The Value of Information in Mobile Ad Targeting

Omid Rafieian* University of Washington

Hema Yoganarasimhan* University of Washington

Abstract

Mobile in-app advertising is now a growing industry. We examine the value of information in improving targeting outcomes in this context. We study a large scale data set (of over 150 million data points across one month) from a leading in-app ad-network in Iran. We first examine which targeting factors improve the targeting outcomes. We build a Machine Learning framework with over 150 features and employ a MART algorithm to train the model. We find that our model improves prediction significantly over the baseline, and performs much better than logistic regression and OLS models. We find that behavioral targeting based on user-level features is more valuable than contextual targeting based on ad-app features. We then use our model to examine how different data-sharing arrangements between the ad network and advertisers will affect an advertisers' ability to do targeted bidding. We show that the least privacy-preserving arrangements are also the most valuable for advertisers. Interestingly, we also find that large advertisers benefit the most from data-sharing arrangements, which raises concerns on data-sharing cabals. Finally, we also examine whether the ad-network is incentivized to share targeting data with advertisers and show that the ad-network may actually prefer to withhold information from advertisers to improve their own revenue since targeted bidding by advertisers softens competition. Thus, by design, the ad-network may be incentivized to preserve users' privacy.

Keywords: mobile, mobile advertising, machine learning, targeting, competition, big data, privacy

*We are grateful to an anonymous firm for providing the data and to the UW-Foster High Performance Computing Lab for providing us with computing resources. We thank the participants of the 2016 Invitational Choice Symposium for their feedback. Please address all correspondence to: rafieian@uw.edu, hemay@uw.edu.

1 Introduction

Smartphone adoption and usage has grown exponentially in the last few years, with more than two billion people owning a smartphone today (Statista, 2016). In 2015, Internet access and usage through mobile devices overtook the traffic from desktops (Chaffey, 2015). Apart from browsing, consumers use these devices for a wide range of activities including navigation, purchase, and entertainment. Indeed, a recent study finds that consumers use mobile phones for an average of 2.8 hours per day. A majority of this time is actually spent outside of the browser on applications, popularly known as "apps" (Perez, 2015). Apps can serve a variety of purposes and there are a numerous categories of apps – social networking apps (Facebook, Instagram), messaging and communication apps (WhatsApp, Snapchat), video channel apps (Youtube, Netflix), game apps (Candy Crush, Pokemon GO), transportation apps (Uber, Lyft), and so on. Indeed, the proliferation and growth of the mobile apps is one of the main drivers behind the widespread adoption of smartphones.

The importance of apps to the mobile eco-system implies that developing and monetizing apps is of interest to many players in this space. There are three well-established monetization strategies – 1) paid model (pay for the app at download or through subscription), 2) freemium model (the basic app is free, but consumers pay for the premium version or for in-app purchases for a better experience), and 3) in-app advertising (the app is free, but consumers are shown advertisements when using the app). In-app advertisements are usually small display advertisements shown at the top or bottom of the screen and are relatively unobtrusive.[1] In-app advertising is the most dominant monetization strategy used by app-developers across the world and are one of the main reasons why a majority of the apps today are free to download. Mobile advertising spending now exceeds 13 Billion US Dollars and a large percentage of this comes from in-app ads (Shaul, 2016). Thus, the success of in-app advertising is crucial to the sustained growth of app markets as well as the smartphone industry.

There are four key players in this marketplace – 1) publishers or app-developers, who develop apps and display advertisements to monetize their app, 2) advertisers, who buy eye-balls or clicks with the goal of attracting consumers, 3) the consumers who obtain free apps in exchange for their eye-balls and personal data, and 4) ad-networks, platforms that match advertisers and publishers. While invisible to consumers, ad-networks are the engines that drive the advertising markets. They are two-sided platforms, where publishers can sell their ad slots (impressions or eye-balls) and

[1] A more recent form of mobile advertisements, known as interstitial advertisements, are more obtrusive. They take over the screen and often play a video for a short period of time. They are typically shown between two levels in a game app or in between a video in an entertainment app. In our setting, there are no interstitial advertisements.

advertisers can bid to show their ads. The ad-network then tries to efficiently match each impression (on a publisher) to an appropriate ad. Any revenue generated is shared between the publishers and the ad-network. A prominent example of such apps is Facebook whose mobile ad revenues reportedly accounted for 80% of its total $5.6 Billion ad revenues, in the fourth quarter of 2015 (Petterson, 2015).

Since advertising is costly for advertisers, it would be more profitable if it has higher response rate. Similarly, higher response rate is desirable for both publishers and platforms, since it directly affects their profits. Further, we can argue that users also prefer to see more relevant ads, instead of seeing random ads. Therefore, the incentives of all four players are aligned with respect to targeting, which can lead to higher response rate. Generally, there are three different targeting approaches in in-app advertising: First, demographic targeting, by which ads are targeted based on user demographics. Second, contextual targeting that matches ads to the context of the apps. Third, behavioral targeting in which users are targeted based on their past behavior.

The main characteristic of ad-networks is the centralized access to the information about all other players, including users, publishers, and advertisers. As such, an effective targeting is achieved at a reasonably low cost, utilizing the centralized data of the ad-network. The reduced cost of targeting is what Goldfarb (2014) calls the main difference between online and offline advertising, and accounts for the rapid growth of online advertising. Hence, understanding the effectiveness of targeting variables and the returns to targeting is essential in this marketplace. While past marketing research has focused on the interplay between targeting and privacy regulations in a field setting (Goldfarb and Tucker, 2011d), there has not been much research on how data could be potentially used for targeting purposes and which targeting variables are more important in terms of returns. We believe that answering these questions is of crucial importance in studying problems related to targeting and privacy.

We first look into this problem from the point-of-view of the data-owner. Therefore, the first question is how to utilize the data in order to better target the ads. For this purpose, we need to have a prediction model, which accurately predicts the outcome of an ad impression (view). After building a state-of-the-art prediction model, we examine how much we can improve an ad's response rate (in this case, click) through targeting. This basically enables us to measure the returns to targeting. Going further, we can also find which features of the data are most helpful in improving targeting by comparing the result of behavioral targeting with contextual targeting.

Having a better knowledge about returns to targeting allows us to investigate substantive questions regarding the ad-network's decision to share the data with advertisers. This notion of data-sharing is directly related to the concept of imperfect targetability proposed by Chen et al.

(2001), because the more information is given to an advertiser, the less imperfect targeting it could do. However, as offered by Chen et al. (2001), imperfect targeting can soften the competition among advertisers, which may not be a desirable property for the ad-network. Moreover, as Levin and Milgrom (2010) mention, one downside of higher ability to target in online auctions is a huge gap between the first and second bidder, indicating a huge inefficiency in the auction. Hence, this is of high strategic importance for the ad-network to know the consequences of sharing data with advertisers.

Additionally, we aim to identify which types of advertisers benefit more from which data-sharing arrangements. As firms have come to realize the value of user-level data, selling and sharing of data have become a common economic activity. For example, many firms like Adobe are building data co-ops so advertisers can pool information and improve their targeting precision (Liyakasa, 2016). However, it is not clear whether these arrangements are incentive compatible in the long run. For example, suppose firm A significantly benefits from sharing data with firm B, while firm B marginally benefits from this data sharing. As a result, considering the long run outcome, firm B may have no incentive to share its data with firm A, even though this arrangement is marginally profitable for it.

To address these questions, we first propose a machine learning framework to predict the click probability for each impression. The main reason we use a machine learning model is the fact that typical econometrics approaches such as fixed effects are usually far from state-of-the-art prediction accuracy (He et al., 2014). To achieve an accurate click prediction, we first define functions that incorporate factors affecting the probability of click. We then generate features based on these functions. We split our data into three datasets: train, cross-validation, and test. We finally train our model, using MART, and evaluate the prediction results on test data.

We apply our machine learning framework to a leading Android in-app advertising platform in Iran. We are provided with their historical data at impression-level, over one-year period, stored on their server on a daily basis. Daily data contain more than 50 million ad impressions on average. Each impression in the data contains the information about time, users, apps, and ads. We conduct our analysis on one month of the data, in October 2015. We then make models of data-sharing arrangements. Inspired by real situations in online advertising, we consider different scenarios with respect to ad-network's decision to share the data with the advertisers. We categorize ads into low, medium, and high type based on their total number of impressions in one month. We find which types of ads benefit more in different data-sharing arrangement scenarios.

The findings of this paper are presented as follows. First, we evaluate our full model, using different methods. We find that Multiple Additive Regression Trees (MART) outperform Logistic

Regression or OLS. Our results for MART indicate state-of-the-art accuracy. Second, we show evidence regarding the data adequacy procedure, suggesting that 200,000 users sample is sufficiently large to capture the desirable variation in the data. Third, we find that using IP as the user identifier instead of Advertising ID would lead to a considerable information loss, since IP is not a stable identifier for users. Fourth, we apply our model to counterfactual data-sharing scenarios and we find that larger advertisers benefit more when the platform allows them to access their own data. However, in case of data-sharing, smaller advertisers benefit more than larger advertisers.

In sum, our paper makes three contributions to the literature. First, we examine the value of information in improving targeting by differentiating between different sorts of targeting. This improvement achieves the state-of-the-art accuracy. Second, our paper makes a methodological contribution to the growing machine learning literature on click prediction. To the best of our knowledge, this is one of the first click prediction models that combine previous theories of advertising with mathematical functions. We also suggest an alternative for high-dimensional fixed-effects models. This contributes to the joint literature on econometrics and machine learning. Third, our paper provides empirical measures that can be taken into account in privacy and data-sharing regulations.

2 Related literature

First, our paper relates to the analytical work on targeting in marketing and economics. Early papers in this area show that imperfect targeting can benefit firms by softening competition (Chen et al., 2001; Iyer et al., 2005). Similarly, Levin and Milgrom (2010) show that increased targeting can give rise to narrow markets in online ad auctions (where the gap between first and second bid is high), which can effectively shrink the ad-platform's profit by precluding it from extracting sufficient rent from the highest-value bidder. Thus, a recurring theme in the analytical papers is that too much targeting can lower both advertiser and platform profits in a competitive setting.

Early empirical papers on targeting mainly focus on modeling consumer response rates. They show that firms can improve customer responsiveness to marketing activities like pricing and promotions by using their customer databases to personalize and target their marketing activities (Rossi et al., 1996; Ansari and Mela, 2003; Chatterjee et al., 2003; Manchanda et al., 2006; Ghose and Yang, 2009). Specific to online advertising, using data from a series of regime changes in advertising regulations, (Goldfarb and Tucker, 2011a) find that lowering targeting reduces consumer response rates. Interestingly, Lambrecht and Tucker (2013) show that re-targeting ads are not always effective. Please see Goldfarb (2014) for an excellent review of targeting in online advertising.

Note that one key difference between the analytical and empirical literature is that the former focuses on profits and market equilibrium, whereas the latter focuses on response rates within a

firm. While targeting can indeed increase consumer response rates, in a competitive market, it can also lead to harder competition among advertisers, which can worsen advertiser profits, but increase platform profits (and vice-versa). By focusing exclusively on response rates, these empirical papers ignore the profitability implications of targeting. Two recent empirical papers try to address this issue by modeling a market-level equilibrium using structural models. Yao and Mela (2011) present a structural model to estimate advertisers' valuations and show that targeting benefits both advertisers and the platform. Similarly, Johnson (2013) finds that both advertisers and publishers are worse off when the platform introduces stricter privacy policies that reduce targeting. In sum, empirical work does not find much evidence to support the theoretical predictions of the negative effects of too much targeting. However, it is not clear whether these findings are context-specific and/or whether they stem from the fact that both these papers only consider very broad targeting strategies, which can be interpreted as "imperfect targeting". In this paper, we shed some light on this issue by using our machine learning model to examine how different levels of targeting can influence competition between advertisers.

Our work also relates to the growing literature on data-sharing and online privacy. Pancras and Sudhir (2007) was one of the first papers in marketing to examine the incentives of data-intermediaries. They find that a monopolist data-intermediary has an incentive to sell its services using nonexclusive arrangements with downstream retailers and use the maximum history available to target consumers. In our setting, the ad-network or platform has full access to all the consumer, advertiser, and publisher data, and can therefore be interpreted as a data-intermediary. We consider different data-sharing arrangements that the platform can offer to advertisers, the value of these arrangements to advertisers. Recent work in this area has also looked at consumers' responsiveness to advertising under different privacy regimes (Goldfarb and Tucker, 2011b,d,c, 2012; Tucker, 2014). We refer readers to Acquisti et al. (2016) for a detailed discussion of consumer privacy issues.

Another stream of work that relates to our paper is the studies on mobile marketing. One of the advantages of mobile marketing is the ability to track consumers' location via their mobile phones. Therefore, advertisers can efficiently target their ads using consumers' location (Luo et al., 2013). A growing body of work has investigated the effectiveness of location-based targeting in mobile advertising (Ghose et al., 2012; Hui et al., 2013; Andrews et al., 2015). Another body of work has focused on specific features of mobile display advertising. For example, Bart et al. (2014) examine the purchase intention for the products that are high on the utilitarian dimension. In addition, Ghose and Han (2014) build a demand estimation for mobile application marketplace. In this paper, we focus on in-app advertising and determine the features affecting click probability.

From a methodological perspective, our paper relates to the literature on predictive machine

learning using stochastic gradient boosting (Friedman et al., 2000; Friedman, 2001; Friedman et al., 2001; Friedman, 2002). More specifically, it pertains to models of click prediction in online advertising. McMahan et al. (2013) discuss the implementation details of such models using case studies from Google, and whereas He et al. (2014) use ad data from Facebook to make some prescriptive suggestions on feature generation, model selection and learning rates, and scalability. There are two main differences between these papers and ours. First, our goal is substantive – we seek to understand and quantify the impact of different types of information in mobile ad targeting, whereas the previous papers are mainly concerned with presenting methods for predicting clicks in a scalable fashion. For instance, on of our primary goals is to understand which type of targeting is more valuable in mobile ads – contextual or behavioral? So while we use the tools proposed in these papers, the tools are not our end goal. Second, unlike these previous papers, we then use our model to then examine the implications of data-sharing arrangements between the advertisers and the platform, and examine the implications of such sharing for targeting by advertisers, competition in the marketplace, and consumer privacy.

Finally, our work adds to the growing literature on applications of machine learning in marketing. Some early prominent works were mainly in the area of conjoint analysis (Toubia et al., 2004, 2003; Evgeniou et al., 2005, 2007). In the recent years, the range of applications as well methods have broadened to include models employ ML techniques to model consideration sets and heuristics (Hauser et al., 2010; Dzyabura and Hauser, 2011), comparisons of SVM models with standard marketing approaches such as logistic regressions (Cui and Curry, 2005; Huang and Luo, 2016), and multi-taste attributes (Liu and Dzyabura, 2016). Our paper also closely relates to Yoganarasimhan (2016), who presents a framework to do personalized search using stochastic gradient boosted trees. We adopt many of the features of her approach in developing our model, such as the feature-generation functional framework, her data preparation techniques that takes advantage of user-level history, and boosted trees for training.

3 Setting and data

3.1 Setting

Our data come from one of the top three IT companies in Iran, which is the leading Android mobile app marketplace with over 85% marketshare in the category (Faucon, 2015). The marketplace generates receives over 20 million weekly views and has more than 17 million active users, of which more than 4 million have made at least one purchase/payment decision with the firm. The firm provides two types of services. First, it functions as an app marketplace, wherein it provides a platform for software developers to publish their apps and consumers to download/purchase these

apps. Second, it also offers an in-app advertising service, through which app-developers can opt-in to show ads to the users of their app and advertisers can bid on impressions. As is standard in this industry, their ad network runs real-time auctions and algorithms to allocate ads across impressions. Our data comes from the second aspect of the firm's business, *i.e.*, their advertising marketplace.

As mentioned in §1, there are four strategic players in this marketplace.

- Users, who are the consumers of apps. They see the ads shown within the apps that they use and may choose to click on the ads.
- Advertisers, who show ads through the marketplace. They design banner ads (texts, pictures, and gifs are supported) and specify their bid as the amount they are willing to pay per click, and can include a maximum budget if they choose to. Currently, advertisers can target their ads based on the following high-level variables – app category, geographical location, connectivity type, time of the day, mobile operators, and mobile brand of the impression. The platform does not support more detailed targeting at this point in time.
- Publishers, who own the apps and decide whether or not to join the ad network. If they choose to join the network, they accrue revenues based on the clicks generated within their app. Publishers earn 70% of the cost of each click in their app (paid by the advertiser), and the remaining 30% is the platform's commission rate.
- The platform, which functions as the matchmaker between users, advertisers, and publishers. It runs a real-time auction for each impression generated by the participating apps and shows the winning ad in each slot. The cost per click that advertisers are charged for is a function of their own bid, other advertisers' bids (in that specific auction), and the auction rules (see the next paragraph for details). The platform uses a CPC pricing mechanism, and therefore generates revenues only when clicks occur.[2]

The platform uses an auction mechanism called *quasi-proportional auctions* in the literature (Mirrokni et al., 2010). The key distinction between quasi-proportional auction and other commonly used auctions (*e.g.*, Generalized Second Price/GSP or Vickrey) is the use of a probabilistic winning rule.[3] Specifically, the platform uses the following allocation rule, where p_i is the probability that a bidder i of the set of all bidders A with bid b_i and quality score s_i wins the auction.

$$p_i = \frac{b_i s_i}{\sum_{j \in A} b_j s_j} \tag{1}$$

[2]An impression lasts 15 seconds. If a user continues using the app beyond 15 seconds, it is treated as a new impression and the platform runs a new auction to determine the next ad to show the user.

[3]The use of real-time auctions to allocate and price ads is common in digital advertising settings. The actual auction mechanism used depends on the platform and its objectives. For example, Google uses Generalized Second-Price (GSP) auction to sell search ads, whereas Facebook uses Vickrey auctions in a social network setting.

The quality score used by the platform is simply the advertiser's eCTR (expected click-through rate). Currently, the platform simply aggregates all total past impressions and clicks for an ad, and uses the ad-specific CTR as the s_i in their auctions. After each impression, the ad-specific CTR is updated based on whether the ad was clicked or not. Thus, the extent of customization in the quality score is quite low. As it is clear from Equation (1), the advertiser who can generate the highest expected revenue for the platform (the one with the highest value of $b_i s_i$) is not guaranteed to win. Rather, his probability of winning is proportional to the expected revenue generated from him. This probabilistic allocation mechanism generates randomization in ads across users and apps, which facilitates our analyses to a great extent.

Quasi-proportional auctions have some advantages compared to the standard second price auction. While it is well-known that a second-price auction with optimal reserve prices is revenue optimal (Myerson, 1981; Riley and Samuelson, 1981), setting optimal reserve prices requires the auctioneer to know the distribution of valuations within each auction. This is not feasible when the valuations are changing constantly and/or the bidders in the system vary widely, as is commonly the case in online ad auctions. This is especially the case with our platform where the market is changing significantly and advertisers are learning their valuations and responding to them as the marketplace evolves. In a *prior-free* setting such as this, Mirrokni et al. (2010) show that quasi-proportional auctions offer better worse-case performance than second-price auctions, especially when bidder valuations are starkly different.[4] For these reasons, the platform has adopted a quasi-proportional auction mechanism and does not employ a reserve price.

3.2 Data

We have data on all the impressions and corresponding clicks (if any) in the platform, for by all the participating apps for a one month period from 30 September 2015 to 30 October 2015. Each impression in the data comes with the following information.

- Time and date: The time-stamp of the impression.
- IP Address: The Internet Protocol address (IP address) associated with the impression, which is essentially the IP of the accessing user's smartphone when the impression happens.
- Advertising ID: The Advertising ID is a user-resettable, unique, anonymous ID for advertising, provided by Google Play services for all smartphones operating on Android.

[4]Consider a simple setting with two bidders A and B, where A has a valuation of \$100 and B \$1. In this case, if the auctioneer has no prior knowledge of the distribution of valuations, he cannot set an appropriate reserve price. Without a reserve price, A will win the auction and pay \$1 in a second price auction, which is significantly lower than her valuation.

- App ID: A unique identifier for apps that advertise through the platform.
- Ad ID: This is an identifier for ads that are shown to the users.
- Click indicator: This variable indicates whether or not the user has clicked on the ad.

The total data we see in this one month interval is quite large. Overall, we observe a total of 1594831699 impressions, indicating that on average, the platform runs more than 600 auctions per second. We also see 14373293 clicks in this time-frame, implying a 0.90% CTR.

3.3 Sampling and data preparation

All supervised machine learning algorithms require at least two sets of data – training data and testing data. Training data is the set of pre-classified data used for estimating the model and inferring the associated parameters. We use k-fold cross-validation to pick the best model (the one with the highest predictive power) based on the training data. However, since this model is optimized based on the training data, it should be tested on a completely different data to evaluate its out-of-sample performance. For this we employ a new dataset, referred to as the test dataset.

To assemble these datasets, we need sufficient history at both the population and user-level to generate features or attributes that will function as inputs in our model. Since we only have a snapshot of the data from the firm, this implies that we have to generate both the training and test datasets from the last few days of data, and use the one month of preceding history to generate the features associated with these impressions (Yoganarasimhan, 2016).

We now discuss the sampling procedure used to generate our training and testing datasets. Sampling is a necessary step when working with big data. A good sampling mechanism needs to satisfy two main requirements – 1) it should contain sufficient information both within and across users, *i.e.*, allow us to generate representative global (population-level) features as well as accurate user-specific features, and 2) should be large enough to take advantage of the size of our data without compromising on the scalability of the model.

To satisfy both these requirements, we sample on users (instead of impressions) because we do not want to lose information at the user-level. Recall that one of our objectives is to examine the effectiveness of behavioral targeting in mobile in-app advertising. To do so, we need the unbroken user history. In fact, we find that user-level information is crucial in predicting the likelihood of a click. Combined with the fact that clicks are relatively rare, the more data we have on a given user, the better we can predict his or her behavior. Further, if we have sufficient between-user variation from a given sample of users, then the marginal value of a new user in improving our global or population-level metrics is low. The intuition is that when we have many users, an additional user is very likely to have similar behavior to those already tracked.

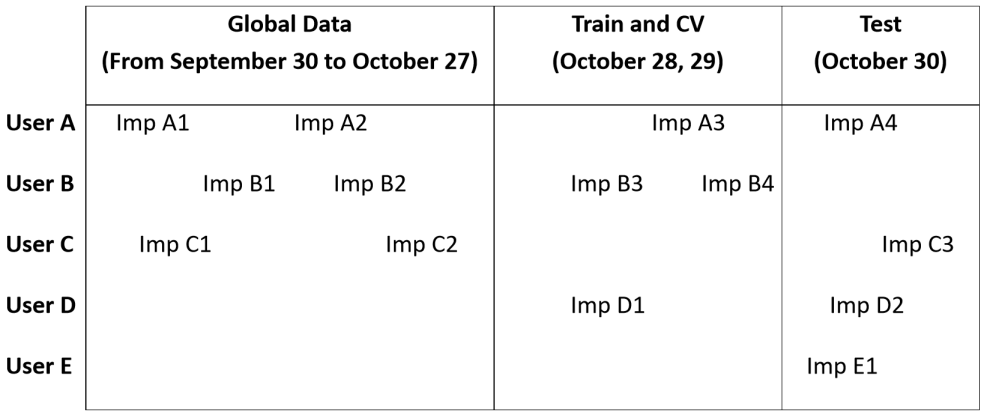


Figure 1: Schema for Data Generation

Accordingly, we draw a sample of 727,354 unique users (out of around 5 million) seen on October 28, 29, and 30 to form our training and test datasets. We then track the impressions containing these users for the last one month to generate the associated features for these data. In §5.3, we formally show that this sample size is sufficient and that increasing the size of the sample further has no real impact on the model performance.

In sum, the process of data preparation starts with sampling users. We draw a sample of users from three days of data kept for prediction purposes, from October 28 to 30. We then track these users over the last one month and make the global data. All the users in global data must be once present in either train and cross-validation data or test data. In total, there are 135,194,585 impressions in global data. We use the sample of October 28 and 29 as the train and cross-validation data, and that of October 30 as the test data. As such, there are 17,733,791 impressions corresponding to the train and cross-validation data, and 9,675,966 impressions corresponding to the test data.

Since the sample is drawn from train and test datasets, there are different situations for users in terms of appearance in different datasets. Some of these situations are illustrated in Figure 1. There are some users available in all three datasets (User A), some only drawn from one of the train or test dataset and available in the global data (User B, User C), and some not available in the Global data (User D, User E).

3.3.1 User identification

User identification and tracking is critical for our purposes. From a methodological perspective, doing so will allow us to predict clicks better, and from a substantive perspective, it is necessary to examine the effectiveness of behavioral targeting. There is no clear "user-identifier" variable in the

data. So we consider two possible identifiers: 1) IP address, and 2) Advertising ID.

While IP address is a well-known tracking metric, it is problematic for two reasons. First, all users behind the same NAT firewall or proxy have the same external IP address. So when we use IP address as the user-identifier, all of them are grouped under the same ID and identified as a single user, which is problematic.[5] Second, IP addresses are generally not static, especially in the case of mobile phones. When a user switches from a WiFi connection to 3G/4G (or vice-versa), the IP address changes. Thus, the same user will show up with different IP addresses in the data.

Alternatively, we can use Advertising ID, which is a user-resettable, unique, anonymous ID associated with a mobile device.[6] It was introduced by Google in 2014 and replaced Andriod ID. The main difference between the two is that Android ID could not be reset by the user, and any data generated by the user could easily be linked to her. Indeed, the move to Advertising ID was a measure to restore privacy controls to the user, allowing privacy conscious users to break the linkage between their activities across time (similar to clearing cookies in the web-surfing context). Thus, if we use this identifier, when a user resets her Advertising ID, she will be interpreted as a new user. Nevertheless, we expect Advertising ID to be a relatively persistent tracking metric since a user needs to be aware of it and actively reset it to change it, whereas IPs change in an ad-hoc fashion every time the user's network connectivity changes.

3.4 Summary statistics

We first look through the time variable. The most crowded time interval in terms of usage is from 6 pm to the end of the day. Namely, around 40% of total impressions are generated after 6 pm. More specifically, the peaks are at 10 and 11 pm, which are the most likely times for people to use their apps. However, not only is the usage at its highest during the night, but the likelihood of click is significantly higher during this period. Figure 2 clearly illustrates the click-through rate is higher in the nights. In fact, the number of clicks generated after noon is three times higher than the number of clicks before noon. The most likely reason explaining such behavior is that people are usually more free at nights and they are more likely to click on an ad and do the further actions.

There are 264 different ads shown in 10789 different apps in our data. However, most impressions come from top apps and ads. Looking at one month of the data, Figure 3a shows the cumulative percentage of total impressions generated by top ads, and Figure 3b shows the same

[5]This problem is exacerbated in Iran, where many websites are censored. To avoid this censorship, many users resort to proxies and/or VPNs, and this leads all these users to show up under the same IP address.

[6]A small number of mobile phone makers are not provisioned to show Google's Advertising ID in their devices and as a result, all the devices made by them are given the same Advertising ID. It is not possible to track individual users for these brands of mobile phones. However, the number of such brands in our data is negligible and we therefore exclude them from our analysis.

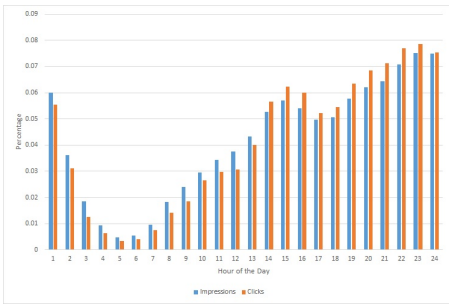
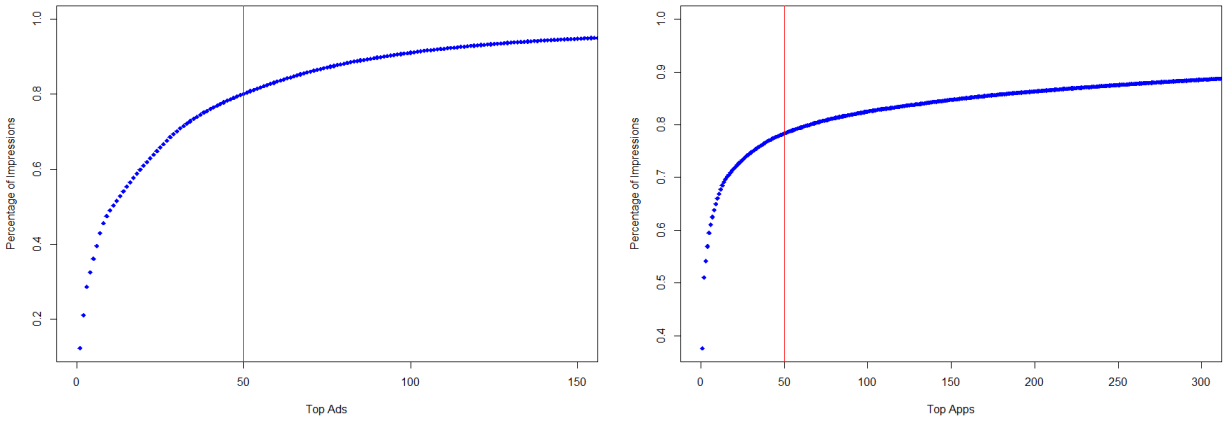


Figure 2: Percentage of Impressions and Clicks by the Time of the Day.

results for top apps. In other words, around 80 percent of all the impressions belong to top 50 ads and top 50 apps.



(a) Ads (b) Apps

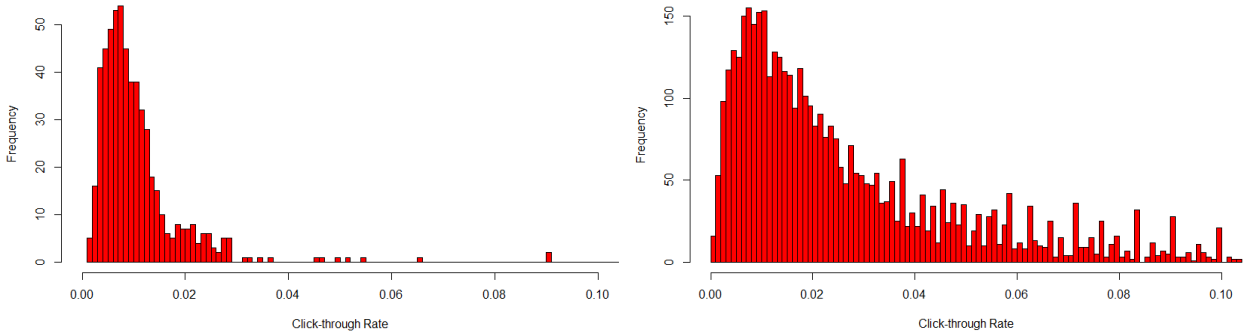
Figure 3: Cumulative Percentage of Impressions Generated by Top Ads and Top Apps

Focusing only on top ads and apps, we present summary statistics for the sample drawn from days October 28, 29, and 30, in Table 1. We report the statistics regarding the interactions between ads, apps, and users. For example, on average, a top ad is shown more than 600,000 times in more than 40 apps containing more than 100,000 users seeing these impressions. Similarly, a top app, serves more than 400,000 impressions of more than 30 ads, on average. The average number of users of an app is around 10,000, of which, 13% have at least clicked once.

There is great heterogeneity among ads and apps in terms of click-through rate. Figure 4a shows the histogram of click-through rate for ads and Figure 4b shows the app specific click-through rate over one month of the data, in October 2015. Comparing two histograms in Figure 4, we find that the variance among apps is higher than the variance among ads. One important factor

Variable	Median	Max	Min
Number of Impressions Showing an Ad	244,045	5,179,823	2,606
Number of Impressions Shown by an App	78,785	10,175,916	43,559
Number of Users Who Have Seen an Ad	64,170	381,262	2,135
Number of Users Who Have Used an App	2,990	143,350	523
Percentage of Users Who Have Clicked in an App	0.12	0.34	0.02
Number of Distinct Apps Showing an Ad	49	50	5
Number of Distinct Ads an App Shows	31	34	29

Table 1: Summary Statistics.



(a) Ads

(b) Apps

Figure 4: Histogram of Click-through Rate for Ads and Apps

driving this considerable gap is the lack of micro-targeting in the platform. In platforms allowing for micro-targeting, each ad is only shown in a few relevant apps, and likewise, apps are showing only a relevant subset of ads. This makes the apps' and ads' click-through rate distributions more identical. However, in our data, an ad is shown in 655 apps on average and an app is showing 44 different ads.

Another important reason explaining why ads are shown in a very broad range of apps is the probabilistic nature of quasi-proportional auctions. Unlike second-price auctions in which the highest score always wins, in quasi-proportional auctions, all bidders have the chance of winning proportional to their score. This, in turn, generates many impressions in which the medium, or even the lowest score ad has been shown. As a result, roughly speaking, each ad is shown in each app, unless the category containing that app is excluded due to the ad's targeting decisions.

Taken together, these two factors produce a high variability in our data, since almost all possible outcomes are observed in some impressions. In this sense, our setting looks similar to a field experiment. Hence, we use the advantage of this level of randomization throughout our paper. This enables us to better capture the app and ad effects.

4 Empirical framework

We now specify the elements of our machine learning framework for targeting. Our problem is one of accurately predicting the probability that an impression i, generated by user U, in app P, for ad A, at time T, global history H, will lead to a click, i.e., $I(C_i) = 1$. Our goal is thus to come up with a classifying algorithm that takes a set of features as input and a set of pre-classified data (training data) as input, and generates as output a probability $p_i(U, P, A, T, H)$ that is as close as possible to the true click probabilities observed in the data.

To formally solve this problem, apart from the data, we thus need to three inputs – 1) Evaluation metric, 2) Feature set, and 3) Classifying algorithm. We now discuss each of these below.

4.1 Evaluation metrics

We now consider different evaluation metrics for our prediction model. Let $\mathbf{p} = (p_1, p_2, \ldots, p_N)$ denote the predicted click probabilities for impressions, and $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ denote the click indicator that we observe for those observations, then the log loss is calculated as follows:

$$LogLoss(\mathbf{p}, \mathbf{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log\left(p_i\right) + \left(1 - y_i\right)\log\left(1 - p_i\right)\right) \tag{2}$$

By definition, $LogLoss$ is the negative log of likelihood for our prediction model. Hence, the higher $LogLoss$ is, the worse our model performs. In order to compare to models in terms of performance, we can divide the $LogLoss$ of one over the other. Obviously, if the fraction is lower than one, it means that the model in numerator performs better, and vice-versa. We call this measure *Normal Entropy*, which is a relative measure of prediction performance. We formalize it as follows:

$$NE(\mathbf{p_A}, \mathbf{p_B}; \mathbf{y}) = \frac{LogLoss(\mathbf{p_A}, \mathbf{y})}{LogLoss(\mathbf{p_B}, \mathbf{y})}, \tag{3}$$

where $\mathbf{p_A}$ and $\mathbf{p_B}$ are respectively the prediction results for models A and B. As such, model B could be seen as the baseline, which in our case could be some simple aggregate measure of CTR. Using $NE(\cdot)$, we can define the *Relative Information Gain* as follows:

$$RIG(\mathbf{p_A}, \mathbf{p_B}; \mathbf{y}) = 1 - NE(\mathbf{p_A}, \mathbf{p_B}; \mathbf{y}) \tag{4}$$

This measure can be interpreted as the percentage improvement we achieve by going from one model to another. This is a typical measure researchers use in applied machine learning models. We also define another evaluation metric, which is the percentage improvement in terms of CTR. For

this purpose, we need to define a targeting rule, by which we can restrict the data points on which we aggregate the CTR.

4.2 Feature Generation

We now discuss feature generation. Features (or attributes/explanatory variables, as we refer to them in the traditional marketing literature) are an important input into all applied machine learning problems. Our goal is to create meaningful features that take advantage of the scale and scope of our data. Features are usually defined with consideration to the main objectives of the model. Since our goal is to accurately predict whether an impression will receive a click or not, our features must capture the factors affecting the probability of click for a given impression.

4.2.1 Feature functions

We follow the functional feature generation framework proposed in Yoganarasimhan (2016). The main advantage of her function-based approach is that it allows us to generate a large and varied set of features using a parsimonious series of functions instead of defining each feature individually.

Recall that each observation in our data is uniquely characterized by four inputs: 1) Time, 2) User, 3) App, and 4) Ad. We therefore utilize the information associated with all of these four inputs to generate features that can inform us of the click probability. As such, let U, P, A, T, and C respectively denote users, apps, ads, hour of the day, and click indicator. We also define the history over which we calculate the functions as H, which is a large set of observations. Using this nomenclature, the feature functions are defined as follows:

1. Impressions (*user*, *app*, *ad*, *time*): This function returns the number of times a given *ad* is **shown** to a specific *user* while using a given *app* in a given instance of *time*. This number is calculated over a pre-specified H. Denoting u, p, a, and t respectively as a given user, app, ad, hour of the day, we define this function as follows:

$$Impressions(u,p,a,t) = \quad _H \mathbb{1}(U=u)\mathbb{1}(P=p)\mathbb{1}(A=a)\mathbb{1}(T=t)$$

   This function can also be invoked with certain input values left unspecified, in which case it is computed by aggregating over all possible values of the unspecified inputs. For example, if we are interested in the number of times that a user has seen a specific ad, we can write:

$$Impressions(u,\_,a,\_) = \quad _H \mathbb{1}(U=u)\mathbb{1}(A=a)$$

   Note that the features generated by this function capture both direct ad exposure effects on user behavior as well as some unobserved ad effects (e.g., an advertiser may not have enough impressions simply because he does not have a large budget).

2. Clicks (*user*, *app*, *ad*, *time*): This function returns the number of times a given *ad* is **clicked** by a specific *user* while using a given *app* in a given instance of *time*. This function is very similar to *Impressions* function, except that it is summed over clicks. As such, we have another indicator added to *Impressions* function, as follows:

$$Clicks(u, p, a, t) = \quad {}_H\, \mathbb{1}(U = u)\mathbb{1}(P = p)\mathbb{1}(A = a)\mathbb{1}(T = t)\mathbb{1}(C = 1),$$

The number of clicks is a good indicator of both ad and app performance. Moreover, at the user-level, the number of clicks captures individual users' propensity to click, as well as her propensity to click within a specific ad and/or app. Thus, this is a very informative metric.

3. CTR (*user*, *app*, *ad*, *time*): This function returns the **click-through rate** of a given *ad* in a given *app* while being used by a specific *user*, in a given instance of *time*. The output is basically calculated dividing the number of clicks by the number of impressions. We can write the equation as follows:

$$CTR(u, p, a, t) = \frac{Clicks(u,p,a,t)}{Impressions(u,p,a,t)}$$

This function is simply a direct combination of the first two. Therefore, it is not necessary to include it in machine learning algorithms like MART which can accommodate non-linear combinations of features without explicit specification by the researcher. However, it is useful to include it explicitly for more traditional methods that cannot perform automatic feature interactions like OLS and logistic regressions.

4. AdCount (*user*, *app*): This function returns the **number of distinct ads** shown to a specific *user* while using a given *app*. Mathematically, we have:

$$AdCount(u, p) = \quad {}_{a\in A}\, \mathbb{1}(Impressions(u, p, a, \_) > 0),$$

where A is the set of all ads. Literature suggests that the variety of ads shown to a user would cause certain types of behavior among consumers (Bauer et al., 1968; Li et al., 2002). We therefore include it in our set of features.

5. AppCount (*user*, *ad*): This function returns the **number of distinct apps** in which a given *ad* is shown to a specific *user*. This functions quite similarly to AdCount. We can define it as follows:

$$AppCount(u, a) = \quad {}_{p\in P}\, \mathbb{1}(Impressions(u, p, a, \_) > 0),$$

where P is the set of all apps. Studies have shown that multichannel usage might lead to different types of consumer behavior (Dijkstra et al., 2005). Hence, we expect a different outcome if a user sees an ad in just one app, rather than seeing in different apps.

6. TimeVariability (*user*): This function measures how different a specific *user* has clicked at different time intervals. Thus, for a given *user*, we use the variance of CTR over time intervals as the measure of time variability. We can write the equation as follows:

$$TimeVariability(u) = \mathrm{Var}_t[CTR(u, \_, \_, t)]$$

The motivation for using this function is to capture different patterns in user behavior and put it as a feature to the model. Since we know that the consumers have different types of behavior at different times, we expect this function to directly explain the variation in clicks.

7. AppVariability (*user*): This function measures how different a specific *user* has clicked in different apps. Thus, for a given *user*, we use the variance of CTR over apps as the measure of app variability. The equation could be written as follows:

$$AppVariability(u) = \mathrm{Var}_p[CTR(u, p, \_, \_)]$$

We know that users respond differently to the same ads in different apps. Therefore, we aim to capture this variation in user behavior into $AppVariability$.

8. Entropy (*user*, *app*): This function measures how diverse a specific *user* has seen the *ads* in a given *app*. For this purpose, we use (Simpson, 1949) measure of diversity. Thus, defining the set of ads shown to a given user while using a given app as A^*, we can write the $Entropy$ function as follows:

$$Entropy(u, p) = \frac{1}{|A^*|} \frac{1}{\sum_{a \in A^*} Impressions(u,p,a,\_)^2}$$

Previous literature has discussed why the diversity of ads matters when we study consumers' response to ads (Li et al., 2002). Moreover, we know that it directly affects short-term and long-term memory of users regarding ads, which would shape their behavior (Sawyer and Ward, 1979; Anderson and Milson, 1989; Sahni, 2015). The entropy metric captures this information.

4.2.2 Feature list and classification

We now use functions defined in the previous section to generate features using different sets of inputs. We present the list of features used in the paper in Table 2, and briefly describe the process of feature generation below.

Table 2: List of Features.

Feature No.	Feature Name	Feature Class
1	Impressions (*user*, _, _, _)	F_B
2	Impressions (_, *app*, _, _)	F_C
3	Impressions (_, _, *ad*, _)	F_C
4	Impressions (_, _, _, *time*)	F_C
5	Impressions (_, *app*, *ad*, _)	F_C
6	Impressions (*user*, *app*, _, _)	F_B, F_C
7	Impressions (*user*, _, *ad*, _)	F_B, F_C
8	Impressions (*user*, *app*, *ad*, _)	F_B, F_C
9	Impressions (*user*, _, _, *time*)	F_B, F_C
10	Clicks (*user*, _, _, _)	F_B
11	Clicks (_, *app*, _, _)	F_C
12	Clicks (_, _, *ad*, _)	F_C
13	Clicks (_, _, _, *time*)	F_C
14	Clicks (_, *app*, *ad*, _)	F_C
15	Clicks (*user*, *app*, _, _)	F_B, F_C
16	Clicks (*user*, _, *ad*, _)	F_B, F_C
17	Clicks (*user*, *app*, *ad*, _)	F_B, F_C
18	Clicks (*user*, _, _, *time*)	F_B, F_C
19	CTR (*user*, _, _, _)	F_B
20	CTR (_, *app*, _, _)	F_C
21	CTR (_, _, *ad*, _)	F_C
22	CTR (_, _, _, *time*)	F_C
23	CTR (_, *app*, *ad*, _)	F_C
24	CTR (*user*, *app*, _, _)	F_B, F_C
25	CTR (*user*, _, *ad*, _)	F_B, F_C
26	CTR (*user*, *app*, *ad*, _)	F_B, F_C
27	CTR (*user*, _, _, *time*)	F_B, F_C
28	AdCount (*user*, _)	F_B
29	AdCount (_, *app*)	F_C
30	AdCount (*user*, *app*)	F_B, F_C
31	AppCount (*user*, _)	F_B, F_C
32	AppCount (_, *ad*)	F_C

Continued on next page

Table 2 – continued from previous page

Feature No.	Feature Name	Feature Class
33	AppCount (*user*, *ad*)	F_B, F_C
34	TimeVariability (*user*)	F_B
35	AppVariability (*user*)	F_B
36	Entropy (*user*, _)	F_B
37	Entropy (_, *app*)	F_C
38	Entropy (*user*, *app*)	F_B, F_C

Each feature is characterized by a function and a set of inputs. For instance, $Impressions(\_, \_, \_, \_)$ is a feature characterized by the $Impressions$ function and can take four possible inputs corresponding to user, ad, app, and hour of the day, which we can either specify or aggregate over. Thus, this function can be used to generate $2^4 = 16$ potential features. However, we do not expect all these features to help improve the predictive power of our model. Hence, we only include those features which clearly help model performance.[7]

In the case of the $Impressions$ function, we start by first generating features with one-element sets for each of users, apps, ads, and time; see Features 1–4 in Table 2. These one-element sets solely capture the effects of the users, apps, ads, and time, respectively, and ignore any potential interactions between them. To take the interactions into the account, we next generate Features 5–9. We consider all the subsets of inputs except the ones capturing either ad-time or app-time interactions because we do not find any evidence that suggests that these interactions matter. We expect all the time effects to be captured in Feature 9. Next, we generate features for $Clicks$ (Features 10–18) and CTR (Features 19–27) using the same sets of inputs. Note that CTR is perfectly determined with $Impressions$ and $Clicks$, and hence is superfluous in machine learning algorithms such as MART. Using other functions defined in the previous section, we generate the next set of features (28–38) shown in Table 2.

Classification of features: To aid our analysis, we categorize features into two (partially overlapping) targeting categories – *Behavioral* (F_B) and *Contextual* (F_C) features. This categorization is shown in the third column of Table 2. Behavioral features are ones that are based on the browsing and click history of the individual user. Contextual features are ones that inform us of the context in which the impression happens (such as the app, the ad, and the time of day of the impression) and

[7]While we winnow down the set of features by experimentation, it is possible to start with a specification that includes all possible features and then employ a *feature selection* wrapper to formally extract the list of most relevant features. This is a common strategy in applied machine learning research. We refer interested readers to Yoganarasimhan (2016) for a detailed discussion of feature selection.

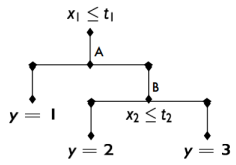


Figure 5: Example of a CART model.

the influence of the context on the probability of clicking. These need not be user-specific, though they might be (e.g., Feature 6 is both contextual and behavioral). In §5.2, we examine the relative value of these two types of features in their ability to predict clicks and improve targeting outcomes.

4.3 MART – Multiple additive regression trees

Finally, we discuss the base machine learning classification algorithm that we use, MART. The following section follows Yoganarasimhan (2016). Note that this algorithm takes as input the training (and validation) data and the set of features discussed above to generate a prediction of click probabilities.

Broadly speaking, MART is a machine learning algorithm that models a dependent or output variable as a linear combination of a set of shallow regression trees (a process known as boosting). In this section, we introduce the concepts of Classification and Regression Trees (CART) and boosted CART (referred to as MART). We present the high level overview of these models here and refer interested readers to Murphy (2012) for details.

4.3.1 Classification and regression trees

CART methods are a popular class of prediction algorithms that recursively partition the input space corresponding to a set of explanatory variables into multiple regions and assign an output value for each region. This kind of partitioning can be represented by a tree structure, where each leaf of the tree represents an output region. Consider a dataset with two input variables $\{x_1, x_2\}$, which are used to predict or model an output variable y using a CART. An example tree with three leaves (or output regions) is shown in Figure 5. This tree first asks if x_1 is less than or equal to a threshold t_1. If yes, it assigns the value of 1 to the output y. If not ($i.e.$, if $x_1 > t_1$), it then asks if x_2 is less than or equal to a threshold t_2. If yes, then it assigns $y = 2$ to this region. If not, it assigns the value $y = 3$ to this region. The chosen y value for a region corresponds to the mean value of y in that region in the case of a continuous output and the dominant y in case of discrete outputs.

Trees are trained or grown using a pre-defined number of leaves and by specifying a cost function that is minimized at each step of the tree using a greedy algorithm. The greedy algorithm implies that at each split, the previous splits are taken as given, and the cost function is minimized

going forward. For instance, at node B in Figure 5, the algorithm does not revisit the split at node A. It however considers all possible splits on all the variables at each node. Thus, the split points at each node can be arbitrary, the tree can be highly unbalanced, and variables can potentially repeat at latter child nodes. All of this flexibility in tree construction can be used to capture a complex set of flexible interactions, which are not predefined but are learned using the data.

CART is popular in the machine learning literature because it is scalable, is easy to interpret, can handle a mixture of discrete and continuous inputs, is insensitive to monotone transformations, and performs automatic variable selection (Murphy, 2012). However, it has accuracy limitations because of its discontinuous nature and because it is trained using greedy algorithms. These drawbacks can be addressed (while preserving all the advantages) through boosting, which gives us MART.

4.3.2 Boosting

Boosting is a technique that can be applied to any classification or prediction algorithm to improve its accuracy (Schapire, 1990). Applying the additive boosting technique to CART produces MART, which has now been shown empirically to be the best classifier available (Caruana and Niculescu-Mizil, 2006; Friedman et al., 2001). MART can be viewed as performing gradient descent in the function space using shallow regression trees (with a small number of leaves). MART works well because it combines the positive aspects of CART with those of boosting. CART, especially shallow regression trees, tend to have high bias, but have low variance. Boosting CART models addresses the bias problem while retaining the low variance. Thus, MART produces high quality classifiers.

MART can be interpreted as a weighted linear combination of a series of regression trees, each trained sequentially to improve the final output using a greedy algorithm. MART's output $L(x)$ can be written as:

$$L_N(x) = \sum_{n=1}^{N} \alpha_n l_n(x, \beta_n) \tag{5}$$

where $l_n(x, \beta_n)$ is the function modeled by the n^{th} regression tree and α_n is the weight associated with the n^{th} tree. Both $l(\cdot)$s and αs are learned during the training or estimation. MARTs are also trained using greedy algorithms. $l_n(x, \beta_n)$ is chosen so as to minimize a pre-specified cost function, which is usually the least-squared error in the case of regressions and an entropy or logit loss function in the case of classification or discrete choice models.

5 Results

5.1 Implementation: training, cross-validation, and tuning of model parameters

For each observation, we have a set of features relating to the four key variables, user, app, ad, and time. Using the features as inputs, we train and validate the model on the train data (which is sampled from October 28^{th} and 29^{th}), and test the model performances on test data (sampled from October 30^{th}). We used the package XGBoost for this purpose.

Since that our training data is used for both training and validation. The validation method we employ is k-fold cross validation, where $k = 2$. Cross validation avoids a common problem in machine learning models – that of over-fitting. Cross-validation partitions the training data into k researcher-specified parts/folds, trains k different models independently on the training data while holding back one of the k^{th} folds, and for each of these training sessions, it picks the model that performs the best on the held-out data (rather than the training data). All the models fitted in the k sessions are then averaged and this is treated as the final model, whose performance is tested on a completely independent data, referred to as the test data.

There are a few key parameters that we need to specify to appropriately tune the MART model. These include the maximum number of trees, the maximum number of nodes per tree, the stopping rule, number of folds (k) using cross validation, and the learning rate. After experimenting with a number of different parameters, we used the following tuning parameters:

- Maximum trees $= 3000$
- Maximum depth of a tree $= 6$
- Stopping rule = Stop adding trees when no improvement in prediction after three rounds
- Number of folds used in cross validation $k = 2$
- Learning rate $= 0.1$

In our training, we find that our optimization algorithm stops at 320 trees (i.e., after adding the 320^{th} tree, the model finds no additional improvement from adding new trees, even though it continues to add three more trees based on our stopping rule). While this combination of tuning parameters gave the best results in terms of model fit and speed of training, we found the model performance to be quite robust to tuning parameters. For instance, reducing the number of trees drastically to seven reduced run time significantly and the model performance was lower by just 1%. Similarly, while lower learning rates are preferred, increasing the learning rate to 1 did not worsen the model performance.

Method	User	Ad	App	Ad-App	User-Time	User-Ad-App	All
MART	0.093	0.007	0.044	0.051	0.112	0.132	0.152
Logistic Regression	0.068	0.008	0.042	0.044	0.079	0.094	0.100
OLS	0.066	0.009	0.044	0.046	0.078	0.091	0.095
Ad-App CTR	0.049	0.049	0.049	0.049	0.049	0.049	0.049

Table 3: *RIG* for different model specifications and feature sets

5.2 RIG improvement

We present our main results on the prediction accuracy of our approach with different model specifications in Table 3. Our main evaluation metric is *Relative Information Gain* (RIG), which we calculate over a baseline model. We use the average probability of click for each observation, i.e., the average CTR for the platform as our baseline prediction. This would be the naive prediction for click probability with absolutely no targeting in place.

The columns in Table 3 refer the set of features used to train the model, as shown below:

- User: 1, 10, 19, 28, 31, 34, 35, 36
- Ad: 3, 12, 21, 32
- App: 2, 11, 20
- Ad-App: 2, 3, 11, 12, 20, 21, 32
- User-Time: 1, 4, 9, 10, 13, 18, 19, 22, 28, 31, 34, 35, 36
- User-Ad-App: 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 14, 15, 16, 17, 19, 20, 21, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38
- All: 1 to 36

Similarly, the rows in Table 3 refer to the optimizer used. We consider four different options:

- MART – the machine-learning algorithm that we described in § 4.3.
- Logistic regression: a binary logit model where we use all the features from the feature set shown in the corresponding column, as well as all possible two-way interactions between those features.
- OLS: a ordinary least squares regression, where we employ all the features shown in the corresponding column, and two-way interactions.
- Ad-App CTR: The average ad-app specific CTR in the training data. Note that this is a constant across all columns because it does not use any features as input.

In the rest of this section, we discuss our key findings.

MART vs. other methods: First, our results indicate that MART performs better than the baseline platform-specific CTR, as well a more targeted Ad-App-specific CTR when we include

all our features (see the last column of Table 3). Second, when we use sufficiently informative features, MART easily outperforms the two most commonly used methods in marketing – Logistic Regression and OLS models. When we use all the available features, the *Relative Information Gain* of MART over Logistic and OLS regressions is more than 5%. Although previous literature has documented that MART outperforms Logistic regression in other settings (He et al., 2014; Yoganarasimhan, 2016), these earlier papers do not examine which types of information/features lead to this improvement. We examine this issue in greater detail and offer some additional insights. Specifically, we find that when we consider only global features, the RIG by MART over Logistic Regression is less than 1%. However, this improvement is around 3%, when we also include real-time or user-specific features. This implies that the user behavior is harder to capture using methods relying on traditional functional reduced-form based approaches. Thus for micro-targeting in real-time, marketers need a rich set of behavioral features as well as a highly optimized machine learning algorithm such as MART.

Behavioral vs. contextual targeting: Next, we examine how much different types of features – starting with contextual (global ad and/or app specific), to behavioral (user-specific), to those that include both types of information (behavioral and contextual) influence the predictive accuracy of our model. Understanding to what extent each variable improves the prediction is critical from a managerial perspective, since storing data could be costly for such platforms. For this purpose, we evaluate the prediction while using different set of features. For example, if we have access only to user information without additional contextual information, we can generate features 1, 10, and 19, while if we have access to ad and app variables, we can generate features 2, 3, 5, 11, 12, 14, 20, 21, 23, 29, 32, and 37.

When we only have access to user-specific information (or pure behavioral targeting with no contextual information), *Relative Information Gain* over the baseline model is 9.3% (column 2 in Table 3). This gain is considerably higher than the case in which we have information regarding both apps and ads but not users (purely contextual features with no behavioral information), which is 5.1% (column 5 in Table 3). Together, these findings suggest that behavioral targeting is more valuable in mobile advertising compared to contextual targeting. While it is possible that additional contextual information (that we do not have) can change the relative ordering of these findings, our findings establish a base case for these comparative results.

App-specific vs. ad-specific features: Third, within contextual information, we examine the relative value of app-specific features to ad-specific features. This is an important issue in in-app mobile ads, and has implications for both advertisers and publishers. Our results indicate that we app-specific features are better at predicting clicks compared to ad-based features (comparing

columns 3 and 4 in Table 3). This finding likely stems from two reasons. First, it could be due to the fact that in-app ads are quite small and cannot convey much information or have significant persuasive quality. Moreover, because all ads are shown to all users because of the randomization from the proportional auction, there is no additional information on user-segments in the ad-specific features. In contrast, in the case of apps, the aggregated contextual features are likely to capture different user interface (UI) and user experience (UX) factors that drive users to click more/less on average in a particular app. Further, the aggregated features for apps contain information regarding their users, since the user segments can vary across apps. Simply put, in our setting users self-select in to apps, but cannot self-select into ads. (This is why the variation in ad-CTR is much lower than the variation in app-CTR.) However, we do find that once we add user-level information to ad-features, they become much more informative and improve the model performance significantly.

Overall model performance: Finally, all elements of our framework combined produce a good predictive power. The RIG of our model is 15.2%, which is considered quite significant in click prediction models. Note that, if we assume a true click probability for each impression between 0 to 0.5, and then generate click outcomes using a Bernoulli distribution with these click probabilities, the RIG of the true probabilities (the perfect estimation) would be no more than 15%.

It is worth noting that although behavioral-targeting variables are more valuable than contextual app-ad variables, we find that we need to use all the features (and their interactions through a non-linear classifier) to obtain the best prediction model. In fact, the substantial improvement in our model is due to using all the interactions between four key variables in our data. Thus, our combined modeling framework that utilizes our complete feature set and a MART classifier is necessary to help the platform and advertisers improve their targeting.

5.3 Sampling and data adequacy

We conducted our analyses using a relatively large sample of users (We sampled 727,354 out of more than 6 million users in train and test data). However, sampling always results in information loss to some extent. In this section, we examine whether or not our sample is adequate, and further, to identify the sample size that minimizes information loss. Thus, we calculate the RIG for different sample sizes. That is, we quantify how much our model gains by using more data points.

We start with 1,000 users and add more data in each step. However, since there is heterogeneity among users, we may randomly find a smaller sample with higher RIG than a larger sample. To minimize the noise in our results, we employ a bootstrap procedure, by which we repeat the sampling for each sample size 10 times. We then calculate the mean and standard deviation of RIG for each sample size.

Table 4 shows the results for RIG over two different baseline models for different sample

Sample Size	RIG over Baseline CTR		RIG over Ad-App CTR	
	Coefficient	Std. error	Coefficient	Std. error
1000	0.101	0.027	0.06	0.032
5000	0.117	0.009	0.071	0.009
10000	0.142	0.006	0.095	0.009
50000	0.142	0.004	0.098	0.005
100000	0.149	0.002	0.104	0.002
200000	0.15	0.002	0.106	0.003
300000	0.151	0.002	0.106	0.002
400000	0.15	0.002	0.106	0.002
500000	0.151	0.001	0.107	0.001
600000	0.151	0.000	0.107	0.000
700000	0.151	0.000	0.108	0.000

Table 4: *RIG* of MART for different sample sizes

sizes. Our first baseline predicts average CTR as the click probability for all the test observation. The second one takes ad and app as inputs and predicts the CTR in this app-ad pair as the click probability. Our results indicate that for sample sizes smaller than 10,000, we have a substantial information loss. However, for anything above 10,000, increasing the sample size slightly improves the prediction, and after 200,000, increasing the sample size does not help improve the prediction results. In other words, the RIG is almost the same as we increase the sample size from 200,000. Thus, we can argue that a sample of 200,000 users out of 4 million, which is approximately 5% of our users, would be sufficient for our purpose.

6 Implications for privacy regulations and data sharing

We now use our modeling framework to examine the implications of changing regulations that protect consumer privacy in this market. We examine two different types of issues – 1) How will changes in data protection and privacy influence advertisers' ability to target consumers? 2) Are the incentives of the different players in this market (platform, advertisers) aligned? Would they all prefer more lax privacy protections and be willing to share data with each other? Specifically, we examine:

- How will strengthening some of the consumer privacy protection affect platform's and advertisers' ability to target consumers?
- If we relaxed the data-sharing restrictions between the platform and the advertisers further, would advertisers be able to improve their targeting? If yes, to what extent and which advertisers benefit the most?
- Does the platform have an incentive to share data with advertisers? Some of the earlier analytical

papers argue that too much targeting can lead to thin markets and soften competition among advertisers, which in turn can hurt platform's profits. Does the data support or refute this hypothesis?

- If we further relaxed the data-sharing restrictions to allow advertisers to share data among each other, how will their targeting change? Would some types of advertisers benefit more than others? Can these data-sharing arrangements be incentive-compatible?

Note that incentives are particularly important in this context because if firms are naturally incentivized to not share data with each other, then there is a market mechanism that is likely to lead to an equilibrium with higher consumer privacy protection. In contrast, if the incentives of the players in terms of data-sharing are more closely aligned, then an external player (such as the government or consumer advocacy groups) may have to impose better privacy regulations that balance consumers' need for privacy with firms' profitability motives.

6.1 Value of user identifiers: IP vs. Advertising ID

User-identification and tracking is at the core of our methodological and substantive contributions. We now examine whether our decision to use Advertising ID (instead of IP address) as our main user-identifier is justified. From a policy perspective, this analysis focuses on key trade-offs and issues at the intersection of consumer privacy and marketing practice. One important question that often comes up in the context of mobile tracking is – if lawmakers were to strengthen consumer privacy laws and prevent the use of a tracking identifier such as Advertising ID, which would force advertisers and the platforms to rely on IP as their mobile-tracking metric, would their ability to target suffer? If so how much?

To answer this question, we re-did all our analysis with IP as the user-identifier instead of Advertising ID. These new results are presented in Table 5. There are major differences compared to Table 3 that are worth noting. First, the overall performance of model drops substantially. This indicates that there is considerable information loss when we move from Advertising ID to IP. Second, we find that although the user information continues to be valuable in this case, the improvement from app-ad features is higher now. This again indicates that there is significant information loss at user-level features with IP as the identifier. As discussed earlier, when we use IPs as the user-identifier, we may treat two different users as one, and we may also observe many different IPs for only one user. As a result, the model using only user-level features is subject to a huge information loss.

We do find that models that interact user-level features with contextual information perform reasonably well (though still worse than Advertising-ID based models). For example, a model that uses user-time interaction performs better than a model that uses only user features. Moreover,

Optimization model	User	Ad	App	Ad-App	User-Time	User-Ad-App	All
MART	0.023	0.009	0.043	0.052	0.047	0.087	0.103
Logistic Regression	0.013	0.009	0.043	0.047	0.032	0.068	0.067
OLS	0.013	0.009	0.044	0.047	0.032	0.065	0.067
Ad-App CTR	0.050	0.050	0.050	0.050	0.050	0.050	0.050

Table 5: *RIG* for different model specifications using IP as the user-identifier

the user-app-ad model performs considerably better than the app-ad model. In fact, when we use the features containing the interactions between user and another variable, the likelihood that our features uniquely relate to users largely increases.

Finally, to ensure that the differences in the results between Tables 3 and 5 are not driven by the fact that they are based on two different samples, we conducted some additional checks using the same dataset. Recall that we had sampled one set of 750,000 unique Advertising IDs and another set of 750,000 unique IPs. The results of Table 3 are based on the former and the results in Table 5 are based on the latter. However, around 20% of these identifiers are mutual, and for these we have the information for on Advertising ID and IP. We now examine the performance of our models on this overlapping dataset and present the results in Table 6. We find that the Advertising ID has a RIG of 5.6% over the IP model even within the same dataset. This reaffirms the importance of using Advertising ID for user-identification in mobile advertising and suggests that IP cannot function as a reasonable substitute.

Optimization model	Advertising ID	IP
MART	0.143	0.092
Logistic Regression	0.092	0.066
OLS	0.092	0.062

Table 6: *RIG* over baseline CTR for different models for the two user identifiers

6.2 Data-sharing arrangements between the platform and advertisers

Thus far, we have evaluated the performance of different prediction models, from a platform's perspective. Our results in previous sections examined how the platform would benefit from different levels of access to the data. In this section, we focus on the advertisers as the main decision makers and investigate how they could utilize the data in different data-sharing scenarios. For example, if the platform provides advertisers with their own impression-level data, how it would affect their targeting decisions.

We first start with describing limitations in data-sharing arrangements. Our focus is on two main types of privacy violation in data-sharing. First, platforms are not usually allowed to share user-level

data with the advertisers. In some cases, however, they can share the aggregated information about users without revealing their identity. Second, regarding advertisers' privacy, sharing the data containing the information about other ads is not allowed. As such, one advertiser can only access their own data.

In what follows, we describe different data-sharing scenarios, each of which reflects one sort of privacy regulation:

1. **Scenario 1:** Advertisers only access their average CTR. No information about the users, apps, and time is provided.

2. **Scenario 2:** Advertisers access their CTR in different apps. This arrangement does not violate privacy regulations, because all the information is provided at the aggregated level.

3. **Scenario 3:** Advertisers access their own impression-level data. The information regarding users is provided. Hence, they access the same variables as what the platform does, but they have no access to other ads' data. Hence, they cannot generate the features that are aggregated over ads. Thus, the arrangement could only be considered as a violation of user privacy.

4. **Scenario 4:** Advertisers access their own impression-level data and the features derived by aggregating over ads. In this scenario, they have the full set of features. Since the features aggregated over ads do not reveal any specific information about ads, this scenario could only be considered as a violation of user information.

5. **Scenario 5:** Advertisers access the full dataset. Hence, they have impression-level data, containing all the information regarding users, apps, ads, and time. This arrangement could be seen as a violation of both users' and ads' privacy.

Considering all the scenarios, we first examine how well advertisers can predict the click probability for their impressions. For each scenario, we use the same test dataset, and evaluate the performance of their prediction model using the metrics we used in previous sections. For scenario 1 and scenario 2, the prediction model is clear. In scenario 1, advertisers simply predict the CTR as the probability of click for all impressions in the test data, since they are not provided with any more information. In scenario 2, they can condition their prediction on the app showing their ad, and predict their CTR in that specific app as the click probability. In scenario 3, however, the advertisers are provided with an impression-level train data. Thus, they can build a learning model, similar to those we used in previous sections. However, they do not access the full data, and as a result, they can only generate features for which the ad is given (features 3, 5, 7, 8, 12, 14, 16, 17, 21, 23, 25,

	Scenario 2	Scenario 3	Scenario 4	Scenario 5
High	0.037	0.098	0.152	0.155
Medium	0.046	0.071	0.098	0.104
Low	0.038	0.063	0.105	0.119

Table 7: RIG for the three tiers of advertisers in different scenarios compared to Scenario 1.

26). In scenario 4, they have their own impression-level data, with the full set of features as it is allowed by the platform. Lastly, in scenario 5, they can use the same MART model as we used, and test the prediction results on their own test data.

Setting the baseline as the model in scenario 1, we calculate the RIG for the models in other scenarios. Our results are presented in Table 7. Primarily, our results indicate that the advertisers have the highest information gain in scenario 5, which is obtained by using the information across other ads. As we expected, RIG in scenario 4 is greater than scenario 3, because advertisers in scenario 4 access to more features. Similarly, the model in scenario 3 performs better than the model in scenario 2, meaning that allowing advertisers to have their own user-level data would result in a better prediction model.

Additionally, we compare the results across different tiers of ads - high, medium, and low - defined based on the number of impressions showing their ad. Our results indicate that larger ads benefit more in scenario 3. One reason is that the larger ads have more impressions. Hence, they can train their model better. However, given our results in section 7.2, we know that sample size does not substantially improve the prediction results if it is sufficiently large. The second reason accounting for the better performance of larger ads is that they capture higher variation in features since they have more audience.

Although the prediction performance is generally a good measure to see how advertisers could utilize the data given to them, it contains no information regarding advertisers' decision variables for targeting. In other words, we cannot necessarily say that since the advertisers have a better click prediction for their observations, they can extract more revenue by targeting. To address this issue, we need to clearly define targeting decisions allowed by the platform.

First, suppose that the platform allows advertisers to target at the impression-level. In other words, advertisers can exactly specify the impressions in which they want their ad be shown and exclude the rest. Mathematically speaking, it is equivalent to condition impressions on the click probability that the model estimates. For example, advertisers can target the impressions for which their model predicts that the click probability is higher than 1%. We calculate the click-through rate for advertisers if they could target upper-median estimated click probabilities. In Table 8, we measure the CTR percentage improvement using such targeting.

	Scenario 2	Scenario 3	Scenario 4	Scenario 5
High	31.6	67.6	75.6	76.5
Medium	32.1	62.8	72.6	74.8
Low	30.0	59.6	72.8	76.8

Table 8: Percentage improvement in CTR for the three tiers of advertisers compared to Scenario 1

The results in Table 8 echo the findings of Table 7. Larger advertisers benefit more when they access their own data in scenario 3. In the two first scenarios, we do not expect to see any significant difference between advertisers of different sizes, as their size does not matter in their prediction. In scenarios 4 and 5, however, there must be some kind of relationship between the size of advertisers and their benefit. We conduct further analyses to get more information on the effect of size and other features of one ad on its performance under different scenarios.

In order to answer the questions that which advertisers benefit the most and why, we conduct a series of regression analysis. Having their improvement metrics (RIG or CTR percentage improvement) in different situations as the dependent variable and the ad specific features as independent variables, we run some linear regression models. The results are presented in Table 9.

Using the RIG of scenario 2 over 1, we find that the coefficient for CTR is significant and negative, meaning that advertisers with higher CTR do relatively worse when they have their CTR in different apps, compared to those with lower CTR. Our results generally show that larger advertisers benefit the most when we move from scenario 1 to 3, because the coefficient for size of the ad is significant and positive. Given the results for RIG of scenarios 4 and 5 over 1, we find that not only does size of the ad matter, but also its CTR significantly affects the performance. In both scenarios 4 and 5, ads of higher CTR do better in terms of click prediction. In scenario 4, the reason is that they have higher clicks to train their data with the full set of features. In scenario 5, however, one likely reason is that if they have higher CTR, their part of data reveals more information regarding clicking behavior, and in turn, the full model has a better prediction performance.

If we set the scenario 3 as the baseline and move from this scenario to scenarios 4 and 5, we observe that advertisers with higher CTR benefit more than those with lower CTR. Interestingly, if we move from scenario 4 to scenario 5, we observe that smaller advertisers benefit the most, since the coefficient for size is significant and negative.

6.3 Data-Sharing arrangements between advertisers

Aside from different privacy regulations that we investigated in previous section, we now consider different data-sharing situations under which advertisers can easily share their data with each other. While they had their own data in scenario 3, they are now able to access one of the other ads and

Dependent Variables	Size		Click-through Rate		Targeting Dummy	
	Coefficient	Std. error	Coefficient	Std. error	Coefficient	Std. error
RIG of Moving from Scenario 2 to 1	0.026	0.125	−2.627**	1.206	0.002	0.009
RIG of Moving from Scenario 3 to 1	0.405***	0.091	0.198	0.876	0.004	0.007
RIG of Moving from Scenario 4 to 1	0.509***	0.121	2.899**	1.169	0.007	0.009
RIG of Moving from Scenario 5 to 1	0.422***	0.130	2.294*	1.252	0.011	0.010
RIG of Moving from Scenario 4 to 3	0.141	0.123	2.914**	1.189	0.004	0.009
RIG of Moving from Scenario 5 to 3	0.047	0.116	2.251*	1.113	0.008	0.008
RIG of Moving from Scenario 5 to 4	−0.097*	0.051	−0.666	0.494	0.005	0.004
$N = 37$, ${R^2}_{2to1} = 0.06$, ${R^2}_{3to1} = 0.33$, ${R^2}_{4to1} = 0.40$, ${R^2}_{5to1} = 0.26$, ${R^2}_{4to3} = 0.13$, ${R^2}_{5to3} = 0.04$, ${R^2}_{5to4} = 0.16$						
Signif. codes: 0.01 '***' 0.05 '**' 0.1 '*'						

Table 9: Linear regression estimates with the RIG as the DV when we move across different scenarios

train a better data. The percentage by which their model would improve is of our interest in this section, and we further investigate it to find out under what conditions sharing is more profitable. Namely, what drives higher RIG when one shares its data with another.

For this purpose, we calculate the RIG for all different data-sharing combinations. We consider top 37 advertisers in our data. Therefore, there are 37×36 data-sharing combinations. Having the RIG of advertiser i when it shares with advertiser j over when it does not (i.e., having its own data), we regress it on general characteristics of sharing, such as the data size or CTR of both i and j. We define $Size_i$ as the percentage of impressions showing ad i, and CTR_i as the click-through rate of ad i. The first column of Table 10 shows the linear regression estimates with the RIG of sharing over not-sharing on the size and the CTR of both parties. As echoed in the previous section, the higher the size of the sharer, the more information gain an advertisers obtains by sharing. Interestingly, we find that the CTR of an advertiser makes the sharing more profitable for itself.

The decision on what advertiser to share with is of great importance for advertisers. They must know what factors are driving the information gain for sharing. We define four more variables. The first variable is the percentage of users that both advertisers have in common. Namely, this is the number of users both i and j have, divided by the number of users i has. We also define the percentage of new users, which is the number of users j has but i does not have, divided by the number of users i has. Similarly, we define the third and fourth variables, the percentage of apps in common, which is the number of apps showing both i and j divided by the number of apps showing i, and the percentage of new apps, which are the number of apps showing j but not i, divided by the number of apps showing i. We add these four variables as independent variables and estimate the results.

The second column of Table 10 shows the results of this regression. We show that $Size_j$ is

Independent Variables	RIG of Sharing over not Sharing		RIG of Sharing over not Sharing	
	Coefficient	Std. error	Coefficient	Std. error
CTR_i	$1.927 \times 10^{-}1$***	$5.334 \times 10^{-}2$	$5.972 \times 10^{-}2$	$4.373 \times 10^{-}2$
CTR_j	$-2.516 \times 10^{-}3$	$5.334 \times 10^{-}2$	$7.317 \times 10^{-}3$	$4.412 \times 10^{-}2$
$Size_i$	$4.178 \times 10^{-}2$***	$5.503 \times 10^{-}3$	$-2.559 \times 10^{-}2$***	$4.434 \times 10^{-}3$
$Size_j$	$2.131 \times 10^{-}2$***	$5.503 \times 10^{-}3$	$-2.163 \times 10^{-}3$	$5.839 \times 10^{-}3$
Percentage of Mutual Users			$-8.529 \times 10^{-}4$	$9.364 \times 10^{-}4$
Percentage of New Users			$2.729 \times 10^{-}4$***	$9.843 \times 10^{-}6$
Percentage of Mutual Apps			$9.896 \times 10^{-}4$*	$5.388 \times 10^{-}4$
Percentage of New Apps			$-2.853 \times 10^{-}4$***	$1.065 \times 10^{-}4$
	$N = 1332$	$R^2 = 0.0546$	$N = 1332$	$R^2 = 0.4094$
Signif. codes: 0.01 '***' 0.05 '**' 0.1 '*'				

Table 10: *Relative Information Gain* of Sharing over not Sharing for Different Specifications

not significant anymore when we control for these four variables. Our results indicate that the advertisers with more new users are more profitable for sharing than the advertisers with mutual users. In contrast, the advertisers with more mutual apps showing them are more profitable than advertisers with newer apps. To summarize, the RIG of sharing is at its highest when the percentage of new users achieved by sharing is the maximum. This sharing obviously benefits both parties, however, the party that has less data benefits more from this sharing contract.

The question of sharing, however, requires more careful consideration, since the extent to which advertisers are allowed to share their data is restricted due to the privacy regulations. For example, advertisers cannot usually match their users since the users are anonymously coded for each advertiser. In this case, they cannot even know the number of users they have in common or the number of new users, unless the ad-network shares such information with them. Our results hold for the cases where advertisers cannot match their users. If they are allowed to match their users, they can then aggregate some features over users and generate more features. Hence, we leave this question for future research.

7 Conclusions

Mobile in-app advertising is now a growing industry. We examine the value of information in improving targeting outcomes in this context. We study a large scale data set (of over 150 million data points across one month) from a leading in-app ad-network in Iran. We first examine which targeting factors improve the targeting outcomes. We build a Machine Learning framework with over 150 features and employ a MART algorithm to train the model. We find that our model improves prediction significantly over the baseline, and performs much better than logistic regressions and OLS models. We find that behavioral targeting based on user-level features is more valuable than

contextual targeting based on ad-app features. We then user our model to examine how different data-sharing arrangements between the ad network and advertisers will affect an advertisers' ability to do targeted bidding. We show that the least privacy-preserving arrangements are also the most valuable for advertisers. Interestingly, we also find that large advertisers benefit the most from data-sharing arrangements, which raises concerns on data-sharing cabals. Finally, we also examine whether the ad-network is incentivized to share targeting data with advertisers and show that the ad-network may actually prefer to withhold information from advertisers to improve their own revenue since targeted bidding by advertisers softens competition. Thus, by design, the ad-network may be incentivized to preserve users' privacy.

References

A. Acquisti, C. R. Taylor, and L. Wagman. The economics of privacy. *Available at SSRN 2580411*, 2016.

J. R. Anderson and R. Milson. Human memory: An adaptive perspective. *Psychological Review*, 96(4):703, 1989.

M. Andrews, X. Luo, Z. Fang, and A. Ghose. Mobile ad effectiveness: Hyper-contextual targeting with crowdedness. *Marketing Science*, 35(2):218–233, 2015.

A. Ansari and C. F. Mela. E-customization. *Journal of marketing research*, 40(2):131–145, 2003.

Y. Bart, A. T. Stephen, and M. Sarvary. Which products are best suited to mobile advertising? a field study of mobile display advertising effects on consumer attitudes and intentions. *Journal of Marketing Research*, 51(3):270–285, 2014.

R. A. Bauer, S. A. Greyser, et al. Advertising in america, the consumer view. 1968.

R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

D. Chaffey. Digesting Mary Meeker's 2015 Internet Trends Analysis, 2015. URL `http://www.smartinsights.com/internet-marketing-statistics/insights-from-kpcb-us-and-global-internet-trends-2015-report/`.

P. Chatterjee, D. L. Hoffman, and T. P. Novak. Modeling the clickstream: Implications for web-based advertising efforts. *Marketing Science*, 22(4):520–541, 2003.

Y. Chen, C. Narasimhan, and Z. J. Zhang. Individual marketing with imperfect targetability. *Marketing Science*, 20(1):23–41, 2001.

D. Cui and D. Curry. Prediction in marketing using the support vector machine. *Marketing Science*, 24(4): 595–615, 2005.

M. Dijkstra, H. E. Buijtels, and W. F. Van Raaij. Separate and joint effects of medium type on consumer responses: a comparison of television, print, and the internet. *Journal of Business Research*, 58(3):377–386, 2005.

D. Dzyabura and J. R. Hauser. Active machine learning for consideration heuristics. *Marketing Science*, 30 (5):801–819, 2011.

T. Evgeniou, C. Boussios, and G. Zacharia. Generalized robust conjoint estimation. *Marketing Science*, 24 (3):415–429, 2005.

T. Evgeniou, M. Pontil, and O. Toubia. A convex optimization approach to modeling consumer heterogeneity in conjoint estimation. *Marketing Science*, 26(6):805–818, 2007.

B. Faucon. Technology Startups Take Root in Tehran. The Wall Street Journal, 2015. URL `http://www.wsj.com/articles/technology-startups-take-root-in-tehran-1424917952`. Online, posted 15-February-2015, retrieved 29-August-2016.

J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

A. Ghose and S. P. Han. Estimating demand for mobile applications in the new economy. *Management Science*, 60(6):1470–1488, 2014.

A. Ghose and S. Yang. An empirical analysis of search engine advertising: Sponsored search in electronic

markets. *Management Science*, 55(10):1605–1622, 2009.
A. Ghose, A. Goldfarb, and S. P. Han. How is the mobile internet different? search costs and local activities. *Information Systems Research*, 24(3):613–631, 2012.
A. Goldfarb. What is different about online advertising? *Review of Industrial Organization*, 44(2):115–129, 2014.
A. Goldfarb and C. Tucker. Search engine advertising: Channel substitution when pricing ads to context. *Management Science*, 57(3):458–470, 2011a.
A. Goldfarb and C. Tucker. Advertising bans and the substitutability of online and offline advertising. *Journal of Marketing Research*, 48(2):207–227, 2011b.
A. Goldfarb and C. Tucker. Online display advertising: Targeting and obtrusiveness. *Marketing Science*, 30 (3):389–404, 2011c.
A. Goldfarb and C. Tucker. Shifts in privacy concerns. *The American Economic Review*, 102(3):349–353, 2012.
A. Goldfarb and C. E. Tucker. Privacy regulation and online advertising. *Management science*, 57(1):57–71, 2011d.
J. R. Hauser, O. Toubia, T. Evgeniou, R. Befurt, and D. Dzyabura. Disjunctions of Conjunctions, Cognitive simplicity, and Consideration Sets. *Journal of Marketing Research*, 47(3):485–496, 2010.
X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–9. ACM, 2014.
D. Huang and L. Luo. Consumer preference elicitation of complex products using fuzzy support vector machine active learning. *Marketing Science*, 35(3):445–464, 2016.
S. K. Hui, J. J. Inman, Y. Huang, and J. Suher. The effect of in-store travel distance on unplanned spending: Applications to mobile promotion strategies. *Journal of Marketing*, 77(2):1–16, 2013.
G. Iyer, D. Soberman, and J. M. Villas-Boas. The targeting of advertising. *Marketing Science*, 24(3):461–476, 2005.
G. A. Johnson. The impact of privacy policy on the auction market for online display advertising. 2013.
A. Lambrecht and C. Tucker. When does retargeting work? information specificity in online advertising. *Journal of Marketing Research*, 50(5):561–576, 2013.
J. Levin and P. Milgrom. Online advertising: Heterogeneity and conflation in market design. *The American Economic Review*, 100(2):603–607, 2010.
H. Li, S. M. Edwards, and J.-H. Lee. Measuring the intrusiveness of advertisements: Scale development and validation. *Journal of advertising*, 31(2):37–47, 2002.
L. Liu and D. Dzyabura. Capturing Multi-taste Preferences: A Machine Learning Approach. Working Paper, 2016.
K. Liyakasa. Adobe Positions Its Cross-Device Co-op As An Alternative To Facebook/Google , 2016. URL `http://adexchanger.com/mobile/adobe-positions-cross-device-co-op-alternative-facebookgoogle/`.
X. Luo, M. Andrews, Z. Fang, and C. W. Phang. Mobile targeting. *Management Science*, 60(7):1738–1756, 2013.
P. Manchanda, J.-P. Dubé, K. Y. Goh, and P. K. Chintagunta. The effect of banner advertising on internet purchasing. *Journal of Marketing Research*, 43(1):98–108, 2006.
H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013.

V. Mirrokni, S. Muthukrishnan, and U. Nadav. Quasi-proportional mechanisms: Prior-free revenue maximization. In *Latin American Symposium on Theoretical Informatics*, pages 565–576. Springer, 2010.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

R. B. Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

J. Pancras and K. Sudhir. Optimal Marketing Strategies for a Customer Data Intermediary. *Journal of Marketing research*, 44(4):560–578, 2007.

S. Perez. Consumers Spend 85% Of Time On Smartphones In Apps, But Only 5 Apps See Heavy Use, 2015. URL https://techcrunch.com/2015/06/22/consumers-spend-85-of-time-on-smartphones-in-apps-but-only-5-apps-see-heavy-use/

T. Petterson. Facebook's Ad Volume Has Grown for the First Time in Two Years, 2015. URLhttp://adage.com/article/digital/facebook-q4-2016-earnings/302378/.

J. G. Riley and W. F. Samuelson. Optimal auctions. *The American Economic Review*, 71(3):381–392, 1981.

P. E. Rossi, R. E. McCulloch, and G. M. Allenby. The Value of Purchase History Data in Target Marketing. *Marketing Science*, 15(4):321–340, 1996.

N. S. Sahni. Effect of temporal spacing between advertising exposures: evidence from online field experiments. *Quantitative Marketing and Economics*, 13(3):203–247, 2015.

A. G. Sawyer and S. Ward. Carry-over effects in advertising communication. *Research in Marketing*, 2: 259–314, 1979.

R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

B. Shaul. Smaato: Share of In-App Mobile Ad Spending Increased 13% in 2015 , 2016. URL http://www.adweek.com/socialtimes/smaato-share-of-in-app-mobile-ad-spending-increased-13-in-2015/636399.

E. H. Simpson. Measurement of diversity. *Nature*, 1949.

Statista. Number of smartphone users worldwide from 2014 to 2019 (in millions), 2016. URL http://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

O. Toubia, D. I. Simester, J. R. Hauser, and E. Dahan. Fast polyhedral adaptive conjoint estimation. *Marketing Science*, 22(3):273–303, 2003.

O. Toubia, J. R. Hauser, and D. I. Simester. Polyhedral methods for adaptive choice-based conjoint analysis. *Journal of Marketing Research*, 41(1):116–131, 2004.

C. E. Tucker. Social networks, personalized advertising, and privacy controls. *Journal of Marketing Research*, 51(5):546–562, 2014.

S. Yao and C. F. Mela. A dynamic model of sponsored search advertising. *Marketing Science*, 30(3):447–468, 2011.

H. Yoganarasimhan. Search personalization using machine learning. *Available at SSRN 2590020*, 2016.