CLICK VOLUME POTENTIAL MAXIMIZATION IN AFFILIATE NETWORK

Krishna Kumar Tiwari and Ritesh Ghodrao

InMobi Technology Services Pvt Ltd, Bangalore, India

ABSTRACT

An affiliate network is all about running advertiser's campaign (acquire new user, download campaign) on multiple/chain of ad-tech companies (aka affiliates), most of the affiliate in affiliate network deals with a huge volume of clicks (pretty much 500M to 1.5B roughly, with click QPS varying from 10K to 25K). Only a small fraction for clicks leads to conversions which leads to revenue to affiliate but hosting a huge volume of clicks costs a lot based on engineering setup. The real challenge here is that we need to maintain the profit after paying for the infra cost, hence it becomes mandatory to optimize on infra cost and revenue equation.

In this paper, we have presented a unique way of modeling the Infra-to-Revenue equation based on click volume and provided a Knapsack way of solving the Infra-to-Revenue equation and maximising our revenue by keeping a constraint on infra cost, which we are calling as CVPM (click volume potential maximization). We have compared CVPM with greedy based optimizations and concluded that CVPM outperforms many of these approaches in most of the real scenarios.

KEYWORDS

Infra cost optimization, Click optimization, 0-1 Knapsack, Adtech optimizations

1. INTRODUCTION

In a typical affiliate ad network, advertisers create campaigns with defined KPIs and payouts, which affiliates pick and run on their available supply. In scenarios where the affiliate cannot meet the advertiser's KPIs, they broker this campaign to other affiliates as offers with some revenue margin. Thus a campaign may actually be picked by a chain of affiliates before reaching the publisher, the indirect affiliates involved are termed as sub-affiliates.

These affiliate platforms simulate a dynamic marketplace by brokering where they are expected to operate at a high scale, maintaining high availability and low latency. This mesh of affiliates and offers helps advertisers achieve better performance on a wide supply with less direct control. The involvement of various affiliates makes the ecosystem highly competitive, where everyone must ensure robust systems with high availability.

Natarajan Meghanathan et al. (Eds) : DMAP, CSITA, ARIN, ISPR, AISO - 2019
pp. 47-53, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91605

The growing number of mobile apps has broadened the potential of mobile apps as advertising avenues, leading to higher offers and affiliates ecosystem. The increasing spends on mobile advertising have also attracted various fraudsters [1], which use different methods to increase the number of clicks. This makes it important to take wiser design decisions and the use of sustainable yet scalable infrastructure. The challenge to support these becomes more grave as the margins involved in affiliate networks are usually minimal. The infrastructure cost usually increases with the volume of clicks, which can range from 20k-40k per second, however, the corresponding revenue may not increase in the same order. Hence, in order to gain maximum profit with these constraints, it is often a better approach to make the best possible trade-off based on underlying infrastructural costs and limit the volume of clicks. This would not only provide better conversion rates to advertisers but also help identify and prevent fraudulent clicks. The prevention of frauds also helps derive genuine offer performance insights with better revenue per click. The ultimate intention is to identify the offers-affiliates which would limit the volume of clicks and help achieve maximum possible revenue.

2. PROBLEM OVERVIEW

We have seen more than one billion (1.4B) clicks in our system in the last month which is around 35% higher compared to last month. Usually in AdTech world, when we have more genuine clicks, ideally it should lead to more conversions hence more revenue but in our case, we realized the revenue increase is pretty much the same (<1% change) compared to old revenue trends hence, handling these extra 35% clicks on this scale is not the right approach.

Let's break down the problem statement in two part

2.1. Infra vs Revenue equation

Each click carries unique identification ids along with information about the time of the click, device details, location, etc. This information has to be securely processed, stored for further reporting and analysis. Thus, processing a high volume of clicks needs more infrastructure, the contribution of fraudulent or extraneous clicks becomes a cost overhead here as the minimal revenues generated by these clicks cannot cover the infrastructure costs.

2.2. Fraud Ratio in the System

Recent advertising fraud research claim that about 15% of advertising spends is lost due to fraud [3]. This approximates to about 1 out of every 5 clicks being fraudulent. While having more clicks may also indicate good performance, it may be exposed to fraud. The AdTech industry overall faces a lot of fraud challenges, for example, bot clicks [2], incentivized traffic, etc. In our case it was majorly affecting our profit, reputation(Advertiser loses trust which affects the reputation) and bad CVRs.

3. Related Work

As we observed from the problem overview section that, the problem which we are trying to approach is coming down as choosing a trade-off point between infrastructure costs and revenues while maintaining minimum fraud in the system. In the AdTech world, this problem is often ignored by many organizations where the primary focus is on revenues as compared to fraud status. The complexity of identifying and blocking offer-affiliates that generate unproductive clicks remains a barrier for several affiliates from adopting cost optimization measures. While several pieces of research are done for minimizing the ad-fraud[4,5,6], there is not much talk of optimizing the infrastructure against revenues. Therefore, there is a dire need for simplified and holistic approaches for measuring the true potential of clicks whilst considering the infrastructural costs and revenue potential.

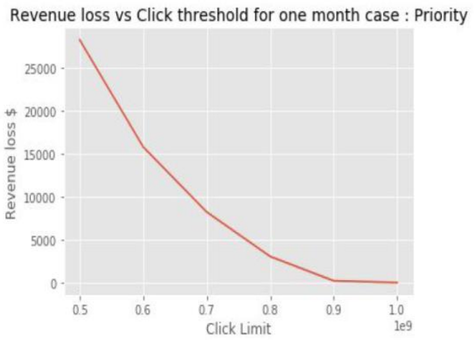
Based on our previous experiment and industry domain knowledge, we identified that organizations approach this problem by manually figuring out performance statistics. The most common indicators are based on Conversion Rates(CVR), Revenue per Click(RPC) for identifying and eliminating the bad click investors from the system..

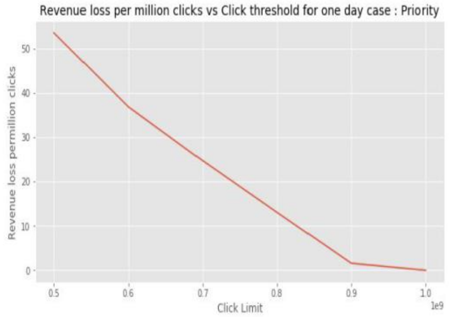
We did an optimization in the past which takes revenue per million click information for each click investor.

- Sort clicks investors(offer-affiliate combination) in revenue per million click order.
- Choose top RPC investors till $\sum_{i=1}^{k} w_i <= W$

For example, if we take the click capacity bucket as W where it can range from 500M to 1Billion, here is the performance of revenue per million click priority approach..

$$W = [500M, 600M, ...1B]$$





It is visible that keeping fewer clicks infra leads to lesser revenue as shown in the above images. he gradient is quite significant and uniform, the revenue impact is much higher across 500Mn to 00Mn, but the revenue loss is notably much lesser compared to 1Bn. This study clearly indicates an almost-linear relationship between the number of clicks and revenue to a certain extent. Thus ess number of clicks definitely lead to lesser revenues, however, the small revenue loss difference between 900Mn and 1Bn clicks encourages the idea of optimizing the number of supported clicks, optimizing the infrastructure cost to maximize the profit margin.

4. Click Volume Potential Maximization Model

4.1. Mathematical Formulation

In an affiliate ad-network, the affiliates try to maximize their ad-clicks, which can be modeled as click investment. An affiliate drives clicks across it's offers based on their supply constraints and revenue potentials, so as to maximize their profit. Let us consider the Offer & Affiliate combination as an item in the system and define:

item[i]= function(offer[j], affiliate[k])
value[i]= revenue(offer[j], affiliate[k])
weight[i]= click_investment(offer[j], affiliate[k])

Maximize

$$\sum_{i=1}^{k} value[i]$$

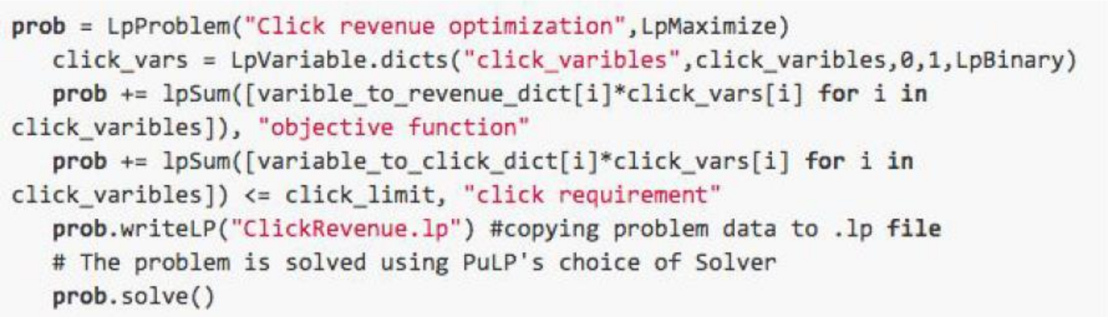
Subject to (given System capacity of handling click volumes)

$$\sum_{i=1}^{k} weight[i] \quad <= W$$

$$\text{k <= len(Items)}$$

Given Revenue vs Infra cost function invest(clicks) = $X.

Finding out the right W which can help to obtain Max(Profit). The Final Objective is to find out the Items which we can remove/block from the system to check click volume and maximize its revenue potential. The pseudocode to do this is mentioned below.

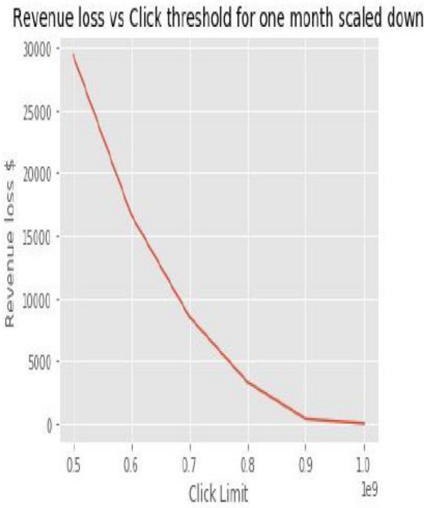


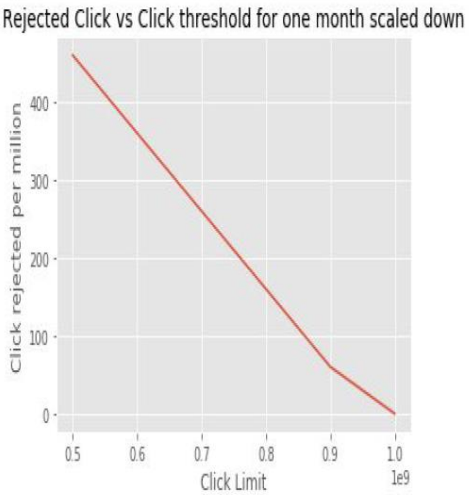
```
prob = LpProblem("Click revenue optimization",LpMaximize)
   click_vars = LpVariable.dicts("click_varibles",click_varibles,0,1,LpBinary)
   prob += lpSum([varible_to_revenue_dict[i]*click_vars[i] for i in
click_varibles]), "objective function"
   prob += lpSum([variable_to_click_dict[i]*click_vars[i] for i in
click_varibles]) <= click_limit, "click requirement"
   prob.writeLP("ClickRevenue.lp") #copying problem data to .lp file
   # The problem is solved using PuLP's choice of Solver
   prob.solve()
```

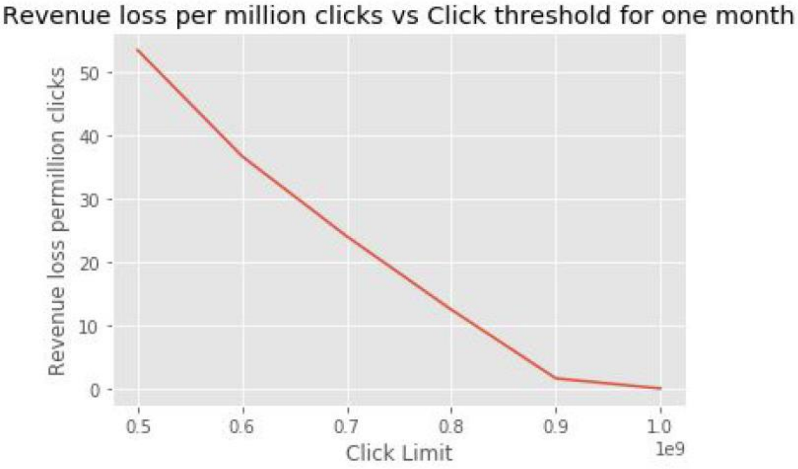
4.2. Experiments

We did multiple experiments of CVPM on past months of data and below is a snapshot of CVPM run on October data.

W = [500M, 600M, ...1B]


Revenue loss vs Click threshold for one month scaled down

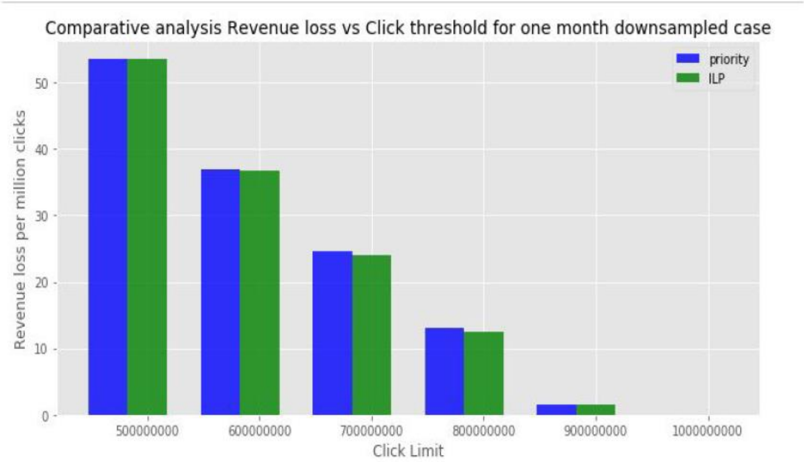

Rejected Click vs Click threshold for one month scaled down

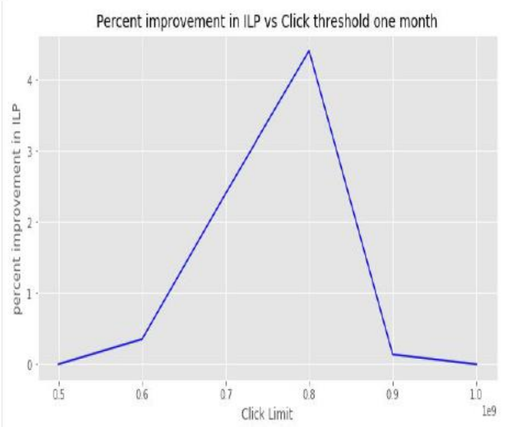

Revenue loss per million clicks vs Click threshold for one month

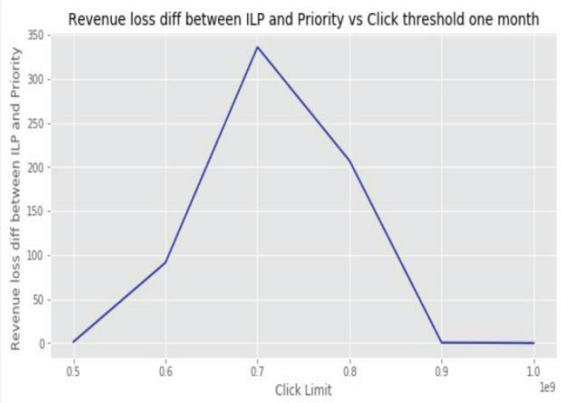
We can see that the number of clicks and revenue is non-linear relation after a certain point as shown in this picture, we have seen the same in greedy approach also, so the point here is we need to come up with a trade-off to choose the right click threshold to maximize the revenue.

4.3. Evaluations

We compared the CVPM approach with priority based greedy approach and found out that CVPM is finding the trade-off rightly which is better by 3-5% in terms of revenue.







In our experiments we found that using CVPM approach helped us to make $350 extra dollars per day compared to priority RPC approach.

5. Conclusions

Based on results obtained after following these approaches, we found that the outcomes of the CVPM model are overall better than traditional approaches. This model also generated results better than the Priority model, approximately to a saving of around 3-5%. The Priority model is a much more direct model for identifying the approximate maximum volume of clicks that can be supported without significant loss in revenue, however, it is not dynamic, and involves affecting all affiliates equally, thereby not considering the false positive. On the other hand, the CVPM approach is easy to implement, dynamic in nature and generates better Infra/Revenue optimization. The CVPM approach provides a granular approach of identifying extraneous click on an offer-affiliate level, which can further be drilled down to a more granular level. For ad-networks operating at similar scales, CVPM provides a new dimension for maximizing profits by not focusing unidirectionally on revenues with more number of clicks. The infrastructure costs are considered and evaluated against the click volume to broadly analyze the click patterns and maximize the potential of clicks.

6. Future Work

This work can be easily extended to the next level by bucketizing items considering sub-affiliates and run these experiments.

Item[i]= F(Offer[j], Affiliate[k], SubAffs(l))

Formulate the problem statement using more features like invalid Clicks, and solving 2-D constraint KNAPSACK Problem.

Acknowledgements

The authors would like to thank the whole Wadogo Team from InMobi and Manik Khandelwal (BITS Pilani).

References

[1] Online Advertising Fraud Neil Daswani, Chris Mysen, Vinay Rao, Stephen Weis, Kourosh Gharachorloo, Shuman Ghosemajumder, and the Google Ad Traffic Quality Team From the forthcoming book, Crimeware, edited by Markus Jakobsson and Zulfikar Ramzan, Copyright (C) 2008 Symantec Press.

[2] The anatomy of Clickbot.A, Daswani, Neil and Stoppelman, Michael, 2007

[3] DECAF: Detecting and Characterizing Ad Fraud in Mobile Apps. Bin Liu, Suman Nath, Ramesh Govindan, Jie Liu

[4] Quantifying Online Advertising Fraud: Ad-Click Bots vs Humans Adrian Neal, Sander Kouwenhoven, 2015

[5] Online Ad-fraud in Search Engine Advertising Campaigns, Andreas Mladenow, Niina Maarit Novak,2015

[6] Detection of Click Spamming in Mobile Advertising, Safiye Şeyma Kaya, Burak Çavdaroğlu,2019

[7] Stochastic Models for Budget Optimization in Search-Based Advertising, S. Muthukrishnan, Martin Pál, Zoya Svitkina, 2007