Targeting and Privacy in Mobile Advertising

Omid Rafieian*  Hema Yoganarasimhan*
University of Washington  University of Washington

June 12, 2017

*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 Avi Goldfarb and Sridhar Narayanan for detailed comments that have improved the paper. We also thank the participants of the 2016 Invitational Choice Symposium, 2016 FTC and Marketing Science Conference, 2016 Big Data and Marketing Analytics Conference at the University of Chicago, 2017 Phd Students Workshop at the University of Washington, and the 2017 Adobe Data Science Symposium for their feedback. Please address all correspondence to: rafieian@uw.edu, hemay@uw.edu.
Abstract

Mobile in-app advertising is now a dominant form of advertising. We propose a machine learning framework to predict targeting effectiveness for in-app ads. Our framework consists of two modules – functions for feature generation and MART-based predictive model. We apply it to a large-scale data with over 150 million impressions from the leading in-app ad-network in a large Asian country. Substantively, we quantify the improvement in targeting effectiveness to be 15.2% over the baseline. We also show that behavioral targeting with user-information is more effective than contextual targeting based on the where-when-which of the impression. Methodologically, we show that MART significantly out-performs econometric models. We use our model to address three policy questions on privacy. First, we show that stronger privacy regulations on user tracking will lead to a approx. 5% loss in targeting effectiveness. We then consider weaker privacy regulations where the ad-network is allowed to share data with advertisers. We find that advertisers can achieve close to the first-best targeting effectiveness without completely compromising privacy. Importantly, we show that data-sharing does not lead to data democratization – all sharing arrangements benefit large advertisers more than small advertisers. Finally, we examine whether the platform has incentives to share data with advertisers. We show that data-sharing leads to targeted bidding by advertisers and thereby softens competition, and that the platform will benefit from limiting targeting and data-sharing. Thus, by design, the ad-network may be incentivized to preserve users’ privacy to some extent.

Keywords: mobile advertising, machine learning, targeting, competition, privacy,
1 Introduction

Smartphone adoption has grown exponentially in the last few years, with more than two billion people owning a smartphone today (Statista, 2016). The average smartphone user now spends over 2.8 hours per day on her phone. In addition to making phone calls and browsing, users also spend a significant amount of time on native applications, popularly known as “apps” (Perez, 2015; Chaffey, 2017). As mobile usage has surged, advertisers have followed the eyeballs. Mobile advertising generated over $36.6 billion dollars in revenues in US in 2016. Indeed, it has now overtaken desktop ad-spend and represents 51% of digital advertising revenues (IAB, 2017).

The rapid growth of mobile advertising partly stems from an ad format unique to the mobile environment – “in-app ads” or ads shown inside apps. In-app advertising now generates over $40 billion dollars worldwide (not just US) and owes its increasing popularity to two factors (Statista, 2017). First, with the proliferation of free apps, in-app advertising has become a mainstream app monetization strategy. Second, in-app ads have had excellent tracking and targeting properties. Advertisers and ad networks have access to a unique device ID associated with the mobile device that they can use for tracking and targeting purposes, referred to as IDFA (ID For Advertisers) in iOS devices and AAID (Android Advertiser ID) in android devices. This device ID is highly persistent and remains the same unless actively re-set by the user. This allows advertisers to stitch together user data across sessions, apps, and even other advertisers (Edwards, 2012).

While the advertising industry has lauded the trackability of in-app ads suggesting that they create value for advertisers by allowing them to target the right consumer, consumers and privacy advocates have derided them citing privacy concerns. Bowing to pressure from consumers, mobile phone makers have already started taking steps to limit user tracking. First, in 2012, Apple went from UDID (Unique Device ID) to IDFA. The latter is user re-settable (much like a cookie), whereas the former was a fully persistent unique device ID that could not be erased by users under any circumstance (Stampler, 2012). In 2013, Android made a similar move from UDID to AAID (Sterling, 2013). While this caused some consternation among advertisers, it nevertheless didn’t upend the mobile advertising eco-system since few users actively re-set IDFAs on a regular basis. However, in mid-2016, Apple took a much stronger stance in support of privacy by allowing users to completely opt out of tracking through LAT (Limit Ad Tracking), wherein a user can simply switch off tracking indefinitely (Gray, 2016). This makes it impossible for advertisers to target ads,

1This is in contrast to browser-based mobile ads, which have poor tracking properties since third-party cookies don’t work in many browsers (especially Safari, which is one of the most popular browsers) and hence tend to be poorly targeted (Sands, 2015).

2When a user turns on LAT, her IDFA simply shows up as a string of zeroes to the advertisers. Thus, all the consumers who have opted into LAT are visible as one large ghost segment to ad networks.
attribute conversions, or re-target and re-engage users. Studies show that nearly 20% of iOS users have already chosen to opt into LAT, raising significant concerns among advertisers about the future of targeted mobile advertising (Seufert, 2016).

This tension between targeting and privacy in mobile advertising is part of a larger ongoing debate over consumer tracking and privacy in digital contexts. Advertisers argue that tracking allows consumers to enjoy free apps and content, and see relevant ads, whereas regulators and consumers demand higher privacy, control over their data, limits on data-sharing among firms, and the ‘right to be forgotten’. In mid-2016, the European Union signed the General Data Protection Regulation (GDPR) agreement, which will go into effect in May 2018 (Kint, 2017). Under the new regulations, consumers will have to opt into (rather than opt out of) behavioral targeting and will need to give explicit permission for their data to be shared across firms. Violators will be fined up to 4% of their global revenues. In the US, under the Obama administration, the Federal Communications Commission (FCC) adopted legislation to limit tracking and data-sharing among large ISPs. The advertising industry lobbied against these rules, and scored a win when these rules were recently relaxed under the Trump administration (Fung, 2017). Nevertheless, polls show that over 92% of US consumers believe that the government should ban tracking and sharing of consumer data (Edwards-Levy and Liebelson, 2017).

Even as this debate rages on, we don’t have a good understanding of how tracking and data-sharing affect targeting effectiveness in mobile advertising, and what the incentives of different players in the advertising industry are. Lack of clear empirical analyses of these issues hampers our ability to have an informed discussion and to form policy on these issues. In this paper, we present a large-scale empirical analysis of mobile targeting effectiveness using machine learning methods. We then examine how changing privacy regulations would affect advertisers’ ability to target, and the incentives of ad-networks to share data with advertisers.

Broadly speaking, we address four key research questions that are of interest to marketers and policy makers. First, we examine how different pieces of information affect targeting effectiveness. Specifically, we estimate the relative value of two types of information – 1) contextual information and 2) behavioral information. The former quantifies the context (when-where-which) of an impression, and the latter uses historic data on an individual user’s past app usage and ad-response for targeting purposes. Contextual targeting is privacy preserving whereas behavioral information is based on user-tracking and therefore impinges on user-privacy. Second, we examine which methodological paradigms perform the best estimating targeting effectiveness. We compare some of the standard methods used in the marketing literature such as OLS and Logistic regressions with machine learning methods such as Boosted regression trees.
Third, we examine how changing privacy regulations will affect advertisers’ ability to target. We consider a counterfactual scenario where privacy regulations are strengthened by policy makers through a mechanism like LAT. In this case user-tracking through device IDs is no longer possible and advertisers have to rely on other mechanisms to track and target users. What is the extent of loss in targeting accuracy under such circumstances? Next, we consider a series of counterfactual scenarios where privacy regulations are weakened such that the ad-network shares increasingly granular information with advertisers. What is the gain in targeting effectiveness under these laxer regulations? Importantly, which types of advertisers benefit more from weaker privacy regulations?

Finally, we examine whether the ad-network has incentives to share data with advertisers and engage in micro-targeting? Starting with Levin and Milgrom (2010)’s conjecture, a growing stream of analytical literature has theorized that high levels of targeting can soften competition between advertisers and thereby reduce ad-network’s revenues. This implies that ad-networks have an incentive to not share all their data with advertisers and not to engage in micro-targeting. If true, this suggests that the ad-network’s and consumers’ incentives are somewhat aligned, wherein ad-networks may be naturally incentivized to protect consumer privacy. In that case, the advertising industry’s claim on the efficacy of self-regulation may hold true. However, there has been no empirical evidence to support these theoretical conjectures so far. Our goal is to examine if there is sufficient empirical evidence to support or reject these hypotheses.

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 perform poorly in terms of prediction accuracy (He et al., 2014). Our framework consists of two components. First, a feature generation framework that relies on a set of functions to generate a large number of features that capture both contextual and behavioral information related to an information. These features serve as the input variables in our prediction model that comes next. We use Multiple Additive Regression Trees (MART) to predict the probability of a click. MART has been shown to be one of the best predictors in the machine learning literature.

We apply our framework to large-scale data from the leading Android in-app advertising platform from a large Asian country, that serves over 50 million ad impressions per day. We conduct our analysis on one month of data, and present the following set of findings.

From a substantive perspective, we find that behavioral features are much more important for effective targeting than contextual features. Contextual features provide only a 5.2% improvement in click prediction, whereas behavioral features provide a 9.2% improvement compared to the baseline. Of course, using both leads to a much larger improvement of 15.2%. Specifically, within contextual
information, we find that app-specific information (where the impression occurs) is the most relevant for targeting, contributing a 4.4% improvement. Ad-specific features are somewhat relevant (contributing 0.7% improvement) followed by time-specific features (when the impression occurs), which contributes a meagre 0.2%. Overall, this suggests that not having behavioral information can severely hamper a firm’s ability to target effectively.

Methodologically, we find that MART significantly outperforms both Logistic and OLS regressions. MART leads to a 15.2% improvement, whereas OLS and Logistic regression lead to 9.5-10% improvements. This is the case, even after we supply the OLS and Logistic models with features generated using our machine learning framework.

From a policy perspective, we find that LAT and similar policies that do not allow for device tracking can have severe negative consequences for targeting. When we turn off device IDs (AAIDs in Android or IDFAs in iOS), and instead rely on IP addresses for behavioral targeting, our overall model performance drops to 10.3%. If we just consider behavioral features, we find that with IP address, behavioral targeting only improves click prediction by 2.3%. This is stark contrast to the 9.2% improvement from behavioral targeting when using device IDs. Next, we consider a set of scenarios where the ad-network is allowed to share data with advertisers. As expected, we find that advertisers’ ability to target increases as the data sharing arrangements become laxer. Nevertheless, we find that the second-laxest data sharing arrangement comes very close to the least lax arrangement in terms of its ability to improve targeting effectiveness. This suggests that even if we allow free data-sharing, the least privacy-preserving arrangements may not be chosen by firms. We also find that there is significant heterogeneity in which advertisers benefit the most from data-sharing. Interestingly, large advertisers benefit when the platform shares data with them compared to small advertisers. This goes against the conjecture of experts who have argued that allowing data-sharing will benefit small firms the most by democratizing access to data.

Finally, we address the question of ad-network’s incentives – “Does the ad-network have incentives to engage in micro-targeting and share targeting data with advertisers? If yes, is there an optimal level of targeting and data-sharing?” We present two approaches to answer this question. First, we derive and estimate measures of competition under different targeting scenarios, and show that competition softens as targeting increases. Next, we develop a stylistic analytical model that fleshes out Levin and Milgrom (2010)’s conjecture, and characterize the ad-network’s revenue, advertiser gain, and total surplus under different targeting scenarios. The analytical model highlights the ad-network’s trade-off between value creation and value appropriation when deciding the optimal level of targeting. We then take this model to data, and empirically show that as targeting increases, the total surplus increases (implying that targeting creates value by matching the best
advertiser and impression). However, the ad-network’s revenues are not monotonic; after a certain level of targeting, they start decreasing. This has two important policy implications. First, it suggests that ad-networks will choose not to engage in micro-targeting, and will not share targeting data with advertisers beyond a certain point. Thus, consumers’ and ad-networks incentives on privacy are somewhat aligned (though not perfectly). Second, given the structure of incentives, it may not be necessary for an external entity such as EU/FCC to impose privacy regulations. To some extent, self-regulation by the industry is feasible.

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,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 advertising. For example, Bart et al. (2014) examine the purchase intention for the products that are high on the utilitarian dimension in mobile display advertising, and Sahni and Nair (2016) exploit the mobile search-ad experiments to empirically investigate theories with regard to native advertising and sponsorship disclosure. More related to the application marketplaces, Ghose and Han (2014) build a demand estimation for mobile application marketplace. In addition, Narang and Shankar (2016) examine the effects of mobile apps on shopper behavior. 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, one 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., 2003, 2004; 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 the leading mobile in-app advertising network of a large Asian country, which has over 85% market-share in the category. Over 10,000 apps and 250 advertisers have joined the ad network. It now serves over 50 million ads per day (i.e., runs more than 600 auctions per second).

3.1.1 Players

We now describe the four key players in this marketplace.

- Consumers: individuals who use apps. They see the ads shown within the apps that they use and may choose to click on the ads.

- Advertisers: firms that 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 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**: app developers who have joined the ad 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.

- **Ad network or Platform**: 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 §3.1.2 for details). The platform uses a CPC pricing mechanism, and therefore generates revenues only when clicks occur.

### 3.1.2 Auction Mechanism

The platform uses an auction mechanism called quasi-proportional auctions in the literature (Mirrokni et al., 2010). The key distinction between a quasi-proportional auction and other commonly used auctions (e.g. second price or Vickrey) is the use of a probabilistic winning rule. 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.

\[
p_i = \frac{b_is_i}{\sum_{j \in A} b_js_j}
\]

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 \( b_is_i \)) is not guaranteed to win. Rather, his probability of winning is proportional to the expected revenue generated from him.

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

---

3. 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.

4. 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.
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 worst-case performance than second-price auctions, especially when bidder valuations are starkly different. For these reasons, the platform has adopted a quasi-proportional auction mechanism and does not employ a reserve price.

3.1.3 Randomness in the Data Generation Process

This probabilistic allocation mechanism generates randomization in ads across users and apps, which facilitates our analyses to a great extent.

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 associated with the impression, which is essentially the IP of the accessing user’s smartphone when the impression occurs.
- Advertising ID: The Advertising ID (also referred to as AAID) is a user re-settable, unique, device ID that is provided by the Android operating system. It is accessible to advertisers and ad networks for tracking and targeting purposes. We use it as the user identifier in our main analysis.
- 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.

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.

Apple’s app store is not available in the country where our data are sourced from. Hence, all smartphones use the Android operating system.
The total data we see in this one month interval is quite large. Overall, we observe a total of 1,594,831,699 impressions and 14,373,293 clicks in this time-frame, implying a 0.90% CTR.

3.3 Sampling and Data Preparation

Supervised machine learning algorithms require three sets of data – training, validation, and test (Hastie et al., 2001). Training data is the set of pre-classified data that is used to fit the model. Machine learning models require the researcher to choose a few ‘tuning’ parameters. These parameters are chosen (or tuned) based on the model’s performance in the training data. So we often end up in situations where the model over-fits on the training data but fits poorly on other data. This is where the validation data comes in – each model fit on the training data is evaluated on the validation data, and the model with the best performance on the validation data is chosen as the final model. Since both training and validation data are used in model specification and selection, we use a completely new data, test data, to evaluate the model’s out-of-sample performance.

Since we only have a snapshot of the data from September 30 to October 30, we have to generate all three datasets from the last few days of data, and use the preceding history to generate the features associated with these impressions (Yoganarasimhan, 2016).

Therefore, as specified in [1] the train and test datasets are the last three days, from October 28 to October 30. We use the first two days (October 28 and 29) for training and validation, and the last day for testing (October 30). We also refer to the preceding history as the global data, by which we can generate features for train and test datasets. The reason why we use the global dataset is to extract some features for impressions we see in either train or test data. For example, the global data would inform us about any impression in the train dataset with respect to the history of the user, ad, app, and time characterizing this impression.

We now discuss the sampling procedure used to generate our training and testing datasets. Sampling is a necessary step when working with large-scale 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.

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 train and test datasets. We then track the impressions containing these users for the last one month to build the global data and generate the associated features for train and test datasets. In §6.2, 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 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 validation data, and that of October 30 as the test data. As such, there are 17,733,791 impressions corresponding to the train and 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).
3.4 Summary Statistics

First, consider the distribution of impressions over ads and apps. We observe a total of 264 unique ads and 10789 unique apps in the data. However, over 80% of the impressions come from top 50 apps and top 50 ads, as shown in Figures 2a and 2b.

We also observe significant heterogeneity in the click-through rates across ads and apps. Figure 3a shows the histogram of click-through rate for ads and Figure 3b shows the app specific click-through rate over one month of the data, in October 2015. Comparing two histograms in Figure 3, we find that the variance among apps is higher than the variance among ads. One important factor 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.

Next, Figure 4 depicts the average percentage of impressions and clicks for each hour of the day. Over 40% usage (or impressions) takes place after 6 pm, with usage peaking at 10-11 pm. Click through rates are also significantly higher during the evenings, with the number of clicks after noon being three times higher than the number of clicks before noon. These patterns may reflect the fact that users are more likely to have leisure time in the afternoons and evenings to spend on apps and click on ads to further explore some new products.

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.

### 3.4.1 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.\(^7\) 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

---

\(^7\)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.
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. 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.

Figure 5 illustrate how our identifiers do not perfectly match. Figure 5a shows that about half of the Advertising IDs correspond to two or more distinct IPs. Given that each Advertising ID can only belong to one user, this figure indicates a significant fraction of users have more than one IP. On the other hand, as shown in Figure 5b, around 80% of IPs correspond to only one Advertising ID. It suggests that although there are different users with the same IP, but this problem is not as prevalent as the problem that one user has multiple IPs. Thus, the case of using IP as the main identifier would be similar to the case that users reset their IDs regularly. Later in this paper, we show what would happen if we use IP as the main identifier.

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

Figure 4: Percentage of impressions and clicks by time of day.
Machine Learning 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 Metric

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:

$$LogLoss(\mathbf{p}; \mathbf{y}) = -\frac{1}{N} \sum_{i=1}^{N} (y_i \log (p_i) + (1 - y_i) \log (1 - p_i))$$

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:

\[ NE(p_A, p_B; y) = \frac{\text{LogLoss}(p_A, y)}{\text{LogLoss}(p_B, y)}, \]  

(3)

where \( p_A \) and \( p_B \) 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:

\[ RIG(p_A, p_B; y) = 1 - NE(p_A, p_B; y) \]  

(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:

$$\text{Impressions}(u, p, a, t) = \sum_{H} \mathbb{1}(U = u) \mathbb{1}(P = p) \mathbb{1}(A = a) \mathbb{1}(T = t)$$

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:

$$\text{Impressions}(u, -, a, -) = \sum_{H} \mathbb{1}(U = u) \mathbb{1}(A = a)$$

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:

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

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:

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

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:

\[
\text{AdCount}(u, p) = \sum_{a \in A} 1(Impressions(u, p, a, _) > 0),
\]

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 function works quite similarly to AdCount. We can define it as follows:

\[
\text{AppCount}(u, a) = \sum_{p \in P} 1(Impressions(u, p, a, _) > 0),
\]

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:

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

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:

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

We know that users respond differently to the same ads in different apps. Therefore, we aim to capture this variation in user behavior into \(\text{AppVariability}\).
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 the \(\text{Simpson} \ 1949\) 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:

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

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 1 and briefly describe the process of feature generation below.

<table>
<thead>
<tr>
<th>Feature No.</th>
<th>Feature Name</th>
<th>Feature Classification</th>
<th>Contextual Features Subclassification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Behavioral</td>
<td>Contextual</td>
</tr>
<tr>
<td>1</td>
<td>Impressions ((user, _, _, _))</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Impressions ((_, app, _, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Impressions ((_, _, ad, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>Impressions ((_, _, _, time))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>Impressions ((_, app, ad, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>Impressions ((user, app, _, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>Impressions ((user, _, ad, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>Impressions ((user, app, ad, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>Impressions ((user, _, _, time))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>10</td>
<td>Clicks ((user, _, _, _))</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Clicks ((_, app, _, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>12</td>
<td>Clicks ((_, _, ad, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>13</td>
<td>Clicks ((_, _, _, time))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>14</td>
<td>Clicks ((_, app, ad, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>15</td>
<td>Clicks ((user, app, _, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>16</td>
<td>Clicks ((user, _, ad, _))</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Feature No.</th>
<th>Feature Name</th>
<th>Feature Classification</th>
<th>Contextual Features Subclassification</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>Clicks (user, app, ad, _)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>18</td>
<td>Clicks (user, __, __, time)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>19</td>
<td>CTR (user, __, __)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>CTR (__, app, __, __)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>CTR (__, __, ad, __)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>CTR (__, __, __, time)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>CTR (__, app, ad, __)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>CTR (user, app, __, __)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>CTR (user, __, ad, __)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>CTR (user, app, ad, __)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>CTR (user, __, __, time)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>AdCount (user, __)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>AdCount (__, app)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>AdCount (user, app)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>AppCount (user, __)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>AppCount (__, ad)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>AppCount (user, ad)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>TimeVariability (user)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>AppVariability (user)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Entropy (user, __)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>Entropy (__, app)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Entropy (user, app)</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Each feature is characterized by a function and a set of inputs. For instance, $\text{Impressions}(\_, \_, \_, \_, \_)$ is a feature characterized by the $\text{Impressions}$ 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.

In the case of the $\text{Impressions}$ 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 I. These one-element sets

---

9While 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.
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 I.

**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 I. 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 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). Next, within, contextual, we further classify the features into three sub-categories (also overlapping) – a) App-specific features ($F_P$), b) Ad-specific features ($F_A$), and c) Time-level specific features ($F_T$). This gives additional information on the context.

In §6.1.1 we examine the relative value of these different 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 6. 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 6, 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 (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:

\[
L_K(x) = \sum_{k=1}^{K} \alpha_k l_k(x, \beta_k)
\]

where \( l_k(x, \beta_k) \) is the function modeled by the \( k^{th} \) regression tree and \( \alpha_k \) is the weight associated with the \( k^{th} \) tree. Both \( l(\cdot) \)s and \( \alpha \)s are learned during the training or estimation. MARTs are also trained using greedy algorithms. \( l_k(x, \beta_k) \) 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 Implementation: training, 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 these features as input, we train the model on train data (sampled from October 28\(^{th}\)), validate it on validation data (sampled from October 29\(^{th}\)), and test the model performances on test data (sampled from October 30\(^{th}\)). We use the package XGBoost for this purpose (Chen and Guestrin, 2016).

One common challenge in implementation of machine learning algorithms is over-fitting. That is, the algorithms are complex enough to perfectly fit the training data and shrink the training error. However, this approach leads to a poor out-of-sample prediction. To avoid over-fitting, we use a simple validation procedure. As such, the validation set is used to estimate prediction error for different sets of tuning parameters and pick the one with best validation performance. We use first half of training and validation data (October 28\(^{th}\)) to fit the model and compute its predictive accuracy on the second half (October 29\(^{th}\)).

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, and the learning rate. After experimenting with a number of different parameters on the validation set, the following tuning parameters gave us the best performance:
• Maximum trees = 3000
• Maximum depth of a tree = 6
• Stopping rule = Stop adding trees when no improvement in prediction after three rounds
• Learning rate = 0.1

In our training, we find that our optimization algorithm stops at 320 trees (i.e., after adding the 320\textsuperscript{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).

6 Results

6.1 RIG improvement

We present our main results on the prediction accuracy of our approach with different model specifications in Table 2. 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 micro-level information available.

There are two main factors that affect the prediction accuracy of our model – 1) the set of features used, and 2) the optimization method used. We show the RIG improvements for different combinations of feature-sets and optimization methods in Table 2.

6.1.1 Value of Different Types of Features

We now examine how much different types of features influence the predictive accuracy of our model, starting with contextual features (that are global in nature and contain aggregate information on ad, app, or time), to behavioral features (that are user-specific), to those that include both types of information (behavioral and contextual). Understanding extent to which each type of information improves prediction is critical from a managerial perspective for three reasons. First, this is a substantively important issue for managers; it can give them some intuition on why/how consumers click on ads and who clicks on their ads. It can also help them decide what types of new information to collect and what type of targeting to spend their resources on. Second, there are different privacy and policy implications associated with different types of targeting. For example, behavioral targeting requires individual-level impression and click data, while contextual targeting can be achieved even with aggregate global data. To evaluate the costs and consequences of storing and using more sensitive data, managers first need accurate measures of the value of more granular data (if any). Third, data storage and processing costs vary across feature types. For example, user-specific behavioral features require more granular information and need real-time updating,
whereas pure-contextual features tend to be more stable, and can therefore be stored as global variables and updated less frequently. In order to decide whether to store and update a feature or not, we need to know its incremental value in improving targeting. Therefore, we evaluate the predictive accuracy of our model for different sets of features.

Recall that we categorized features into two broad overlapping categories – 1) Behavioral, denoted by $F_B$ and 2) Contextual, denoted by $F_C$. Within contextual, features are further sub-categorized as App-specific ($F_P$), Ad-specific ($F_A$), and Time-specific features ($F_T$). Based on this categorization, we define the following eight combinations of feature sets.

1. User: This set consists of purely behavioral features, with no contextual information. Formally, we define this as the set features in $F_B$ but not in $F_C$ (1, 10, 19).

2. Ad-App-Time: This set consists of purely contextual features, with no behavioral information. We define this as the set of features in $F_B$ but not in $F_C$ (2, 3, 4, 11, 12, 20, 21, 23, 29, 32, 37).

Next, we consider three sub-categorizations of pure contextual features as follows:

3. Ad: Ad-specific contextual features that are neither app nor time specific. Formally, these consist of features in $F_A$ but not in $F_B$, $F_P$, and $F_T$ (3, 12, 21).

4. App: Contextual features that are app-specific, but neither ad nor time specific. Formally, these consist of features in $F_P$ but not in $F_B$, $F_A$, and $F_T$ (2, 11, 20).

5. Time: Contextual features that are time-specific, but neither ad nor app specific. Formally, these consist of features in $F_T$ but not in $F_B$, $F_A$, and $F_P$ (4, 13, 22).

Finally, we consider combinations of behavioral and contextual features.

6. User-Time: Consist of features in $F_B$ or $F_T$ but not in $F_A$ and $F_P$ (1, 4, 9, 10, 13, 18, 19, 22, 34).

7. User-Ad-App: Consist of features in $F_B$, $F_A$, and $F_P$ but not in $F_T$ (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, 35, 36, 37, 38).

8. All: Consist of all the features (1 to 38).

The columns in Table 2 show the RIG improvements with these eight different sets of features.

**Behavioral vs. Contextual Features:** As we can see from Table 2, when we only have access to user-specific information (pure behavioral targeting with no contextual information, as shown
in Column 2 in Table \[2\] User), RIG over the baseline model is 9.3%. This gain is considerably higher than 5.1%, which is the improvement when we use all three aspects of contextual information (ad, app, time), but no behavioral information (Column 3 in Table \[2\] Ad-App-Time). 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:** Next, within contextual features, we examine the relative value of app-specific and ad-specific features. This is an important question that is of relevance to both advertisers and publishers in mobile in-app ads. Our results indicate that we app-specific features are better at predicting clicks compared to ad-based features (comparing columns 4 and 5 in Table \[2\]). 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 don’t find much value in pure time-specific features without any other information. This could due to the fact that there is significant heterogeneity across consumers in when and how they use their mobile apps and click on ads.

**Combination of features:** Interestingly, time-specific features do add value when we combine them with user-specific features (Column 7; User-Time). As mentioned earlier, this could be because user-time features can capture the heterogeneity in how users use behave across time. Similarly, we find that when we combine user-level information with app-ad-specific information,
the improvement in the model’s predictive ability is quite significant.

6.1.2 MART vs. other methods

We also compare different optimization methods and their role in improving the predictive accuracy of our model. In Table 2, 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.

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 2). 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 RIG 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.

6.1.3 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 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 comes from the interactions between four key variables (user, ad, app, time) 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 outcomes.

6.2 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. We also report the average sample size of train and test data respectively as $\bar{N}_{\text{train}}$ and $\bar{N}_{\text{test}}$ in Table 3.

Table 3 shows the results for RIG over two different baseline models for different sample sizes. Our first baseline predicts average CTR as the click probability for all the test observation.

![Figure 7: RIG of different user sample sizes over two different baselines.](image)
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.

7 Implications for Privacy Regulations, Micro-Targeting, and Data Sharing

We now use our modeling framework to examine the implications of changing consumer privacy regulations in this market.

1. In §7.1, we examine how policy changes in privacy regulations would affect the advertisers’ ability to target. We consider a situation where consumer privacy protection is strengthened through a ban of user tracking through Advertising ID. We quantify the effect of this change on the platform’s and advertisers’ ability to target consumers.

2. In §7.2, we study the impact of weaker consumer privacy protection compared to the current restrictions imposed by the platform. We consider a series of increasingly lax data-sharing arrangements that the platform can make with its advertisers and examine how they would affect the advertisers’ ability to target. Can such arrangements improve advertisers’ targeting outcomes? If yes, which types of advertisers would benefit the most from them?
3. In §7.3 we examine whether the platform has an incentive to share targeting data with advertisers. Some 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. We empirically examine whether the data support or refute this hypothesis. Note that incentives are particularly important in this context because if platform is naturally incentivized to not do micro-targeting and/or share data advertisers, then we may naturally converge to an equilibrium with higher consumer privacy protection. In contrast, if the platform is incentivized to use store and share micro-level consumer data, 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.

7.1 Value of User Identifiers: IP vs. Advertising ID

User-identification and tracking are at the core of targeting technology in the digital advertising industry. As discussed earlier, mobile in-app advertisements rely on a device-specific ID (that is re-settable by the user) for tracking purposes. Privacy advocates have demanded that mobile phone makers disable such tracking, whereas advertisers have argued that this would lead to significant loss in their targetability (and hence revenues). As discussed earlier, Apple has already taken a step in this direction with the introduction of LAT.

We now empirically examine what would be the extent of loss in targeting ability if lawmakers were to strengthen consumer privacy laws and prevent the use of a tracking identifier? Under this new regime, advertisers and the platforms would have to rely on IP as their mobile-tracking metric. Would this adversely affect the advertisers’ and platform’s ability to target, and if yes how much? From a policy perspective, this analysis focuses on key trade-offs and issues at the intersection of consumer privacy and marketing practice.

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 4. There are major differences compared to Table 2 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 contextual 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, 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 identify users largely increases.

Finally, to ensure that the differences in the results between Tables 2 and 4 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 727,354 unique Advertising IDs and another set of 727,354 unique IPs. The results of Table 2 are based on the former and the results in Table 4 are based on the latter. However, around 20% of these identifiers are mutual, and for these we have the information on both Advertising ID and IP. We now examine the performance of our models on this overlapping dataset and present the results in Table 5. 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.

Table 5: RIG for different models using two different user identifiers (Advertising ID or IP)

### 7.2 Data-sharing arrangements between the platform and advertisers

Ad-networks usually share information with the advertisers to help them manage their campaign more effectively with respect to their objectives. This is part of a service that ad-networks are giving to their customers (advertisers). However, the extent to which the ad-networks should be allowed to share the information with the advertisers is of high importance for privacy advocates as well as all the parties involved in the mobile in-app advertising (i.e. ad-networks, advertisers, publishers, and users). The reason is that some data-sharing arrangements can potentially violate privacy rights of the parties.
We focus on two major types of privacy violation in data-sharing. The first one is the violation of users’ privacy. As such, ad-networks are not usually allowed to share user-level data with advertisers. In some cases, however, they can share the aggregated information about users without revealing their identity. For example, ad-networks usually share information on a segment of users interested in a particular context. The second type of privacy violation is with respect to advertisers’ privacy. That is, sharing data of advertiser A with advertiser B violates advertiser A’s privacy.

Given the concerns associated with such privacy violations, governments regulate to protect privacy rights of the parties. In what follows, we first define five hypothetical data-sharing scenarios, each would reflect the result of a specific privacy regulation. We then examine advertisers’ ability to target and compare their performance across different scenarios. Further, we seek to find the ad characteristics accounting for the performance of advertisers under each scenario.

We begin with describing five hypothetical scenarios, as presented below:

1. **Scenario 1**: Advertisers only access their average CTR. No information about the users, apps, and time is provided. Thus, no privacy violation is expected in this scenario.

2. **Scenario 2**: Advertisers access their average CTR in different apps. This arrangement does not violate privacy regulations, because all the information is provided at the aggregate level. Again, no privacy violation is expected.

3. **Scenario 3**: Advertisers access their own impression-level data. It means that they observe the variables, time, user, app, and click for each impression within their ad. However, they do not access other ads’ impression level data. Thus, they cannot generate the features that are aggregated over ads. This arrangement could only be considered as a violation of user privacy.

4. **Scenario 4**: Advertisers access their own impression-level data and the features obtained 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 is not a violation of advertisers’ privacy. Thus, like scenario 3, 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 advertisers’ privacy.

Given the information available in each scenario, we build a prediction model and evaluate its performance on a separate test set for each advertiser. We contend that a better performance on
the same test set indicates a greater ability to target. Note that it does not necessarily imply that the main objective of all advertisers is to have a higher CTR. Rather, since the number of clicks is the main driver of their costs, having a more accurate click prediction helps them target more effectively. Assuming this, the challenge is to build a prediction model an advertiser can make, given the constraints in each scenario. The prediction models for the first two scenarios are clear. In scenario 1, advertisers simply predict the average 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 average CTR in that specific app as the click probability.

In scenario 3, however, the advertisers are provided with their own impression-level train data. Thus, they can build a prediction model, similar to what we built in §4. However, since they do not access the data of other ads, they are not able to generate the features aggregated over the ads. For example, they cannot generate Impressions(user, __, __), since this feature is aggregated over ads. The features they can generate are those for which the ad is given, including features 3, 5, 7, 8, 12, 14, 16, 17, 21, 23, 25, 26, and 32.

In scenario 4, they have their own impression-level data, with the full set of features as it is allowed by the ad-network. Lastly, in scenario 5, they can use the same MART model as we used, and test the prediction results on their own test data. Scenarios 4 and 5 are the same with respect to the set of features they use. The only difference is in the size of data. In scenario 4, advertisers are just allowed to train the full model on their own data, while in scenario 5, advertisers train the model on the full dataset including all the ads.

Setting the baseline as the model in scenario 1, we calculate the RIG of advertiser’s model in scenarios 2 to 5 over the baseline. It is worth noting that we test the prediction models under different scenarios on the same test dataset. We take the average RIG for different tiers of ads - large, medium, and small - defined based on the size of their data. Two general patterns could instantly be observed in Figure 8. First, the increasing trend indicates that RIG increases as we move from one scenario to the next one with less data restrictions. Namely, advertisers achieve greater ability to target when the data-sharing arrangements are more lax. Second, larger advertisers seem to benefit more in all scenarios with impression-level data-sharing. The reason stems from the fact that larger advertisers access to a more granular data with higher variation.

Taking a closer look, however, Figure 8 illustrates the extent to which advertisers in different tiers benefit by moving from one more restrictive scenario to a more lax one. This can be captured by the slope of such movements. For example, the slope for large advertisers is flatter than the slope for small advertiser when moving from scenario 4 to 5. This would capture the competitive advantage
advertisers get from moving from one scenario to another. Thus, although larger advertisers benefit more than smaller advertisers in scenario 5, they may prefer to be in scenario 3 or 4, since their competitive advantage is the highest in these scenarios.

Although Figure 8 provides some evidence on how advertisers differentially benefit when data restrictions are removed, the use of average RIG over one tier of ads would be misleading. Further, it does not fully address which data characteristics of an advertiser are accountable for a greater ability to target. We intend to answer these questions by conducting a series of linear regression models. Each model represents one moving scenario (e.g. from scenario 3 to 5). We calculate RIG for all ads in this moving scenario, and regress it on two data characteristics of the ads: size and CTR. The results are presented in Table 6.

As shown in columns 2, 3, and 4 of Table 6, the size of an advertiser positively affects its ability to target, when the ad-network shares impression-level data with advertisers. However, this effect may be due to different factors in each case. For columns 2 and 3, we know that the larger advertisers are provided with a larger data set. Thus, data granularity would explain why larger advertisers perform better. In column 4, however, we know that all advertisers have the same data. The reason why larger advertisers still benefit more when they have the full data set is that the model relies more on their data, since their impressions constitute a bigger portion of the full data set. Thus, the model fits better on their impressions.
Although the larger advertisers benefit more in all the scenarios under which they are given impression-level data, they may want to have their privacy rights preserved in order to keep their competitive advantage in scenario 4. As indicated in column 7, larger advertisers benefit less as we move from scenario 4 to 5. It offers some evidence that advertisers differentially benefit from different levels of information sharing. In addition to the size, we examine the effects of CTR on advertisers’ ability to target. We find specific evidence that in scenario 4 (columns 3 and 5), not only does size of the ad matter, but also its CTR positively affects the performance. One reason is that advertisers are provided with many features in this scenario, and with more clicks, they can build a better prediction model. In column 6, we also find the marginally significant positive effect of CTR on the ability to target as we move from scenario 3 to 5. 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 on their test data.

7.3 Platform’s Incentives to Target and Share Data with Advertisers

7.3.1 Background

While the previous sections focused on how the platform’s as well as advertisers’ ability to target changes with the extent of information they have access to and the privacy regulations in place, it was silent on the fundamental question of – “Does the platform have incentives to do micro-targeting and share targeting data with advertisers? If yes, is there an optimal level of targeting and data-sharing?”

These questions have been the focus of a growing body of literature on targeting in Internet ad auctions. [Levin and Milgrom (2010)] were one of the first to conjecture that too much targeting can
thin markets which in turn would soften competition and make the platform worse off. Since then this idea has been explored in a series of theory papers. Two key themes emerge from this literature. The first is a characterization of the platform’s trade-offs with increasing targeting. On the one hand, enhanced targeting increases social welfare by improving the match between advertisers and consumers. On the other hand, it softens competition when the heterogeneity in advertisers’ match for a given consumer (or impression) is high. This in turn can reduce platform revenues by allowing advertisers to appropriate the increase in social surplus at the expense of the platform. This leads to the theoretical prediction that under some circumstances, standard second-price auctions lower revenues for the platform (Bergemann and Bonatti, 2011; Fu et al., 2012; Amaldoss et al., 2015; Hummel and McAfee, 2016). The second broad theme relates to data-sharing, privacy, and targeting. In an early paper, Milgrom and Weber (1982) identified the classic Linkage Principle which suggests that the auctioneer/platform can increase its revenue from second price auctions if the bidders’ valuations (or match for an impression) are positively correlated. However, in Internet ad auctions, such positive correlation is less likely. Thus, sharing data with advertisers can lead to targeted bidding strategies on the advertisers’ part, which in turn can hurt platform profits (Ganuza, 2004; De Corniere and De Nijs, 2016; Marotta et al., 2017).

In spite of the increasing interest from the theoretical side, so far there has been no empirical evidence to support the idea that targeting in real ad auctions can exceed optimal levels and/or that reducing targeting can improve platform revenues. Our contribution to this debate is to present empirical evidence from a large-scale mobile advertising platform on the extent to which micro-targeting can affect competition and revenues in ad auctions.

The rest of this section proceeds as follows. We first present a simple example in §7.3.2 to fix ideas and highlight the platform’s trade-off. In §7.3.3 we characterize some model-free measures of competition under different targeting strategies and present the empirical distributions of these measures in our data. Next, we present a stylistic model of platform revenues under different targeting/data-sharing strategies. We then take this model to the data in §7.3.5 and present some empirical results on measures of competition and platform revenues. We conclude with some caveats regarding our analysis and a brief discussion on optimal mechanism design for mobile ad auctions.

### 7.3.2 A Simple Example

Consider a platform with two advertisers ($a_1$ and $a_2$) competing for two impressions by two users ($u_1$ and $u_2$). Assume that the platform uses a second-price auction with Cost Per Impression (CPI) pricing strategy. In this mechanism, the highest bidder wins the impression and pays the bid of the second-highest bidder for the impression. These auctions have the useful property of truthful
Table 7: Example depicting two scenarios: 1) No targeting and 2) Perfect Targeting

<table>
<thead>
<tr>
<th>No Data Disclosure (or No Targeting)</th>
<th>Full Data Disclosure (or Perfect Disclosure)</th>
</tr>
</thead>
<tbody>
<tr>
<td>For both impressions:</td>
<td>For User 1’s impression:</td>
</tr>
<tr>
<td>( a_1 ) bids: ( b_{11} = b_{21} = 0.3 )</td>
<td>( b_{11} = 0.5, b_{12} = 0.1 )</td>
</tr>
<tr>
<td>( a_2 ) bids: ( b_{12} = b_{22} = 0.2 )</td>
<td>( a_1 ) wins impressions for ( u_1 )</td>
</tr>
<tr>
<td>( a_1 ) wins both impressions</td>
<td>Pays 0.1 for the impression</td>
</tr>
<tr>
<td>Pays 0.2 per impression</td>
<td></td>
</tr>
<tr>
<td>Advertiser ( a_1 )’s Surplus ( R_{a1} = 0.2 )</td>
<td>Advertiser ( a_1 )’s Surplus ( R_{a1} = 0.4 )</td>
</tr>
<tr>
<td>Advertiser ( a_2 )’s Surplus ( R_{a2} = 0 )</td>
<td>Advertiser ( a_2 )’s Surplus ( R_{a2} = 0.2 )</td>
</tr>
<tr>
<td>Platform’s Revenue ( R_P = 0.2 + 0.2 = 0.4 )</td>
<td>Platform’s Revenue ( R_P = 0.1 + 0.1 = 0.2 )</td>
</tr>
<tr>
<td>Total surplus ( R_T = 0.6 )</td>
<td>Total surplus ( R_T = 0.8 )</td>
</tr>
</tbody>
</table>

The other commonly used pricing strategy is Cost Per Click (CPC). Both CPI and CPC mechanisms generate the same revenues for the platform and advertisers under different targeting strategies.

Further assume that the advertisers are symmetric (i.e., their valuation of a click is the same, and hereafter normalized to 1). Equation (6) shows the match values between the advertisers and users, which can also be interpreted as the eCTR of an impression for the advertiser-user pair. Notice that advertiser \( a_1 \) has a better match with user \( u_1 \) and advertiser \( a_2 \) with user \( u_2 \).

We now consider the advertiser’s bidding strategy and outcomes under two regimes – 1) No data disclosure by the platform, and 2) Full disclosure of match values by the platform. The results

\[
eCTR = \begin{pmatrix} 0.5 & 0.1 \\ 0.1 & 0.3 \end{pmatrix} \rightarrow \begin{pmatrix} 0.3 \\ 0.2 \end{pmatrix}
\]

Further assume that the advertisers are symmetric (i.e., their valuation of a click is the same, and hereafter normalized to 1). Equation (6) shows the match values between the advertisers and users, which can also be interpreted as the eCTR of an impression for the advertiser-user pair. Notice that advertiser \( a_1 \) has a better match with user \( u_1 \) and advertiser \( a_2 \) with user \( u_2 \).

We now consider the advertiser’s bidding strategy and outcomes under two regimes – 1) No data disclosure by the platform, and 2) Full disclosure of match values by the platform. The results

\[10\text{It is worth noting that there are other deterministic auction mechanisms, such as first-price auction, which are revenue equivalent to second-price auction. Thus, for the sake of simplicity, we focus our attention on the second-price auction in our analysis, but the results are generalizable to other auctions as well.}\]
from these two scenarios are laid out in Table 7 and discussed below:

- No data disclosure – Here advertisers do not have information about their match with each user and can only estimate the aggregate match value over both users. Then, A and B’s expected match over the two users are $0.3$ and $0.2$, respectively as shown in Equation (6). Because this is a second price auction, advertisers simply bid their expected valuations. Thus, A wins both impressions and pays the next highest bid, $b_{21} = b_{22} = 0.2$ for each impression. Therefore, the total revenue of platform is $R_P = 0.4$, and that of the advertisers is $R_{a1} = 0.2$ and $R_{a2} = 0$, and the total surplus is $R_T = 0.6$.

- Full data disclosure – Since advertisers now have information on their match for each impression, they submit “targeted bids” that reflect their valuations as shown in Table 7. Therefore, the advertiser who values the impression more wins it. However, because of the asymmetry in advertisers’ valuation over impressions, the competition over each impression is softer. This ensures higher advertiser revenues, with $R_{a1} = 0.4$ and $R_{a2} = 0.2$. However, the platform’s revenue is now lower, with $R_P = 0.2$. Thus, even though ads are matched more efficiently and the total surplus generated in the system is higher, the platform extracts less revenue compared to the case of imperfect targeting.

This example illustrates the trade-off between value creation and value appropriation for the platform, and highlights why fine-grained targeting may not always be aligned with the platform’s incentives.

### 7.3.3 Measures of Competition

To empirically examine how data disclosure and targeting affect competition, we first need to characterize competition under different targeting strategies. Intuitively, if data disclosure reveals significant heterogeneity among advertisers’ valuations of impressions, then advertisers may differentiate themselves by targeting their bids, which in turn would soften competition.

As highlighted in §7.3.2, the distribution of match values or the eCTRs of advertisers within an impression is informative of the extent of competition. We therefore analyze the nature of the competition through measures based on this variable. Let $\mathbf{m}_\tau$ as the vector of match values for advertisers for a targeting strategy $\tau$ for a given impression.

$$\mathbf{m}_\tau = (m_{\tau 1}, m_{\tau 2}, ..., m_{\tau A}),$$

where $m_{\tau a}$ indicates the match value for ad $a$ in targeting area $\tau$. Without loss of generality, we assume that $m_{\tau 1} \geq m_{\tau 2} \geq \cdots \geq m_{\tau A}$. While there are different interpretations of the competition based on the distribution of $\mathbf{m}_\tau$, we focus on two main metrics:

1. $\sigma^2_\tau$, the variance in the match values. Higher variance in the distribution of $\mathbf{m}_\tau$ implies that
Figure 9: Examples of different types of competitive scenarios

competitors are more differentiated, implying softer competition. In contrast, if advertisers’ match values are close, then they will all compete equally for the impression and the competition is more intense. However, if the variance of $m_{\tau}$ is high,

2. Our second measure is $\rho_{\tau} = \frac{m_{\tau 1}}{m_{\tau 2}}$, the ratio of the highest to the second highest match value for the impression for an impression targeting area $\tau$. This measure is important in a competitive context, because a sufficiently large ratio would indicate that the highest would win easily, since the rest would not want to compete. Further, this measure is important from a theoretical viewpoint, because it reflects the gap between the highest and second highest valuations in an auction, which a well-known source of inefficiency in auctions with deterministic winning rule (Milgrom and Weber [1982]).
In Figure 9 we present the distributions of match values to illustrate different types of competition for four example impressions. In each sub-figure, we sort all the match values for that specific impression in decreasing order and distinguish the top two with a different shape and color. Figure 9a is a scenario where the match values for all the advertisers are quite close, and both of our metrics are low here. This represents a highly competitive scenario that is stable. Even if one of the advertisers were not to bid, the overall scenario would not change much. In Figure 9b, the match values are segregated into different levels, implying a high variance, but the ratio of the top two bids is still small. In Figure 9c, the ratio of the top two bids is high, implying soft competition between the top two advertisers. However, if the topmost advertiser drops out, the rest are concentrated and the variance is relatively low, implying that soft competition is not so stable here. Finally, in Figure 9d, both metrics are relatively high. This case represents a stable low competition scenario.

7.3.4 A Stylistic Model of Targeting and Competition

While the model-free measures discussed above can give us some insight into the extent of competition in the marketplace, we now specify a stylistic and simple model to capture the relationship between targeting, platform revenues, and social surplus for a given targeting strategy. As before, we consider a second-price auction with CPI pricing.

Consider a platform that receives $I$ impressions and serves $A$ advertisers. Let $m_{ia}$ denote the match value (or eCTR) of ad $a$ for impression $i$. For simplicity, we assume that advertisers are symmetric in their valuation for a click (henceforth normalized to 1). Thus, the expected value of an impression for an advertiser under perfect targeting is simply its match value of that impression. Now denote match value matrix $M$ as:

$$M = \begin{bmatrix}
m_{11} & m_{12} & \ldots & m_{1A} \\
m_{21} & m_{22} & \ldots & m_{2A} \\
\vdots & \vdots & & \vdots \\
m_{I1} & m_{I2} & \ldots & m_{IA}
\end{bmatrix}$$

(8)

If the platform reveals $M$ to advertisers, they place targeted bids for each impression $i$ equal to $b_{ia} = m_{ia}$. Each impression $i$ is then sold to the advertiser with the highest match value (or bid) which is given by $\max_a m_{ia}$. The total surplus generated by the platform is:

$$S_P = \sum_{i=1}^{I} \max_a m_{ia}$$

(9)

In contrast, if the platform conceals information, and only provides advertisers their aggregate
match values over all impressions \( \bar{m}_a = \frac{1}{I} \sum_{i=1}^I m_{ia} \), then all advertisers place non-targeted bids over all impressions where \( b_a = \bar{m}_a \). In this case, the advertiser with highest match value aggregated over all impressions wins all impressions and the total surplus generated by the platform is:

\[
S_0 = \max_a \sum_{i=1}^I m_{ia}
\]

(10)

We can easily show that \( S_P \geq S_0 \), indicating that perfect targeting will increase the total surplus generated in the market. However, it is not clear how total surplus is distributed between advertisers and the ad-network\(^\text{11}\). Under no targeting, ad-network’s revenue for each impression is the same, equals to the second highest match value aggregated over all impressions. Hence, the winner pays

\[
a_0^{(2)} = \max_{a \backslash a_0^{(1)}} \frac{1}{I} \sum_{i=1}^I m_{ia}
\]

per impressions, meaning that the total revenue will be:

\[
R_0 = \max_{a \backslash a_0^{(1)}} \sum_{i=1}^I m_{ia}
\]

(11)

For the case of perfect targeting, however, the payment differs per each impression. For impression \( i \), the highest match value pays the second highest match value, which equals to

\[
a_p^{(2)} = \arg\max_{a \backslash a_p^{(1)}} m_{ia}
\]

This results in the following revenue:

\[
R_P = \sum_{i=1}^I \max_{a \backslash a_p^{(1)}} m_{ia}
\]

While we can theoretically show that \( S_P \geq S_0 \), comparing \( R_P \) and \( R_0 \) is more of an empirical question. In fact, the inequality could be in both directions, if we do not impose any assumption on the distribution of match values.

### 7.3.5 Empirical Analysis of Competition and Micro-Targeting

To apply this theoretical model to data, we first need to generate the match-value matrix \( M \) for a large sample of impressions \( I \). We do that as follows. First, we draw a sample of 20000 users from the test data and identify all the impressions generated by these users in the test data, which gives us 391069 impressions. Note that in the data, only one of the advertisers was actually shown to the user. However, using our model, we can generate predictions of eCTRs (or match values) for all the advertisers. To the extent that ads have been sufficiently randomized across users, these counterfactual eCTR predictions should be realistic.

Once we have the empirical \( \hat{M} \) matrix for these \( I = 391069 \) impressions, we can generate the

\(^{11}\)Please note that 70% of the advertiser’s payment goes to the publisher and the remaining 30% goes to the ad-network.
<table>
<thead>
<tr>
<th>Targeting Level</th>
<th>Average Ratio</th>
<th>Max Ratio</th>
<th>Avg. Variance</th>
<th>Surplus</th>
<th>Revenue</th>
<th># Targeting Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impression</td>
<td>1.212</td>
<td>58.87</td>
<td>8.08 × 10⁻⁵</td>
<td>0.01426</td>
<td>0.01178</td>
<td>391069</td>
</tr>
<tr>
<td>User</td>
<td>1.209</td>
<td>14.95</td>
<td>3.63 × 10⁻⁵</td>
<td>0.01399</td>
<td>0.01189</td>
<td>20000</td>
</tr>
<tr>
<td>App</td>
<td>1.116</td>
<td>13.99</td>
<td>7.78 × 10⁻⁶</td>
<td>0.0114</td>
<td>0.01072</td>
<td>49</td>
</tr>
<tr>
<td>Time</td>
<td>1.123</td>
<td>5.47</td>
<td>6.18 × 10⁻⁶</td>
<td>0.01132</td>
<td>0.01029</td>
<td>24</td>
</tr>
<tr>
<td>No Targeting</td>
<td>1.114</td>
<td>4.61</td>
<td>5.88 × 10⁻⁶</td>
<td>0.01112</td>
<td>0.01025</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 8: Measures of competition, platform revenues, and total surplus for different levels of targeting.

revenue and surplus for different targeting strategies. For the purpose of this exercise, we consider five targeting scenarios that the platform can offer:

- **No targeting** – Here advertisers only know their expected match value over all impressions. This is the strategy currently implemented by the platform.
- **Time-based targeting** – In this case, the advertiser sees her expected match value for all impressions at a given point in time. Since we partitioned time into 24 discrete hours, this boils down to 24 targeting areas from an advertiser’s perspective.
- **App-based targeting** – Here, the platform aggregates an advertiser’s match value over all the impressions within a specific app, and gives her an app-specific eCTR. Since we have 49 apps, the advertiser can choose targeted bids for each specific app.
- **User-based targeting** – In this case, the platform gives the advertiser her expected CTR for each user. For the sample considered here (20,000), the advertiser can set user-specific bids for each of these users.
- **Impression-based targeting** – This is the most granular level of targeting. In this case.

**Measures of Competition:** We first discuss the empirical distributions of the two measures of competition described in §7.3.3 – the ratio between the highest and second highest match values and the average variance of match values, within an impression. The results are shown in Columns 4-7 of Figure 8. As we conjectured, the mean of ratio measure \( \rho_r \) across all impressions (Column 4) exhibits an increasing trend with the extent of targeting. The maximum ratio across all impressions is also higher with more targeting (Column 5). The variance of match values within an impression (\( \sigma^2 \)) also exhibits a similar pattern, with the average variance increasing with more targeting. Together, these results suggest that as we increase targeting, the competition between advertisers become softer.

**Platform Revenue and Social Surplus:** Next, in Columns 2-3 we present the results on social surplus and revenue (which are shown per impression). As our theory model predicts, the social surplus is increasing with the extent of targeting. In our setting, going from the current scenario
(of no targeting) to one with full impression-level targeting can increase the social surplus by 28% which is significant. However, note that platform revenues exhibit more of an inverted U-shaped curve. The platform revenues are maximized when it does not allow targeting at the impression level, but instead restricts it to users. This suggests that the platform has an incentive to withhold impression-specific click (and hence targeting) data from advertisers. Thus, from the platform’s perspective, complete data-sharing and micro-targeting are not optimal. Therefore, the advertisers’ and the platform’s incentives are not perfectly aligned in this respect. Indeed, the platform’s incentives to restrict detailed data-sharing is somewhat aligned with that of consumers, who demand privacy.

In sum, our analyses is the first to present empirical evidence to support the the theoretical conjectures on the incentives of platforms to limit targeting and data sharing. Our findings provide some support to the advertising industry’s claim that external regulation may not be necessary to reduce tracking and targeting, and that the industry can self-regulate.

8 Conclusions

Mobile in-app advertising is now a dominant form of advertising. We propose a machine learning framework to predict targeting effectiveness for in-app ads. Our framework consists of two modules – functions for feature generation and MART-based predictive model. We apply it to a large-scale data with over 150 million impressions from the leading in-app ad-network in a large Asian country. Substantively, we quantify the improvement in targeting effectiveness to be 15.2% over the baseline. We also show that behavioral targeting with user-information is more effective than contextual targeting based on the where-when-which of the impression. Methodologically, we show that MART significantly out-performs econometric models. We use our model to address three policy questions on privacy. First, we show that stronger privacy regulations on user tracking will lead to a approx. 5% loss in targeting effectiveness. We then consider weaker privacy regulations where the ad-network is allowed to share data with advertisers. We find that advertisers can achieve close to the first-best targeting effectiveness without completely compromising privacy. Importantly, we show that data-sharing does not lead to data democratization – all sharing arrangements benefit large advertisers more than small advertisers. Finally, we examine whether the platform has incentives to share data with advertisers. We show that data-sharing leads to targeted bidding by advertisers and thereby softens competition, and that the platform will benefit from limiting targeting and data-sharing. Thus, by design, the ad-network may be incentivized to preserve users’ privacy to some extent.
References

A. Edwards-Levy and D. Liebelson. Even Trump Voters Hate This Bill He Just Signed, 2017. URL http://www.huffingtonpost.com/entry/trump-online-privacy-poll_us_58e295e7e4b0f4a923b0d94a
S. K. Hui, J. J. Inman, Y. Huang, and J. Suher. The effect of in-store travel distance on unplanned spending:


N. S. Sahni. Effect of temporal spacing between advertising exposures: evidence from online field experiments.


E. B. Seufert. IDFA Zeroing is the Massive Change to Mobile Advertising That No One is Talking About, 2016. URL http://mobiledevmemo.com/idfa-zeroing-massive-change/


