Information Systems Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Matching Mobile Applications for Cross-Promotion

Gene Moo Lee, Shu He, Joowon Lee, Andrew B. Whinston

informs
http://pubsonline.informs.org/journal/isre

INFORMATION SYSTEMS RESEARCH
Vol. 31, No. 3, September 2020, pp. 865–891
ISSN 1047-7047 (print), ISSN 1526-5536 (online)

Matching Mobile Applications for Cross-Promotion

Gene Moo Lee,[a] Shu He,[b] Joowon Lee,[c] Andrew B. Whinston[d]

[a] Sauder School of Business, University of British Columbia, Vancouver, British Columbia V6T 1Z2, Canada; [b] Department of Operations and Information Management, University of Connecticut, Storrs, Connecticut 06269; [c] College of Social Science, Hansung University, Seoul 136-172, Korea; [d] Department of Information, Risks, and Operations Management, McCombs School of Business, The University of Texas at Austin, Austin, Texas 78712

Contact: gene.lee@sauder.ubc.ca, https://orcid.org/0000-0003-0657-6898 (GML); shu.he@uconn.edu, https://orcid.org/0000-0001-5899-5299 (SH); jwlee1015@gmail.com (JL); abw@uts.cc.utexas.edu, https://orcid.org/0000-0002-7371-8991 (ABW)

Received: July 31, 2017
Revised: December 21, 2018; October 16, 2019
Accepted: December 15, 2019
Published Online in Articles in Advance: August 26, 2020

https://doi.org/10.1287/isre.2020.0921

Abstract. The mobile applications (apps) market is one of the most successful software markets. As the platform grows rapidly, with millions of apps and billions of users, search costs are increasing tremendously. The challenge is how app developers can target the right users with their apps and how consumers can find the apps that fit their needs. Cross-promotion, advertising a mobile app (target app) in another app (source app), is introduced as a new app-promotion framework to alleviate the issue of search costs. In this paper, we model source app user behaviors (downloads and postdownload usages) with respect to different target apps in cross-promotion campaigns. We construct a novel app similarity measure using latent Dirichlet allocation topic modeling on apps' production descriptions and then analyze how the similarity between the source and target apps influences users' app download and usage decisions. To estimate the model, we use a unique data set from a large-scale random matching experiment conducted by a major mobile advertising company in Korea. The empirical results show that consumers prefer more diversified apps when they are making download decisions compared with their usage decisions, which is supported by the psychology literature on people's variety-seeking behavior. Lastly, we propose an app-matching system based on machine-learning models (on app download and usage prediction) and generalized deferred acceptance algorithms. The simulation results show that app analytics capability is essential in building accurate prediction models and in increasing ad effectiveness of cross-promotion campaigns and that, at the expense of privacy, individual user data can further improve the matching performance. This paper has implications on the trade-off between utility and privacy in the growing mobile economy.

History: Alessandro Acquisti, Senior Editor; Pei-Yu Chen, Associate Editor.
Supplemental Material: The online appendices are available at https://doi.org/10.1287/isre.2020.0921.

Keywords: mobile applications • cross-promotion • matching • search cost • two-sided platform • topic modeling • machine learning • deferred acceptance • algorithm • mobile analytics

1. Introduction

The mobile applications (apps) market is one of the most successful software markets in recent years (Petsas et al. 2013, Bresnahan and Greenstein 2014, Yin et al. 2014). Mobile platforms, such as iOS and Android, allow third-party developers to introduce their apps into the market. Millions of mobile apps are available in app markets, such as Apple's App Store and Google's Play Store.[1] Billions of people have adopted smartphones and tablets as their main internet devices.[2] As a two-sided platform, the mobile app market connects app developers and users, which helps reduce search costs (Rochet and Tirole 2003, Hagiu 2006). To alleviate the information asymmetry, app markets provide in-house rankings, consumer reviews and ratings, and keyword search functionality.

Despite the existing features, the two-sided mobile app market with large numbers of players on both sides can face increased search costs in the absence of an efficient market design (Evans et al. 2011). In fact, measurement studies on mobile app markets found evidence of a "winners-take-all" phenomenon (Petsas et al. 2013, Zhong and Michahelles 2013). An industry report indicates that 54% of the total app store revenue goes to only 2% of the developers.[3] This is in sharp contrast to other online platforms, such as video streaming (Anderson 2006), auctions (Hu and Bolivar 2008), retail (Linden et al. 2003), and books (Oestreicher-Singer and Sundararajan 2012), which exhibit long-tail demand curves. In a long-tail market, a large number of less popular products constitute a significant portion of the total market share (Anderson 2006, Brynjolfsson et al. 2010).

We argue that a lack of efficient market design and subsequent increased search costs can be the cause of demand concentration in the mobile app market (Evans et al. 2011). App developers face challenges to spotlight their new apps in the presence of millions of other apps. Currently, app visibility heavily relies on app markets' in-house ranking systems (Carare 2012, Ifrach and Johari 2014), which list the most popular and fastest-growing apps in each app genre. Without a proper search mechanism, consumers are mainly exposed to the top-ranked apps, which cover only a small fraction of the entire market. The dependency on rankings can further reinforce the demand concentration. Hence, developers are actively exploring various user acquisition channels, including mobile display ads (MDAs; Qiu et al. 2017), purchased downloads (Li et al. 2016), price promotions (Askalidis 2016, Li et al. 2017), and search ads.[4]

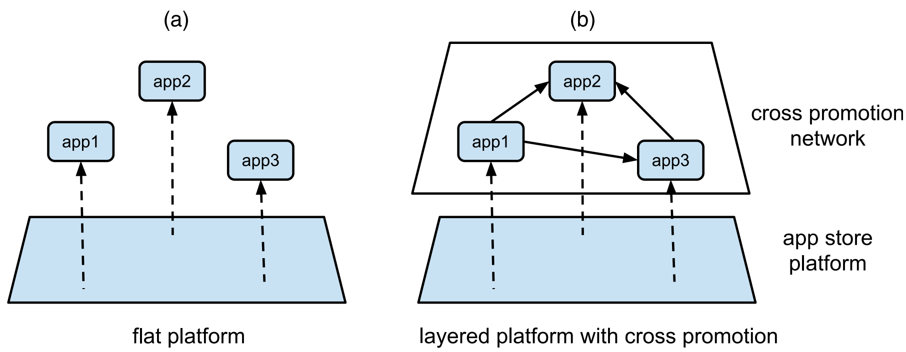
Cross-promotion (CP) is introduced as a new form of targeted advertising in the mobile app market. In a CP campaign, an app (which we term *target app*) is exposed to potential users who are already using another app (which we term *source app*). Essentially, consumers can be targeted based on their app usage behaviors (i.e., the source app they are using). CP campaigns incentivize users to install and use the target apps by providing in-app rewards (e.g., free game items) that can be used in the source apps. Both source and target app developers are motivated to participate in CP campaigns for the following reasons: (1) because CP is a case of in-app advertising, source app developers can monetize their app visibility,[5] and (2) target app developers can reach the right users by selecting the right source apps as CP campaign partners. Conceptually, a CP network adds an extra layer of communication between app developers on top of the existing app platform, as illustrated by Figure 1. With effective app matching, a CP network can help users find adequate apps and provide developers with an effective user acquisition channel, thus reducing search costs in the market.

In fact, CP has been successfully implemented in the industry. For example, Tapjoy[6] has attracted over 10,000 apps with more than 520 million monthly users and now has been integrated with Google AdMob.[7] According to Tapjoy's report, the CP network can significantly increase user retention, user sessions, and in-app purchases.[8] Social media apps, such as Facebook[9] and Twitter,[10] which can be regarded as source apps in a broader definition of CP, have introduced similar app-promotion mechanisms. Despite the continued growth of the CP market, the mechanism and effectiveness of this promotion are not well understood in the academic research and industry compared with other app marketing strategies.[11]

The objective of this paper is to systematically understand the CP app-promotion framework by analyzing data collected from a large-scale experiment conducted in a mobile app ad platform in Korea. This platform shares a similar design with Tapjoy in that source app users can earn rewards by downloading corresponding target apps. In the experiment, 41,294 CP campaigns were conducted on 215 mobile apps (including 127 source and 128 target apps) from November 2013 to May 2014. The campaigns involved approximately 400,000 users whose detailed behaviors (downloads and postdownload usages) were tracked by an app analytics tool. One unique feature of this experiment is that source and target apps were *randomly* matched, enabling us to explore consumers' preferences toward different app matches and to design an app-matching platform to improve the CP ad effectiveness.

To identify the determinants of CP ad effectiveness, we apply multiple econometric models (e.g., linear regression, tobit model, hurdle model, and two-stage model) to analyze source app users' decisions regarding target app download and postdownload usage with respect to different source–target app matches. Our models incorporate a novel app similarity measure using topic modeling (Blei et al. 2003) while controlling for the source and target apps'

Figure 1. (Color online) Conceptual Illustration of Cross-Promotion Network and App Store Platform



characteristics (e.g., app functional diversity, developer's experience, file size, ranking, etc.). The empirical results show that source–target matching plays a pivotal role in CP ad effectiveness: user retention rates and their postdownload usage are high in well-matched CP campaigns. The results also show that although people prefer to download target apps that are different from source apps (variety seeking), they spend more time on target apps that are similar to source apps (consistency seeking). This can be evidence of people's two opposite behaviors that are well documented in the marketing and behavioral economics literature (Simonson 1990, Oliver 1999, Adomavicius and Tuzhilin 2005, Johnson et al. 2006, Adomavicius et al. 2015). In addition, we empirically evaluate the effects of CP participation on source and target apps' performance by difference-in-differences (DID) models. The results show that CP campaigns can significantly improve target apps' performance, especially for initially lower ranked apps. In other words, the CP can help mitigate the demand concentration issue in the app market.

Based on insights from the empirical results, we propose an app-matching system based on prediction models (on app downloads and usages) and matching algorithms. The current industry practice of source–target app matching is mostly based on the public information available in the app market (e.g., app genres, ranks, reviews), and we argue that the resulting matchings can be suboptimal.[12] Our machine-learning models use different levels of information (e.g., public features, analytics features, and individual user features) to predict the ad effectiveness for possible app matches. The cross-validation results show that an accurate download prediction model can be constructed with privacy-preserving app features, whereas individual user features are necessary for accurate postdownload usage prediction. Using the prediction models, we simulate different app-matching algorithms, including a generalized deferred acceptance algorithm (Gale and Shapley 1962, Roth 1984, Roth 2008). The matching simulation results show that app analytics features (Han et al. 2016) are essential in improving the CP ad effectiveness and that, at the expense of privacy, the use of individual user data can induce further improvements (Rafieian and Yoganarasimhan 2018b).

Our work has important implications for both academia and industry. To the best of our knowledge, this is the first empirical study on the CP framework. In that, this research contributes to the growing literature on the mobile app market, mobile advertising, and multiproduct promotion. Our empirical results show the existence of consumers' variety-seeking behavior discrepancy with digital products. Furthermore, the proposed app-matching platform has practical implications by improving CP ad effectiveness, which can potentially strengthen the vitality of the mobile app ecosystem.

The remainder of this paper is organized as follows. In Section 2, we review the literature and discuss our contributions. We describe the data and variable construction in Sections 3 and 4, respectively. In Sections 5 and 6, we empirically explore consumers' preferences on apps using multiple econometric models and study the overall impact of CP on participating apps and the whole app market. In Section 7, we design an app-matching platform for the CP network using machine-learning models and matching algorithms. Section 8 concludes the paper with directions for future research.

2. Literature Review

In this section, we review related studies in the information systems, marketing, and economics literature. Specifically, our work is related to research strands on the mobile app market, multiproduct promotion, and consistency- and variety-seeking behavior.

2.1. Mobile App Market and Mobile Advertising

With its rapid growth, the mobile app market has attracted increasing attention from the information systems and economics literature. One strand of research is to investigate the effect of app rankings and app characteristics on apps' demand. Using the daily top 100 apps data in Apple's App Store, Carare (2012) and Ifrach and Johari (2014) investigated the causal relation between best-seller rank information and app demand. Other characteristics, such as rating, file size, age, version, app genre, in-app purchase, and in-app advertising, are also shown to influence app's demand (Ghose and Han 2014, Lee and Raghu 2014).

With a larger number of apps in the market, the market experiences a huge search cost (Levin 2011, Ershov 2016). In such a crowded market, app developers have adopted different advertising strategies: mobile display ads (Qiu et al. 2017), purchased downloads (Li et al. 2016), price promotions (Askalidis 2016, Li et al. 2017), and search ads.[13] CP is a relatively new app-promotion strategy in the mobile app market, which has not been empirically investigated by researchers. The only exception is an analytical paper (Guo et al. 2019) that analyzes when app developers should offer reward advertising rather than direct selling of premium content. Compared with other promotion mechanisms, CP achieves a significantly large number of downloads because of the incentives provided to the source app users who download the target app. To a certain extent, CP and the purchased download share common characteristics in that both adopt an incentive mechanism, but a major difference is that in the CP framework, both the promoting and the promoted parties are mobile apps.

Mobile apps have also become one of the fastest-growing ad platforms.[14] There is a growing literature on mobile targeting, considering various factors, such as location, time, weather, trajectory, and product characteristics (Ghose et al. 2012, 2019; Luo et al. 2013; Bart et al. 2014; Molitor et al. 2015). More recently, Rafieian and Yoganarasimhan (2018b) investigate how incorporating customers' behavioral features can significantly improve the display ad effectiveness with machine-learning methods. In another paper of Rafieian and Yoganarasimhan (2018a), they also explore the impact of previous ads' diversity on the consumer's response to the current ad. Our paper contributes to this strand of literature by empirically evaluating how the CP ad effectiveness changes when different levels of information (from public data to app- and individual-level usage data) is used in mobile app targeting.

2.2. Multiproduct Promotion

A CP campaign involves multiple products (source and target apps). Thus, our paper is related to the literature on multiproduct promotion mechanisms, such as cross-selling, product bundling, and recommender systems. Table 1 summarizes representative studies of multiproduct promotions and articulates the unique aspects (incentives and sequential consumptions) of CP compared with other multiproduct promotions.

Cross-selling is the practice of selling related products or services to existing consumers, which is a well-studied strategy in the marketing literature (Kamakura 2008). Existing studies have focused on how to improve the effectiveness of cross-selling campaigns by accurately predicting the purchase probability of new products using customer information (Kamakura et al. 2004; Li et al. 2005, 2011). In predicting the new-product adoption, these papers use consumer demographics as well as purchase history and product usage patterns. Similarly, our paper uses app download and postdownload usage data for predictions. However, two main differences are apparent between these works and ours. First, cross-selling is mainly conducted among products within a *single* company, whereas CP allows multiple app publishers to promote their apps to others' customers. As a result, incentive compatibility is a critical issue in CP matching, whereas one only needs to consider the focal firm's profit in conducting cross-selling campaigns. The second difference is the number of products involved in the promotions. Most cross-selling studies involve a handful of products, and it is easy to recognize cross-product relationships. But hundreds of thousands of apps are seen in mobile app markets. To address the scalability issue in matching a larger number of apps, we build a text-mining approach to quantify app similarity based on textual descriptions.

Product bundling is the sale of two or more separate products or services in one package (Stremersch and Tellis 2002). Prior studies focused on companies' optimality of bundling and bundling price (Venkatesh and Mahajan 2009, Bhargava 2012). In addition, other papers consider how product bundling can be a strategy for new-product introduction or market penetration (Stremersch and Tellis 2002, Lee and O'Connor 2003, Nalebuff 2004). Related to our paper, Soman and Gourville (2001) analyze consumers' behaviors after their product bundling purchase, showing that they are less likely to consume the purchased products. CP is different from product bundling in two aspects: (1) consumers do not install two products (source and target apps) at the same time, and (2) the benefit to the consumer is the in-app incentives that can be used in the source app rather than total price discount.

CP also shares some characteristics with recommender systems, which is a widely adopted strategy

Table 1. Literature Review on Multiproduct Promotion

Research strands	Examples	If incentivized	If sequential	Representative papers
Cross-selling	Smartphone + insurance, credit card + instant loans	No	Yes	Kamakura et al. (2004) Li et al. (2005) Kamakura (2008) Li et al. (2011)
Product bundling	Travel package, PC installed with Windows Operating System	Yes	No	Soman and Gourville (2001) Stremersch and Tellis (2002) Lee and O'Connor (2003) Nalebuff (2004) Venkatesh and Mahajan (2009) Bhargava (2012)
Recommender system	Netflix, Amazon	No	Yes or no	Resnick and Varian (1997) Burke (2002) Adomavicius and Tuzhilin (2005) Schafer et al. (2007) Lops et al. (2011)
Cross-promotion	Mobile app market	Yes	Yes	Guo et al. (2019) Our paper

in online platforms. There are abundant studies developing new algorithms to provide customized recommendations to help customers find the most satisfying products (Resnick and Varian 1997, Adomavicius and Tuzhilin 2005). There are three main approaches in recommender systems: usage-based collaborative filtering (Schafer et al. 2007), content-based filtering (Lops et al. 2011), and hybrid recommender systems (Burke 2002). Our proposed CP matching framework adopts the ideas of using the product content (e.g., app descriptions) and user behavior (e.g., app download and usage patterns) to produce app matches. However, CP framework is distinct in that there is contract between app developers, whereas no such contract exists among recommended products in recommender systems.

2.3. Consistency- and Variety-Seeking Behavior

Understanding people's decisions in choosing multiple products helps us improve ad effectiveness. Previous studies in the marketing and behavioral economics literature found abundant evidence of people's two opposite behaviors: consistency and variety seeking. On the one hand, marketing studies show that consumers have loyal behaviors toward products and companies and exhibit stable preferences (Oliver 1999, Johnson et al. 2006, Fong 2017). For example, studies show that people usually stick to a certain taste in selecting products in online shopping (Linden et al. 2003), music streaming (Hariri et al. 2012), and mobile app usage (Natarajan et al. 2013). This stable preference is the basis for recommender systems and the proposed CP framework, in which new products are recommended based on consumers' previous purchases (Adomavicius and Tuzhilin 2005).

Another stream of studies, on the other hand, suggests that people are motivated to choose variety in product choice (Simonson and Winer 1992, Read and Loewenstein 1995). McAlister (1982) explains multiple reasons for the variety-seeking behavior, including managing satiation, which infers that consumers' utility from the consumption of similar items declines to meet internal needs, and being motivated by external reasons. Researchers conducted laboratory experiments to find that people prefer greater variety when making sequential decisions of individual items over time (Simonson and Winer 1992, Read and Loewenstein 1995). More recently, Adomavicius and Tuzhilin (2005) extended the research domain on consumers' variety-seeking behaviors from consumable goods to digital goods, showing that the difference in variety-seeking preferences between bundled and unbundled consumptions is eliminated in the digital market.

Mobile apps can be regarded as durable goods because they are used over time. We contribute to this consumer behavior literature by providing empirical evidence of the existence of consistency- or variety-seeking behaviors in the mobile app context.

3. Data

3.1. Data Description

We use a secondary data set provided by IGAWorks, a leading mobile ad platform in Korea. The firm's product line includes a mobile app analytics tool called Adbrix and an app monetization platform supporting various promotions, such as MDAs and CPs. It has the largest mobile ad network in Korea, with thousands of mobile apps and more than 2.4 million users.

The data consist of three components: app metadata, funnel data, and usage data. First, the app metadata include descriptive information about 383,896 mobile apps in three major app markets in Korea: Apple's App Store, Google's Play Store, and SK Telecom's T Store.[15] Each app record contains the app's name, text description, screenshots, developer, registration time, price, average rate, and file size, which is publicly available information in the app markets. Because app ranking is an important factor in visibility (Carare 2012, Ifrach and Johari 2014), we collected daily rankings from the three aforementioned app marketplaces. The ranking data cover 250,422 apps (both free and paid) in 51 different app genres.[16]

Next, the funnel data provide information on mobile app ad promotions, which include CPs and MDAs, that the mobile ad platform has executed with its clients (app developers) from November 2013 to May 2014. Specifically, for the CPs, the data cover 41,294 campaigns involving 127 source apps (the media in which the ads are placed), 128 target apps (apps to promote), and about 400,000 users. The funnel data keep track of the user acquisition channels: CP, MDA, or organic growth.[17] For each user acquired through a CP campaign, the associated source app is recorded in the data.

Lastly, the usage data include the detailed app usage records of 1.1 million users with 679 apps that adopted the Adbrix analytics tool.[18] The individual user-level data include daily app session times (i.e., how long a consumer uses an app in a given day), daily connection counts (i.e., how many times a consumer executes an app in a given day), and daily buy activities (i.e., the number of in-app purchases made by a consumer in a given day).

3.2. User Acquisition Channels and Ad Effectiveness

To compare different user acquisition channels, we conduct a model-free analysis on the ad effectiveness in terms of download and postdownload usage. From the app usage and funnel data, we collect three groups of users based on the acquisition channels: organic downloads, MDAs, and CPs. Then we count the total number of downloads for each channel and the average number of downloads for each app. In addition, we gather postdownload usage data from consumers of each acquisition channel. An alternative ad effectiveness measure is the conversion rate (i.e., the probability of download given an ad impression). However, because of the absence of ad impression data, we cannot measure this metric.

The results are shown in Table 2. Among the analytics-enabled 679 apps, 137 have conducted MDA campaigns and 128 have participated in CP campaigns as target apps. Although our data source did not share the campaign cost data, Facebook's pricing model of cost per action (which is close to CP) is 10 times more expensive than that of cost per click (which is close to MDA) according to an industry report.[19] Although app developers can select the types of campaigns to conduct, our data show that the characteristics of apps conducting MDA and CP campaigns are comparable (see details in Online Appendix A). Notably, CP campaigns have generated a total of 1.3 million new app installations, whereas MDA campaigns only acquired fewer than a half million. In terms of per-app downloads, a CP campaign, on average, acquired 10,469 downloads, which is about three times as many downloads as an MDA campaign (3,507). Moreover, the number of CP-induced downloads is even larger than that of organic downloads (9,638). Increases in downloads can significantly improve an app's ranking and visibility in the market.

Postdownload user engagement can be measured with various metrics, such as session time, connection counts, and purchase activities. In our analysis, we focus on session time because (1) purchase activities are only available in apps with in-app purchase options, and (2) connection counts and session time are highly correlated. On average, a user who organically downloaded an app uses it for 3.6 hours, but the numbers are 2.411 and 0.203 for MDA and CP, respectively. Clearly, the users acquired by CP campaigns are the least active group, which aligns with concerns about the *free-rider* problem (i.e., a situation in which a user downloads a target app just to obtain the incentives in the source app rather than through genuine intention to use the target app). This can be a critical issue for both the CP platform and participating target apps because the CP campaigns yield a low return on investment in the presence of the free-rider problem.[20] We find consistent results for the other postdownload engagement metric, connection counts.

One of the purposes of this paper is to design an advanced app-matching platform to improve the ad effectiveness of CP campaigns. To test the feasibility, we conduct an in-depth analysis on CP campaigns. As the CP platform creates many-to-many matchings between source and target apps, one target app's ads are placed in multiple source apps. We divide a target app's users according to specific source apps. Then, for each source–target match, we calculate the average postdownload usage levels and identify the 1%, 5%, and 10% best source apps for each target app.

We find that the top 1% matches are 10 times more effective than the average ones and that the top 10% are seven times more effective than the average. The results also show that the top 5% matches outperform the MDA. Based on these observations, we argue that the source–target app matches in CP campaigns are critical to acquire potentially active users. Hence, it is important to understand consumers' preferences on app download and postdownload usage and find determinants of the ad effectiveness of CP campaigns.

4. Variable Construction

In this section, we define our main dependent variables, the ad effectiveness of CP campaigns, and then review various app- and user-level characteristics that can have impacts on the ad effectiveness in different app matches. Notably, some variables can be collected from the public domain (e.g., file size, app genre, price, etc.), whereas others can only be constructed from the app analytics tools (e.g., number of daily active users). Among the variables, subscripts i, j, and t denote source app, target app, and day, respectively. The summary statistics of the main variables are shown in Table 3. The variables are divided into the dependent and independent variables, which are further grouped by public and analytics features by the dependency on the app analytics and machine

Table 2. Ad Effectiveness by User Acquisition Channel

Acquisition channel	No. of apps	Total downloads	Per-app downloads	Total hours	Per-user hours
Overall	679	8,365,289	12,320	25,314,835	3.026
Organic downloads	679	6,544,655	9,638	23,842,404	3.643
Mobile display ads	137	480,504	3,507	1,158,486	2.411
Cross-promotions	128	1,340,130	10,469	313,944	0.203
Cross-promotions (10%)	128	183,706	1,435	52,829	1.451
Cross-promotions (5%)	128	60,151	469	18,173	2.437
Cross-promotions (1%)	128	38,755	303	9,620	2.743

Table 3. Summary Statistics: App-to-App Matching Experiment

Variable	Description	Obs	Mean	SD	Min	Max
Dependent variables: Ad effectiveness						
dn_ijt	No. of downloads of target app *j* for source app *i* on day *t*	41,294	2.613	11.514	0	344
dn_ratio_ijt	Download ratio for target app *j* for source app *i* on day *t*	41,294	0.0117	0.0567	0	1
sess_ijt	Session time on target app *j* of an average user from source app *i* on day *t* (log)	41,294	1.203	3.456	0	19.485
conn_ijt	Connection count on target app *j* of an average user from source app *i* on day *t* (log)	41,294	0.152	0.577	0	8.010
Features from public information (public)						
size_i	File size of source app *i* (megabytes)	41,294	14.509	14.978	1.6	158
size_j	File size of target app *j* (megabytes)	41,294	35.769	44.582	1.7	337
if_rank_it	One if source app *i* is listed on ranking of day *t*, zero otherwise	41,294	0.978	0.147	0	1
rank_it	Daily ranking for source app *i* on day *t* (log)	41,294	3.132	1.706	0	6.526
if_rank_jt	One if target app *j* is listed on ranking of day *t*, zero otherwise	41,294	0.858	0.349	0	1
rank_jt	Daily ranking for target app *j* on day *t* (log)	41,294	1.623	1.513	0	6.265
dev_it	No. of other apps of source app *i* developer before day *t*	41,294	5.121	12.982	0	153
dev_jt	No. of other apps of target app *j* developer before day *t*	41,294	10.518	16.928	0	63
holiday_t	One if day *t* is a national holiday, zero otherwise	41,294	0.092	0.289	0	1
Features from app analytics and machine learning (analytics)						
cross_ijt	No. of days since target *j* started its CP campaign on source *i* as of day *t*	41,294	15.183	23.469	1	166
user_ijt	No. of active users from source app *i* who view target app *j*	41,294	1,871.121	4,548.281	0	94,948
avg_scr_it	Average screen size of source app *i*'s daily active users on day *t* (K pixel)	41,294	1,165.5	195.9	97.5	2,073.6
avg_sess_it	Average session time on source app *i* before day *t* (log)	41,294	5.976	6.687	0	15.934
display_jt	One if target app *j* runs MDA campaign on day *t*, zero otherwise	41,294	0.336	0.472	0	1
topic_ij	App topic similarity based on topic model	41,294	0.270	0.290	0	0.999
topic_sq_ij	Squared value of app topic similarity	41,294	0.157	0.233	0	0.999
entropy_i	Functional diversity of source app *i* based on topic model	41,294	0.972	0.485	0	2.107
entropy_j	Functional diversity of target app *j* based on topic model	41,294	0.882	0.577	0	2.107

Note. Obs, observations; SD, standard deviation.

learning. App genre distributions of source and target apps are shown in Online Appendix A.1.

4.1. Measures of CP Ad Effectiveness

To measure the effectiveness of CP campaigns, we use three dependent variables: (1) the app download numbers (*dn_ijt*) or app installation rate (*dn_ratio_ijt*, i.e., the probability that the target app is installed, given an ad impression),[21] (2) the postdownload total session time (*sess_ijt*), and (3) the postdownload total execution count (*conn_ijt*) on a particular day *t* for an app pair (*i*, *j*). The first variable evaluates users' short-term download decisions, whereas the latter two assess users' long-term postdownload usage decisions. Because the distributions of these variables are highly skewed (see Table 3), we use the log-transformed postdownload dependent variables in the analysis.

The CP platform needs to optimize ad effectiveness considering two different, possibly competing objectives from source and target app developers. Source app developers will seek to maximize the number of app downloads because they receive a cut on each target app download transaction. In the meantime, the objective of target app developers is to maximize both the number of downloads and postdownload usage. More downloads may lead to a higher app rank, which can significantly increase the app's visibility. Greater usage will bring in more in-app purchases, more advertising revenue, and a larger network effect (Ghose and Han 2014, Ifrach and Johari 2014). From the CP platform's perspective, one might want to maximize the number of downloads because of the cost-per-install payment scheme. However, from a long-term perspective, the ad platform may also need to consider postdownload target app usage to attract more target app participations in the future. Considering stakeholders' different perspectives, it is crucial to design a matching market that can resolve the issue of objective alignment.

4.2. App-Level Characteristics

A large body of work (Ghose and Han 2014, Lee and Raghu 2014, Yin et al. 2014) shows that mobile users have preferences over apps with various characteristics, leading to different aggregated market shares. The time-invariant app-level characteristics used in our analysis include functional diversity and app genres for both source and target apps. For time-varying app variables, we have app visibility in the market, app developer's experience, the engagement level and users' device characteristics for the source app, and whether it has another type of campaign for the target app.

Functional diversity is an important feature for an app. A simple way to capture this is the file size, because a more complex app tends to have a larger file size.[22] In that, we measure source and target apps' file sizes in megabytes (MB; *size_i* and *size_j*).[23] We complement the app size feature with the topic diversity based on the description of each app. The idea is that apps with more functions may indicate various topics in their descriptions. Using the latent Dirichlet allocation (LDA) model from Section 4.3, we measure the entropy of topic vectors of source and target apps (*entropy_i* and *entropy_j*).

To control for the visibility of source and target apps, we use each app's daily ranking within each genre (e.g., games, social networking, music) during the CP campaigns. An app's daily ranking is based on various app characteristics, including average app store rating, review volume, download and install counts, uninstalls, and so on. Thus, it can serve as a proxy for the app's popularity and demand. Because some apps are new in the market and have not appeared in the in-house rankings (maximum rank is 1,499), we add two variables to control for the ranking information of each source and target app: whether the app is available on the in-house ranking on day t (*if_rank_it* and *if_rank_jt*) and the log-transformed ranking number if it is among the top 1,499 apps (*rank_it* and *rank_jt*; Garg and Telang 2013).[24]

Another variable that might reflect an app's quality is its developers' prior experience. Presumably, app developers who have already released multiple apps before may have more resources and knowledge on app development and promotion. Hence, we use the number of other apps the developer of each source and target app has developed on day t, which is denoted as *dev_it* and *dev_jt*.[25]

User engagement level in source apps may have impacts on the CP campaign's performance. On the one hand, active user engagement in the source app may have positive effects because those users are incentivized to download target apps to receive rewards for better source app usages. On the other hand, active source app users may be reluctant to shift to the new target app. To capture each source app's user engagement level, we use the log-transformed daily average session time users spent on source app i (*avg_sess_it*) before day t. We do not use activity data on day t to avoid potential endogeneity issues.

We also control for daily active users' average screen size (*avg_scr_it*) of source app i on day t. We include this in our analysis for two reasons. First, mobile devices with a larger screen size are generally more expensive and have more advanced features. Thus, this variable can capture source app users' demographics to a certain degree. Second, the screen size may impact users' app usage pattern. For example, users tend to use more apps if they have larger screens.[26]

Lastly, many app developers conduct simultaneous promotions in multiple channels, such as MDAs. We use a dummy variable indicating whether target app j conducted an MDA campaign on day t (*display_jt*).

4.3. App Pair-Level Characteristics

For the app pair-level characteristics, we measure app similarity between source and target apps, which is time invariant. One may use app genre to do the matching, but this approach can only generate a binary variable. Thus, we collect text descriptions of apps and measure pairwise app similarity using the LDA topic model. For time-varying variables, we also control for the CP campaigns' durations and the number of source app users who viewed a target app in the CP campaign.

As discussed in Section 2.3, consumers exhibit consistency- and variety-seeking behaviors. In this study, we seek to investigate users' trade-offs between diversification and inertia in mobile app choices. In the CP campaign term, we analyze the effect of source–target similarity on app downloads and postdownload usage behaviors.

App developers provide detailed app descriptions in the app market so that potential users can understand the app's features. In this sense, an app's description can be a good representation of its features. Similar descriptions indicate common features of the apps, such as app genre and usage scenarios. To measure the horizontal differentiation of the apps' content, we can leverage various text mining approaches using keywords (e.g., term frequency–inverse document frequency (TF-IDF; Salton and Buckley 1988), topic models (e.g., LDA; Blei et al. 2003), and vector space models (e.g., doc2vec; Le and Mikolov 2014). We decided to use the LDA topic model in the main analysis for the following reasons: (1) the LDA topic model overcomes the data sparsity issue (which TF-IDF suffers from), (2) the resulting topics and their keywords are human interpretable (whereas the dimensions obtained from the doc2vec model are not), and (3) the LDA topic model is widely acceptable in various research contexts, such as scientific articles (Griffiths and Steyvers 2004, Wang and Blei 2011), music (Hariri et al. 2012), social media (Ramage et al. 2010, Weng et al. 2010, Lee et al. 2016, Shin et al. 2020), and business descriptions (Shi et al. 2016). Nonetheless, as robustness checks, we also conducted empirical analyses using alternative text-mining approaches and found consistent results, as shown in Online Appendix E.

We apply the LDA topic model on the app description corpus of 95,956 mobile apps in the Korean market (Blei et al. 2003). As a natural language-processing technique, LDA allows a set of documents to be explained by "hidden topics," which are sets of related keywords. In our context, each app description is a mixture of a small number of app features, and each word in the description is a realization of the app's features. For details on LDA, see Blei (2012). One important hyperparameter to choose in LDA is the number of topics, which can be decided based on quantitative and qualitative criteria. We varied the number of topics from 10 to 200, finding that 100 is the optimal choice in consideration of the trade-off between intertopic similarity and model fit.

The quantitative evaluation in the hyperparameter selection is described in Online Appendix B. For the qualitative evaluation, we report the 100-topic model in Tables 3 and 4 of Online Appendix B. Because the constructed topics are in Korean keywords, they are translated into English. The representative topics in the Korean app market include music (topics 0 and 27), social networks (topics 1, 14, 25, and 41), children (topics 6 and 34), religion (topic 11), games (topics 16 and 27), foreign-language education (topics 19 and 33), e-commerce (topics 18 and 29), and utilities (topics 10 and 13). We believe that the resulting topics provide a reasonable overview of the app market.

Based on the LDA model, app i's description can be represented by a topic vector $\boldsymbol{V_i} = < V_{i,1}, V_{i,2}, \ldots, V_{i,K} >$, where K is the number of topics, $V_{i,k}$ is the nonnegative weight on the kth topic, and the sum of weights is one ($\sum_{k=1}^{K} V_{i,k} = 1$). Given a pair source app i and target app j and their respectively topic vectors $\boldsymbol{V_i}$ and $\boldsymbol{V_j}$, we define app similarity $P(i, j)$ (*topic_ij*) to be the cosine similarity of the two topic vectors, where the resulting values range from zero to one.[27] For extreme cases, $P(i, j) = 0$ if the two apps do not share any common topics and $P(i, j) = 1$ if the two apps share identical topics. To analyze curvilinear relations, we also include the squared term (*topic_sq_ij*) in the analysis.

The durations of CP campaigns may also affect ad effectiveness. For example, if the campaign has executed for a long time, the number of daily installs may diminish over time. On the contrary, multiple exposures to the same ad may lead to target app installation (Xu et al. 2014). Thus, we control for the number of elapsed days since target app j's promotion has been conducted on source app i as of day t (*cross_ijt*). The CP campaign's performance will also depend on the number of source app users who viewed a target app in the CP campaign (*user_ijt*). Because the impression data are not available, we made an assumption that a daily active user of a source app had an equal probability of viewing a specific target app among all target apps in the CP market.

4.4. Time Characteristics

We control for time effects in the empirical analysis. Because people may have more casual time during holidays and weekends, we add categorical variables for each day of the week (*weekday_t*) and for each month (*month_t*) and a dummy variable indicating whether day t is a national holiday for Korea (*holiday_t*) in the model.

5. Empirical Analysis

In this section, we conduct empirical analysis using multiple models to estimate the effect of an app match and the corresponding source–target app similarity on CP ad effectiveness. In our data, a CP campaign is identified as a (source app, target app, day) triple. We use three dependent variables for ad effectiveness of CP campaigns: *download rate*, *postdownload session time*, and *postdownload connection count*. It is essential for us to explore the impact on downloads and postdownload usages together for the following reasons. First, a high number of downloads can help the focal app obtain high visibility (ranking), which will further attract more users and ads. Second, an app will have a long-term growth potential when enough active users are acquired. We first evaluate the impact of app matchings on CP outcomes using each measurement separately. Then we consider the relationship between the download and postdownload usages to understand the comprehensive effect.

5.1. Model Specifications

We first conduct separate analyses for the three dependent variables. Our first model specification is a linear model with source app, target app, and time fixed effects. The main independent variables of interest are *topic_ij* and *topic_sq_ij*, as we explore how the source–target app similarity influences the CP ad effectiveness. We also control for other app and time characteristics (discussed in Section 4). Then we try tobit and hurdle models because of the large portion of zero values in the postdownload usage variable, with app fixed effects or app genre fixed effects based on the models' abilities to converge. Lastly, we developed a two-stage model to incorporate the relationship between user's download and postdownload usage. The two dependent variables are actually tightly connected to each other as a source app user makes sequential decisions: first, whether to download the target app and, second, how much time to spend on the downloaded app. Essentially, a user's postdownload usage behavior depends on the first-step download decision because we cannot observe counterfactual postdownload usage of the users who chose not to download the target app. However, in our collected data, we do not observe which specific source app users were exposed to the target app because of the absence of ad impression data. Thus, we apply an aggregated Heckman's sample-selection model (Heckman 1979) with a pair of apps as the unit of analysis. Readers can find the details of our two-stage model in Online Appendix D. We also control for source and target app genre fixed effects.

5.2. Estimation Results

The estimation results for the download rate, postdownload session time, and postdownload connection count using different models are located in Tables 4, 5, and 6, respectively. The results from our two-stage model are listed in Table 7. For some of the postdownload specifications (e.g., Tables 5 and 6, columns (5) and (6)), we did not include *topic_sq_ij* to avoid a potential multicollinearity issue. We observe an inverted-U relationship between the source–target app topic similarity and the download rate in all specifications, reaching the optimal matching at around 0.4–0.5 (which is around the 70th percentile of the overall topic similarity distribution) for all specifications. This could be the evidence of the variety-seeking behavior in the app download behavior (McAlister 1982, Simonson 1990). In other words, users want to download new apps that are neither identical nor too different from those that they are already using.

Considering the postdownload usage stage, however, we find a strictly positive relationship between source–target app similarity and postdownload usage in all specifications, which can be explained by the consistency-seeking behavior. For some of the specifications (Table 5, columns (1) and (3), and Table 6, columns (1) and (3)), although the coefficient of *topic_ij* is negative, the magnitude is quite small, and it is not statistically significant. Hence, it won't substantially influence the overall trend. With other variables controlled for, the more similar the target app is to the source app, the more usage it may attain from CP campaigns. This indicates that even with the exploration of new app genres, users' intrinsic preferences in app consumption do not change significantly (Oliver 1999, Adomavicius and Tuzhilin 2005, Johnson et al. 2006). A plausible explanation is that users tend to choose more diversified products because of the uncertainty on the target app characteristics at download decision time. But once the target app characteristics are observed by users, consistency-seeking behavior prevails. This phenomenon has been well documented in the marketing literature (Simonson 1990, Simonson and Winer 1992).

To better understand the marginal and overall effects of *topic_ij* on the postdownload usage behavior, in Figure 2, we visualize how the download and usage measures change when app similarity increases by 0.01 at different values of *topic_ij* as well as the overall impact of app similarity at different values.[28] For both panels, the x-axis represents the source–target app similarity (*topic_ij*). In terms of the y-axis, the left panels show the percentage increase or decrease of downloads or postdownload session times when *topic_ij* is increased by 0.01 at each level. And the right panels show the overall impacts. For example, the top left panel shows that the number of downloads will increase by about 0.005% if the app similarity increases from 0.25 to 0.26. An inverse U-shape relationship between app similarity and number of downloads can be inferred from the fact that when the app similarity is larger than 0.50, the percentage change will be smaller than zero in the top-left panel. This inverse U-shape can be directly observed in the top-right panel. The bottom-right panel shows that app similarity has strictly positive effects on postdownload app usage time.

5.3. Other Control Variables

We also find interesting results from some control variables. The coefficients of *display_ijt* are positive and significant for both download and postdownload usage decisions, indicating that display ad campaigns would increase the visibility of the focal target app and motivate new users' download and postdownload decisions.

Source app developers' experience levels (*dev_it*) are found to have negative impacts on CP ad effectiveness. The results show that when source app developers are more experienced and established producers (higher values of *dev_it*), users are less likely to download and use the new target app. A plausible explanation is that because experienced

Table 4. Empirical Results on Download Rate

	Linear		Tobit	Hurdle	
	(1)	(2)	(3)	(4)	(5)
Variables	*dn_ratio_ijt*	*dn_ratio_ijt*	*dn_ratio_ijt*	*dn_ratio_ijt*	selection
topic_ij	1.120***	0.263***	3.718***	0.954***	0.617***
	(0.199)	(0.0668)	(0.695)	(0.162)	(0.0894)
topic_sq_ij	–1.111***		–3.402***	–0.784***	–0.366**
	(0.239)		(0.823)	(0.193)	(0.111)
display_jt	0.936***	0.934***	4.746***	–0.0522	0.213***
	(0.0464)	(0.0464)	(0.198)	(0.0351)	(0.0199)
cross_ijt	0.000855	0.000914	0.0134***	–0.00477***	0.00352***
	(0.000733)	(0.000733)	(0.00231)	(0.000457)	(0.000316)
size_i				–0.0116***	–0.00206***
				(0.00143)	(0.000679)
size_j				–0.000444	–0.000564***
				(0.000327)	(0.000181)
entropy_i				–0.776***	–0.149***
				(0.0295)	(0.0161)
entropy_j				0.0440	–0.162***
				(0.0434)	(0.0200)
dev_i	–0.00328*	–0.00304	–0.0509***	–0.00592	–0.0145***
	(0.00196)	(0.00196)	(0.0127)	(0.00498)	(0.00131)
dev_j	–0.0637***	–0.0640***	–0.151***	0.000328	–0.0141***
	(0.0139)	(0.0139)	(0.0343)	(0.00121)	(0.000691)
avg_sess_it	–0.0102	–0.0110	–0.0453*	–0.0265***	0.0123***
	(0.00884)	(0.00883)	(0.0270)	(0.00240)	(0.00137)
avg_src_it	–1.17e-8	–1.65e-8	–5.16e-8	–1.54e-6***	1.81e-7***
	(6.76e-8)	(6.76e-8)	(3.76e-7)	(6.09e-8)	(2.31e-8)
user_ijt	–4.28e-06	1.55e-07	1.97e-05	–0.000124***	9.78e-06***
	(6.75e-06)	(6.74e-06)	(2.34e-05)	(4.18e-06)	(1.98e-06)
if_rank_it	0.326***	0.304***	1.380***	0.112	0.427***
	(0.123)	(0.123)	(0.428)	(0.117)	(0.0561)
rank_it	–0.0459**	–0.0408*	–0.241***	0.230***	–0.107***
	(0.0209)	(0.0209)	(0.0723)	(0.0138)	(0.00671)
if_rank_jt	1.472***	1.472***	7.127***	–0.458***	1.243***
	(0.0673)	(0.0674)	(0.296)	(0.0740)	(0.0370)
rank_jt	–0.393***	–0.394***	–1.520***	0.0353***	–0.188***
	(0.0182)	(0.0182)	(0.0647)	(0.0136)	(0.00829)
Constant	–9.343***	–9.277***	–14.34***	–1.856***	–1.193***
	(0.203)	(0.202)	(0.831)	(0.204)	(0.114)
App FEs	Yes	Yes	Yes		
App genre FEs				Yes	Yes
Time FEs	Yes	Yes	Yes	Yes	Yes
Observations	41,294	41,294	41,294	41,294	41,294

Notes. This table represents the empirical results showing how source app users' target app download behaviors vary with different source–target app pairs. Variables *topic_ij* and *topic_sq_ij* are the source–target topic similarity and its squared term, respectively. Other control variables include source app *i*'s and target app *j*'s time-varying app characteristics, the day's characteristics, and time-varying dyadic characteristics defined by source app *i* and target app *j*. In columns (1) and (2), the data are analyzed by the panel data fixed effects linear model. In the column (3), the data are analyzed by the tobit model. In columns (4) and (5), the data are analyzed by the hurdle model. FEs, fixed effects.
$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

developers would create higher-quality source apps, their users are less likely to shift to new apps. For the coefficient of *dev_jt*, we find consistent results for the download decision because it is plausible that variety-seeking users are more likely to download and try out creative apps from new developers. The two-stage model results show that established app publishers may have more knowledge on customer retentions.

Considering the effects of ranking information about source and target apps, we controlled for four

Table 5. Empirical Results on Postdownload Session Time

	Linear		Tobit		Hurdle	
	(1)	(2)	(3)	(4)	(5)	(6)
Variables	*sess_ijt*	*sess_ijt*	*sess_ijt*	*sess_ijt*	*sess_ijt*	selection
topic_ij	0.394 (0.260)	0.578*** (0.0874)	−0.228 (1.839)	4.730*** (0.559)	0.927*** (0.139)	0.402*** (0.0358)
topic_sq_ij	0.238 (0.356)		6.099*** (2.121)			
display_jt	0.894*** (0.0603)	0.895*** (0.0604)	3.231*** (0.363)	3.190*** (0.363)	0.366*** (0.102)	0.193*** (0.0247)
cross_ijt	−0.00120 (0.00105)	−0.00121 (0.00105)	0.00754 (0.00562)	0.00742 (0.00562)	−0.00859*** (0.00125)	0.00255*** (0.000377)
size_i			−0.112*** (0.0324)	−0.107*** (0.0322)	0.00849** (0.00348)	−0.00309*** (0.000921)
size_i			−0.0129*** (0.00349)	−0.0129*** (0.00349)	−7.77e-05 (0.00115)	−0.000876*** (0.000229)
entropy_i			−2.202*** (0.391)	−2.372*** (0.389)	0.141* (0.0799)	−0.160*** (0.0188)
entropy_j			−0.526 (0.394)	−0.780** (0.383)	−0.647*** (0.132)	−0.0508** (0.0252)
dev_it	0.00426** (0.00193)	0.00421** (0.00192)	−0.239*** (0.0333)	−0.243*** (0.0340)	0.00921 (0.00748)	−0.0125*** (0.00192)
dev_jt	−0.0264 (0.0181)	−0.0263 (0.0181)	−0.125*** (0.0136)	−0.125*** (0.0136)	−0.00277 (0.00364)	−0.00860*** (0.000922)
avg_sess_it	−0.0185 (0.0120)	−0.0184 (0.0120)	0.263*** (0.0334)	0.271*** (0.0334)	0.00861 (0.00721)	0.0178*** (0.00175)
avg_scr_it	−1.53e-07*** (5.36e-08)	−1.52e-07*** (5.36e-08)	−2.30e-06*** (6.97e-07)	−2.11e-06*** (6.88e-07)	2.07e-07 (1.74e-07)	8.15e-08*** (2.97e-08)
user_ijt	6.60e-05*** (1.61e-05)	6.51e-05*** (1.61e-05)	0.000392*** (3.30e-05)	0.000387*** (3.29e-05)	2.61e-06 (7.49e-06)	2.71e-05*** (2.28e-06)
if_rank_it	0.241 (0.166)	0.245 (0.166)	6.571*** (1.161)	6.556*** (1.164)	−1.231*** (0.311)	0.577*** (0.0697)
rank_it	−0.0229 (0.0258)	−0.0240 (0.0257)	−1.228*** (0.175)	−1.276*** (0.173)	0.0820** (0.0350)	−0.137*** (0.00860)
if_rank_jt	1.084*** (0.0795)	1.084*** (0.0795)	16.55*** (0.763)	16.56*** (0.763)	1.063*** (0.198)	1.071*** (0.0512)
rank_jt	−0.359*** (0.0240)	−0.358*** (0.0240)	−1.961*** (0.156)	−1.944*** (0.156)	−0.160*** (0.0353)	−0.127*** (0.0106)
Constant	0.782*** (0.255)	0.768*** (0.255)	−117.4 (382.5)	−117.0 (615.0)	9.395*** (0.913)	−2.860*** (0.201)
App FEs	Yes	Yes				
App genre FEs			Yes	Yes	Yes	Yes
Time FEs	Yes	Yes	Yes	Yes	Yes	Yes
Observations	41,294	41,294	41,294	41,294	41,294	41,294

Notes. This table represents the empirical results showing how source app users' target app postdownload average session time vary with different source–target app pairs. Variables *topic_ij* and *topic_sq_ij* are the source–target topic similarity and its squared term, respectively. Other control variables include source app *i*'s and target app *j*'s time-varying app characteristics, the day's characteristics, and time-varying dyadic characteristics defined by source app *i* and target app *j*. In columns (1) and (2), the data are analyzed by the panel data fixed effects linear model. In column (3) and column (4), the data are analyzed by the tobit model. In columns (5) and (6), the data are analyzed by the hurdle model. FEs, fixed effects.

$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

ranking-related variables: whether source and target apps are among the top 1,499 in the store ranking (*if_rank_it*, *if_rank_jt*) and the log-transformed ranking (*rank_it*, *rank_jt*). For all specifications, we find that higher-ranked target apps would have better CP ad effectiveness for both download and postdownload usage because the ranking information is a comprehensive measure of the focal app's quality,

Table 6. Empirical Results on Postdownload Connection Count

	Linear		Tobit		Hurdle	
	(1)	(2)	(3)	(4)	(5)	(6)
Variables	*conn_ijt*	*conn_ijt*	*conn_ijt*	*conn_ijt*	*conn_ijt*	selection
topic_ij	−0.0565	0.0500***	0.0842	0.551***	0.180***	0.351***
	(0.0516)	(0.0171)	(0.246)	(0.0751)	(0.0401)	(0.0289)
topic_sq_ij	0.138**		0.579**			
	(0.0701)		(0.290)			
display_jt	0.111***	0.112***	0.371***	0.367***	0.150***	0.214***
	(0.0101)	(0.0101)	(0.0477)	(0.0477)	(0.0276)	(0.0200)
cross_ijt	−6.09e-05	−6.83e-05	−0.00113	−0.00118	−0.00158***	0.00338***
	(0.000184)	(0.000184)	(0.000795)	(0.000794)	(0.000376)	(0.000317)
size_i			−0.0106**	−0.0102**	0.00241**	−0.00245***
			(0.00447)	(0.00447)	(0.00103)	(0.000685)
size_j			−0.00207***	−0.00208***	−0.000628**	−0.000559***
			(0.000452)	(0.000452)	(0.000247)	(0.000183)
entropy_i			−0.269***	−0.285***	−0.00612	−0.142***
			(0.0530)	(0.0527)	(0.0214)	(0.0155)
entropy_j			−0.161***	−0.185***	−0.111***	−0.145***
			(0.0514)	(0.0499)	(0.0342)	(0.0196)
dev_it	0.000521	0.000492	−0.0337***	−0.0340***	0.00197	−0.0144***
	(0.000375)	(0.000375)	(0.00424)	(0.00430)	(0.00205)	(0.00132)
dev_jt	−0.00786***	−0.00782***	−0.0213***	−0.0214***	−0.00292***	−0.0134***
	(0.00240)	(0.00240)	(0.00174)	(0.00174)	(0.000929)	(0.000697)
avg_sess_it	0.000482	0.000575	0.0279***	0.0287***	0.00493***	0.0133***
	(0.00215)	(0.00215)	(0.00441)	(0.00440)	(0.00189)	(0.00137)
avg_scr_it	−1.38e-08*	−1.32e-08*	−9.05e-08	−7.33e-08	3.20e-08	1.82e-07***
	(7.96e-09)	(7.95e-09)	(9.18e-08)	(9.10e-08)	(4.09e-08)	(2.32e-08)
user_ijt	6.48e-06**	5.93e-06**	4.87e-05***	4.84e-05***	1.74e-05***	8.55e-06***
	(2.64e-06)	(2.64e-06)	(4.41e-06)	(4.40e-06)	(2.55e-06)	(1.98e-06)
if_rank_it	−0.0326	−0.0299	0.883***	0.883***	−0.0850	0.481***
	(0.0312)	(0.0311)	(0.164)	(0.164)	(0.0898)	(0.0563)
rank_it	0.00827*	0.00764*	−0.165***	−0.169***	−0.0210**	−0.109***
	(0.00456)	(0.00453)	(0.0231)	(0.0229)	(0.00906)	(0.00668)
if_rank_jt	0.214***	0.214***	2.241***	2.243***	0.206***	1.248***
	(0.0131)	(0.0131)	(0.0973)	(0.0974)	(0.0416)	(0.0376)
rank_jt	−0.0445***	−0.0445***	−0.343***	−0.342***	−0.0513***	−0.187***
	(0.00370)	(0.00370)	(0.0207)	(0.0207)	(0.00927)	(0.00834)
Constant	0.124***	0.116***	−14.74	−14.71	0.199	−1.513***
	(0.0411)	(0.0410)	(25.31)	(24.83)	(0.125)	(0.116)
App FEs	Yes	Yes				
App genre FEs			Yes	Yes	Yes	Yes
Time FEs	Yes	Yes	Yes	Yes	Yes	Yes
Observations	41,294	41,294	41,294	41,294	41,294	41,294

Notes. This table represents the empirical results showing how source app users' target app postdownload average connection counts vary with different source–target app pairs. Variables *topic_ij* and *topic_sq_ij* are the source–target topic similarity and its squared term, respectively. Other control variables include source app *i*'s and target app *j*'s time-varying app characteristics, the day's characteristics, and time-varying dyadic characteristics defined by source app *i* and target app *j*. In columns (1) and (2), the data are analyzed by the panel data fixed effects linear model. In column (3) and column (4), the data are analyzed by the tobit model. In columns (5) and (6), the data are analyzed by the hurdle model. FEs, fixed effects

*$p < 0.1$; **$p < 0.05$; ***$p < 0.01$.

and customers prefer to download and use higher-ranked apps. However, we find less consistent results for the source app rank. This is not very surprising because the number of active users has already been controlled for in the analysis, which is a decisive factor of the ranking. This might also be partially due to two opposite effects: (1) the users of high-quality source apps are more likely to download the target

Table 7. Empirical Analysis Using Two-Stage Models

	Session time		Connection counts	
	(1)	(2)	(3)	(4)
Variables	*dn_ijt*	*sess_ijt*	*dn_ijt*	*conn_ijt*
topic_ij	0.914***	0.848***	1.109***	1.802***
	(0.0132)	(0.179)	(0.0131)	(0.220)
topic_sq_ij	−0.774***	1.572***	−0.896***	0.0424
	(0.0141)	(0.200)	(0.0140)	(0.227)
display_jt	0.103***	2.598***	0.129***	0.516***
	(0.0265)	(0.0410)	(0.0270)	(0.0434)
cross_ijt	−0.00114***	−0.0224***	−0.00111***	−0.0322***
	(0.0000510)	(0.000752)	(0.000502)	(0.00769)
size_i	−0.367***	1.093***	−0.296***	−0.559***
	(0.00433)	(0.0545)	(0.00437)	(0.0688)
size_j	−0.0340***	−0.0994***	−0.296***	0.352***
	(0.00125)	(0.0184)	(0.00437)	(0.0189)
entropy_i	−0.412***	−0.682***	−0.475***	−0.592***
	(0.00362)	(0.0620)	(0.00374)	(0.0836)
entropy_j	−0.0392***	−0.708***	−0.0555***	−0.298***
	(0.00306)	(0.0521)	(0.00304)	(0.0526)
dev_i	−0.00786***	−0.0515***	−0.00791***	−0.120***
	(0.000366)	(0.00520)	(0.00358)	(0.00532)
dev_j	−0.00603***	0.0107***	−0.00801***	0.0460***
	(0.000974)	(0.00142)	(0.000974)	(0.00168)
avg_sess_it	0.0331***	0.0157***	0.0328***	0.131***
	(0.00247)	(0.00278)	(0.000251)	(0.00500)
avg_scr_it	−0.168***	0.289***	−0.173***	0.957***
	(0.0319)	(0.0375)	(0.0320)	(0.0435)
if_rank_it	0.311***	1.712***	0.127***	1.819***
	(0.0139)	(0.174)	(0.0110)	(0.176)
rank_it	0.0843***	−1.017***	0.0783***	−0.132***
	(0.00133)	(0.0168)	(0.0139)	(0.0196)
if_rank_jt	0.389***	2.202***	0.422***	2.592***
	(0.00494)	(0.0680)	(0.0505)	(0.0840)
rank_jt	−0.0208***	−0.501***	−0.0417***	−0.0940***
	(0.00986)	(0.0520)	(0.00104)	(0.0158)
Constant	−4.036***	14.336***	−2.459***	5.810***
	(0.118)	(0.539)	(0.0447)	(0.666)
App genre FEs	Yes	Yes	Yes	Yes
Ranking	Yes	Yes	Yes	Yes
Time FEs	Yes	Yes	Yes	Yes
Observations	41,294	41,294	41,294	41,294

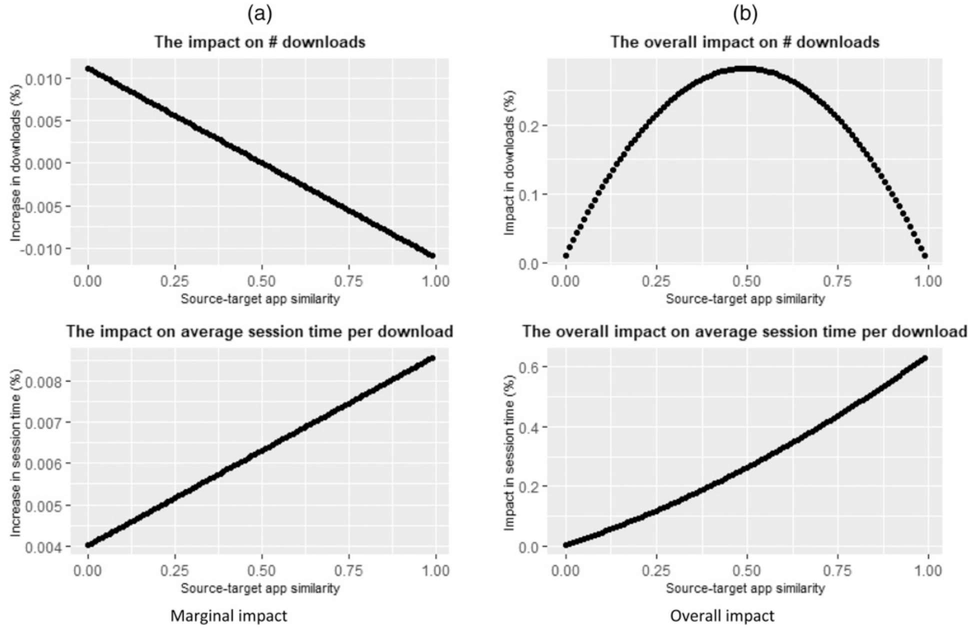
Notes. This table represents the empirical results from the two-stage model. Variables *topic_ij* and *topic_sq_ij* are the source–target topic similarity and its squared term, respectively. Other control variables include source app *i*'s and target app *j*'s time-varying app characteristics, the day's characteristics, and time-varying dyadic characteristics defined by source app *i* and target app *j*. In columns (1) and (2), the postdownload usage measure is session time. In columns (3) and (4), the postdownload usage measure is connection count. FEs, fixed effects.

$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

app to collect an in-app benefit, and (2) users who are satisfied with the source apps are less likely to migrate to the target apps.

For the model specifications other than the linear and download tobit models, the results also show the impacts of app file sizes (*size_i* and *size_j*) and functional diversity measures (*entropy_i* and *entropy_j*) on users' app consumption behaviors. For most specifications, users' probability of downloading the target app decreases as the target app size or functional diversity measure increases. Given that most target apps are free, users' primary costs to adopt and use an app are data usage and the storage space required for the new app installations. Thus, target apps with

Figure 2. The Marginal and Overall Impact of Source–Target App Similarity on Download and Postdownload Usage



larger file sizes may be less attractive in the download decision and the overall usage patterns. In addition, when the source app has more features (which can be represented by a larger size and a higher functional diversity measure), new target apps will be less attractive. Based on the results of the two-stage model, we can see that for a given download, postdownload usage will increase if the target app has more features because these downloaded targets may have attractive functionalities for users.

Lastly, we observe that users with larger screen sizes (*avg_scr_it*) are less likely to download target apps in most model specifications. A plausible explanation could be that the users with larger-screen mobile phones would have higher income, thus finding in-app rewards to be less attractive. Based on the two-stage model, conditional on a download decision, users' postdownload usages would increase if the average screen size is larger. One potential reason is that a larger screen would bring customers a better user experience.

6. Impact on Source Apps and Market Demand Distribution

In this section, we take one step back and investigate the effect of CP campaigns on the whole app market. One important question on CP framework is whether it will alleviate the winner-take-all phenomenon discussed in Section 1. Specifically, we are interested in whether lowly ranked apps are actually benefiting more from CP campaigns. If this is the case, the campaigns are not only helping target apps but also contributing to the long-tail demand distribution of the app market. Another important aspect of the CP framework is its impact on source apps. If the campaign's outcome comes at the expense of the source apps' performance because of substitution effects (i.e., users stop using source apps after downloading target apps), source app developers may be reluctant to join this promotion. Thus, we investigate how the rankings of source and target apps, as well as download and usage measures, will be influenced by CP participation.

We use staggered DID models to evaluate the effects. In our data set, 679 apps have adopted the app analytics tool, which allows us to monitor individual users' downloads and usage activities. We use the analytics-enabled apps without CP participation as the control group. The apps that have participated in the CP as either source or target apps are in the treated group.[29] For each treated app, we create dummy variables to indicate whether day t is after its initial participation in a CP campaign (*after_it* and *after_jt*). To evaluate the impact of CP campaign participation on an app's subsequent performance, we use the daily ranking of the app (source or target) as the dependent variable.[30] However, one limitation of the ranking data are that the rankings are reported up to a certain number (1,499 in our data), leaving a large number of missing values. We address this data issue with three approaches. In the first model (Log_rank), following Smith and Telang (2009), we replace the missing values with the maximum rank (1,499) and use the natural logarithm–transformed ranking as the dependent variable. To control for app-level unobservable characteristics, we use a panel data fixed-effects linear model for the empirical analysis. In the second model (Top 50), we use a dummy variable *top50* to indicate whether the app's ranking is within the top 50. Lastly, we also use a tobit model (Log_rank with tobit) with app dummies to analyze the data with censored ranking (which has the upper limit of 1,499). In addition, we add an interaction (*after_rank_jt*) between the after dummy and the apps' average ranking before their CP participation to explore the moderating effect of initial rankings of the apps. For the control variables, we have a display ad dummy (*after_display_t*) and time fixed effects (e.g., day of week, month, and year). We separately evaluated the effect on source and target apps. The number of apps included in the analysis has decreased from 679 to 643 and 615 because some apps only served as either a source or a target app. Note that in the Top 50 model, the positive coefficient of the after dummy indicates app performance improvement, whereas it is the opposite in the other two models.

The results are reported in Table 8. In all specifications, it is shown that the apps participating in CP campaigns as target apps experience ranking improvement. From the perspective of source apps, we do not find that they suffer from substitute effects (e.g., ranking does not systematically change after CP participations), except for the probability of being ranked in the top 50, which is only significant at 10%. This can partially explain why source apps are willing to promote other apps to their customers. Lastly, based on the interaction term's coefficients, we find that target apps with initial lower rankings tend to benefit more from CP campaigns. This may be evidence that CP campaigns could potentially mitigate this market's winner-take-all issue. In sum, the results indicate that CP campaigns provide (1) a good monetization tool without performance compromise for source apps, (2) a user acquisition channel for target apps, and (3) a mechanism to improve the diversity of mobile app markets.

The results for download and usage measures are presented in Table 9. We use the log-transformed number of downloads and postdownload usage measures with linear regressions. We find interesting results for source apps: CP participation (1) does not impact their new user acquisitions and (2) can even lead to a longer session time and more connection counts. A plausible explanation is that as users get in-app rewards by downloading target apps, they tend to spend more time in the source apps. This provides a rationale for source app to participate in CP campaigns. From the perspective of target apps, as we expected, both their number of downloads and in-app usage increase after CP participation.

7. Matching Platform Design

Based on the insights from the empirical results, we design a CP matching platform that matches source and target apps to maximize the ad effectiveness of CP campaigns. In industry practice, one approach is to use a real-time bidding system that allows target apps to bid on the right source apps based on their expected ad performances. For this approach to work, target app developers should have the capability of predicting the ad performance on alternative source apps. However, as will be shown in Section 7.2, accurate ad performance prediction requires detailed app- and user-level data on a large number of apps, which are not available to individual app developers. We argue that the CP platform has the informational advantage to collect comprehensive analytics data on the whole app market. Thus, we propose a centralized CP matching platform that can produce efficient app matches for multiple CP campaigns.

We first describe the overall design of the information system for the app-matching platform. The central tasks of the matching platform are (1) to collect public and analytics information on the source and target apps in the CP market and (2) to match source and target apps that will produce the best match outcomes in terms of expected downloads and postdownload usages. In doing so, the matching platform should accurately predict the outcome of different possible matches and choose the optimal one. We build supervised machine-learning-based predictive models using detailed app- and user-level features (as described in Section 4). Notably, some app features are publicly available in the app markets (which we term *public features*), whereas other dynamic app features (which we term *analytics features*) can only be constructed by the matching platform that has access to the detailed app analytics data (e.g., app execution time, in-app purchase). In addition, *individual features* (e.g., device model, platform, carrier), which can create privacy concerns, are also only available to the platform. Using different combinations of features, we train machine learning models. Lastly, we run simulations to estimate the ad effectiveness outcomes based on different features and matching algorithms (e.g., random, greedy, deferred acceptance). By comparing the outcomes of different models, we can show the marginal benefit of using additional features on the matching performance. We find that the inclusion of analytics and individual features significantly improves the ad effectiveness of CP campaigns in terms of both download and postdownload usages.

7.1. Information System Design

Figure 3 illustrates the design of an information system to support the matching market for CP campaigns.

Table 8. The Impact of CP Campaigns on Source and Target Apps' Rankings

	Log_rank			Top 50			Log_rank with tobit		
	Target apps		Source apps	Target apps		Source apps	Target apps		Source apps
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Variables	*log_rank_jt*	*log_rank_jt*	*log_rank_jt*	*top50_jt*	*top50_jt*	*top50_it*	*log_rank_jt*	*log_rank_jt*	*log_rank_jt*
after_jt	−1.583***	0.0276		0.172***	−0.0171		−2.590***	−2.786***	
	(0.353)	(0.167)		(0.0664)	(0.0730)		(0.0159)	(0.0173)	
after_rank_jt		−0.00290***			0.000340***			−0.000613***	
		(0.000272)			(8.80e-05)			(3.93e-05)	
after_it			0.0149			−0.0838*			−2.103***
			(0.301)			(0.0478)			(0.0159)
after_display_t	−0.247	−0.190	0.0547	0.0934	0.0866	−0.115**	−1.007***	−1.094***	−2.413***
	(0.371)	(0.164)	(0.0798)	(0.143)	(0.118)	(0.0576)	(0.0918)	(0.0943)	(0.0941)
App FEs	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Time FEs	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Constant	5.312***	5.390***	5.358***	0.142***	0.133***	0.104***	4.936***	4.923***	4.925***
	(0.0782)	(0.0702)	(0.0738)	(0.0196)	(0.0187)	(0.0171)	(0.0610)	(0.0534)	(0.0565)
Observations	91,188	91,188	85,656	91,188	91,188	85,656	98,367	98,367	98,367
R^2	0.067	0.139	0.014	0.027	0.048	0.012			
Number of apps	643	643	615	643	643	615	643	643	615

Notes. This table represents the empirical analysis to show how the performance of source apps and target apps are influenced by the CP participation. Variable *after_rank_jt* is the interaction between the *after_jt* dummy and each app's average ranking before the CP participation; *after_display_t* is the dummy that indicates whether this app has display ads in day *t*. In columns (1)–(3), the dependent variable is the log-transformed ranking of the focal app, and the model is the linear panel data model with fixed effects. In col-umns (4)–(6), the dependent variable *top50* is a dummy that represents whether the app is a top 50 app and the model is the linear panel data model with fixed effects. The results with the conditional logit model with fixed effects are quantitatively equivalent. In columns (7)–(9), the dependent variable is the log-transformed ranking of the focal app and the model is the tobit model with app dummies. FEs, fixed effects.

$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

The system consists of (1) the back-end subsystem, which keeps collecting public app market data and app analytics data, and (2) the front-end subsystem, which conducts the real-time matching process between source and target apps on a daily basis.

The back-end subsystem collects (1) public information on the mobile apps in different app stores (e.g., Apple's App Store, Google's Play Store, and SK Telecom's T Store) and (2) app analytics data. The data collector, written in Python, uses the app market application programming interfaces to collect various app characteristics, such as app prices, app genres, app descriptions, availability of in-app purchases, reviews, ratings, rankings, file sizes, developers, and updates. Then, using the collection of app descriptions, we build topic models to quantify pairwise app similarity using a Scala LDA implementation.[31] The analytics data provide more detailed information on the status of an app, which includes an active user list, per-user app usage activities (e.g., connection count, session time, in-app purchases), and mobile device information (e.g., Android/iOS platform, device model, wireless carrier). Notably, we classify the analytics data into aggregated (e.g., summary statistics of all users) and individual data. Recorded in a relational database, the collection of public and analytics data points is then used as a training data set to build machine-learning models on ad effectiveness. Given a source app user, the model predicts whether the user will download a target app and, if so, further predicts the degree to which the user will use the downloaded app. Note that the machine-learning training process is independent of the daily data updates from the front-end subsystem. Section 7.2 describes the details of the prediction models.

The front-end subsystem dynamically generates matches between source and target apps. The inputs of the matching process are (1) ad campaign requests from target apps, (2) available source apps, and (3) predicted ad effectiveness provided by the back-end subsystem. In the matching engine, we use a generalized deferred acceptance algorithm (Gale and Shapley 1962, Roth 1984, Roth 2008) to propose app matches (described in Section 7.3.3). The app developers are informed of the resulting matches, and the CP campaigns will be launched. In industry practice, all the data points are updated on a daily basis, and thus the matchings are updated at the same frequency. Because the number of apps involved in the CP promotions is only on the order of hundreds, most of the matching algorithms can be executed within a few seconds on a desktop computer with reasonable performance. But we also simulate more crowded

Table 9. The Impact of CP Campaigns on Source and Target Apps' New Downloads and Postdownload Usages

	Download			Session time			Connection counts		
	Target apps		Source apps	Target apps		Source apps	Target apps		Source apps
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Variables	*log_dn_jt*	*log_dn_jt*	*log_dn_jt*	*sess_jt*	*sess_jt*	*sess_it*	*conn_jt*	*conn_jt*	*conn_it*
after_jt	2.140***	1.341**		6.717***	4.624**		3.432***	2.370**	
	(0.470)	(0.652)		(1.421)	(2.129)		(0.683)	(0.986)	
after_rank_jt		0.00144**			0.00377**			0.00191**	
		(0.000604)			(0.00180)			(0.000911)	
after_it			0.602			3.019**			1.806***
			(0.483)			(1.496)			(0.692)
after_display_t	0.619	0.591	0.593	1.274	1.199	1.895**	0.480	0.442	0.773**
	(0.518)	(0.426)	(0.460)	(1.048)	(0.825)	(0.944)	(0.465)	(0.347)	(0.301)
App FEs	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Time FEs	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Constant	1.860***	1.821***	1.706***	6.754***	6.653***	5.981***	2.859***	2.807***	2.801***
	(0.148)	(0.143)	(0.123)	(0.428)	(0.412)	(0.394)	(0.179)	(0.171)	(0.165)
Observations	91,188	91,188	85,656	91,188	91,188	85,656	98,367	98,367	98,367
R^2	0.050	0.057	0.007	0.073	0.078	0.043	0.089	0.096	0.046
Number of apps	643	643	615	643	643	615	643	643	615

Notes. This table represents the empirical analysis to show how the number of downloads and postdownload usages are influenced by the CP participation. Variable *after_rank_jt* is the interaction between the *after_jt* dummy and each app's average ranking before the CP participation; *after_display_t* is the dummy that indicates whether this app has display ads in day *t*. In columns (1)–(3), the dependent variable is the log transformed number of daily downloads of the focal app. In columns (4)–(6), the dependent variable is the log transformed total session time the focal app's users spent on it. In columns (7)–(9), the dependent variable is the log transformed total connection counts from the focal app's users executed on it. FEs, fixed effects

$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

markets with thousands of apps and explore the scalability of the algorithms. In addition, we measure the marginal benefits of adding different levels of features (public, analytics, and individual) on the prediction accuracy and ultimately ad performance.

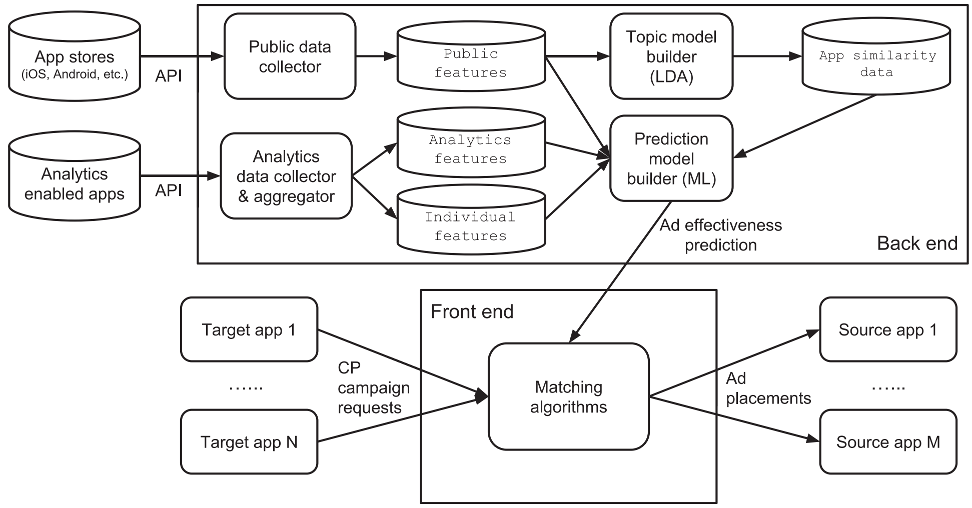
7.2. Building Prediction Models Using Machine Learning

7.2.1. General Setup. We build supervised machine-learning models to predict CP ad effectiveness in terms of downloads, postdownload session time, and postdownload connection count. The unit of analysis is (day t, source app i, target app j, user u). In other words, for user u using source app i on day t, the model predicts the likelihood of download and postdownload usage with respective to a target app j. Because download action is a binary decision, we apply classification algorithms from the machine-learning literature. For postdownload usage prediction, we apply regression algorithms because the outcome variables are in continuous values. We use Python packages Keras[32] and Tensorflow[33] for neural network models and Scikit-learn[34] for the rest of the machine-learning models. We collect the CP ad effectiveness outcome and app features according to the variable definitions from Section 4. With extensive cross-validations, we choose the best model and hyperparameters, which are then used for matching simulations.

7.2.2. Feature Construction. To train a classification model, one needs to have both positive and negative samples. In our context, the data set should include the ad impressions that are successful ($dn\_uijt = 1$: a user from a source app downloaded the target app) and those that are not ($dn\_uijt = 0$: a user from a source app was exposed to the target app but decided not to download it). However, because we do not have impression data, we can only observe positive samples (26,668 target app installation events). Thus, to artificially generate negative samples, we make an assumption that a 10% random sample of daily active users of a source app is exposed to a specific target app.[35] The total number of observations for the download prediction data (dn_uijt), including the negative samples, is about 1.3 million. We note that negative sample generation was only used to generate a training data set for the download prediction model but not in the postdownload usage prediction. Thus, the number of observations for the postdownload usage (e.g., $sess\_uijt$, $conn\_uijt$) is 26,668.

Depending on their roles in the CP matching platform, stakeholders have different access levels of information. For example, an individual app developer can only

Figure 3. Information System Design for the App-Matching Platform



access information from the public app market and its own app analytics data. Thus, we believe that the matching process of the current industry practice (e.g., Tapjoy) can be simulated with the case that only public features are available. By contrast, the CP platform has access to public app market data as well as app analytics data from all the apps that adopted the analytics tool.

The variables constructed in Section 4 are used as public and analytics features. We further define individual user-level variables to control for individual heterogeneity. The summary statistics of public and analytics features are listed in Table 3, and those of individual user level features are listed in Table 10.[36] First, following the idea of source–target app similarity, we calculate the topic similarity between a target app and the apps used by a user using different weighting schemes: unweighted average (*topic_avg_uj*), session-weighted average (*topic_sess_uj*), and connection-weighted average (*topic_conn_uj*). Second, to control for user activity levels, we count the number of unique apps a user used in a day (*unique_app_daily_u*), the total number of unique apps a user used in the previous month (*unique_app_total_u*), the app session time a user spent in a day (*sess_daily_u*), the total app session time a user spent in the previous month (*session_total_u*), the app connection count a user had in a day (*conn_daily_u*), the total app connection count a user had in the previous month (*conn_total_u*), the number of in-app purchases a user made in the previous month (*iap_total_u*), and the number of conversions by a user (*conversion_total_u*). Note that these individual user variables are measured based on individual user activities recorded by the app analytics tool and that they can create privacy concerns. Because mobile app users' preferences can change dynamically, we use one-month data (March 15–April 14, 2014) to construct the individual user features and then use the subsequent month data (April 15–May 14, 2014) to generate the dependent variables of the matching simulation.

7.2.3. Download Prediction Model. The outcome variable (*dn_uijt*) and features are constructed according to the definitions, as described previously. Then the best-machine learning model and hyperparameters are selected from extensive cross-validation experiments. We split the 1.3 million observations into 60% for training, 30% for validation, and 10% for testing. To mitigate the potential biases in data splitting, we report the average test performance over five different training/validate/test sets. For the download model evaluation metrics, we use precision ($\frac{TP}{TP+FP}$), recall ($\frac{TP}{TP+FN}$), and F_1 score ($2 \times \frac{precision \times recall}{precision + recall}$), where TP is true-positive count, FP is false-positive count, and FN is false-negative count. Following Davis and Goadrich (2006), we also report the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC) because our data set has a class imbalance issue (where there are significantly more negative samples than positive ones).[37] For all five metrics, higher values are preferred.

For model selection, we use representative classification algorithms, such as logistic regression (Logit), Lasso, k-nearest neighbors (KNN), support vector classifier (SVC), random forest (RF) classifier, feedforward neural network (FFNN), and convolutional neural network (CNN). AUPRC was used as the target metric in hyperparameter tuning. The hyperparameters tested in the cross-validation follow:[38]

- KNN: # of neighbors = {1, 3, 5, 7, 9, 11}
- SVC: kernel function = {radial basis function, linear, polynomial, sigmoid}
- RF: # of trees = {20, 30, 40, 50, 100}, max tree depth = {1, 2, 4, 8, 16, 32, 64}

Table 10. Summary Statistics: App Matching Simulation

Variable	Descriptions	Obs	Mean	SD	Min	Max
	Dependent variables: ad effectiveness					
dn_uijt	One if source app i's user u downloaded target app j on day t, zero otherwise	1,318,980	0.020	0.140	0	1
sess_uijt	Postdownload total session time on target app j of user u from source app i on day t (log)	26,668	1.242	3.650	0	19.618
conn_uijt	Postdownload total connection count on target app j of user u from source app i on day t (log)	26,668	0.472	0.654	0	8.018
	Individual user features (individual)					
topic_avg_uj	Average topic similarity between user u's apps and target app j	1,161,534	0.434	0.390	0.000	1.000
topic_sess_uj	Session-weighted topic similarity between user u's apps and target app j	925,282	0.475	0.413	0.000	1.000
topic_conn_uj	Connection-weighted topic similarity between user u's apps and target app j	1,159,698	0.449	0.408	0.000	1.000
uniq_app_daily_u	No. of distinct apps user u uses daily	1,318,980	1.045	1.154	0	110.36
uniq_app_total_u	No. of distinct apps user u uses in total	1,318,980	32.387	35.786	0	3,421
sess_daily_u	Average session time of user u (minutes)	1,318,980	17.366	41.823	0	7,903
sess_total_u	Total session time of user u (minutes)	1,318,980	538.340	1296.511	0	244,999
conn_daily_u	Average connection count of user u	1,318,980	3.409	27.159	0	7,440
conn_total_u	Total connection count of user u	1,318,980	105.693	841.925	0	230,649
iap_total_u	No. of in-app purchases by user u	1,318,980	0.018	0.135	0	4
conversion_total_u	No. of conversions by user u	1,318,980	0.282	1.478	0	29
platform_android_u	One if user u's platform is Android, zero otherwise	471,562	0.999	0.036	0	1
platform_ios_u	One if user u's platform is iOS, zero otherwise	471,562	0.000	0.022	0	1
platform_others_u	One if user u's platform is among others	471,562	0.001	0.028	0	1
model_samsung_u	One if user u's device is a Samsung model, zero otherwise	471,562	0.197	0.398	0	1
model_apple_u	One if user u's device is an Apple model, zero otherwise	471,562	0.000	0.016	0	1
model_others_u	One if user u's device is among others, zero otherwise	471,562	0.802	0.398	0	1
carrier_skt_u	One if user u's carrier is SK Telecom, zero otherwise	471,562	0.461	0.498	0	1
carrier_kt_u	One if user u's carrier is Korea Telecom, zero otherwise	471,562	0.253	0.435	0	1
carrier_lgt_u	One if user u's carrier is LG Telecom, zero otherwise	471,562	0.242	0.428	0	1
carrier_others_u	One if user u's carrier is among others, zero otherwise	471,562	0.044	0.205	0	1

Note. For the variable naming convention, u, i, j, and t stand for user, source app, target app, and day, respectively. Obs, observations; SD, standard deviation.

- FFNN: # of layers = {2, 3, 4, 5, 6, 7, 8}, # of nodes in a layer = {8, 16, 32, 64, 128, 256}
- CNN: # of filters = {15, 20, 25, 30}, kernel size = {1, 3, 5, 7, 9}, optimizer = {adam}

Using our computer server with a 2.86-GHz central processing unit, the average training times in each hyperparameter combination were 6.22, 13.93, and 8.27 minutes for the RF, FFNN, and CNN models, respectively. The total hyperparameter tuning process required 18.16, 40.65, and 16.55 hours for the three models, respectively.

Panel A of Table 11 shows the download prediction results of different models when all features (public, analytics, and individual) are used. We omit the results on Logit, KNN, and SVC because of their suboptimal performances. The results show that RF is the best model for download prediction (AUPRC of 52.77%, AUROC of 92.34%, and F_1 score of 45.83%).

Using the best RF model, we next estimate the marginal effect of different feature groups. To measure the impact of the most basic features (public), we include naive models that do not require any features: the first model (Rand_Down) predicts a user to download a target app with a probability of 50%, and the second model (Avg_Down) predicts a user to download a target app with a probability of 2% (the mean value of *dn_uijt*). Panel B of Table 11 shows that (1) the prediction model can achieve AUPRC of up to 33% even with the privacy-preserving features (public and analytics) and that (2) the inclusion of individual-level features can boost the prediction accuracy by 11%. The result also shows that the marginal effects of public, analytics, and individual features are 19%, 12%, and 19% in the AUPRC, respectively. We observe qualitatively similar results with the F_1 and AUROC measures.

Table 11. Prediction Performance on Download

Panel A: Model comparison (features: public + analytics + individual)						
Model	Best hyperparameters	Precision	Recall	F_1 score	AUROC	AUPRC
Lasso	—	69.79	14.85	24.49	83.99	30.76
CNN	#filters = 20; kernel size = 7	66.70	23.97	33.42	86.93	35.86
FFNN	#layers = 3; #nodes = 64	79.41	32.02	45.46	90.16	48.63
RF	Depth = 16; #trees = 100	84.73	31.41	45.83	92.34	52.77
Panel B: Feature group comparison						
Model	Feature group	Precision	Recall	F-1 Score	AUROC	AUPRC
Avg_Down	No features	2.10	2.06	2.08	50.08	2.02
Rand_Down	No features	2.01	49.82	3.86	49.96	2.02
RF	Public	70.40	7.85	14.12	74.85	21.44
RF	Public + analytics	75.16	19.93	31.50	81.20	33.60
RF	Public + analytics + individual	84.73	31.41	45.83	92.34	52.77

7.2.4. Postdownload Usage Prediction Model. Following similar procedures in building download prediction models, postdownload usage prediction models are developed with feature construction, model selection, hyperparameter tuning, and cross-validation. There are three major differences in the usage prediction model. First, we use regression models because postdownload usage metrics are in continuous values. Second, because postdownload usage is highly skewed, we apply the log transformation to the session time and connection count. Third, because postdownload usage prediction is only relevant to the users who downloaded target apps, we use the 26,668 observations with download actions ($dn\_uijt = 1$).

For model comparison, we use regression algorithms such as linear regression (Linear), Lasso, RF regressor, FFNN, and CNN. The model evaluation metrics are coefficient of determination (R^2), mean absolute error (MAE), and root mean square error (RMSE). A higher value is preferred for R^2, and a lower value is preferred for MAE and RMSE. The specifications of the hyperparameters we tested in the cross-validation follow:

- RF: # of trees = {20, 30, 40, 50, 100}, max tree depth = {1, 2, 4, 8, 16, 32, 64}
- FFNN: # layers = {2, 3, 4, 5, 6, 7, 8}, # nodes in a layer = {16, 32, 64, 128, 256}, optimizer = {*rmsprop*}
- CNN: # of filters = {15, 20, 25, 30}, kernel size = {1, 3, 5, 7, 9}, optimizer = {*rmsprop*}

The average model training times in each hyperparameter combination are 4.77, 4.78, and 4.77 seconds for RF, FFNN, and CNN models, respectively. In total, the hyperparameter tuning process only took 13.94, 13.97, and 9.55 minutes for the three models, respectively.

Panels A and C of Table 12 show the postdownload usage prediction results of different models when all features (public, analytics, and individual features) are used. The results show that RF has the best prediction performance for both postdownload session time ($R^2 = 0.22$ and MAE = 2.44) and connection count ($R^2 = 0.39$ and MAE = 0.39).

Using the best RF model, we estimate the marginal effect of different feature groups on postdownload usage prediction. To measure the impact of the most basic features (public), we again include naive models that do not require any features: in the first benchmark (*Avg_Sess*, *Avg_Conn*), we use the average value over all apps as the "prediction" of each user's postdownload usage, and in the second benchmark (*Avg_Sess_Src*, *Avg_Conn_Src*), we use the average value over the specific source app as the prediction of each user's postdownload usage.[39] Panels B and D of Table 12 show the results on different feature groups. Interestingly, we find that the prediction model works well (R^2 = 0.22 and 0.39) only when all the features are available. However, when the models do not use individual user features, the prediction accuracy degrades significantly (R^2 values are in the range between 0.00 and 0.08). This result implies that it is necessary to have individual user-level data in predicting postdownload usage levels.

7.3. App Matching Simulation

7.3.1. General Setup. With the large number of apps available in the app market, there is a significant search cost for app developers on both sides (sources and targets) to find optimal CP campaign counterparts in an efficient manner. We propose a centralized matching platform to assign the matchings between source and target apps based on their predicted preference lists and different matching algorithms. Notably, for each simulation, we use two different models for matching simulation and the subsequent evaluation processes. To measure the impact of different feature groups, matches are generated using

Table 12. Prediction Performance on Postdownload Usage

Panel A: Model comparison, metric: *sess_uijt*, features: public + analytics + individual				
Model	Best hyperparameters	R^2	MAE	MSE
Lasso		0.03	3.16	5.23
CNN	#filters = 25; kernel size = 6	0.15	2.72	4.90
FFNN	#layers = 2; #nodes = 16	0.19	2.49	4.80
RF	#trees = 100; depth = 8	0.22	2.44	4.69
Panel B: Feature group comparison: metric: *sess_uijt*				
Model	Feature groups	R^2	MAE	MSE
Avg_Sess	No features	0.00	3.41	5.61
Avg_Sess_Src	No features	0.00	1.98	5.58
RF	Public	0.08	2.93	5.11
RF	Public + analytics	0.05	2.96	5.19
RF	Public + analytics + individual	0.22	2.44	4.69
Panel C: Model comparison, metric: *conn_uijt*, features: public + analytics + individual				
Model	Best hyperparameters	R^2	MAE	MSE
Lasso		0.00	0.70	0.95
CNN	#filters = 30; kernel size = 4	0.32	0.45	0.78
FFNN	#layers = 4; #nodes = 16	0.35	0.43	0.76
RF	#trees = 100; depth = 8	0.39	0.39	0.74
Panel D: Feature group comparison: metric: *conn_uijt*				
Model	Feature groups	R^2	MAE	MSE
Avg_Conn	No features	0.00	0.70	0.98
Avg_Conn_Src	No features	0.00	0.68	1.13
RF	Public	0.02	0.70	0.94
RF	Public + analytics	0.00	0.71	0.95
RF	Public + analytics + individual	0.39	0.39	0.74

prediction models, some of which use partial feature groups. To evaluate the performance of the resulting matches from different algorithms and prediction models, we always use the most accurate prediction model (i.e., the RF model using full features: public + analytics + individual).

7.3.2. Utility Estimation of App Matches. In the simulation, a critical operationalization step is to construct the preference list for each app based on utility values, considering both cost and benefit for source and target app developers to participate in the CP campaign. Specifically, for target app developers, the cost is the payment for each successful download, and the benefit is the corresponding number of downloads and postdownload usage gained from the CP campaign. For source app developers, the benefit is the cut on each download transaction, whereas the potential cost is the possible loss of active users to the target apps. As shown in Section 6, though, source apps do not suffer from such issues.

Different payment exchanges are available between source and target apps. The first possible contract is that a source app developer receives a fixed price per target app download. This contract may not be the most efficient scheme in terms of maximizing the effectiveness of the ad strategy because it does not consider the various levels of benefit that a single download can potentially achieve. For instance, some source app users may keep using the target app after downloading, whereas free-riding users may download the target app only for the reward without usage intention. It is obvious that the former download is more valuable for target app developers. Therefore, target app developers should be willing to pay a higher price for such downloads with actual usage. Thus, the second potential contract is benefit contingent, with the price based on potential usage of the target app as a result of the download. Following the literature, we assume that source and target app developers would share the benefit at a fixed fraction (Sørensen 2007).

In our simulation design, we incorporate the two payment schemes in the utility value estimation. The first one is the fixed-price model that was being used by our data source CP platform. For this, we use the *download*-maximization strategy, where the utility of a match for the target app is the expected number of downloads. Second, we also consider the benefit-contingent scheme with the *usage*-maximization strategy, where the utility of a match is the expected usage of the acquired users. In both scenarios, we leverage the prediction models to predict the download probability and postdownload usage level and use them as the utility values in the preference list.

In CP industry practice (e.g., Tapjoy), target app developers make bids to source apps based on their expected utilities. In this case, app developers only have access to public app market data. We simulate this scenario by using prediction models trained only on public data. By contrast, arguably, the CP platform can make a better-informed decision because it has comprehensive analytics data, which scenario is modeled by using the prediction models trained with a full set of features (public + analytics + individual).

7.3.3. Matching Algorithms. Given the constructed preference list, the next step is to apply matching algorithms to generate source–target matches. Following the matching literature, we use two deferred acceptance–based algorithms. The deferred acceptance algorithm was first introduced by Gale and Shapley (1962) for one-to-one and many-to-one matching games in the contexts of marriage and college admissions. The algorithm has been applied to other matching markets, such as resident–hospital assignments in the United States and school admissions in the cities of New York and Boston. It is shown that the outcome of the algorithm is a *stable* matching, which guarantees that the matched pairs do not have incentives to deviate from the given matches.

For the CP campaign case, each target app can be advertised on multiple source apps, and each source app can promote several target apps at the same time. Thus, we extend the deferred acceptance algorithm to allow many-to-many matching. The quota (the number of target apps that can be matched to a source app) is an important parameter of the deferred acceptance algorithm, and fairness is essential when we evaluate the performance of a matching as well as the long-term success of the market. Therefore, the number of matches each source and target app has should be balanced. We thus set a quota for each target app as the number of ad slots available on all source apps divided by the number of target apps. Following CP industry practice, each source app can be matched to at most seven target apps in our simulation.[40] In addition, because either counterpart of the two-sided market can make the proposal, we use source app– and target app–proposing deferred acceptance algorithms (*src_da_match* and *tgt_da_match*). Online Appendix G provides the details of the generalized deferred acceptance matching algorithms and examines their computational scalability.[41]

For performance evaluation, we compare the deferred acceptance algorithms with two naive algorithms: random matching (*random_match*) and greedy matching (*greedy_match*). With the random matching algorithm, for a given source app, we randomly select seven target apps. In the greedy matching algorithm, the centralized decision maker—the matching platform in our case—first sorts all possible source–target pairs based on predefined utility values (either download or usage). Then, in a greedy fashion, pairs will be matched until all the demands are met (e.g., all target apps are fully matched). This greedy algorithm is different from the deferred acceptance algorithm because the resulting matches may not be guaranteed to be stable.

7.3.4. Simulation Results. The final step of the simulation is to evaluate the matches generated by different matching algorithms and different features. For a given hypothetical match made by the algorithm, we use the most accurate prediction model (RF with public + analytics + individual features) to predict the download and postdownload usage metrics.[42] Then we aggregate each app pair's predicted values (e.g., number of downloads and postdownload usages) to the day level. If the algorithms are to maximize the total downloads, we keep download performance as the ad effectiveness metric. Similarly, we report the postdownload usage performance when the usage-maximization strategy is used. As mentioned in Section 7.2.2, we use the last 30-day data for simulation and report the average daily performance over the 30-day period. Specifically, for the download metric, we count the total app downloads (*dn_uijt*) from all CP campaigns in a given day. For the postdownload usage metrics, we aggregate the session time (*sess_uijt*) or connection count (*conn_uijt*) from all the users who downloaded an app through any CP campaign from a given day.

The results are summarized in Table 13. The first notable result is that the inclusion of analytics and individual features significantly improves the matching performance. In terms of the download metric, the analytics features induce marginal performance improvements of 39% in all three nonrandom algorithms (*greedy_match*, *src_da_match*, *tgt_da_match*), and the marginal effect of individual features is around 28%. For the connection metric, the marginal benefits are in a similar range (22%–38%). However, in the session hours metric, the marginal benefits were smaller (analytics 4.5%, individual 1%).

Interestingly, when only public features are used to train the machine-learning model, the performances of nonrandom algorithms can be even worse

Table 13. Matching Simulation Results

Feature groups	*random_match*	*greedy_match*	*src_da_match*	*tgt_da_match*
Panel A: Metric = total downloads from a day				
Public	8,587.5	6,513.9	6,484.3	6,510.4
Public + analytics	8,565.2	9,065.3	9,056.3	9,050.8
Public + analytics + individual	8,392.7	11,586.3	11,586.8	11,586.9
Panel B: Metric = total session times from the users acquired in a day				
Public	10,629.9	14,803.5	15,533.1	15,550.1
Public + analytics	10,429.1	15,406.0	16,254.2	16,251.6
Public + analytics + individual	9,694.5	16,064.7	16,414.8	16,412.9
Panel C: Metric = total connections from the users acquired in a day				
Public	1,909.6	1,516.6	1,441.6	1,442.4
Public + analytics	1,876.6	1,856.7	1,986.6	1,987.0
Public + analytics + individual	1,902.6	2,355.0	2,426.1	2,427.0

(in terms of downloads and connections) than those of random matching algorithms. This can be evidence that the industry practice that only relies on public information may not be optimal. This also implies that when using machine-learning approaches in the CP matching process, it is essential to have detailed analytics data on the app usage characteristics to have accurate prediction models and ultimately efficient app matches.

In terms of different matching algorithms, the target-proposing deferred acceptance algorithm (*tgt_da_match*) shows the best performance, whereas the differences among the three nonrandom algorithms are marginal. Overall, compared with the worst specification (*random_match*), the best specification (deferred acceptance with full features) shows 78%, 69%, and 68% improvement in the download, session, and connection metrics, respectively.

8. Concluding Remarks

The success of the mobile app market can be attributed to its nature as a two-sided market (Rochet and Tirole 2003). However, search cost issues arise as the market becomes increasingly crowded with millions of apps and billions of users (Hagiu 2006, Evans et al. 2011). Cross-promotion, advertising an app (target app) within another app (source app), is introduced as an innovative app promotion mechanism to address this issue. Based on the data from a large-scale random matching experiment, we identified significant factors (including the topic model-based app similarity) that contribute to better app matching in CP campaigns. We document the preference discrepancy over app users' variety-seeking download behaviors and consistency-seeking postdownload usage decisions. To further improve the app-matching process in CP campaigns, we propose an app-matching platform using machine-learning models and matching algorithms. Our simulation results show that the ad effectiveness of CP campaigns can be significantly improved with the use of analytics and individual user features.

We believe that our paper contributes to both the academic literature and industry practice. From the academic perspective, our paper contributes to the information systems and marketing literature on mobile ecosystem and mobile advertising. In addition, because CP campaigns are involved with multiple products, this paper makes contributions to the research strand on multiproduct promotions. Compared with other related marketing strategies, such as cross-selling, product bundling, and recommender systems, we articulate the unique aspects of the new promotion framework: sequential purchases, products from different companies, and incentives for user acquisition. The paper also contributes to the broader literature on targeted advertising by empirically measuring the improved matching efficiency with analytics and individual data (Marotta et al. 2018).

Our paper also has managerial implications for app developers and platform managers. Our empirical results provide direct guidance to app developers considering CP as one of their ad strategies. Specifically, our results show that app developers should choose the right partner apps to maximize the ad effectiveness. With more knowledge on consumers' preferences on app consumption, they will be able to choose the source app that better fits their preference. From the mobile app platform's perspective, our paper has set an example of how to use app analytics data and machine-learning models to improve ad effectiveness and enhance the diversity of the market. Lastly, by comparing ad effectiveness with and without user behavior data, we discussed the trade-off between the cost of privacy and the benefit of market efficiency (Rafieian and Yoganarasimhan 2018b).

Our paper can have multiple extensions to address its current limitations. First, we can improve the empirical and simulation analyses with additional data sources. Specifically, the absence of ad impression data prevents us from conducting individual-level empirical analysis. For the same reason, we had to make some assumptions in the simulation analysis. We may collect more data (e.g., ad impression and app reviews) on the app market to conduct more detailed analysis at the individual level. For the prediction models, we may consider constructing more individual user features for improved model accuracy.

Second, our simulation used either a fixed unit price for download (download maximization) or the fixed sharing fraction among source or target apps (usage maximization). We could further improve the matching efficiency using auction models, as for Hatfield and Kominers (2017). In this case, the unit price is endogenously defined based on the bargaining power of both source and target apps. One requirement of this auction model is that each app developer needs to have an accurate understanding of the potential outcome when the app is matched to a specific app. However, in a related case of search ads auctions, Eliaz and Spiegler (2016) found that companies tend to bid on too many keywords because of the vague demand functions of their potential customers. In the CP market context, our proposed machine-learning approach can be used to accurately estimate the demand functions.

Lastly, our empirical and simulation analyses are based on observational data. In the econometric analyses, we tried our best to address the potential endogeneity issue by conducting the random matching experiment and using DID model with tight control variables. However, the results will be more reliable and convincing by conducting randomized field experiments to test the causal effects of using different app-matching algorithms and prediction models.

Endnotes

[1] See Statista, "Number of Apps Available in Leading App Stores as of 4th Quarter 2019," https://goo.gl/rjsjqK (accessed February 12, 2020).

[2] See Statista, "Number of Smartphone Users Worldwide from 2016 to 2021 (in Billions)," https://goo.gl/ykdQHc (accessed February 12, 2020).

[3] See TechCrunch, "The Majority of Today's App Businesses Are Not Sustainable," https://goo.gl/ZsdVzV (accessed February 12, 2020).

[4] See Adweek, "Apple's Search Ads Are Generating Conversion Rates Higher than 50%," https://goo.gl/sdXr6z (accessed February 12, 2020). The authors thank an anonymous reviewer for introducing this search ad mechanism.

[5] The majority of mobile apps are free to download, and the main revenue sources of free app developers are in-app purchases and in-app advertising. See https://goo.gl/sgDDya (accessed February 12, 2020).

[6] Tapjoy is a CP platform rewarding users with virtual currency for other app games in exchange for watching ads or downloading new apps. See https://www.tapjoy.com/in-app-advertising/ (accessed February 12, 2020).

[7] See HBS Digital Initiative, "Is Tapjoy Finding the Key to Mobile App Monetization?," https://goo.gl/9JMmZd (accessed February 12, 2020).

[8] See the data source: https://medium.com/tapjoy (accessed February 12, 2020).

[9] See Twitter, "Mobile App Promotion Guide," https://goo.gl/7do9JD (accessed February 12, 2020).

[10] See Facebook, "App Install Ads," https://goo.gl/FM77wa (accessed February 12, 2020).

[11] Tapjoy employs real-time bidding to meet the supply and demand of CP campaigns, which requires the target app developers to understand how to choose the right source app. Another CP platform, Tap for Tap, matches up apps based on a number for factors, such as app genre, app rating, and reviews. See https://goo.gl/iVkf2R (accessed February 12, 2020).

[12] See Business of Apps, "Promoting Your Mobile Apps Using Cross Promotion Networks," https://goo.gl/iVkf2R (accessed February 12, 2020).

[13] See Adweek, "Apple's Search Ads Are Generating Conversion Rates Higher than 50%," https://goo.gl/Dqx7ZL (accessed February 12, 2020).

[14] The Interactive Advertising Bureau (IAB) Internet Advertising Revenue Report shows that mobile ads have been growing substantially from 2012 to 2017, and it accounts for 57% of all digital ad revenue.

[15] SK Telecom is the largest telecommunications operator, and T Store is its app store.

[16] Note that more than 130,000 apps did not make it to the app store ranking.

[17] In industry practice, an app user is considered *organic* if the app's installation cannot be attributed with any promotions.

[18] Adbrix is a mobile app analytics tool available to any developer. A part of Adbrix-adopted apps conducts MDA or CP campaigns.

[19] See Word Stream, "The Comprehensive Guide to Online Advertising Costs," https://www.wordstream.com/blog/ws/2017/07/05/online-advertising-costs (accessed February 12, 2020).

[20] In Online Appendix A.3, we conduct additional analyses to show that the postdownload usage difference between the users from CP campaigns and MDA campaigns is not due to the intrinsic characteristics of the involved apps.

[21] Because the download ratio variable ranges from zero to one, we log transformed the download ratio. In addition, for observations whose download ratios are zero, we use the minimum value of all available log download ratio numbers.

[22] We acknowledge that simple apps with multimedia content may also have large file sizes.

[23] App updates can cause file size changes, but our data show that most apps do not have updates during CP campaigns. Thus, we use the initial file sizes on the first day of the CP campaign.

[24] In Table 3, target apps are on average ranked higher than source apps. But we note that there are more missing rankings for target apps (only 35,424 observations) than for source apps (40,387 observations). If the missing rankings are replaced with the maximum ranking (1,499), the average rankings of source and target apps are 99 and 235.

[25] In our data-collection period, 34 source and 27 target app developers released new apps. Thus, we measure developer experience levels as time-varying variables.

[26] See http://info.localytics.com/blog/large-screen-devices-show-34-more-time-in-app (accessed February 12, 2020).

[27] The choice of cosine similarity is not critical because other alternative similarity measures produce consistent results.

[28] We thank anonymous reviewers for the great suggestions.

[29] Based on the results in Table 1 and Figure 2 in Online Appendix A, we found that the analytics-enabled apps are very similar to the apps in CP campaigns in terms of their intrinsic app characteristics. To further test the validity of the control group, we conducted parallel tests by adding a set of interaction terms of biweekly dummies and whether the app conducted CP campaigns. The results are in Online Appendix F.

[30] We believe that ranking is a comprehensive measure for app success because it is decided by multiple factors, such as recent downloads, ratings, and reviews.

[31] See Stanford Topic Model Toolbox, http://nlp.stanford.edu/software/tmt/tmt-0.4/ (accessed February 12, 2020).

[32] See Keras: The Python Deep Learning Library, https://keras.io/ (accessed February 12, 2020).

[33] See TensorFlow: An Open Source Machine Learning Framework for Everyone, https://www.tensorflow.org/ (accessed February 12, 2020).

[34] See Scikit-learn: Machine Learning in Python, https://scikit-learn.org/ (accessed February 12, 2020).

[35] As a result, we have a 2% click-through rate in the augmented data for download prediction, which is consistent with an industry report. See https://www.acquisio.com/blog/agency/what-is-a-good-click-through-rate-ctr/ (accessed February 12, 2020).

[36] Note that the variables in the two tables are constructed from the same CP campaign data with the same source and target apps.

[37] We thank an anonymous reviewer for suggesting use of the AUPRC measure.

[38] Note that Logit and Lasso models do not require hyperparameter tuning.

[39] We did not use a target app usage-based prediction model because new target apps do not have sufficient usage data.

[40] According to our data source for the random matching experiment, each target app is randomly matched with all available source apps, and one user will be shown seven target apps at one time.

[41] We thank an anonymous reviewer for the suggestion to examine the scalability of the matching algorithms.

[42] Note that the matches are still generated from different matching algorithms and different features.

References

Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowledge Data Engrg.* 17(6): 734–749.

Adomavicius G, Bockstedt J, Curley SP (2015) Bundling effects on variety seeking for digital information goods. *J. Management Inform. Systems* 31(4):182–212.

Anderson C (2006) *The Long Tail: Why the Future of Business is Selling Less of More* (Hachette Books, New York).

Askalidis G (2016) The impact of large scale promotions on the sales and ratings of mobile apps: Evidence from Apple's app store. Working paper, Northwestern University, Evanston, IL.

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

Bhargava HK (2012) Retailer-driven product bundling in a distribution channel. *Marketing Sci.* 31(6):1014–1021.

Blei DM (2012) Probabilistic topic models. *Comm. ACM* 55(4):77–84.

Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. *J. Machine Learn. Res.* 3(1):993–1022.

Bresnahan T, Greenstein S (2014) Mobile computing: The next platform rivalry. *Amer. Econom. Rev.* 104(5):475–480.

Brynjolfsson E, Hu Y(J), Smith MD (2010) Long tails vs. superstars: The effect of information technology on product variety and sales concentration patterns. *Inform. Systems Res.* 21(4):736–747.

Burke R (2002) Hybrid recommender systems: Survey and experiments. *User Model. User-Adapted Interaction* 12(4):331–370.

Carare O (2012) The impact of bestseller rank on demand: Evidence from the app market. *Internat. Econom. Rev.* 53(3):717–742.

Davis J, Goadrich M (2006) The relationship between precision-recall and roc curves. *Proc. 23rd Internat. Conf. Machine Learn.* (Association for Computing Machinery, New York), 233–240.

Eliaz K, Spiegler R (2016) Search design and broad matching. *Amer. Econom. Rev.* 106(3):563–586.

Ershov D (2016) The effect of consumer search costs on entry and quality in the mobile app market. Working paper, University of Toronto, Toronto.

Evans DS, Schmalensee R, Noel MD, Chang HH, Garcia-Swartz DD (2011) *Platform Economics: Essays on Multi-Sided Businesses* (Competition Policy International, Boston).

Fong NM (2017) How targeting affects customer search: A field experiment. *Management Sci.* 63(7):2353–2364.

Gale D, Shapley LS (1962) College admissions and the stability of marriage. *Amer. Math. Monthly* 69(1):9–15.

Garg R, Telang R (2013) Inferring app demand from publicly available data. *MIS Quart.* 37(4):1253–1264.

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

Ghose A, Goldfarb A, Han SP (2012) How is the mobile Internet different? Search costs and local activities. *Inform. Systems Res.* 24(3):613–631.

Ghose A, Li B, Liu S (2019) Mobile targeting using customer trajectory patterns. *Management Sci.* 65(11):5027–5049.

Griffiths TL, Steyvers M (2004) Finding scientific topics. *Proc. Natl. Acad. Sci. USA* 101(Suppl. 1):5228–5235.

Guo, Hong, Xuying Zhao, Lin Hao, De Liu (2019) Economic analysis of reward advertising. *Production Oper. Management* 28(10): 2413–2430.

Hagiu A (2006) Pricing and commitment by two-sided platforms. *RAND J. Econom.* 37(3):720–737.

Han SP, Park S, Oh W (2016) Mobile app analytics: A multiple discrete-continuous choice framework. *MIS Quart.* 40(4):983–1008.

Hariri N, Mobasher B, Burke R (2012) Context-aware music recommendation based on latent topic sequential patterns. *Proc. ACM Conf. Recommender Systems* (Association for Computing Machinery, New York), 131–138.

Hatfield JW, Kominers SD (2017) Contract design and stability in many-to-many matching. *Games Econom. Behav.* 101:78–97.

Heckman JJ (1979) Sample selection bias as a specification error. *Econometrica* 47(1):153–161.

Hu W, Bolivar A (2008) Online auctions efficiency: A survey of eBay auctions. *Proc. Internat. Conf. World Wide Web* (Association for Computing Machinery, New York), 925–934.

Ifrach B, Johari R (2014) The impact of visibility on demand in the market for mobile apps. Preprint, submitted June 24, https://ssrn.com/abstract=2444542.

Johnson MD, Herrmann A, Huber F (2006) The evolution of loyalty intentions. *J. Marketing* 70(2):122–132.

Kamakura WA (2008) Cross-selling: Offering the right product to the right customer at the right time. *J. Relationship Marketing* 6(3–4):41–58.

Kamakura WA, Kossar BS, Wedel M (2004) Identifying innovators for the cross-selling of new products. *Management Sci.* 50(8):1120–1133.

Le Q, Mikolov T (2014) Distributed representations of sentences and documents. Xing EP, Jebara T, eds. *Proc. Internat. Conf. Machine Learn.*, *Proc. Machine Learn. Res.* 32(2):1188–1196.

Lee GM, Qiu L, Whinston AB (2016) A friend like me: Modeling network formation in a location-based social network. *J. Management Inform. Systems* 33(4):1008–1033.

Lee, G, Raghu TS (2014) Determinants of mobile apps' success: Evidence from the app store market. *J. Management Inform. Systems* 31(2):133–170.

Lee Y, O'Connor GC (2003) New product launch strategy for network effects products. *J. Acad. Marketing Sci.* 31(3):241–255.

Levin JD (2011) The economics of internet markets. NBER Working Paper No. 16852, National Bureau of Economic Research, Cambridge, MA.

Li MM, Huang Y, Sinha A (2017) Data-driven promotion planning for mobile applications. Working paper, University of Michigan, Ann Arbor.

Li S, Sun B, Montgomery AL (2011) Cross-selling the right product to the right customer at the right time. *J. Marketing Res.* 48(4):683–700.

Li S, Sun B, Wilcox RT (2005) Cross-selling sequentially ordered products: An application to consumer banking services. *J. Marketing Res.* 42(2):233–239.

Li X, Bresnahan T, Yin P-L (2016) Paying incumbents and customers to enter an industry: Buying downloads. Working paper, Stanford University, Stanford, CA.

Linden G, Smith B, York J (2003) Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.* 7(1):76–80.

Lops P, De Gemmis M, Semeraro G (2011) Content-based recommender systems: State of the art and trends. *Recommender Systems Handbook* (Springer, Boston), 73–105.

Luo X, Zhang J, Duan W (2013) Social media and firm equity value. *Inform. Systems Res.* 24(1):146–163.

Marotta V, Zhang K, Acquisti A (2018) The welfare impact of targeted advertising. Working paper, Carnegie Mellon University, Pittsburgh, PA.

McAlister L (1982) A dynamic attribute satiation model of variety-seeking behavior. *J. Consumer Res.* 9(2):141–150.

Molitor D, Reichhart P, Spann M, Ghose A (2015) Measuring the effectiveness of location-based pull advertising: A randomized field experiment. Working paper, Fordham University, New York.

Nalebuff B (2004) Bundling as an entry barrier. *Quart. J. Econom.* 119(1):159–187.

Natarajan N, Shin D, Dhillon IS (2013) Which app will you use next? Collaborative filtering with interactional context. *Proc. ACM Conf. Recommender Systems* (Association for Computing Machinery, New York), 201–208.

Oestreicher-Singer G, Sundararajan A (2012) Recommendation networks and the long tail of electronic commerce. *MIS Quart.* 36(1): 65–84.

Oliver RL (1999) Whence consumer loyalty? *J. Marketing* 63(4):33–44.

Petsas T, Papadogiannakis A, Polychronakis M, Markatos EP, Karagiannis T (2013) Rise of the planet of the apps: A systematic study of the mobile app ecosystem. *Proc. ACM Internet Measurement Conf.* (Association for Computing Machinery, New York), 277–290.

Qiu Y, Gopal A, Hann I-H (2017) Logic pluralism in mobile platform ecosystems: A study of indie app developers on the iOS app store. *Inform. Systems Res.* 28(2):225–249.

Rafieian O, Yoganarasimhan H (2018a) How does variety of previous ads influence consumer's ad response? Working paper, University of Washington, Seattle.

Rafieian O, Yoganarasimhan H (2018b) Targeting and privacy in mobile advertising. Working paper, University of Washington, Seattle.

Ramage D, Dumais ST, Liebling DJ (2010) Characterizing microblogs with topic models. *Proc. Internat. AAAI Conf. Web Social Media* (AAAI Press, Palo Alto, CA), 130–137.

Read D, Loewenstein G (1995) Diversification bias: Explaining the discrepancy in variety seeking between combined and separated choices. *J. Experiment. Psych. Appl.* 1(1):34–49.

Resnick P, Varian HR (1997) Recommender systems. *Comm. ACM* 40(3):56–58.

Rochet J-C, Tirole J (2003) Platform competition in two-sided markets. *J. Eur. Econom. Assoc.* 1(4):990–1029.

Roth AE (1984) The evolution of the labor market for medical interns and residents: A case study in game theory. *J. Political Econom.* 92(6):991–1016.

Roth AE (2008) What have we learned from market design? *Econom. J.* 118(527):285–310.

Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. *Inform. Processing Management* 24(5):513–523.

Schafer JB, Frankowski D, Herlocker J, Sen S (2007) Collaborative filtering recommender systems. Brusilovsky P, Kobsa A, Nejdl W, eds. *The Adaptive Web: Methods and Strategies of Web Personalization* (Springer, Berlin), 291–324.

Shi Z, Lee GM, Whinston AB (2016) Toward a better measure of business proximity: Topic modeling for industry intelligence. *MIS Quart.* 40(4):1035–1056.

Shin D, He S, Lee GM, Whinston AB, Cetintas S, Lee K-C (2020) Enhancing social media analysis with visual data analytics: A deep learning approach. *MIS Quart.* Forthcoming.

Simonson I (1990) The effect of purchase quantity and timing on variety-seeking behavior. *J. Marketing Res.* 27(2):150–162.

Simonson I, Winer RS (1992) The influence of purchase quantity and display format on consumer preference for variety. *J. Consumer Res.* 19(1):133–138.

Smith MD, Telang R (2009) Competing with free: The impact of movie broadcasts on DVD sales and Internet piracy. *MIS Quart.* 33(2): 321–338.

Soman D, Gourville JT (2001) Transaction decoupling: How price bundling affects the decision to consume. *J. Marketing Res.* 38(1): 30–44.

Sørensen M (2007) How smart is smart money? A two-sided matching model of venture capital. *J. Finance* 62(6):2725–2762.

Stremersch S, Tellis GJ (2002) Strategic bundling of products and prices: A new synthesis for marketing. *J. Marketing* 66(1): 55–72.

Venkatesh R, Mahajan V (2009) The design and pricing of bundles: A review of normative guidelines and practical approaches. *Handbook of Pricing Research in Marketing* (Edward Elgar, Cheltenham, UK), 232–257.

Wang C, Blei DM (2011) Collaborative topic modeling for recommending scientific articles. *Proc. ACM Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York), 448–456.

Weng J, Lim E-P, Jiang J, He Q (2010) TwitterRank: Finding topic-sensitive influential Twitterers. *Proc. ACM Internat. Conf. Web Search Data Mining* (Association for Computing Machinery, New York), 261–270.

Xu L, Duan JA, Whinston AB (2014) Path to purchase: A mutually exciting point process model for online advertising and conversion. *Management Sci.* 60(6):1392–1412.

Yin P-L, Davis JP, Muzyrya Y (2014) Entrepreneurial innovation: Killer apps in the iPhone ecosystem. *Amer. Econom. Rev.* 104(5): 255–259.

Zhong N, Michahelles F (2013) Google Play is not a long tail market: An empirical analysis of app adoption on the Google Play app market. *Proc. 28th Annual ACM Sympos. Applied Comput.* (Association for Computing Machinery, New York), 499–504.