Towards Comprehensive Repositories of Opinions

Han Zhang, Kasra Edalat Nejad, Amir Rahmati, and Harsha V. Madhyastha
University of Michigan

ABSTRACT

Despite the popularity of recommendation services (such as Yelp, Healthgrades, and Angie's List), for a majority of entities listed on these services, one has to rely on opinions shared by a few users. We argue that this paucity of reviews for most entities stems from the fact that the vast majority of users largely consume opinions shared by others but seldom post reviews themselves. Therefore, leveraging the trend that services are increasingly accessed from a client-side app rather than over the Web, we propose augmenting recommendation services to implicitly infer any user's opinions based on observations of the user's activities. Implicit inference of many of a user's recommendations are feasible due to the rich sensory capabilities of smartphones and wearables as well as the digital footprints left behind by many activities in the physical world. However, implicit inference of opinions is inherently uncertain and automated sharing of inferences raises significant privacy and security concerns. In this paper, we discuss how to tackle these challenges so that users looking for recommendations can draw upon a more comprehensive set of opinions than is the case today.

1 Introduction

Motivation: Need for recommendations. All of us rely on recommendations from others for a variety of purposes such as knowledge discovery (e.g., web pages and books), entertainment (e.g., songs and movies), service provider selection (e.g., doctors and electricians), and travel (e.g., hotels and restaurants). To cater to this need for recommendations, several online service enable users to discover opinions shared by others. As evidence of their utility, many services that focus primarily on crowdsourcing of reviews (such as Yelp, Angie's List, TripAdvisor, and Healthgrades) feature in Alexa's list of top 500 websites and they each have annual revenues in the order of hundreds of millions.

Problem: Lack of reviews. However, pretty much all of these popular services suffer from a common problem: most of the entities listed on these sites have very few reviews. For example, we find that the median number of reviews is less than 5 for doctors listed on Healthgrades and less than 10 for service providers (such as electricians, plumbers, and gardeners) listed on Angie's List. Thus, users are often forced to choose from a set of entities based on a small set of potentially unrepresentative reviews for each. A user's confidence in a discovered recommendation critically relies on aggregating a large number of opinions [3].

One could imagine that the paucity of reviews for an entity is because only a few users have interacted with it and only these few users have an opinion to share, but we find that this is not the case. For services that do present data about both the number of reviews and the number of user interactions, we find a significant discrepancy between the two. For example, the median number of reviews/comments for an app on Google Play or for a video on YouTube are at least an order of magnitude lower than the medians for the number of installs of the app and the number of views of the video.

Root cause: Most users are passive consumers. Thus, we are stuck in an undesirable status quo where, despite a large number of users potentially having interacted with any particular entity, each of us is having to make decisions based on opinions shared by a handful of users for most entities [11]. We observe that the root cause for this unfortunate state of affairs is that existing services place the onus on users to share their opinions. While letting users choose what they share enables them to protect their privacy and reputation by sharing only that subset of their opinions they are comfortable revealing, in practice, most opinions go unshared because users either do not consider the task of posting a review worth the effort or simply forget to do so.

Vision: Opinion discovery and sharing without user input. We propose that recommendation sharing providers (RSPs) not rely only on explicit user input, but also attempt to *implicitly* infer any user's opinions and *automatically* share these inferred opinions with others. While implicit inference of a user's likes and dislikes is already the norm for interactions online (e.g., Netflix and Amazon make recommendations based on a user's viewing/purchase history), our proposal is to also track and infer users' opinions about their interactions in the physical world. This is now feasible due to the digital footprints from many user activities (e.g., phone calls, payments), sensory capabilities of smartphones and wearable devices, and advances in machine learning. For example, monitoring a user's location can reveal the restaurants she frequents, and the dentists and plumbers she would recommend can be inferred from her phone call history. Thus, by employing implicit inference and automated sharing, RSPs can significantly increase the number of users whose opinions a user can draw upon for a typical entity.

HotNets-XV, November 09 - 10, 2016, Atlanta, GA, USA

 ISBN 978-1-4503-4661-0/16/11…$15.00

DOI: http://dx.doi.org/10.1145/3005745.3005765

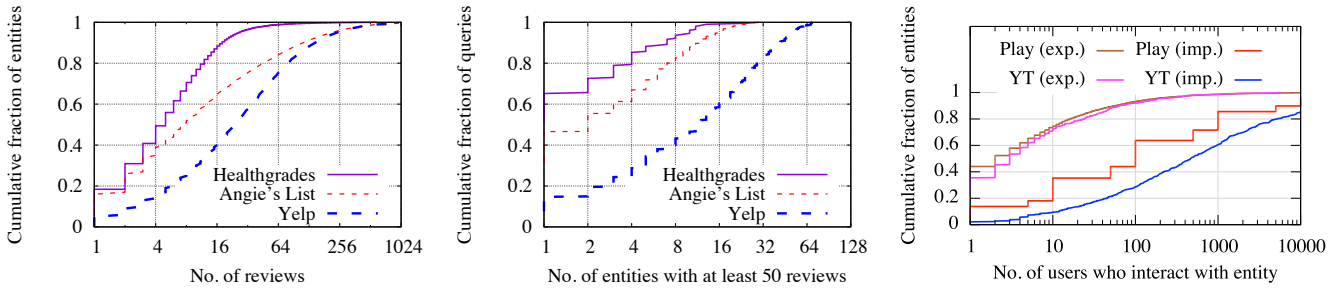

Figure 1: **(a) Distribution across entities of number of reviews. (b) Distribution across queries of the number of matching entities with 50 or more reviews. (c) Comparison on Google Play and YouTube (YT) of the number of users who explicitly interact (e.g., post review, +1) versus number of users who implicitly interact (e.g., install app, view video).**

Service	# of Categories	# of Entities
Yelp	9	24,417 restaurants
Angie's List	24	26,066 service providers
Healthgrades	4	24,922 doctors

Table 1: **Summary of measurements.**

However, extending the functionality of recommendation services as we envision above entails many new challenges. First, inferring a user's opinion about an entity based only on observations of the user's interactions with that entity is inherently associated with uncertainty as compared with relying on the user's explicit input. Second, even if accurate inference of opinions is feasible, users will be wary about their activities being continually monitored and having to trust RSPs to protect their privacy. These privacy concerns are likely to put off the sizeable fraction of users who seldom share their own opinions on existing services. Lastly, RSPs will have to deal with new kinds of fraud; instead of detecting fake reviews either by analyzing their content [20, 19] or by detecting groups of colluding accounts [18, 32], an RSP will have to identify suspicious activity patterns aimed at getting the RSP to infer fake recommendations.

In this paper, we present a sketch of solutions to these unique challenges that need to be tackled to enable implicit inference and automated sharing on recommendation services. On the one hand, to deal with uncertainty, we propose accounting for the "effort" a user puts in when choosing to interact with a particular entity. On the other hand, we discuss how the use of anonymity-preservation techniques to protect privacy can be balanced with the need to detect fraudulent activity. Put together, with these modifications to recommendation services, all users can benefit from a more comprehensive collection of opinions than is the case today.

2 Motivation

We begin by describing our measurements of three popular recommendation services and our takeaways from analyzing these measurements.

Measurements. We crawl reviews from three services—Yelp, Angie's List, and Healthgrades—all three of which feature in Alexa's top 500 websites. Yelp is used largely for sharing reviews of restaurants, Healthgrades for doctors, and Angie's List for various kinds of service providers such as electricians, plumbers, and gardeners.

On all three services, we issue a number of queries and crawl the reviews associated with each of the results. Each query comprises the combination of a zipcode within the US and a category. We focus on locations where the number of reviews are likely to be high by using the most populous zipcode in each of the 50 states in the US. In each zipcode, the set of categories we query for varies across services: on Yelp, we query for 9 popular cuisines; on Healthgrades, we query for 4 types of doctors (dentists, family medicine, pediatrics, and plastic surgery) for which users are likely to rely on recommendations found online; and, on Angie's List, we query for all 24 types of service providers listed on the site. Table 1 summarizes the queries we used and the total number of entities we discovered across all of our queries.

Lack of reviews. First, we examine the number of reviews per entity (restaurant, doctor, or service provider). Figure 1(a) shows that, on all three services, a large fraction of entities have very few reviews. The median number of reviews is 8, 5, and 25 on Angie's List, Healthgrades, and Yelp. This is despite each of these recommendation services being the most popular for the types of entities it caters to and despite our focus on the most populous locations in the US.

The low number of reviews for most entities implies that, for any query issued by a user, the user can form an informed opinion only of a small number of results. Figure 1(b) shows that, for the median query in our measurements, the number of results with at least 50 reviews is 12 on Yelp, 2 on Angie's List, and 1 on Healthgrades, all of which constitute a small fraction of the total number of results that match the median query. For example, though Yelp returns 127 Chinese restaurants near zipcode 19120 (Philadelphia), only 4 of these results have 50 or more reviews. Similarly, Healthgrades lists 248 dentists near zipcode 11368 (New York), but only 13 have over 50 reviews. Thus, for most queries, a user has to either go with a result that has a limited set of reviews or choose from a small set of options that do have a sizeable number of reviews.

Passive consumers dominate. Next, we examine if the low number of reviews for an entity is typically because only few

users have interacted with the entity and are capable of writing a review. For this, we turn our attention to two other services—Google Play and YouTube; unlike Yelp, Angie's List, and Healthgrades, which can only monitor the set of entities for which a user views reviews on their service, Google Play and YouTube can also identify the number of users who interact with each entity (i.e., download an app or view a video). We randomly selected 1000 apps on Google Play and 1000 videos on YouTube. For every selected entity, we crawled the number of users who have explicitly contributed feedback (in the form of a review, comment, rating, favorite, like, etc.), and the number who have interacted with the entity (viewed a video or downloaded an app).

As we see in Figure 1(c), the discrepancy between the number of users who have interacted with each entity and those who have explicitly provided feedback is more than an order of magnitude. On services such as Yelp, Angie's List, and Healthgrades, we expect the discrepancy between those who interact and those who provide feedback to be likely worse because users interact with the entities listed on these sites (restaurants, service providers, and doctors) offline and need to remember to return to the online service and provide their input. Thus, if the opinion of even a fraction of those who have interacted with an entity but not provided feedback can be implicitly inferred, these numbers suggest that the number of opinions that users can draw upon for a typical entity can be dramatically increased.

3 Overview

RSPs could address the paucity of reviews by recruiting more users or by incentivizing users to post more reviews. While these strategies may help, the current state of affairs is despite each of the recommendation services we studied already having tens of millions of active users [7, 2, 4] and RSPs having gone to great lengths to entice users [12]. Alternatively, if an RSP attempts to increase the chances of its users posting reviews by reminding them to do so, for the types of entities we are considering (doctors, plumbers, restaurants, etc.), an RSP will need the ability to track a user's interactions in the physical world in order to even identify when a user should be sent a reminder.

Therefore, we believe a more prudent course of action would be to figure out how to benefit from the experiences of the silent majority.

3.1 Architecture

We envision RSPs re-architecting their services such that they not only accept reviews from users like they do today, but also enable users to discover recommendations implicit from other users' activities. The high-level principle is that any form of explicit input required from users—even a simple Yes/No to approve sharing of inferred opinions—will limit user participation. Hence, RSPs must completely automate the process of gathering input from their passive users. A similar rationale led to the development of the Portable People Meter [13], a device that monitors the radio and television channels that users are listening to or viewing, without relying on explicit input from users.

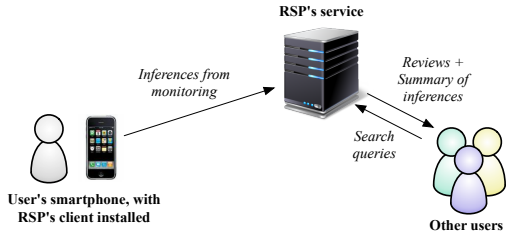

Figure 2: Envisioned architecture of a recommendation service.

We envision recommendation services will be designed as shown in Figure 2, with the following modifications to the RSP's client and service.

Inferring user-entity interactions. First, RSPs will need to modify their smartphone apps to monitor the user's interactions with entities of interest. Note that many popular recommendation services such as Yelp and TripAdvisor already have tens of millions of users who use their smartphone app; Yelp even gets 70% of its search queries from smartphone users [7]. To monitor users' interactions with entities of interest, RSPs' apps will need persistent access (even when the user is not interacting with the app) to inputs such as location, phone call history, and emails. An app can then map these sensitive inputs to the corresponding entities (e.g., map location to restaurant or phone number to dentist).

Inferring user opinions. Given the increasing array of sensors on wearable devices (e.g., heart rate monitors on smartwatches [9]), an RSP may be able to infer a user's opinion about an entity by monitoring the user's emotions when interacting with the entity. In this paper, we restrict our consideration to more modest means of inferring a user's recommendations: by observing repeated interactions between the user and a entity. To identify repeated interactions, an RSP will need to maintain a history of every user's interactions.

Enabling recommendation discovery. An RSP's service can enable its users to benefit from the inferences made by its app by modifying its search interface. For every search result, the RSP can show not only reviews explicitly contributed by users but also a summary of inferred opinions. Unlike the use of collaborative filtering [30] to suggest recommendations based on the entities that a user has interacted with, a search-based interface is more widely applicable. For example, any particular user is likely to have interacted with only one or at most a few doctors and plumbers, preempting the inference of the user's preferences.

3.2 Challenges

Modifying recommendation services as above will require RSPs to tackle several new challenges that they do not have to in their current avatar where they only focus on sharing reviews posted by users.

- **Uncertainty:** The proposed changes to RSPs are worthwhile only if users' opinions can be accurately inferred. In contrast to explicitly provided input, the process of im-

plicitly inferring a user's opinion is inherently uncertain. After all, the RSP would be attempting to infer what is in the user's mind based on observations of the user's interactions with entities. Prior work on implicitly inferring a user's level of interest in a particular entity has focused only on web pages [17] and videos [14] seen online; the features used for inference in these cases (e.g., mouse scrolling and eyeball tracking) are unavailable when inferring a user's opinions about entities in the physical world.

- **Privacy:** Implicit inference and automated sharing raise a number of privacy concerns for users: 1) for RSP-provided apps to monitor users' interactions with real-world entities, they have to be permitted access to sensitive data such as location and phone call history, 2) RSPs need to maintain a history of user-entity interactions in order to identify recommendations, and 3) in order to ensure that decision overhead does not limit opinion sharing, inferences made will have to be shared with others without requiring approval from the user. RSPs risk losing users unless these privacy concerns are adequately addressed.
- **Security:** Reliance on implicit inference also exposes recommendation services to new types of fraud. Instead of posting fake reviews, fraudulent users can attempt to get an RSP to infer fake recommendations by exposing the RSP-provided app to artificial user activity.

4 Design

We next discuss the high-level principles that an RSP can apply to tackle each of the previously described challenges.

4.1 Handling uncertainty

As mentioned previously, we anticipate RSPs inferring a user's recommendation of an entity based on observations of repeated interaction between the user and the entity. However, repeated interaction is of course not always a sign of endorsement; an RSP should not attribute loyalty to what is laziness or compulsion. For example, a user's repeated phone calls to a plumber may be because the plumber did a poor job to begin with. Or, a user may frequent a restaurant only because it is one of the few close to where the user works that satisfy the user's dietary restrictions.

We envision RSPs taking one of two approaches to deal with such uncertainty in implicitly inferring users' opinions.

Effort is endorsement. One approach would be to infer a predictive classifier that takes as input observations of a user's interactions with an entity and either outputs a numerical rating between 0 and 5 or declares it infeasible to accurately gauge the user's opinion.[1] To learn such a classifier, an RSP can gather training data by correlating observations of user-entity interactions with user-provided ratings for the subset of users who do provide explicit input. However, the main question in using such an approach is: what are the input features for the classifier?

[1] Since implicit inference of opinions will never be perfect, an RSP must strive to identify instances when accurate inference is infeasible and choose to avoid making a judgement about the user's opinion in such cases.

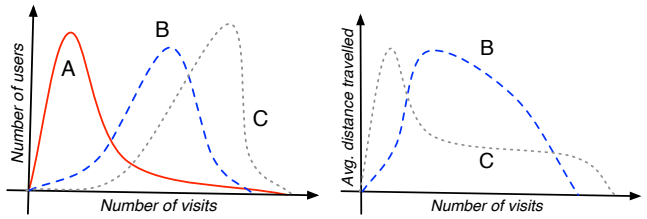

Figure 3: Examples of visualizations comparing users' interactions with different entities.

We believe the key features are of three kinds: 1) features that quantify the effort the user puts in to interact with an entity, e.g., the distance traveled by a user to visit a dentist; 2) features that reveal whether the user tried out many options before settling on a choice or has stuck with a choice merely due to laziness in finding an alternative, e.g., a user's repeated interactions with an electrician mean more if he has availed the services of other electricians previously; and 3) features that quantify the number of other similar options from among which the user selected the entity with which she repeatedly interacts, e.g., nearby restaurants with similar attributes (cuisine, price level, parking, etc.). All of these features are undoubtedly tricky to get right, e.g., the user may have interacted with a different electrician only because she moved to a different city, and gauging the similarity of restaurants depends on a large number of dimensions that are hard to compare such as the menu and ambiance. However, these high level principles are likely to hold an RSP in good stead in selecting input features for its opinion predictor.

Comparative visualizations. An alternative approach to deal with the uncertainty of inferring recommendations would be to *not* attempt to infer the opinions of *individual* users. Instead, the aggregate statistics about users' interactions with an entity can often be quite revealing.

Specifically, when a user issues a search, apart from displaying the set of entities that match the query, an RSP's search interface can also show to the user visualizations that compare users' interactions with those entities. For example, Figure 3(a) compares the histograms of the number of visits per user across three dentists. Such a visualization would make clear that dentist A has very few repeat patients compared to dentists B and C. However, as discussed previously, repeated interactions do not necessarily denote endorsement. Hence, additional visualizations would be vital such as the one in Figure 3(b), which shows that the average distance travelled is more strongly correlated with the number of visits for dentist B than dentist C.

When a set of users interact with the same entity as a group (e.g., visit a restaurant together), an RSP must explicitly account for such instances to ensure that the collective recommendation power of groups does not artificially inflate the aggregate activity associated with an entity.

4.2 Protecting privacy

Even if an RSP is successfully able to manage the uncertainty associated with implicit inference of opinions to provide value for its users, they may still be concerned about using the RSP's modified client due to privacy concerns.

Anonymous uploads. We anticipate any user's primary concerns to be that she has to give an RSP's app access to sensitive data such as location and phone call history, and that the RSP shares its inferences with others without seeking the user's approval. If an RSP uses histograms of inferred ratings or visualizations of aggregate user interactions to export its inferences to users, no information about any individual user is revealed to other users of the service. However, an RSP could potentially share its inferences with third-parties such as advertisers [6], or it could change its interface in a manner that enables other users to infer the entities with which a particular user has interacted [15].

We propose that RSPs address this potential privacy concern by ensuring that it is impossible for RSPs themselves to identify for any user the set of entities with which the user has interacted. For this, an RSP's app should locally map the inputs that it is privy to to the corresponding entities and anonymously upload its inferences to the RSP's service. To prevent the use of aggregate information for potential de-anonymization [24], for every entity with which a user interacts, the app should upload its inferences on an independent anonymous channel, assuming the underlying anonymity network ensures that any two anonymous channels are unlinkable.

Note that, since there is no need for real-time dissemination or discovery of recommendations in the domains we are considering (restaurants, doctors, service providers, etc.), an RSP's app can upload all of its inferences asynchronously, thereby preventing timing attacks. Moreover, having its app anonymously upload its inferences does not hurt an RSP's typical revenue model of showing ads to its users when they browse its service for recommendations.

Privacy-preserving storage of activity history. Recall that, in order to infer recommendations based on repeated interactions, for every entity that a user has interacted with, the RSP needs to store a sequence of interactions, with a number of features associated with each interaction (e.g., duration of interaction, time since last interaction, distance travelled since previous stationary spot, etc.). To protect the user's privacy from the RSP, it appears intuitive to store the user's history locally on her device. But, leakage of this history (e.g., when a user's device is stolen or compromised) could lead to undesirable consequences such as identity theft or stalking. Moreover, storing histories that span several years (e.g., to infer recommendations of rarely used service providers such as dentists and plumbers) is at odds with the current trend among software firms of minimizing the amount of data they store about their users [10].

We believe the solution is for any RSP to store only a recent snapshot of any user's inferred interactions on her device and store the rest of the user's long-term history at the RSP's servers. When a user's device is stolen or compromised, only the user's recent interactions are leaked, similar to how a limited history of a user's web page visits [5] and phone calls [8] would be revealed on today's phones. It is crucial here that, for every (user, entity) pair for which it stores a historical record, the server must not be able to identify the user.

We must address two key challenges in storing interaction histories anonymously at the RSP's servers. First, to protect user anonymity irrespective of the external information available to an RSP [25, 24, 15], the RSP's servers should store a separate interaction history for every (user, entity) pair in a manner such that a user's histories for two different entities are unlinkable. Such unlinkability will ensure that, even if the RSP does have information about some of a user's interactions with a specific entity, the RSP can at most learn about the user's other interactions with that entity, but no information about other entities with which the user has interacted will be revealed. Second, to safely allow users to update their interaction histories without violating user anonymity, authentication of a user's request to update a specific record should be based on an identifier that is unlinkable to either the user or the user's devices.

Our design for storage of histories at the RSP's servers satisfies the above-mentioned two properties as follows. When a user u first installs the RSP's app, the app picks a random number, say R_u, and stores this locally on the user's phone. Thereafter, whenever the app infers the user's interaction with an entity e, it anonymously requests the RSP's servers to add a new record to the history associated with ID $hash(R_u, e)$; if the server is not already storing a history with this identifier, it initializes a new interaction history for entity e and associates it with the specified ID. On the user's device, the RSP's app purges an entry from the user's history once the entry is older than a configurable threshold.

This solution for storing any user's interaction histories at an RSP's servers ensures that the RSP cannot link histories for two different entities stored by the same user. In addition, the client's algorithmic generation of the ID associated with every entity preempts the need for the client to locally store a (entity, ID) mapping, which would reveal all the entities that the user has ever interacted with if data stored on the user's phone is leaked. Even if the value of R_u for a user is leaked, one cannot use it to access the user's information from the server, because the RSP's service only need support requests to update histories but not to retrieve them.

A malicious user could attempt to corrupt others' histories by attempting to guess the R_u value for other users. An RSP can however limit the impact of such attacks by handing out blindly signed tokens [16] at a limited rate to every device and require that every device present a valid token when anonymously uploading information.

4.3 Detecting fake activity

Once RSPs begin to use implicit inferences to inform users' choices, those with vested interests will look to exploit this new vector for influencing users.

Sophisticated adversaries could get an RSP to infer fake recommendations either by modifying the RSP's app (or reverse engineering the app's protocol with the RSP's service) to upload fake information or by providing fake sensor inputs to the unmodified app. To combat such attacks, RSPs can employ remote attestation [31, 26] to confirm that the client has not been modified and use techniques for trustworthy sensing [22, 21, 29, 23, 33] to ensure that the sensor inputs received by the client are legitimate.

However, even without modifying an RSP's client or tampering with the inputs it receives, a fraudulent user can lead the client to infer fake recommendations by generating user activity that appears to indicate significant engagement between the user and an entity. For example, to provide evidence of recommending a particular electrician, a user could simply make several back-to-back phone calls to the electrician, hanging up immediately after calling but resulting in a record in the phone's call history. Similarly, any employee at a restaurant can use his presence at the restaurant daily as evidence of his approval of the restaurant.

To tackle such sources of fraud, an RSP's implicit inference of a user's recommendation of an entity should verify whether the user's engagement with that entity reflects that of a typical user. In the above examples, the service should verify that a user's phone calls to an electrician and visits to a restaurant are appropriately spaced apart and are of reasonable duration. This would greatly raise the bar for generating fake recommendations as fraudulent users will have to incur significant cost and effort to mimic the activities of a typical user. For example, to generate a fake recommendation for a particular dentist, a user will need to be at the dentist's office for reasonable periods of time over several years.

For an RSP to generate a profile of a typical user's activities, we observe that the vast majority of users are not malicious; this is why we can still trust an entity's average rating on any existing recommendation sharing service when the number of reviews for that entity is high. Therefore, since the history of interactions for every (user, entity) pair is stored on an RSP's servers, it can merge these individual histories to generate a profile of the typical user. For example, an RSP that enables discovery of recommendations for service providers can use its knowledge of the observed distribution of gaps between interactions with the same provider to detect fraud when a user's frequency of interaction is significantly greater than is typical for a user. Though it is hard to evaluate whether the interactions between a user and an entity are fake if the number of interactions is small, such an interaction history will have limited influence on others.

Discarding interaction histories that significantly deviate from the activity patterns of the typical user will not completely eliminate fake recommendations, but will help dissuade all but the most concerted malicious users by requiring that they put in significant effort to have an impact.

5 Discussion

Trust model. RSPs stand to profit from learning more about their users, and the closed source nature of the client app for most RSPs makes it hard to verify their privacy claims. Therefore, it would be ideal if the mechanisms that protect user anonymity are implemented in the smartphone OS, so as to make it infeasible for an RSP's client to compromise user privacy. But, until there are many apps that seek to implicitly infer and automatically share opinions, OS developers have little incentive to support such apps.

Transparency. An RSP must ensure that any user of its app has visibility into the inferences the app has made about the user's activities. Exposing inferences to users will not only assuage potential fears about not knowing what an RSP has inferred about them, but also enable users to correct inaccurate inferences made by the RSP. While getting a user to vet every inference made about her activities is impractical, as doing so will nullify the benefits of implicit inference, the need for inferences to be corrected will likely arise often given the challenges in accurately making inferences without user input (Section 4.1).

Location tracking. Any RSP that attempts to infer a user's interactions based on the user's location (e.g., to identify restaurants the user visits) will have to address concerns about energy efficiency. It can do so by exploiting cues from sensors such as the accelerometer [27] (e.g., to sample the user's location only when the user has been stationary for a few minutes and to resample only if the user moves) and by leveraging WiFi and cellular information [28], not only the GPS.

Incentives. Some RSPs are accessed primarily over the Web, e.g., Angie's List has 10-12M users visiting its website every month [2] but only up to 500K users have installed its Android app [1]. In such cases, a user is more likely to install the app if she herself benefits from it, and her incentive is not just to help other users by enabling the RSP to infer her opinions. For example, for any search query issued by a user, the RSP could tailor results based on the user's history.

6 Conclusions

In summary, we call for a redesign of services that enable recommendation discovery based on opinions shared by users. Motivated by the observation that most entities have very few reviews on existing services, we argue that there exists a need and that it is feasible today to make inferences about users' opinions by passively monitoring their activities. We presented various techniques that RSPs will need to employ to tackle the challenges associated with realizing our vision. In combination, our prescribed redesign of recommendation services will enable all of us to benefit from each others' implicit, but often not explicitly stated, opinions.

7 References

[1] Angie's List - Android Apps on Google Play. https://play.google.com/store/apps/details?id=com.angieslist.android.activity.

[2] Angie's List tears down paywall, eyes Facebook competition. http://seekingalpha.com/article/3961669-angies-list-tears-paywall-eyes-facebook-competition.

[3] Fair comment | The economist. http://www.economist.com/node/13174365.

[4] Healthgrades - Wikipedia. https://en.wikipedia.org/wiki/Healthgrades.

[5] How much Safari history is kept on iPad? http://apple.stackexchange.com/questions/181481/how-much-safari-history-is-kept-on-ipad.

[6] How private are health-tracking apps on your phone? http://health.usnews.com/wellness/articles/2016-07-13/how-private-are-health-tracking-apps-on-your-phone.

[7] An introduction to Yelp metrics. http://www.yelp.com/factsheet.

[8] iPhone 5S recent call history. https://discussions.apple.com/thread/6356506?tstart=0.

[9] Nissan launches Nismo smartwatch for drivers. http://www.bbc.com/news/technology-23964797.

[10] What's driving silicon valley to become 'radicalized'. https://www.washingtonpost.com/news/the-switch/wp/2016/05/24/what-is-driving-silicon-valley-to-become-radicalized/.

[11] Yelp and the "1/9/90 rule". https://www.yelpblog.com/2011/06/yelp-and-the-1990-rule.

[12] Yelp's five-star growth engine. http://growthhackers.com/growth-studies/yelp.

[13] V. Aijala, G. Cohen, J. Jensen, W. Lynch, and J. Urbi. Method and apparatus for encoding/decoding broadcast or recorded segments and monitoring audience exposure thereto, 1996. US Patent 5,579,124.

[14] X. Bao, S. Fan, R. R. Choudhury, A. Varshavsky, and K. Li. Your reactions suggest you like the movie: Automatic content rating using reaction sensing. 2013.

[15] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov. "You might also like:" Privacy risks of collaborative filtering. In *IEEE S&P*, 2011.

[16] D. Chaum. Blind signatures for untraceable payments. In *CRYPTO*, 1983.

[17] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *IUI*, 2001.

[18] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Exploiting burstiness in reviews for review spammer detection. In *ICWSM*, 2013.

[19] S. Feng, R. Banerjee, and Y. Choi. Syntactic stylometry for deception detection. In *ACL*, 2012.

[20] S. Feng, L. Xing, A. Gogar, and Y. Choi. Distributional footprints of deceptive product reviews. In *ICWSM*, 2012.

[21] P. Gilbert, L. P. Cox, J. Jung, and D. Wetherall. Toward trustworthy mobile sensing. In *HotMobile*, 2010.

[22] H. Liu, S. Saroiu, A. Wolman, and H. Raj. Software abstractions for trusted sensors. In *MobiSys*, 2012.

[23] W. Luo and U. Hengartner. Veriplace: A privacy-aware location proof architecture. In *SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010.

[24] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets (how to break anonymity of the Netflix prize dataset). In *IEEE S&P*, 2008.

[25] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE S&P*, 2009.

[26] M. Nauman, S. Khan, X. Zhang, and J.-P. Seifert. Beyond kernel-level integrity measurement: Enabling remote attestation for the Android platform. In *Trust and Trustworthy Computing*. 2010.

[27] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In *MobiSys*, 2010.

[28] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan. Energy-efficient positioning for smartphones using Cell-ID sequence matching. In *MobiSys*, 2011.

[29] S. Saroiu and A. Wolman. Enabling new mobile applications with location proofs. In *HotMobile*, 2009.

[30] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, 2001.

[31] D. Schellekens, B. Wyseur, and B. Preneel. Remote attestation on legacy operating systems with trusted platform modules. *Science of Computer Programming*, 74(1):13–22, 2008.

[32] G. Wang, S. Xie, B. Liu, and P. S. Yu. Review graph based online store review spammer detection. In *ICDM*, 2011.

[33] Z. Zhu and G. Cao. APPLAUS: A privacy-preserving location proof updating system for location-based services. In *INFOCOM*, 2011.