Click Volume Potential Maximization Model in Affiliate Network

Krishna Kumar Tiwari, Ritesh Ghodrao

InMobi Technology Services Pvt Ltd

Outline

The Problem

Our click hosting platform serves advertisements, captures the clicks and store them.
The cost of infrastructure naturally remains a significant factor in our click hosting cost given its scale. During peak times, we received around 1.4Bn ad-clicks on our platform.

Motivation:

- The increase in revenue was minimal as the clicks nearly doubled from 700Mn to 1.5Bn.

Extraneous clicks subsequently lead to:

- Poor conversion rates
- Reduced profit margin
- Impacts reputation & advertiser's trust

Mathematical formulation

Our affiliate partners are doing click investments on offers to maximize their profits. Considering the Offer and Affiliate combination as an item in system, we can define the following:

Item[i]= combination(Offer[j], Affiliate[k])

Value[i]= revenue(Offer[j], Affiliate[k])

Weight[i]= click_investment(Offer[j], Affiliate[k])

Mathematical formulation

- Given system's capacity of handling maximum click volumes **W**, the objective is to maximize

$$\sum_{n=i}^{k} Values[i]$$

Subject to,

$$\sum_{n=i}^{k} Weight[i] <= W \textbf{ and } k = length(Item).$$

Basis the Revenue vs Infra cost function obtained, invest **X** clicks

- Finding out the appropriate **W** can help to obtain max(Profit). This can be extended to find Items which can be removed/block from system to check click volume and maximize it's revenue potential .

Key Levers

- Data sufficiency at offer-affiliate combination
- Conversion Rate (CVR)
- Revenue Per Click (RPC)
- Infra-Revenue equations
- Dynamicity of model

Approaches

Naive Approach

1. Figuring out thresholds based on CVR, RPC
2. Blocking offer-affiliate combinations that breach calculated thresholds

DRAWBACKS

Can not be scaled dynamically for varying data set and thresholds.

Naive Approach ++

1. This is primarily based on revenue per million clicks
2. Optimize number of clicks based on maximum revenue per million clicks at Item level
3. Sort the items in order of decreasing RPMC
4. Choose top RPC Items till $\sum_{n=i}^{k} w[i] <= W$
5. The remaining items are offer-affiliate combinations that can be dropped

CVPM Model

Applying Integer Linear Programming to solve 0-1 KNAPSACK problem defined in mathematical formulation

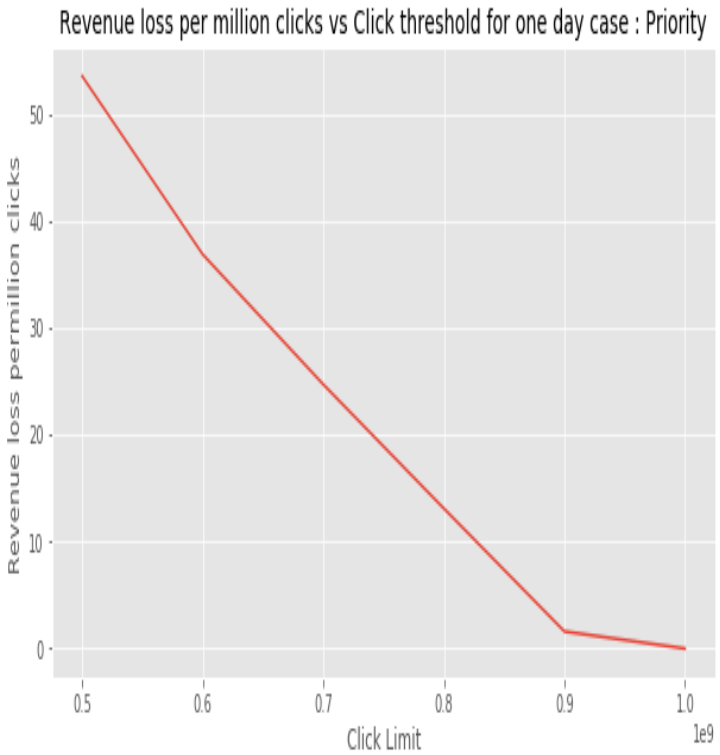
1. Select subset which provides maximum of click volume possible, i.e. **W**
2. Optimize for max(Profit) given total click investment as **X**
3. Find and block Item combinations that do not appear in result subset

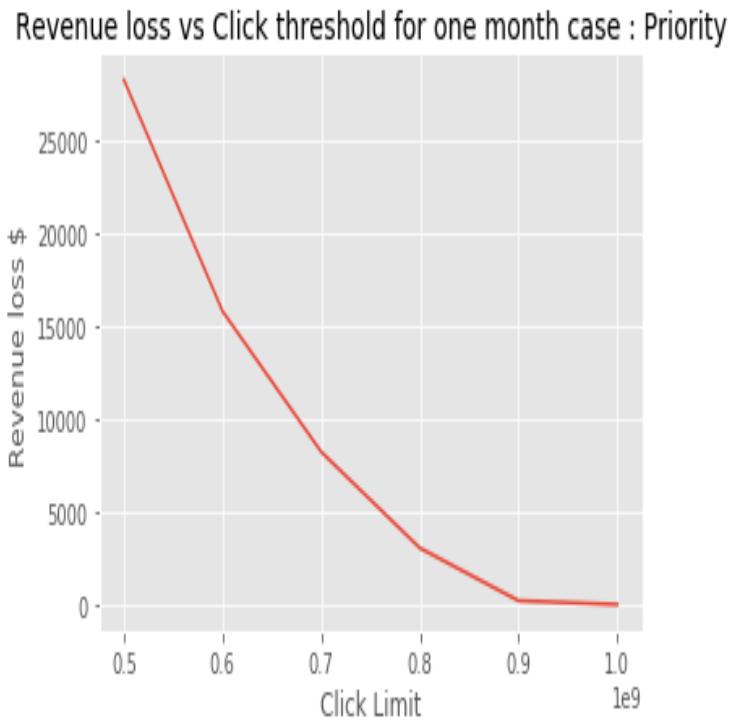
CVPM Model

```
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()
```

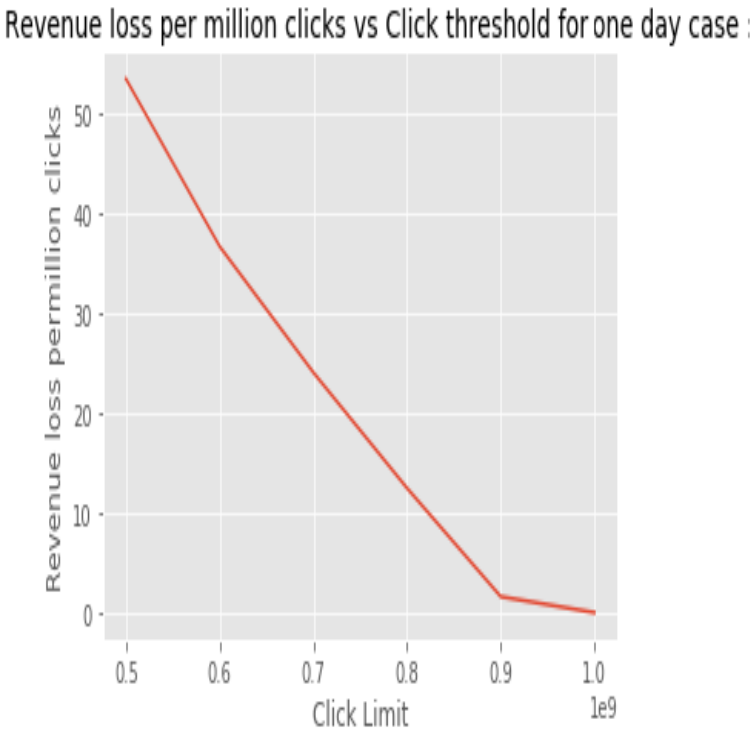
Results

MRPC Priority Model



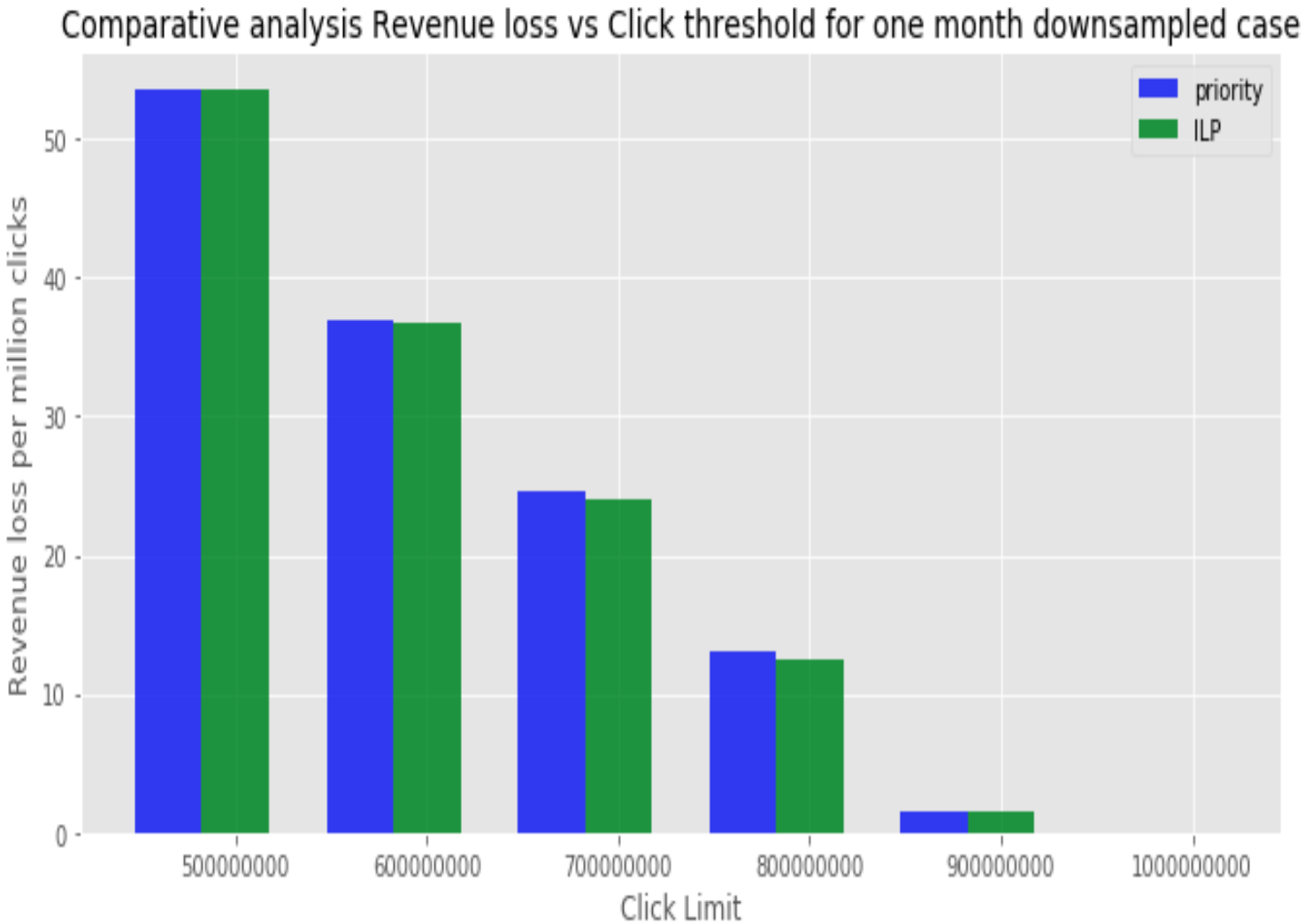


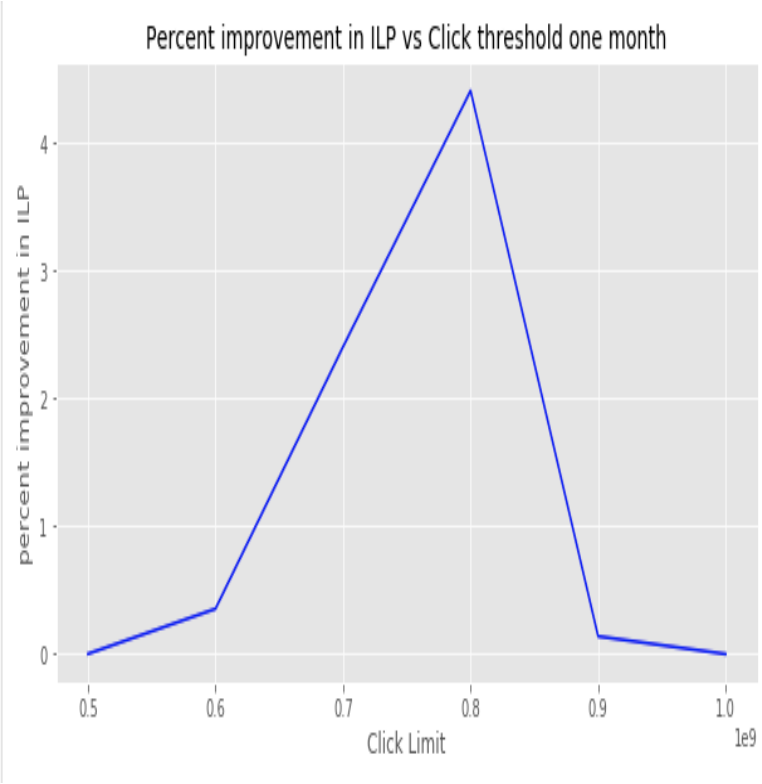
CVPM Model

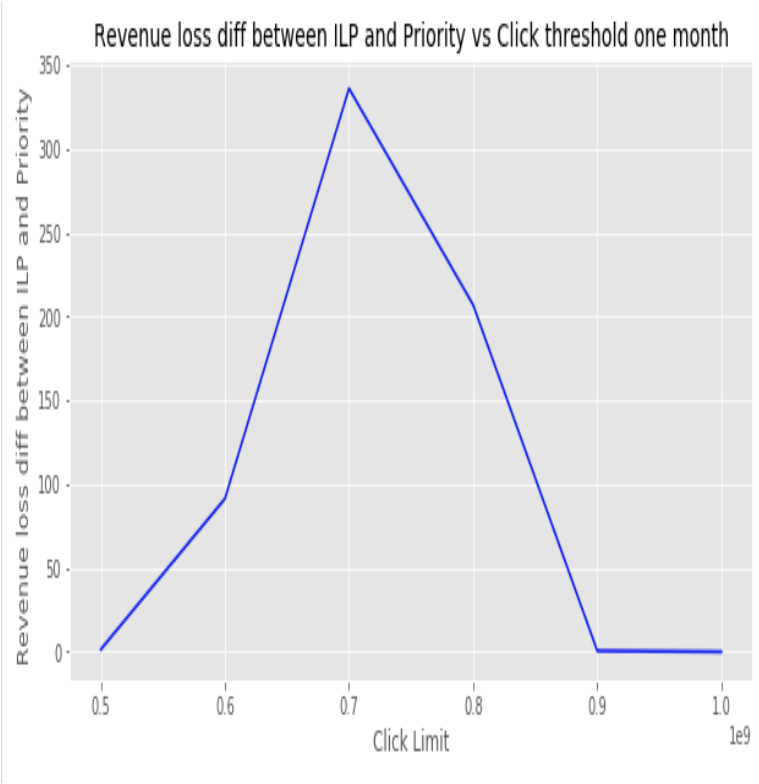

Revenue loss per million clicks vs Click threshold for one day case
Revenue loss permillion clicks
0
10
20
30
40
50
0.5
0.6
0.7
0.8
0.9
1.0
1e9
Click Limit


Revenue loss vs Click threshold for one month scaled down
Revenue loss $
0
5000
10000
15000
20000
25000
30000
0.5
0.6
0.7
0.8
0.9
1.0
1e9
Click Limit

Comparison


Comparative analysis Revenue loss vs Click threshold for one month downsampled case
priority
ILP
Revenue loss per million clicks
0
10
20
30
40
50
500000000
600000000
700000000
800000000
900000000
1000000000
Click Limit


Percent improvement in ILP vs Click threshold one month
percent improvement in ILP
0
1
2
3
4
0.5
0.6
0.7
0.8
0.9
1.0
Click Limit
1e9


Revenue loss diff between ILP and Priority vs Click threshold one month
Revenue loss diff between ILP and Priority
0
50
100
150
200
250
300
350
0.5
0.6
0.7
0.8
0.9
1.0
Click Limit
1e9

Conclusion

We found that the CVPM model is slightly better than traditional approaches. It also out-performs the Priority model by a 4-5% margin.

- Provides better Infra-Revenue optimization
- Can be dynamically modelled for different thresholds
- Easier to implement and interpret

Future Work

- The feature set can be extended to inculcate more features like invalid clicks, etc. and formulated to a 2-D constraint Knapsack problem.
- In addition to current offer and affiliate combination, a more granular selection can be considered in Items set.

Thank You !