of decision strategies*?

3.3 Crowd-based predictions

The privacy predictions are based on a linear regression model that takes a set of privacy preferences and produce a prediction for the preference a given user would give to a given location. Each preference is relevant to a given location, l_k, which is a longitude/latitude pair, clustered within a perimeter of 30 m (the radius of an average house). Crowd preferences are represented as a set of scores $s_l^u \in [0,1]$, where each score is a number between 0 and 1 (0—not willing to share, 1—very willing to share), given by an individual user u visiting the location l_k. We define a summary function, S, that averages the scores given for a set of locations $\mathcal{L}$ by the set of users that visited the locations $U_{\mathcal{L}}$:

$$S(\mathcal{L}) = \frac{1}{|\mathcal{L}|} \sum_{u_i \in \mathcal{L}} s_k^{u_i}$$

Based on the general summary function, we define three specific summaries that averages the scores for places, semantic categories and users. We construct a selection function that selects all preferences in locations which are distanced r meters from the desired location:

$$S_{place}(l_k) = S\{l_i | dis(l_i, l_k) < r\}$$

To implement semantic-based prediction, we assign a semantic category to each location $\mathcal{C} : l_k \rightarrow \{c_1, c_2, \ldots, c_m\}$. In this paper, we look at three simple semantic categories: $range(\mathcal{C}) = \{home, work, other\}$. The semantic prediction selection function selects all locations that have the same category as the designated location:

$$S_{semantic}(l_k) = S\{l_i | \mathcal{C}(l_i) = \mathcal{C}(l_k)\}$$

To personalize predictions, we define a *user bias*, a concept that quantitatively captures the user's privacy tendencies, reflecting the difference between the user's privacy preferences and the preferences of the general population. A higher bias means a higher deviation from the mean. We denote the bias of a user u by σ_u, and define it as a subtraction of the of the user's average score, μ_u from μ, the average score of the general population: $\sigma_u = \mu_u - \mu$.

We define the expected privacy preference for a user u and a location l_k as a linear combination model of the predictive variables (note that σ_u is taken as a constant as it is unrelated to any single location). The values p_k^u are fitted using a multivariate linear regression, where α, β and γ are the regression coefficients and ϵ is the error variable (representing the noise in the model).

$$p_k^u = \alpha \cdot S_{place}(l_k) + \beta \cdot S_{semantic}(l_k) + \gamma \cdot \sigma_u + \epsilon_k$$

In the equation above, each selection function returns the average of the preferences for the place, semantic category and user, respectively. In Sect. 5, we show how the prediction model fares with actual data.

3.4 Decision strategies

The decision engine takes the prediction generated by the prediction engine and computes a decision based on a decision strategy, a construct that is used to configure the decision algorithm and to set the extent of manual intervention the decision algorithm will yield. Strategies are represented using two threshold values t_{manual} and $t_{disclose}$, each in the range of the [0,1], and $t_{disclose} \geq t_{manual}$. We define the decision engine as a function that discloses the location if the prediction is higher than $t_{disclose}$ that asks the user to manually decide if the prediction is between $t_{disclose}$ and t_{manual}, and denies the location if the prediction is lower than t_{manual}. By limiting the possible strategies to those that can be represented using the two threshold values, we are able to quantitatively analyze the strategies' properties and performance.

Let us exemplify the way the model of the decision engine works using the following scenario. A mobile application requests a location, l_k. The prediction for that location is 0.85. If the prediction is lower than t_{manual}, the location is disclosed, if the ratio is between t_{manual} and $t_{disclose}$, then the user is asked to weigh in on the decision, and if the ratio is higher than $t_{disclose}$, then the location is disclosed without any user intervention. For example, if $t_{disclose} = 0.8$, then the location will be disclosed. Setting the thresholds can yield dramatically different behavior. If the strategy is set as: $\mathcal{S} = \{t_{manual} = 0, t_{disclose} = 1\}$, then the decision engine is fully manual, as all predictions will be above the manual threshold and below the disclose threshold. In that case, the outcome of our example scenario is manual. If the strategy is set as: $\mathcal{S} = \{t_{manual} = 0.5, t_{disclose} = 0.5\}$ then the decision is fully automated as all ratio values will be either smaller than t_{manual} or higher than $t_{disclose}$. In a semi-automatic strategy, the threshold values will be set up in some distinctive way that would reflect the desired amount and nature of user intervention.

In evaluating the strategies, we had devised and implemented three strategies that represent the three basic modes of decision making:

- Fully manual strategy (**M**): all requests are decided as manual, such that $\mathcal{S} = \{t_{manual} = 0, t_{disclose} = 1\}$.
- Fully automatic strategy (**A**): all requests are decided automatically, such that $\mathcal{S} = \{t_{manual} = x, t_{disclose} = x\}$
- semi-automatic (**SM**): in this strategy, $t_{disclose} = t_{manual} + \Delta$, where Δ is a "gap" constant that is set up to the average standard deviation of all disclosure rates in the crowd model. In our experiments, $\Delta = 0.229$.

The automatic and manual strategies were designed to understand the boundaries of the automation/accuracy tradeoff. The semi-automatic strategy represents the baseline for comparison for both of the tradeoff extremes, searching for a balance between accuracy and automation using different approaches for decision making.

4 Evaluation methodology

In this section, we provide an overview of our study, and we describe the software that was used to conduct the study, and define the analysis methods used in the evaluation.

4.1 Experimental setup

The crowdsourcing methodology was evaluated in a 2-week user study in which we asked 30 participants (16 females and 14 males), to provide privacy preferences for the locations they had visited. Participants were recruited from the university population using flyers and mailing lists. For their efforts, participants received a 40$ compensation voucher at the end of the study. The university itself is located at the heart of a large urban area. All the participants had visited the university at least 2 times throughout the course of the study. Figure 2 depicts the client software that was used in the study. The client was developed in Java and runs on the Android operating system (versions 2.0 to 2.3). At the beginning of the study, the

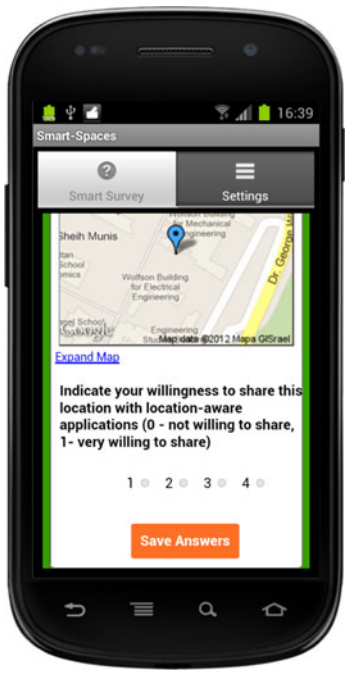

Fig. 2 The client software used for the privacy preferences survey. The software presents the survey interface, including a map of the surveyed location and the related question

participants were asked to install our client software on their own phones, using their own data plans.

The client software pops up an online surveys that were presented to the participants between two to three times a day, in which they were asked to provide privacy preferences for the places they visited. The survey system was designed to support maximal coverage of the places visited by the participant while preserving battery and data usage. Each survey was based on the last location that was visited by the participant, tracked by GPS and WiFi positioning. In order to preserve battery life, the locations were observed in 15-min intervals. To omit in-transit locations, which are not comparable between participants, the application had only selected locations in which the participant had stayed for at least 20 min. In order to survey as many locations as possible, if a particular location was surveyed at least 2 times during the study, the application did not survey that location again, and instead selected other locations that were not surveyed before. As a result, some much-visited locations, such as the participant's home, was omitted gradually from the survey, and the time that passed between the actual visit and the survey was between 0 and 6 h. The survey included a map of the location and a reference to the time in which the participant had visited the location.

The participants were asked to indicate how willing they are in sharing particular locations with location-aware applications using a 1–4 Likert scale. In the pre-study survey, our research assistant had explained to the participants that the sharing is relevant to three location-aware applications: Google Latitude, Facebook Places and Localmind. The participants received explanations regarding how these applications work and what they can potentially do with the location information. While the explanations can potentially bias the participants' privacy approach, we believed that without any prior information about emerging location sharing technologies, participants would not been able to make a knowledgable decision regarding their location sharing. We had picked these three applications specifically, to provide a mix of applications that use location in a social context (e.g., Latitude and Places), to people in the users' social network (e.g., Facebook Places) and to strangers in an anonymous way (e.g., Localmind). At the end of the study, participants were asked to annotate all the locations they visited with the semantic categories: each location was displayed to the participant on a map, and the user selected the appropriate category.

4.2 Data analysis

We divide our analysis into three parts: preprocessing, analyzing prediction results and analyzing decision strategies results. In preprocessing, the scores are normalized to a [0,1] range by dividing the Likert score by 4. The second step in preprocessing is to semantically analyze the locations as described in Sect. 3.3. To analyze the prediction results, we compare the actual scores made by participants to the predicted scores produced to different prediction methods. In order to measure accuracy, we define a loss function for penalizing errors in prediction. Given p_k^u as the prediction for user u and location k, and s_k^u as the actual score for that user and that location, we define the loss function as the squared error: $e_k^u = (s_k^u - p_k^u)^2$. This type of loss function was chosen as it quantifies how different is the prediction from the actual score, because both the score and the prediction are on a continuos scale, and because it penalizes gross errors.

In our analysis, we focus on evaluating the accuracy of the prediction methods, which range from being relatively simple to more complex. We base the notation on the linear regression model presented in Sect. 3.3:

- *Simple:* This method is based solely on the place-based prediction, predicting the preference according to the preferences of other users for the same place.
- *Semantic:* This method is based on a combination of the place and semantic methods, predicting the preferences according to the preferences for the same place, or for the same type of place.
- *Semantic/biased:* This method is based on all three selection functions and is called biased as it takes the user bias as part of the model.

Decision strategies, which use the predictions to make a sharing decision, are analyzed on two dimensions: accuracy and automation. This methodology simulates how strategies impact the user experience, in terms of the satisfaction from the framework's decisions and the burden required from the user. If a decision engine involves the user in every decision, it might get perfect results, but would compromise the usability of the application through excessive user burden. On the other hand, if an engine requires no manual intervention, its accuracy can be mediocre. The objective of our evaluation methodology is to enable us to characterize how well a strategy fits in this tradeoff between accuracy and automation, and help us identify good strategies that balance these two important aspects.

We assume that we run a decision engine on a set of predictions. The outcome of the decision process can be either a positive decision (disclose), negative (denied) or manual. We evaluate the decision strategy by comparing it to the score actually given by the user. We define a binary loss function that evaluates the decisions. We assume that the user is always satisfied with the result of a manual decision, and therefore manual decisions are always considered true. Therefore, information retrieval

categorization can be used on the output of the decision engine, resulting in four categories: true positives (*tp*), false positives (*fp*), true negatives (*tn*) and false negatives (*fn*). We are able to categorize observations by comparing the prediction to the actual preference of the study participant. On the basis of this categorization, we employ standard information retrieval measurements, namely precision, recall and accuracy. We evaluate the overall accuracy using the standard accuracy function used in information retrieval, giving equal weights to both measures: $accuracy(S) = \frac{tp+tn}{tp+tn+fp+fn}$.

To evaluate the user involvement for each of the strategies, we count the number of manual decisions and the number of automatic decisions. We define the user involvement of a given strategy as the ratio between automatic decisions and the overall number of decisions. We denote by a the number of decisions taken autonomically by the decision engine, and by m the number of decisions sent to the user. We define the automation measure as: $automation(S) = \frac{a}{m+a}$.

The two measures, accuracy and automation, reflect, respectively, how well an algorithm decides regarding a set of locations, and how much user intervention is required. To evaluate the overall performance of an algorithm, configured by a strategy, we developed a simple combined measure, which we call the combined score (or "combined" for short). We define the score as follows: $combined(S) = \alpha \cdot automation(S) + (1 - \alpha) \cdot accuracy(S)$. The combined score is a sum of automation and accuracy, weighted by a coefficient $\alpha \in [0, 1]$ which sets the ratio between the two measures. For example, when $\alpha = 0$, a high score would be given to a strategy with high accuracy with no regard to automation. When $\alpha = 1$, the only meaningful measure would be automation, and when $\alpha = 0.5$, equal importance would be given to both measures.

5 Results

In this section, we explain the results in the order of the crowdsourcing process: first, we describe the results of the prediction methods for privacy methods and then the results of the decision strategies. Overall, around 500 locations were ranked by the study participants. The mean score is 0.67, and the standard deviation is 0.19.

5.1 Evaluating prediction methods

Our first set of results, presented in Fig. 3, compares the accuracy of three prediction methods: simple, semantic and semantic/biased, a composition of semantic and user-based prediction. The methods are compared according to the error they produce when predicting the value of a given location request. The differences between the methods are significant, using a Kruskal–Wallis test, with $\chi^2 = 133.80$ and $p < 0.0001$. The simple method, which uses privacy predictions from other users for the same location, was found to be the least efficient, with a median squared error of 0.3 (out of 1). The simple method is useful for a minority of the locations ranked by participants. The semantic method outperforms the simple method by an order of magnitude, predicting almost 75 % of the locations with an accuracy a squared error of 0.05. The biased method raises this proportion to 80 %.

Analyzing the properties of the multiple linear regression model described in 3.3 provides several insights into the robustness of semantic and user-based predictions. The model fits the data well: $Adj.\ R^2 = 0.32$, $p < 0.001$. Table 1 presents the standardized beta coefficients of the three predictive variables. Each variable is defined using the preference selection function. The regression model regresses over those three averages. If the place was not visited by any other participants, then the average overall score was assigned to the prediction for the place variable. About 60 % of all locations were not visited by a single participant, while a third of the remaining 40 % were visited by only two participants. Therefore, the prediction power of the place variable is rather limited in the general place. Specifically, the method fared low even with our study population, which were concentrated in a tight geographical perimeter. On the contrary, the semantic variable provides superior prediction power as it provides some prediction for every given location. The semantic model explains about 35 % of the variability even with our three simple categories: "home", "work" and "other".

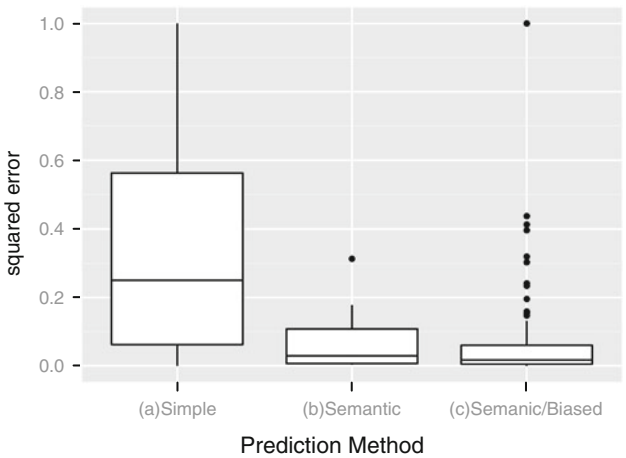

Fig. 3 Comparison of three privacy prediction methods according to the error they produce at the Y axis. The *middle line* represents the median point, the *box borders* represent the 1st and 2nd quartiles around the median, and the *black points* represent outliers. A *higher value* signifies a higher error value

Figure 4 provides an insight into the effectiveness of semantic-based prediction: the score distribution patterns of the three semantic categories are very different and have completely different score patterns. The semantic method bridges the main limitation of the simple method, namely the low probability of covering a location by adequate number of other users. The average score and the distribution are quite different for all categories, which makes the prediction quite feasible. Users are more likely to share their work location, less likely to share their home locations and are even less likely to share other locations. These results are very much in accordance with previous research that explored empirical models of location sharing [25]. A more in-depth analysis of the "other" category might increase the number of significant patterns and the overall efficiency of the semantic method.

The user bias summarizes the average privacy preferences of the user for all location, apart from the location that we wish to predict. The average participant had provided privacy preferences for 15 distinct locations, which means that the prediction was based on average of preferences for 14 other locations. The effectiveness of the bias, which explain around 40 % of the variability of the score, reflects the individual approaches toward privacy. Different users have different scoring patterns: some user's average score 0.3, while others have an average score of 0.8. Adding the user's average score to the prediction model personalizes the prediction according to the user's score distribution.

5.2 Evaluating decision strategies

The decision strategies, which take the crowd prediction and produce a location disclosure decision, differ significantly with respect to their accuracy and automation. The differences between the strategies are significant when comparing their automation and accuracy. Results were obtained in two-sample independent t tests with unequal variances, with $p < 0.001$ for every strategy pair. Figure 5 describes the accuracy of each strategy, for different threshold t_{manual} values. As we assume that all manual decisions are correct (true positives and true negatives), it is not surprising that the manual strategy (M) has perfect accuracy. The fully automatic strategy (A) is producing reasonable results, when compared to similar information retrieval problems. It exhibits a maximal accuracy of 0.8, with precision of 0.83 and recall of 0.801. However, the automatic strategy is outperformed by the semi-automatic approach, as it forwards the decision to the user to manually intervene when deciding on some of the location contexts.

The semi-automatic approach reaches the best accuracy when the threshold is set to 0.7. At that point, the approach provides an accuracy of 0.9 with precision of 0.9 and recall of 0.98. This boost in accuracy can be explained by turning to manual intervention, which results in true positives and true negatives, thus increasing both the recall and precision of the strategy. Furthermore, as the decision algorithm outputs location contexts with medium grades as manual

Table 1 An analysis of the three prediction variables, showing their beta coefficients and p values

Variable	β	SE	p
$S_{semantic}$	0.359	0.03	$p < 0.001$
S_{user}	0.40	0.08	$p < 0.001$
S_{place}	0.08	0.15	$p = 0.19$

With the three variables, the model fits 81 % of the information required to predict privacy preference

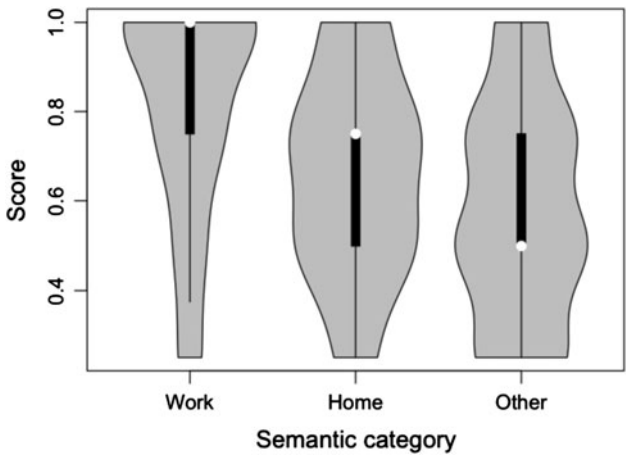

Fig. 4 The score density for each of the semantic categories. Higher score represents higher willingness to share the location. *White points* denote the median (1 for work, 0.75 for home, and 0.5 for other) and the *black line* represents the two quartiles around the median. The body of the graph depicts the distribution graph for each category

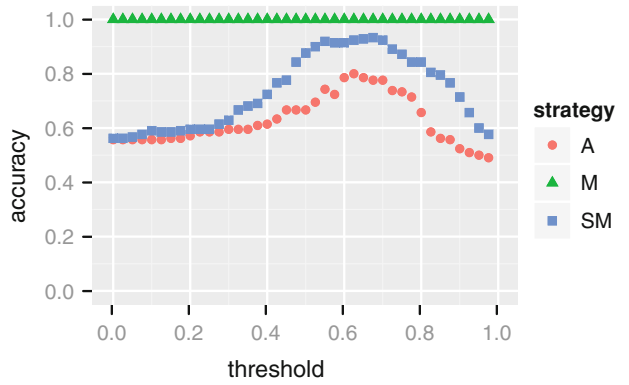

Fig. 5 *Accuracy*: The accuracy versus the threshold value t_{manual} for each of the three strategies. *Each point* represents the average accuracy of all observations for a given threshold. The manual strategy has perfect accuracy as we assume that the user's decisions are always accurate

results, they receive a boost exactly where automatic algorithms will risk outputting a decision that will be either false positive or false negative.

Figure 6 presents the levels of automation for each of the strategies. Naturally, the fully manual approach has no automation, and at the same time, the fully automatic strategy has maximal automation regardless of the threshold. The automation of the semi-automatic strategy is based on the threshold value. We see that the highest levels of automation are received in either a very low threshold (where most requests are denied) or a very high threshold (where most requests are allowed.) The reason for this phenomena is the variability of the scoring patterns. The variability of the scores is highest when the threshold is between 0.675 and 0.825. In this range, the threshold captures the scores that are consternated around the mean, where the variability of the participants' scores is maximal. In this range, the locations are not very private (e.g., home) or very public (e.g., the university lawn), but contain a mixture of private and public locations, resulting in a higher variability. The semi-manual strategy deals well with high variability of the scores, as lower-scored locations can be delegated to the manual decision. At the 0.675–0.825 threshold range, the semi-automatic strategy delegates 1 out of 10 requests to the user. As a result, the accuracy increases and at the same time the automation decreases.

To design a usable and trustworthy system, automation and accuracy should be balanced. Figure 7 shows the combined score for all strategies, with a variable threshold and five sub-diagrams according to a variable α value. The combined score is configured by the α coefficient, where $\alpha = 1$ gives all the weight on accuracy and $\alpha = 0$ gives all weight to automation. The sub-diagrams in Fig. 7 are ordered from left to right according to the α values, ranging from giving full weight to automation (on the left) and full weight to accuracy (on the right). Here, we see a generalization of the results shown in Figs. 5 and 6, expressing an accuracy-automation tradeoff. The manual strategy has constant accuracy, and therefore it is dependent only on the α value, having the worst combined score when $\alpha = 0$ and the best combined score when $\alpha = 1$. Similarly, the combined score of the automatic strategy is linearly decreasing as it is dependent on the proportion of accuracy in the overall score. The tradeoff is particularly telling when it comes to the semi-automatic approaches. The semi-automatic strategy trades automation with accuracy and is less sensitive to the α value than the automatic approach. The variable approach outperforms the constant strategy when automation plays a meaningful role in the combined score ($0.58 < \alpha < 0.7$), as it is relatively less dependent on manual input to provide accuracy. The Pareto optimality of the both the accuracy and the automation is achieved when $\alpha = 0.62$.

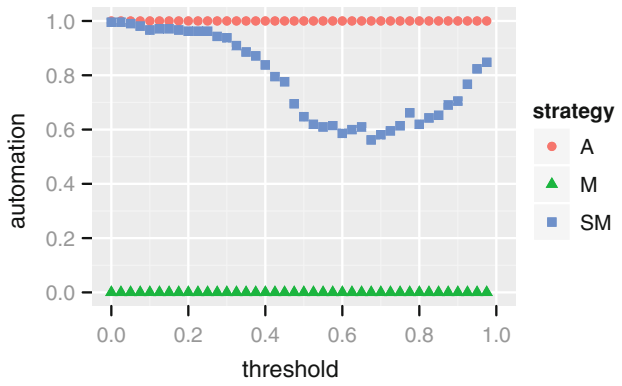

Fig. 6 *Automation*: The automation versus the threshold value t_{manual} for each strategy. *Each point* represents the average automation for all observations for a given threshold. The automatic strategy has full automation for any threshold while the manual strategy has none

6 Discussion

In this section, we take an overview of our findings with respect to the open issues faced by the research community. Specifically, we examine three issues: the viability of crowdsourcing for privacy preference, the tradeoff between automation and accuracy and the third is ad hoc decision strategies for context sharing. We follow this discussion with a sub-section about limitations of our study and issues regarding crowd privacy.

Crowdsourcing is used mainly to build a general knowledge model, which is not personalized to specific user models. However, our results section shows that we can predict some of the privacy preferences of individual users. That does not mean that crowdsourcing can be used directly to decide instead of users. But our results do show that mixing automatic and manual decision making can achieve relatively high accuracy with reasonable level of user burden (i.e., asking the user to decide on one out of ten decisions.) Our results show that semantic analysis and personalization are crucial for predicting privacy preferences. As the literature of privacy preferences repeatedly tell us, people have distinct privacy preferences that follow profiles such as those described by Alan Westin: privacy fundamentals, privacy pragmatists and unconcerned [27]. The α value, which adjusts the tradeoff between automation and accuracy, can be used to personalize the user experience for different categories of users. For example, the system will set α to 0.9 for a privacy fundamentalist and to 0.5 for an unconcerned user.

The second issue is managing the tradeoff between automation and accuracy in questions related to privacy. The research community and the media are well aware that

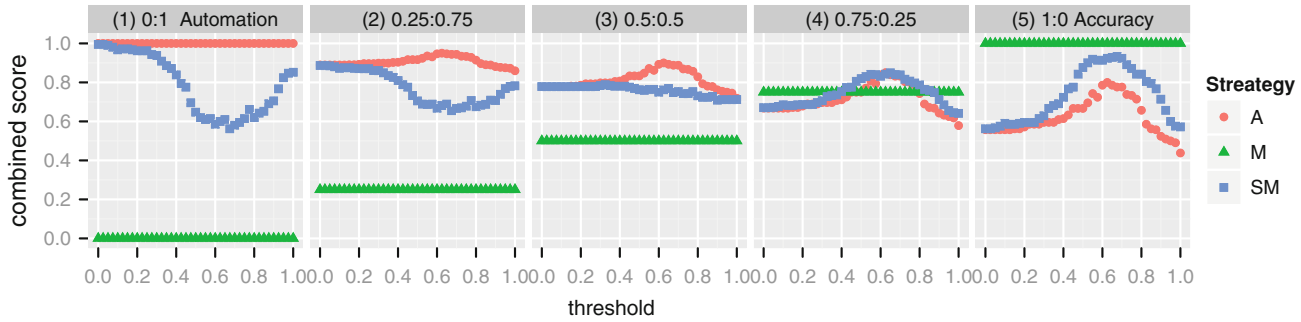

Fig. 7 The combined score versus a variable threshold for each of the strategies, given according to 5 α values from 0 (on the *left*) to 1 (on the *right*). The combined score weights accuracy by α and weights automation by $1 - \alpha$, such that: $combined = \alpha{\cdot}automation + (1 - \alpha){\cdot}accuracy$. In each of the five α value settings, we show the combined score depending on a moving threshold

one of the main problems in managing privacy is its substantial requests on the user's time and effort [5, 6, 23]. The Super-Ego framework manifests this tradeoff as part of its inherent mechanism, making it possible to configure the relations between automation and accuracy. The simple framework we propose for evaluation presented in Sect. 4.2 can be used to systematically evaluate the tension between automation and accuracy in user studies and for different algorithms.

The third issue relates to using Super-Ego in solutions for managing context information using a combination of ad hoc decision strategies solutions and pre-defined rules. Several works have shown how pre-defined specifications of rules, conflict specifications and roles can effectively be employed in privacy-sensitive context awareness [5, 7, 14, 26]. Combining rule-based decision models with ad hoc decision models, such as Super-Ego, can result in usable privacy management systems. In a combined approach, the user's known restrictions can be expressed directly using a rule-based interface. Unexpected situations, which are not well defined by rules, can be handled by Super-Ego, with its combination of automatic and manual decision processes.

6.1 Limitations and future work

The approach we present in this paper is limited in several ways. First, Super-Ego requires knowledge about location disclosure behavior from the general population. While widespread adoption of Super-Ego can eventually lead to create such a knowledge base, it is currently nonexistent. Moreover, in this paper, we do not resolve potential privacy risks that stem from sharing historical location context disclosure decisions. The second limitation is the architecture of Super-Ego, which requires all context requests to go through a single filtering layer. This property can eventually limit the applicability of the approach. The third limitation regards the methods we use to analyze locations and scores. In this work, we had shown that the overall accuracy and performance of crowdsourcing is promising, even with very simple methods. A combination of more sophisticated methods can yield better predictions and handle situations in which malicious parties try to game the system through wrongful input. The fourth limitation is the equal weight we give to false positives and false negatives. In most scenarios, the outcome of a wrongful disclosure of a private location can be considered more harmful than denying a mobile application of a location. Fifth, we do not take into account in the decision process the different applications that request the information, and the different uses the applications might use the location context for. Finally, it is important to emphasize that we have not tested how crowdsourcing would be used by actual users. Our user study merely predicted their preferences rather than testing how suggestions would be used in the real world.

6.2 Crowd privacy

This paper aims to improve users' privacy management, but if privacy crowdsourcing frameworks are implemented naively, they can create new privacy risks to the users. Storing the preference information on a centralized server can expose the users to security and privacy breaches, and in any case requires a great deal of trust. Defining and proving privacy protection measures for privacy crowdsourcing is part of our future work, but we can point out to a number of methods that can be used to improve the privacy of the system. First, a distributed architecture can be used to remove some of the privacy threats. The individual privacy bias, the $S_{user}(u)$ part of the prediction regression, can be computed and stored on the user's client software and not be distributed to the server. Second, summarizing techniques can be used to cloak the identity of users in other elements of the crowdsourcing data. The preferences for the semantic location type, that is,

$S_{semantic}(l_k)$, can be summarized and represented by the mean value and standard deviation of preferences for every semantic location type.

The preferences for the unique locations are posing the most serious challenge in protecting users' privacy, as these preferences cannot distributed or summarized. However, *k*-anonymity measures for location-based services, as suggested by Gedik and Liu [12], can be used to provide a guaranteed level of anonymity to users. The system can be tuned to save preference information for locations that were visited by k or more unique users, giving users a $k-1$ protection for being recognized. While *k*-anonymity methods are vulnerable in situations where the sensitive information is not diverse enough [18] , this is not the case in this particular problem. The privacy preferences information contains only the sharing score for a particular location, and the score can be easily diversified without losing the original preference distribution.

7 Conclusions

In this paper, we present a method for helping users manage their privacy in context-aware systems by employing crowdsourcing architecture. We have implemented and evaluated our approach for a particular type of context: locations sensed by the mobile device and disclosed to mobile applications. We present an architecture in which a filtering layer, called Super-Ego, is placed between the mobile platform's operating system and mobile applications that require the user's context information. As mitigating the task of deciding about the release of every location to the user will compromise the usability of the framework, we develop and evaluate a model for decision strategies that combine automated methods and manual intervention.

The evaluation of Super-Ego is based on a user study in which 30 participants have provided privacy preferences for a period of 2 weeks. The empirical evaluation portrayed the tradeoff between accuracy and automation with respect to different decision strategies, including manual, automatic and various semi-automatic approaches. While a fully automated approach delivers reasonable results (predicting the privacy preferences for 80 % of the locations), they are less accurate than manual and semi-automatic approaches. Our results show that having the user interfere in even a small part of the location contexts boosts the accuracy of the decision process. While user intervention reduces the automation of a given strategy, it is possible to quantify and adapt the strategy to the desired amount of user involvement.

The findings in this paper open several possibilities for future work, apart from working on the limitations presented in Sect. 6.1. One course of research involves exploring additional dimensions of crowdsourcing, including more sophisticated context sensing using the user's activities and more detailed privacy preferences collection. Another possibility involves a richer obfuscation methods and preference analysis methods. The second possibility is investigating the impact of accuracy models that provide different weights to positive and negative errors. For example, studying the difference between models that are less tolerant to releasing unwanted locations than to blocking legitimate locations.

References

1. Anthony D, Kotz D, Henderson T (2007) Privacy in location-aware computing environments. IEEE Pervasive Comput 6(4):64–72
2. Arthur C (2011) iPhone keeps record of everywhere you go. The Guardian, Wednesday, 20 April
3. Balebako R, Leon PG, Mugan J, Acquisti A, Cranor LF, Sadeh N (2011) Nudging users towards privacy on mobile devices, in workshop on persuasion, influence, nudge and coercion through mobile devices (PINC). In: Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems (CHI EA '11), New York, NY, USA, ACM
4. Barkhuus L, Brown B, Bell M, Sherwood S, Hall M, Chalmers M (2008) From awareness to repartee: sharing location within social groups. In CHI'08, pp 497–506
5. Benisch M, Kelley P, Sadeh N, Cranor L (2010) Capturing location-privacy preferences: quantifying accuracy and user-burden tradeoffs. Pers Ubiquit Comput 15(7):679–694
6. Bilton N (2010) Price of facebook privacy? Start clicking. New York Times Article, New York
7. Costa P, Almeida J, Pires L, van Sinderen M (2008) Evaluation of a rule-based approach for context-aware services. In: Global telecommunications conference, 2008. IEEE GLOBECOM 2008. IEEE, pp 1–5, 30 2008-Dec 4
8. Cranshaw J, Mugan J, Sadeh N (2011) User-controllable learning of location privacy policies with gaussian mixture models. In: Proceedings of the twenty-fifth conference on artificial intelligence (AAAI-11)
9. Dey A, Abowd G, Salber D (2001) A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human Comput Interact 16(2–4):97–166
10. Do TMT, Blom J, Gatica-Perez D (2011) Smartphone usage in the wild: a large-scale analysis of applications and context. In: Proceedings of the 13th international conference on multimodal interfaces, ICMI'11, pp 353–360. ACM
11. Duckham M, Kulik L (2005) A formal model of obfuscation and negotiation for location privacy. In: Gellersen H, Want R, Schmidt A (eds) Pervasive computing, (Lecture Notes in Computer Science), vol 3468. Springer, Berlin, pp 243–251
12. Gedik B, Liu L (2008) Protecting location privacy with personalized k-anonymity: architecture and algorithms. IEEE Trans Mobile Comput 7(1):1–18
13. Heipke C (2010) Crowdsourcing geospatial data. ISPRS J Photogramm Remote Sens 65(6):550–557. ISPRS Centenary Celebration Issue
14. Hesselman C, Eertink H, Wibbels M (2007) Privacy-aware context discovery for next generation mobile services. In: International

symposium on applications and the internet workshops, 2007. SAINT workshops 2007. IEEE Computer Society 2007, Hiroshima, Japan, 15–19 January 2007. doi: http://www.computer.org/csdl/proceedings/saintw/2007/2757/00/27570003-abs.html
15. Hong JI, Landay JA (2004) An architecture for privacy-sensitive ubiquitous computing. In: Proceedings of the 2nd international conference on mobile systems, applications, and services, MobiSys '04, pp 177–189, New York, NY, USA, ACM
16. Iachello G, Smith I, Consolovo S, Abowd G, Hughes J, Howard J, Potter F, Scott J, Sohn T, Hightower J, LaMarca A (2005) Control, deception, and communication: evaluating the deployment of a location-enhanced messaging service. In: Ubicomp'05, Springer, pp 213–231
17. Khalil A, Connelly K (2006) Context-aware telephony: privacy preferences and sharing patterns. In: CSCW'06
18. Machanavajjhala A, Kifer D, Gehrke J, Venkitasubramaniam M (2007) L-diversity: privacy beyond k-anonymity. ACM Trans Knowl Discov Data (TKDD) 1(1):1–52
19. Nissenbaum H (2004) Privacy as contextual integrity. Wash Law Rev Assoc 79:119–158
20. Palen L, Dourish P (2003) Unpacking "privacy" for a networked world. In: CHI'03, pp 129–136, New York, NY, USA, ACM
21. Priedhorsky R, Terveen L (2008) The computational geowiki: what, why, and how. In: Proceedings of the 2008 ACM conference on computer supported cooperative work, CSCW'08, pp 267–276, New York, NY, USA, ACM
22. Roche S, Propeck-Zimmermann E, Mericskay B (2011) Geoweb and crisis management: issues and perspectives of volunteered geographic information. GeoJournal 1–20. doi:10.1007/s10708-011-9423-9
23. Sadeh N, Hong J, Cranor L, Fette I, Kelley P, Prabaker M, Rao J (2009) Understanding and capturing people's privacy policies in a mobile social networking application. Pers Ubiquit Comput 13(16):401–412
24. Toch E (2011) Super-Ego: a framework for privacy-sensitive bounded context-awareness. In: Proceedings of the 5th ACM international workshop on context-awareness for self-managing systems (CASEMANS'11), August
25. Toch E, Cranshaw J, Drielsma PH, Tsai JY, Kelley PG, Springfield J, Cranor L, Hong J, Sadeh N (2010) Empirical models of privacy in location sharing. In: Proceedings of the 12th ACM international conference on ubiquitous computing, Ubicomp'10, pp 129–138, New York, NY, USA, ACM
26. Tuttlies V, Schiele G, Becker C (2009) End-user configuration for pervasive computing environments. In: International conference on complex, intelligent and software intensive systems, 2009. CISIS'09, pp 487 –493, March
27. Westin A (1967) Privacy and freedom. The Bodley Head, London
28. Ye M, Yin P, Lee W-C, Lee D-L (2011) Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, SIGIR'11, pp 325–334, New York, NY, USA, ACM
29. Zheng VW, Zheng Y, Xie X, Yang Q (2010) Collaborative location and activity recommendations with gps history data. In: Proceedings of the 19th international conference on World wide web, WWW'10, New York, NY, USA. ACM, pp 1029–1038