CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Architecture of in-app ad recommender system
Authors	Mukherjee, Anik;Sundarraj, R. P.;Dutta, Kaushik
Publication date	2016-05
Original Citation	Mukherjee, A., Sundarraj, R. P. & Dutta, K. 2016. Architecture of in-app ad recommender system. In: Parsons, J., Tuunanen, T., Venable, J. R., Helfert, M., Donnellan, B., & Kenneally, J. (eds.) Breakthroughs and Emerging Insights from Ongoing Design Science Projects: Research-in-progress papers and poster presentations from the 11th International Conference on Design Science Research in Information Systems and Technology (DESRIST) 2016. St. John, Canada, 23-25 May. pp. 59-67
Type of publication	Conference item
Link to publisher's version	https://desrist2016.wordpress.com/
Rights	©2016, The Author(s).
Download date	2026-11-08 09:14:30
Item downloaded from	https://hdl.handle.net/10468/2567

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Architecture of In-App Ad Recommender System

Anik Mukherjee*, RP Sundarraj*, Kaushik Dutta$

*Department of Management Studies, Indian Institute of Technology, Madras
Sardar Patel Road, Chennai, Tamil Nadu 600036 – India.
$Department of Information Systems and Decision Sciences, Muma College of Business, University of South Florida, United States of America
{*anikit.jgec@gmail.com; *rpsundarraj@iitm.ac.in; $duttak@usf.edu }

Abstract: *Increased adoption of smartphones has caused mobile advertising to be the second-most revenue-generating medium among all forms of existing online advertising. Application (henceforth called app) developers try to monetize their apps by selling in-app ad-spaces to the advertisers (or ad-agencies) through various intermediaries such as ad-networks. Surveys, however, indicate that mobile ad campaigns are not as successful as they can be, in part due to inappropriate audience targeting, and in turn, user-apathy toward such ads. This motivates the need for a system, where both advertisers and mobile-app developers gain from the in-app advertising eco-system. In this paper, we propose an architecture of design-science artifacts for an ad-network, to meet the objectives of both these stakeholders.*

Keywords: Negotiation, Recommender System, Integer Programming Model.

1. Introduction

Online advertising has been growing on importance during the last decade. It comes in different forms such as display, digital audio and video, banner, social media, and mobile, and of these, display and mobile are among the most popular. An Internet Advertisement Bureau (IAB) report[1] indicates that more than 30% of the total online advertising revenue comes from mobile advertising, this being second only to the 38% market share of display advertising. In terms of revenue, too, online advertisement has exceeded the $50 billion mark in 2015, which suggests that mobile-ad revenue contributes more than $15 billion[1].

Research on display advertising has been dealt with from multiple perspectives, for example, from: (a) the publisher; (b) advertiser and (c) ad-networks. However, these results are not translatable to the mobile situation, owing to some fundamental differences. First, in the display scenario, typically, the advertisement placement happens through off-line negotiation. For example, if one needs to put an advertisement at CNN.com, s/he needs to negotiate directly with the Turner on the cost of putting such an advertisement. Second, display web sites resort to other ways of generating revenue such as selling user data, registration fee to access premium content, and fee to put up the content (such as a pharma company putting ads in WebMD.com). In contrast, the majority of mobile apps are free, and rely on advertisement-based revenues.

[1] Available at: http://www.iab.com/wp-content/uploads/2015/10/IAB_Internet_Advertising_Revenue_Report_HY_2015.pdf

Additionally, mobile apps market is populated by mobile apps developed by not so well known publishers that do not have the resources to negotiate ad-pricing with individual advertises or their agencies. Therefore, they resort to programmatic ad-buying through ad-networks. However, the lack of any standardized framework that would address the need of publishers, advertisers and ad networks all together had made the mobile ad ecosystem unreliable[2,3]. Most of the ad-networks deliver ads that are optimized for advertisers (giving maximum click or maximum impressions) and does not address the need of the publishers. In this paper, we propose the design of an architectural framework for an ad-network that seeks to balance the needs of both publishers and advertisers. Rather than treating the interest of the publishers and advertisers dis-jointly, in our proposed framework, we address them together, to determine which advertisement to deliver when and at what price.

2. Literature review

Keyword search is one way of audience targeting[4] in website, where keywords are extracted from website contents and publishers bid for key words. According to Hermann et. al. [1], if one website contains certain keywords, advertisement targeted to those keywords will be displayed in the website. Keyword based advertisement is not applicable to in app advertisement. In the past, researchers had tried to solve web-site advertising problem from scheduling problem's perspective. Menon et al. [2] has proved web based advertising problem to be NP-Hard problem. Deane et al. [3] has followed artificial intelligence based technique to solve online ad-scheduling problem. However, in mobile apps ad-placement problem is not only a scheduling problem, as discussed later in the paper it needs to address several other aspects such as audience targeting, price negotiation, target negotiation etc. In mobile programmatic ad delivery platform, ad-networks act as the bridge between advertiser and publisher to make automated in-app advertising happen in real time. However, there is paucity of research in this domain. Recently, [4][5] proposed a mathematical model and rule generation approach to maximize the revenue of publishers. In this paper, we aim to extend that work to the next level by proposing the architecture of entire in-app advertising eco-system that addresses publishers', ad-networks' and advertisers' needs jointly.

3. Design Science Research Problem Statement and Objective

Of the five design steps [6][7] (Problem Statement, Objective, Design Artifacts, Demonstration and Evaluation), in this work-in-progress paper, we have addressed the first three. Further, part of our proposed architecture has been evaluated in [5][4].

3.1. Problem Statement

[2] Available at: http://www.emarketer.com/Article/Marketers-Wasting-Money-on-Mobile-Ad-Clicks/1009351

[3] Available at: http://www.mobilemarketer.com/cms/news/research/9015.html

[4] Available at: https://www.google.com.sg/adwords/

Advertisers and *publishers* represent the two main stakeholders in the online advertising eco-system. Ad agencies help interested advertisers to register in their ad-networks to display their ads in different online websites or media.

A stream of ads from advertisers (or their agencies) is coming to ad-networks to be pushed into multiple available ad-slots available to publishers. The number of available slots at any instant depends on the number of users of apps at that instant. Owing to limited number of slots, not all the ads can be picked up simultaneously. Hence, we need to select ads in such a way that it contributes enough to the return-of-advertisement-dollar to advertisers (measured by number of clicks or number of target impressions), as well as to the revenue generation of publishers. Moreover, the problem is complex as well, as given below.

Issue 1: Impression Requirement Negotiation

In order to do effective app-targeting, apps need to be ranked on some measure. However, what is unique about apps, as compared to websites, is that this ranking can be very unstable [8]. The number of users of a particular app varies widely even within a day, and except for a few apps, the others loose their popularity after a few days. This makes smartphone app-targeting difficult. Thus, advertisers cannot negotiate for impression requirements and price over a long scheduling horizon, as is the case with websites. Instead, they need to negotiate for a small time-horizon of, say, a few hours. Short time-frames can lead to inefficient negotiation, in turn resulting in a loss to both parties. Thus, any system that is created must address the issue of short-term, albeit effective, negotiation.

Issue 2: Efficient ad-selection for publishers

From the publisher perspective, revenue must be maximized, by allowing advertisers to push ads in their online available ad-spaces. When an ad is placed in a slot in an app, the publishers of that app gets revenue. Additionally, if the user of the app clicks on the placed ad, there is possibility of additional revenues to publishers. However, all decisions regarding the selection of ads are to be made by ad-networks in the order of tens of milliseconds; IAB report[1] states that the time must be under 50 milliseconds. Given the complexity of the problem, optimal ad-selection can be a challenge.

Issue 3: Proper Audience Targeting for advertisers

Researchers have done extensive research in website advertising where cookies keep track of the behavior of web site users. The absence of cookies in mobile apps is a key issue. Audience targeting is a crucial part for advertisers to maximize their return-on-ad-spend.

3.2. Objective of the problem

The overarching objective of our research is to develop an automated in-app advertisement system for ad-networks, by considering the perspectives of multiple stakeholders. To address the three issues given above, we define three specific objectives:

➔ *Objective 1*: Ad-Agency Publisher Negotiation

➔ *Objective 2*: Publisher's Ad-Selection
➔ *Objective 3*: Online Ad Recommendation

To address the *Issue1* of our problem, we aim to negotiate with the ad-agencies and publishers in *Objective1* whereas *Objective2* addresses the *Issue2* by selecting ads in such a way that it generates maximum revenue of the publishers. Last challenge of our problem is efficient audience targeting. *Objective3* of our problem is to develop an Online Ad-recommendation engine to solve *Issue3*. This directly depends on the output of first two objectives and its own advertiser based internal algorithm. Once the *Objective3* is solved, our system places the best ad in the corresponding mobile app.

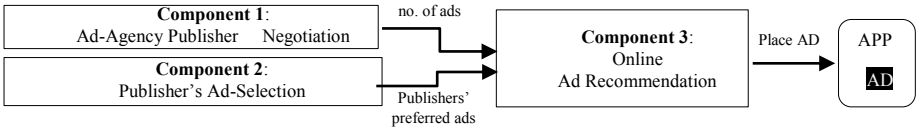


Figure 1: High Level Architecture of In-App Recommender System

The high level architecture of our system is given in Figure 1. The detailed architecture of our system is presented in Section 4. According to Figure 1, *Component 1* and *Component 2* provide inputs to C*omponent 3* which decides the single best ad to be pushed into each individual online available ad-slot within 50 milliseconds. First two components run offline based on machine learning based algorithm. The last component of our system forwards ads based on greedy recommendation algorithm in real time. This happens online. *Objective 1* of the system is partly implemented in our previous works [5] [4].

4. Design Science Artifacts

Artifacts are the core components of a DSR-based system. According to [6], *artifacts*, broadly include, constructs, models, algorithms and system instantiation; the artifacts we have proposed herein are algorithms pertaining to each of the three components in Figure 1.

4.1. Component 1: Ad-agency Publisher Negotiation

In in-app advertising, a direct-buying mechanism[5] is used to decide the number of ads that needs to be displayed by a publisher. On behalf of advertisers, ad-agencies buy available online ad-spaces. This is often done through a negotiation process [9]. Given the advances in negotiation support systems, our architecture entails a negotiation engine described below.

4.1.1. Artifact 1: Advertiser-Publisher Negotiation

The advertiser-publisher negotiation artifact is a multi-agent negotiation system where ad-agency represents buyer and publisher represents seller (see Figure 2). Since

[5] http://www.vantagelocal.com/display-advertising-direct-buy-vs-rtb/

negotiation can be a long-drawn process and can also be involved, we propose that an automated negotiation-system be used for this purpose. As shown in the figure, in the online negotiation agent, multiple ad-agencies negotiate with publishers via negotiating agents known as ad-agent and pub-agent, respectively. The well-known alternate-

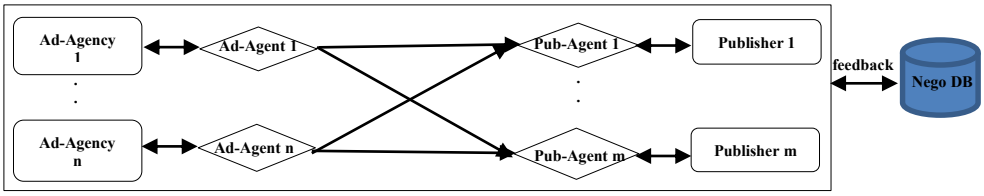


Figure 2: Online Negotiation Agents interaction

offers protocol [10] is used. Agents are guided by different parameters to make the trade happen. Parameters associated with the buyer include click-through-rate, conversion rate, and expected number of audience reachable. Example-parameter for the seller include monetary value for the publisher, minimum/maximum number of ads advertiser is willing to display, and payment for impressions. Details on how to structure electronic-negotiation can be found, for example, in [11], [12].

4.2. Component 2: Publisher's Ad-Selection Component

Once the required number of ads is decided by the negotiation engine, we need to develop offline methods to support the online component in Section 4.3. We adopt a two-step procedure for this purpose [5, 6]: a revenue-based optimization model whose results are used by a rule-generation system.

4.2.1. Artifact 2: Revenue-based Optimization Model

The revenue-based model is used to generate selection/rejection decisions for past dataset. The immediate past is used to make the results relevant for practical application. An Integer Linear Programming (ILP) model with revenue maximization objective is used. The model's revenue function includes Click-through cost, impression cost and conversion cost. Constraints include: (a) limitations on the availability of the total number of ad-spaces; (b) pacing constraints to spread the budgetary spending across different periods of the planning horizon; (c) impression requirements; and (d) a constraint to ensure that a particular ad-space can have no more than one ad at any given time. Details of the model can be found in [4], [5]; the model is NP-complete.

4.2.2. Artifact 3: Rule Generation

The term rule refers to an authoritative statement of what to do or not to do in a specific situation issued by an appropriate person or body. In our context, the rationale for the rule-generation is two-fold. First, rules represent a practical approach to provide a real-time decision in the allowed short time-window, especially given the complexity of the optimization model. Second, our rule generation is based on the optimization model's results (with the immediate past data), thereby seeking to ensure the accuracy and usefulness of the rules to the extent possible. We generate two kinds of rules:

positive rules (based on selected ads) and negative rules based on rejected ones. The rule generation artifact consists of three main parts [5, 6]: (a) Data Classification Engine; (b) Rule Engine; and (c) Rule Optimization Engine.

4.2.2.1. Data Classification Engine:

Preliminary observation with the optimization model's results indicated that the selection/rejection decision depends on the value ranges of the data. In our case, each tuple of the dataset represents individual ad with six different attributes such as Click-through Rate (CTR), Click Price (CTRPRICE), Conversion Rate (CONV), Conversion Price (CONVPRICE), Price for each impression (IMPPRICE), Winning bid price of ad (BID). We use K-Means clustering to divide each attribute of the dataset into predefined clusters.

4.2.2.2. Rule Engine

The clustered data set is fed into the rule engine. The apriori algorithm, drawn from the area of machine learning, is used by the rule engine. The apriori algorithm is used, for example, in market-basket analysis to find out the most-sold products of a store/company, based on the probability of occurrence of that product in store's transactional database. This helps decision makers to pinpoint the reason behind an ad campaign for that product. We apply this idea to find out the patterns among the aforementioned six different attributes and generate rules of the form X, Y ➔ Z, where the antecedents X, Y imply that Z has occurred. Two control parameters are used to moderate the number and quality of the rules: *support* representing the percentage of times in which the antecedents have occurred and *confidence,* which stands for the probability of occurrence of the rule. Thus, based on support and confidence values, the set of rules can change.

4.2.2.3. Rule Optimization

From the previous discussion, we can see that lower (higher) support and confidence values implies higher (lower) number of rules. The robustness of our proposed architecture is dependent on the quality of rules generated. Large number of rules could tend to improve accuracy (i.e., quality of the selected ads), but also increase decision-making time. This implies that the support and confidence values must be optimized. Genetic algorithm based rule optimization technique can be used [13] to come up with the best set of rules from among the pool of rules.

4.3. Component 3: Online Ad Recommendation

Online Ad Recommendation engine is the last component of in-App advertising architecture. It consists of two sub-components: Recommender System which works from the advertiser's perspective and greedy algorithms which picks up the one final best ad based on outputs of all other components and few criteria.

4.3.1. Artifact 4: Recommender System

According to Forbes[6], advertisers are spending money on mobile advertising without getting any significant profit. Thus, the recommendation system is used to help advertisers; specifically, from the real-time stream of ads, it recommends ads to ad-spaces. It takes into account context-specific information, and hence, can help draw audience attention; in turn, this provides the possibility of increasing revenue. Ads are ranked by the ad-scoring algorithm inside recommendation engine, based on similarity of ad with respect to the targeted in-app ad-space. The "Top-N" ads are then sent to the rule-based greedy algorithm for the next level of filtering.

4.3.2. Artifact 5: Rule Based Greedy Algorithm

The rule-based system takes three inputs: (a)negotiated number of impressions from negotiation engine; (b) the ads after applying the rules from the (offline) rule-generation artifact on real-time stream of ads; and (c) the ads from the recommendation system. The greedy algorithm applies the rules and checks negotiation results to make an intermediate select/reject decision on each incoming ad. This decision is compared with that of the recommendation systems and priority will be given to that which appears in both sets, thereby attempting to account for the priorities of both the advertiser and the publisher. The detailed architecture summarizing all the discussion is given in Figure3.

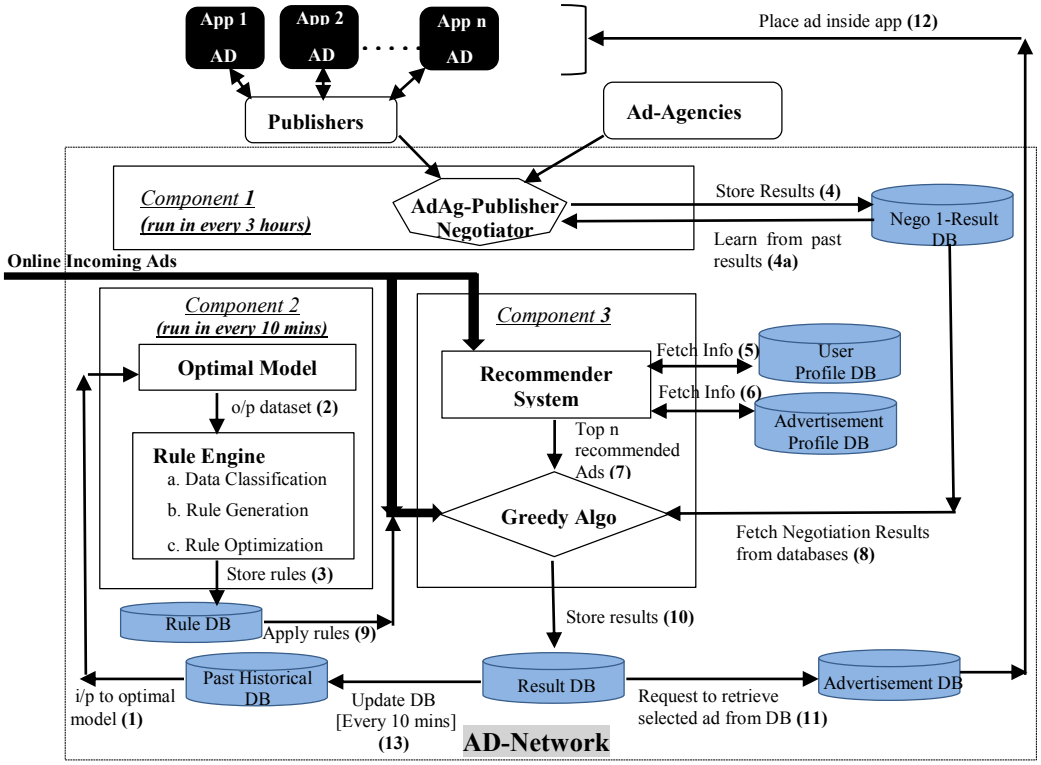


Figure 3: Architecture of In-App Ad Recommender System

[6] Available at: http://www.forbes.com/sites/baininsights/2012/07/03/when-is-advertising-a-waste-of-money/#67b8000f70c2

5. Conclusions

In this research we present the design of an architectural framework for mobile ad network that addresses the need for publishers (revenue maximization) and the need for advertisers (maximization of the return on advertisement dollar) simultaneously. Unlike past research that has dealt with either publisher or ad network or advertisers, in this research we take a holistic approach by proposing a rule based negotiation engine. We have described the key component of this complex system. The future research on this will focus on developing the details of the individual components. In future we also intend to implement and evaluate the proposed framework in a simulated environment and compare with the existing silo based approach.

6. References

1. Hermann, C., Francine, C., Ayman, F., GREENE, D.: Server-based keyword advertisement management, http://www.google.com/patents/US20050144064, (2005).
2. Menon, S., Amiri, A.: Scheduling Banner Advertisements on the Web. INFORMS J. Comput. (2004).
3. Deane, J.: Hybrid genetic algorithm and augmented neural network application for solving the online advertisement scheduling problem with contextual targeting. Expert Syst. Appl. 39, 5168–5177 (2012).
4. Mukherjee, A., Sundarraj, R.P., Dutta, K.: Online Programmatic Ad-Placement for Supply Side Platform of Mobile Advertisement : An Apriori Rule- generation Approach. Workshop on Information Techologies and Systems. (2015)
5. Mukherjee, A., Sundarraj, R.P., Dutta, K.: AN ONLINE ALGORITHM FOR PROGRAMMATIC ADVERTISEMENT PLACEMENT IN SUPPLY SIDE PLATFORM. Pacific Asia Conference on Information Systems.(2015)
6. Hevner, A.R., Chatterjee, S.: Design Research in Information Systems. Des. Res. Inf. Syst. 28, 75–105 (2004).
7. Peffers, K.E.N., Tuunanen, T., Rothenberger, M.A.M.A., Chatterjee, S.: A design science research methodology for information systems research. J. Manag. Inf. Syst. 24, 45–77 (2007).
8. Petsas, T., Papadogiannakis, A., Polychronakis, M., Markatos, E.P., Karagiannis, T.: Rise of the planet of the apps. In: Proceedings of the 2013 conference on Internet measurement conference - IMC '13. pp. 277–290. ACM Press, New York, New York, USA (2013).
9. Tawfik Jelassi, M., Foroughi, A.: Negotiation support systems: an overview of design issues and existing software. Decis. Support Syst. 5, 167–181 (1989).
10. Rubinstein, B.Y.A.: Perfect Equilibrium in a Bargaining Model Author (s): Ariel Rubinstein Published by : The Econometric Society PERFECT EQUILIBRIUM IN A BARGAINING MODEL. Econometrica. 50, 97–109 (2014).
11. Kersten, G.E., Lai, H.: Negotiation support and e-negotiation systems: An overview. Gr. Decis. Negot. 16, 553–586 (2007).
12. Sundarraj, R.P., Mok, W.W.H.: Models for human negotiation elements: Validation and implications for electronic procurement. IEEE Trans. Eng. Manag. 58, 412–430 (2011).
13. Christian, A.J., Martín, G.P.: Optimization of association rules with genetic algorithms. Proc. - Int. Conf. Chil. Comput. Sci. Soc. SCCC. 193–197 (2011).