Crowdsourced App Review Manipulation

Shanshan Li James Caverlee Wei Niu Parisa Kaghazgaran
Department of Computer Science and Engineering, Texas A&M University
College Station, Texas, USA
{ssli,caverlee,niu,kaghazgaran}@tamu.edu

ABSTRACT

With the rapid adoption of smartphones worldwide and the reliance on app marketplaces to discover new apps, these marketplaces are critical for connecting users with apps. And yet, the user reviews and ratings on these marketplaces may be strategically targeted by app developers. We investigate the use of crowdsourcing platforms to manipulate app reviews. We find that (i) apps targeted by crowdsourcing platforms are rated significantly higher on average than other apps; (ii) the reviews themselves arrive in bursts; (iii) app reviewers tend to repeat themselves by relying on some standard repeated text; and (iv) apps by the same developer tend to share a more similar language model: if one app has been targeted, it is likely that many of the other apps from the same developer have also been targeted.

KEYWORDS

app reviews; manipulation; crowdsourcing; user behavior

1 INTRODUCTION

Mobile app marketplaces like Google Play and Apple's App Store serve as the nexus for many of our online experiences. With the rapid adoption of smartphones worldwide and the reliance on app marketplaces to discover new apps, these marketplaces are critical for connecting users with apps [5]. A key factor driving user engagement with apps is user reviews and ratings. Indeed, previous research has explored text mining to identify fine-grained app features mentioned in reviews [4] and how user reviews can improve app retrieval through methods that exploit reviews [7, 9]. From a software engineering perspective, researchers have developed methods to extract informative reviews that can help developers respond to user feedback [2] and observed that apps that respond to user feedback (e.g., via bugs or desired features suggested in user reviews) in future releases do indeed increase their ratings over time [2, 8].

And yet, these user reviews and ratings may be strategically targeted by app developers to artificially promote their own apps (or potentially, to demote the apps of competitors). Indeed, seminal work by Chandy et al. [1] identified spam app reviews which aim to deceive users to download harmful apps or impede them from downloading benign apps. Follow on research has explored collusion in app rating systems [12, 14], explored the use of incentivized review marketplaces to attack the trustworthiness of app reviews [13], and found evidence of app popularity manipulation [15].

> Amazing Party Going App!! ★★★★★
> *After downloading and using this app I started thinking about all the times in the past I could have used this amazing app. I love the party finder and text blocker and call blocker features more than anything. Drunk texting and dialing issues are a problem of the past. Definitely a great phone tracker as well. Great app if you enjoy the nightlife, drinking, pre game parties, or the frat life. The app also helps you keep track of your friends with the friends finder feature, and who hasn't lost a drunk friend a time or two. Amazing ideas and user friendliness. I will not go out without having this app ready again!*

Figure 1: Example of a suspicious review.

In this paper, we investigate a complementary attack vector on the trustworthiness of app reviews: crowdsourcing platforms that allow a single manipulator to martial a crowd of human review writers to target app reviews. Such crowd-based manipulation has been identified as a serious threat to the viability of many systems that rely on user-generated content [3, 6, 10, 11], but there has been little if any study of crowdsourced targeting of app reviews. We present our initial investigation into the use of crowdsourcing platforms to launch targeted review manipulation on these app marketplaces. Our overarching goal is to study if these platforms are susceptible to crowdsourced attacks – Do crowdsourced reviews bypass review filters to actually be posted? Are reviews positive or negative? Do they actually impact the aggregate ratings of apps? Are there correlations among a developer's apps in terms of being targeted for manipulation? Does the platform's "related apps" feature expose users to more targeted apps?

Toward answering these questions, we sample 100+ targeted apps from a popular crowdsourcing platform (and a control group of randomly selected apps from the App Store) and make the following observations: (i) we find that apps targeted by crowdsourcing platforms are rated significantly higher on average than other apps, indicating that app manipulation is focused on app promotion, rather than in punishing the apps of competitors; (ii) the reviews themselves arrive in bursts, and have an immediate positive impact on the average ratings of the apps; (iii) the patterns of linguistic evolution suggest that app reviewers tend to repeat themselves by relying on some standard repeated text; and (iv) apps by the same developer tend to share a more similar language model: if one app

SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan

DOI: http://dx.doi.org/10.1145/3077136.3080741

has been targeted, it is likely that many of the other apps from the same developer have also been targeted; in contrast, we find the platform's "related apps" feature tends to identify apps that have not been targeted.

2 DATA

We focus on crowdsourced manipulation of apps launched from the Microworkers crowdsourcing platform. Microworkers is similar in style to Amazon Mechanical Turk, where requesters may post tasks and workers can select from a variety of tasks to perform. In practice, workers can earn around $1.50 for a review. We crawled all tasks on Microworkers from October 2016 to February 2017. We identified 114 unique iOS apps from Apple's App Store and 51 Android apps from the Google Play store. A sample review on a targeted app is shown in Figure 1. In this paper, we focus on iOS apps; hence, we crawled the metadata associated with each of the 114 apps (e.g., the developer, the version number), the reviews associated with each app, and the reviewers of each app. In total, we collected 50,461 reviews. We refer to these apps as **targeted apps**.

As a point of comparison, we also select a set of random apps. Note that the targeted apps are not themselves randomly distributed throughout the App Store – most of the targeted apps are in the category Games, as highlighted in Table 1. To control for variations in review types across categories, we randomly sampled 485 apps while following the category distribution of the targeted apps. In total, we collected 142,400 reviews for these random apps. We refer to these apps as **random apps**.

Category	Fraction
Games	77.0%
Photo & Video	3.9%
Stickers	3.9%
Productivity	1.9%
Health & Fitness	1.9%
Business	0.9%
All Others	11.4%

Table 1: Targeted apps category distribution

3 INVESTIGATION

Impact of Crowdsourced Targeting on Ratings. We begin by examining the ratings of apps that have been targeted. Are crowdsourcing platforms promoting apps or demoting the apps of competitors? Figure 2 shows the distribution of ratings across both the targeted and random apps. From random apps, fewer than 10% of the apps boast a rating of 5.0, with a non-trivial number receiving ratings of 2.0 to 4.5. In contrast, the targeted apps are overwhelmingly rated as 4.5 or 5.0 on average, indicating that the goal of crowdsourced manipulation is to boost the ratings of one's own apps.

Indeed, we can measure the direct impact of crowdsourced targeting by measuring the average rating *before* and then *after* the crowdsourcing task was posted. We see in Figure 3 the average rating distribution of targeted apps before and after they were targeted

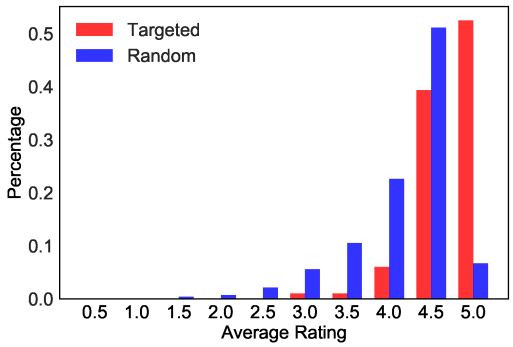

Figure 2: Average ratings for targeted and random apps.

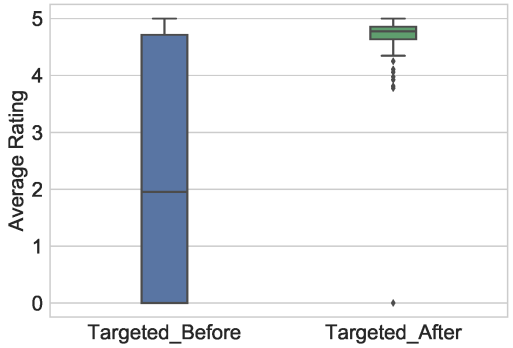

Figure 3: Ratings before and after promotion.

for promotion. The impact is clear: before targeting, apps have a median rating of around 2.0 with a wide distribution of ratings. After targeting, the apps are skewed upward to a very narrow range in the 4.0 to 5.0 range. Hence, the current goal of crowdsourced app manipulation is self-promotion (rather than competitor demotion) and the impact is a clear improvement of app rating.

Review Burstiness. Next, we turn to the burstiness of reviews. Since crowdsourcing platforms can organize large numbers of workers in a short time, we expect that targeted apps may receive bursts of reviews in a short time window. Hence, we measure the standard deviation of the review time for each targeted app as shown in Figure 4. This figure compares the distributions between the review time pattern for targeted apps and random apps. In this case, a small standard deviation corresponds to many reviews being posted in a short time window, whereas a larger standard deviation corresponds to reviews posted over a long time period (and hence, lacking burstiness). We can observe that the distribution for the timing of reviews of targeted apps is upper-left-skewed indicating that these reviews tend to be posted in bursts.

Similarly, we can measure the impact of promotion by measuring the burstiness before and after these apps have been targeted for promotion on the crowdsourcing platform. Figure 5 shows that the "after" curve is skewed up and to the left, indicating that these apps' reviews are burstier after promotion than before.

Reviewer activity. Given the set of 114 targeted apps, we identify all reviewers who have reviewed at least two promoted apps.

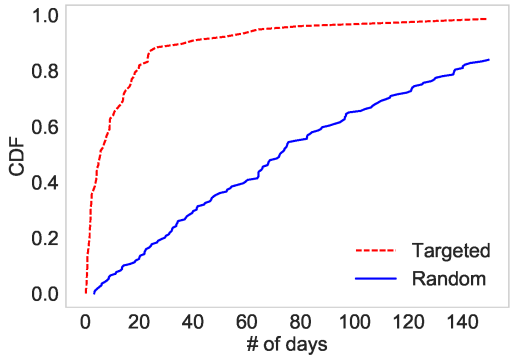

Figure 4: Burstiness of reviews: The standard deviation of review time for targeted and random apps.

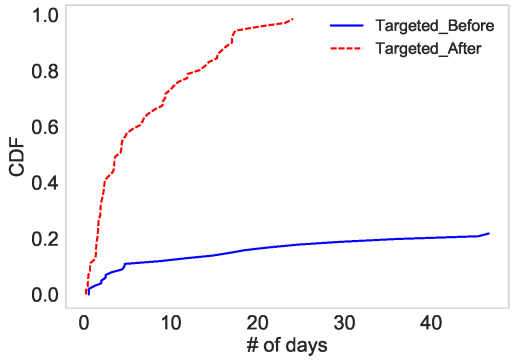

Figure 5: Burstiness before and after promotion.

The intuition is that a reviewer who has reviewed more than one promoted app is most likely a worker from a crowdsourcing platform. We then find how many apps these suspicious reviewers have reviewed. We find many reviewers have rated tens of targeted apps. There are even five reviewers who reviewed 70 or more promoted apps. Therefore, active reviewers of targeted apps are likely suspicious reviewers.

We further collect those suspicious reviewers who reviewed at least two targeted apps. Totally, we find 328 such reviewers. We then collect all of the apps from the reviews they have reviewed, including apps that are not in our known targeted app set. We find that the highest number of apps reviewed by a suspicious reviewer is close to 400, indicating the professionalization of these crowdsourcing platforms for impacting a large number of apps.

Self-similarity of reviewers. Since many reviewers are clearly reviewing only for compensation, perhaps their level of effort is not as strong as a legitimate reviewer. Hence, we consider the self-similarity of a reviewer's reviews. By measuring the average Jaccard similarity between each two sequential reviews for each suspicious and random reviewer, we arrive at Figure 7. We can see that reviewers of random apps demonstrate less self-similarity (left skewed distribution), whereas the reviewers of targeted apps engage in less lexical variation (the distribution is skewed more rightward); hence, these reviewers tend to mimic themselves by using repeated terms and phrases in their reviews. Since suspicious reviewers may control multiple accounts, we are interested to explore in our future work whether we can uncover such cliques through additional self-reviewer and cross-reviewer similarity analysis.

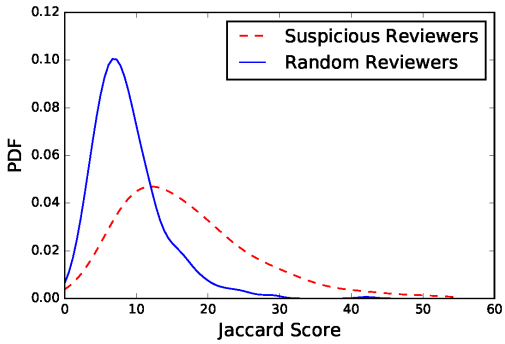

Figure 6: Self-similarity distribution for reviewers with at least 20 reviews.

Additionally, we can measure the evolution of these suspicious reviewers by measuring the self-similarity at different stages of their "life" in the App Store. Here, we divide a user's reviews into five equal stages ordered by time. In each stage, the average similarity between the last review and every other review is considered as the self-similarity score for that stage. In this way, we can measure the initial state of the reviewer (Stage 1) versus the final state of the reviewer (Stage 5). As we see in Figure 8, suspicious reviewers begin their life appearing to be more like a legitimate reviewer but then evolve right-ward, meaning they begin to recycle terms and phrases in their reviews. This suggests an opportunity to identify reviewers who begin to engage in such promoted review efforts.

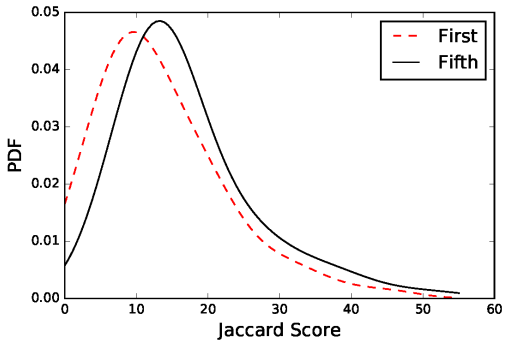

Figure 7: Evolution of self-similarity distribution for suspicious reviewers.

Developers vs. "Related Apps". Finally, we consider the neighborhood around these targeted apps. Do apps by the same developer also appear to be targeted? What about for apps in the platform's "related apps" feature? Since users may explore these neighborhoods for new apps to download, we are interested if these neighborhoods expose users to even more targeted apps. For this experiment, we

measure the Kullback-Leibler (KL) divergence between the unigram language model for the reviews of a targeted app versus either the language model of (i) all the other apps created by the same developer; or (ii) all of the "related apps" (a.k.a."customers also bought") suggested by the App Store. We measure the KL-divergence for all of our targeted apps and random apps.

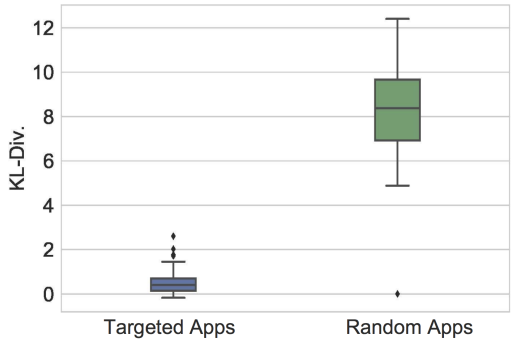

Figure 8: KL-divergence for apps from the same developer.

The boxplots in Figure 9 shows the distribution for apps by the same developer. The left boxplot summarizes each targeted app's *KL(Targeted||AlsoDeveloped)*, where *Targeted* and *AlsoDeveloped* are the respective word probability distributions in the target app's reviews and its developer's other apps' reviews. The right box plot can be interpreted analogously, but for random apps. Since *KL(Targeted||AlsoDeveloped)* provides a quantitative estimate of how much the reviews for a targeted app linguistically differ from reviews for apps by the same developer, we can conclude that reviews of targeted apps and the developer's other apps are more similar than of reviews of random apps. This is possible evidence that developers may be targeting many of their own apps. Since our sampling method of crawling crowdsourcing platforms provides a partial window into what apps are being targeted, this style of finding nearby apps may uncover additional apps that have been targeted.

In contrast, we find in Figure 10 that the KL-divergence for "related apps" is similar for both targeted and random apps. We interpret the similar boxplots to mean that there is less chance of other targeted apps infecting the related apps functionality, suggesting that app recommendations are more difficult to manipulate.

4 FUTURE WORK

This initial examination of crowdsourced app reviews has shown both the impact of these reviews on app marketplaces, but also some clues toward mitigating their impact. In our continuing work, we are expanding our collection of targeted apps to include more App Store apps as well as apps from the Google Play store. We are also exploring machine learning models for identifying hidden targeted apps, reviews, and reviewers. We are especially interested in creating evolutionary user models to capture more fine-grained changes in reviewer styles over time toward automatically detecting reviewers who engage in occasional review-for-pay jobs.

Acknowledgement. This work was supported in part by AFOSR grant FA9550-15-1-0149.

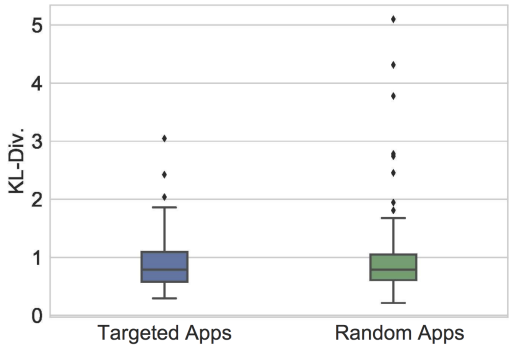

Figure 9: KL-divergence for "related apps".

REFERENCES

[1] Rishi Chandy and Haijie Gu. 2012. Identifying spam in the iOS app store. In *ACM WebQuality*.
[2] Ning Chen and et al. 2014. AR-miner: mining informative reviews for developers from mobile app marketplace. In *ACM International Conference on Software Engineering*.
[3] Amir Fayazi, Kyumin Lee, James Caverlee, and Anna Squicciarini. 2015. Uncovering crowdsourced manipulation of online reviews. In *ACM SIGIR*.
[4] Emitza Guzman and Walid Maalej. 2014. How do users like this feature? a fine grained sentiment analysis of app reviews. In *IEEE Requirements Engineering Conference*.
[5] Isabel Kloumann and et al. 2015. The Lifecycles of Apps in a Social Ecosystem. In *WWW*.
[6] Kyumin Lee, Prithivi Tamilarasan, and James Caverlee. 2013. Crowdturfers, Campaigns, and Social Media: Tracking and Revealing Crowdsourced Manipulation of Social Media. In *ICWSM*.
[7] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. 2014. New and improved: modeling versions to improve app recommendation. In *ACM SIGIR*.
[8] Fabio Palomba and et al. 2015. User reviews matter! tracking crowdsourced reviews to support evolution of successful apps. In *IEEE International Conference on Software Maintenance and Evolution*.
[9] Dae Hoon Park and et al. 2015. Leveraging user reviews to improve accuracy for mobile app retrieval. In *ACM SIGIR*.
[10] Jonghyuk Song, Sangho Lee, and Jong Kim. 2015. Crowdtarget: Target-based detection of crowdturfing in online social networks. In *CCS*.
[11] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. 2012. Serf and turf: crowdturfing for fun and profit. In *WWW*.
[12] Zhen Xie and Sencun Zhu. 2014. Grouptie: toward hidden collusion group discovery in app stores. In *ACM Conference on Security and Privacy in Wireless & Mobile Networks*.
[13] Zhen Xie and Sencun Zhu. 2015. AppWatcher: Unveiling the underground market of trading mobile app reviews. In *ACM Conference on Security and Privacy in Wireless & Mobile Networks*.
[14] Zhen Xie, Sencun Zhu, Qing Li, and Wenjing Wang. 2016. You can promote, but you can't hide: large-scale abused app detection in mobile app stores. In *ACM ACSAC*.
[15] Hengshu Zhu, Hui Xiong, Yong Ge, and Enhong Chen. 2013. Ranking fraud detection for mobile apps: A holistic view. In *ACM CIKM*.