Programmatic optimization of ad pods for maximizing consumer engagement and revenue

Niranjan Kumawat
n.kumawat@samsung.com
Samsung Research Institute
Bangalore, Karnataka, India

Manu Vajpai
manu.vajpai@samsung.com
Samsung Research Institute
Bangalore, Karnataka, India

Nitish Varshney
nitish.var@samsung.com
Samsung Research Institute
Bangalore, Karnataka, India
Samsung Ads

- We are the fourth largest advertiser in the world.
- Samsung Ads can reach over 200 million devices across Smart TV, connected TV devices and cross-media solutions.
- There are 30 million Samsung Smart TVs registered in the US, and 73 million globally.
- 75% of audiences in the Samsung Smart TV ecosystem are watching some form of linear TV.

Introduction to ad pods in Connected TVs

- Podding is similar to linear ads on TV
- Inserted dynamically into video content on CTV
- Bundling ads together reduces consumer fatigue because of littered ads
- Also increases revenue by increasing impression opportunities
- Higher revenue can be obtained with some pods with costs in brand conflicts and fatigue
- Optimal pods balance consumer engagement and revenue

Disclaimer: Brand names (unless owned by Samsung) have no affiliation with this work and are used only for representational purposes
**Ad-podding RTB auction**

- **Real-Time Bidding process:**
  - Device requests for bid on ad slot
  - Exchange (or SSP) requests bids with device/user info
  - Advertisers (DSPs) submit bids with category and domain information
  - Bids are evaluated and top bidder(s) get to display ads
  - All allocated ads are collated into pods and inserted into video
  - Selection of ads for podding is as per business objectives, formulated as constraints
  - This is a Multi-Objective Knapsack problem

\[
\begin{align*}
\max & \sum_{i=1}^{K} p_i x_i, \quad \text{s.t.} \quad \sum_{i=1}^{K} d_i x_i \leq D \\
\text{where,} & \quad x_i = \begin{cases} 
0, & \text{if } b_i \notin S \\
1, & \text{if } b_i \in S 
\end{cases}
\end{align*}
\]

\[\forall b_p, b_q \in S, \quad C_p \cap C_q = \emptyset \text{ or } A_p \cap A_q = \emptyset\]

- \(a_i\) : identifier of the linear ad
- \(p_i\) : cost to display \(a_i\) or it’s bidding price.
- \(d_i\) : duration of \(a_i\).
- \(C_i\) : set of IAB categories of \(a_i\).
- \(A_i\) : set of ad domains of \(a_i\).
Possible constraints on pod selection

● Maximizing sum of bid cpms of constituent ads (revenue maximization)
● Low similarity between IAB categories of constituent ads to minimize conflicts
● Low similarity between ad domains to minimize over-exposure
● Capping ad frequency for constituent ads across pods
● Enforcing specific distribution of ads by duration/categories (e.g. shorter ads at end)
● Requests by DSPs to always display the entire pod with their ads or to not pod their ads at all
Approaches to solving MOK problem

**Available Approaches**
- Exact solutions
  - Dynamic programming
  - Backtracking
- Heuristics
  - ACO
  - Evolutionary
- Learning based solutions
  - Pointer networks
  - Reinforcement learning
- Tailored greedy approaches

**Selected Approaches**
- Evolutionary algorithms (heuristic)
- Backtracking (exact)
- Dynamic programming (exact)
- Greedy algorithm (heuristic)

Some works have combined multiple approaches to address their problems
Experimental setup

- Two datasets:
  - In-house: 7 days of auction logs with 15 extracted IAB categories, duration, bid value
  - YouTube ad dataset: Public domain dataset without bid information. Modelled with viewership. 5000 ads with metadata.
- Pods constructed from ads in datasets
  - Sampling in heuristics, ordered selection in exact and greedy
- Constraints on IAB categories and revenue (sum of bids in pods)
- Repetition of IAB category is (mild) engagement loss
- Evolutionary algorithms as heuristic approach
  - Linearly increasing generations with pod size
  - Very high penalty for exceeding capacity
  - Moderate-to-high penalty for similarity of constituent ads
- Dynamic programmatic and backtracking as exact solutions
  - Used as benchmarks for profit, engagement and computational complexity
- Our greedy solution
Greedy solution

- We have developed two heuristics for greedy sampling of slots
  - PDR - Price to duration ratio
  - PDRwP - Price to duration ratio with price prioritized
- PDR selects slot with optimum balance of price and duration
- PDRwP gives less preference to very long ads, further preventing fatigue
- PDRwP can be tuned to also remove very small ads

\[ PDR_i = \frac{p_i}{d_i} \]
\[ PDRwP_i = p_i(1 + \frac{1}{d_i}) \]
Estimating performance of algorithms

- Percentage deviation in profit - measure non-optimal profit
- If profit is non-optimal then so is engagement
- Similar to Mean-Absolute Percentage Error (MAPE)
- 1000 iterations for each approach
- EA is farthest from optimal, in spite of tuning, while DP is optimal
- Greedy approaches, in particular PDRwP, perform near-optimal
- Some solutions for PDRwP are non-optimal (rounding error)

\[ Dev = \frac{1}{R} \sum_{t=1}^{R} \left( \frac{\text{abs}(\alpha_t - \beta_t)}{\beta_t} \right) \times 100 \]

<table>
<thead>
<tr>
<th>Percentile</th>
<th>DP</th>
<th>EA</th>
<th>G-PDR</th>
<th>G-PDRwP</th>
</tr>
</thead>
<tbody>
<tr>
<td>50\textsuperscript{th}</td>
<td>0.0</td>
<td>47.57</td>
<td>11.84</td>
<td>0.0</td>
</tr>
<tr>
<td>95\textsuperscript{th}</td>
<td>0.0</td>
<td>111.97</td>
<td>33.11</td>
<td>14.38</td>
</tr>
<tr>
<td>99\textsuperscript{th}</td>
<td>0.0</td>
<td>133.64</td>
<td>41.94</td>
<td>29.05</td>
</tr>
</tbody>
</table>

In-house

<table>
<thead>
<tr>
<th>Percentile</th>
<th>DP</th>
<th>EA</th>
<th>G-PDR</th>
<th>G-PDRwP</th>
</tr>
</thead>
<tbody>
<tr>
<td>50\textsuperscript{th}</td>
<td>0.0</td>
<td>152.73</td>
<td>15.26</td>
<td>0.0</td>
</tr>
<tr>
<td>95\textsuperscript{th}</td>
<td>0.0</td>
<td>294.80</td>
<td>79.95</td>
<td>0.0</td>
</tr>
<tr>
<td>99\textsuperscript{th}</td>
<td>0.0</td>
<td>405.45</td>
<td>87.62</td>
<td>0.0</td>
</tr>
</tbody>
</table>

YouTube

\( \alpha \) is profit from the selected approach

\( \beta \) is profit from optimal approaches

\( R \) is indexing over calculations
Distribution of errors

- Density of percentage deviation will give a holistic overview of performance
- If spread away from 0 is high - poor performance
- Greedy approaches are near-optimal
# Computational complexity in RTB context

<table>
<thead>
<tr>
<th>Bids in pod</th>
<th>Dataset</th>
<th>Backtracking</th>
<th>DP</th>
<th>EA</th>
<th>G-PDR</th>
<th>G-PDRwP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>In-house</td>
<td>0.108 ± 0.0</td>
<td>0.256 ± 0.0</td>
<td>46.7 ± 0.235</td>
<td>0.019 ± 0.0</td>
<td>0.020 ± 0.0</td>
</tr>
<tr>
<td></td>
<td>YT</td>
<td>0.143 ± 0.0</td>
<td>0.409 ± 0.001</td>
<td>48.2 ± 0.18</td>
<td>0.027 ± 0.0</td>
<td>0.028 ± 0.0</td>
</tr>
<tr>
<td>10</td>
<td>In-house</td>
<td>1.53 ± 0.004</td>
<td>0.914 ± 0.0</td>
<td>95.4 ± 0.512</td>
<td>0.045 ± 0.0</td>
<td>0.045 ± 0.0</td>
</tr>
<tr>
<td></td>
<td>YT</td>
<td>1.05 ± 0.003</td>
<td>0.730 ± 0.003</td>
<td>98.8 ± 0.421</td>
<td>0.038 ± 0.0</td>
<td>0.038 ± 0.0</td>
</tr>
<tr>
<td>15</td>
<td>In-house</td>
<td>3.63 ± 0.013</td>
<td>1.17 ± 0.002</td>
<td>172 ± 0.308</td>
<td>0.055 ± 0.0</td>
<td>0.055 ± 0.0</td>
</tr>
<tr>
<td></td>
<td>YT</td>
<td>9.35 ± 0.015</td>
<td>1.48 ± 0.038</td>
<td>140 ± 0.488</td>
<td>0.049 ± 0.0</td>
<td>0.051 ± 0.0</td>
</tr>
<tr>
<td>20</td>
<td>In-house</td>
<td>22.9 ± 0.061</td>
<td>2.13 ± 0.013</td>
<td>203 ± 0.786</td>
<td>0.067 ± 0.0</td>
<td>0.067 ± 0.0</td>
</tr>
<tr>
<td></td>
<td>YT</td>
<td>23 ± 0.153</td>
<td>2.33 ± 0.01</td>
<td>195 ± 0.407</td>
<td>0.076 ± 0.0</td>
<td>0.074 ± 0.001</td>
</tr>
<tr>
<td>25</td>
<td>In-house</td>
<td>89 ± 0.290</td>
<td>2.12 ± 0.005</td>
<td>261 ± 2.17</td>
<td>0.083 ± 0.0</td>
<td>0.082 ± 0.0</td>
</tr>
<tr>
<td></td>
<td>YT</td>
<td>86.2 ± 1.09</td>
<td>2.27 ± 0.046</td>
<td>248 ± 1.17</td>
<td>0.080 ± 0.0</td>
<td>0.081 ± 0.0</td>
</tr>
<tr>
<td>30</td>
<td>In-house</td>
<td>191 ± 1.11</td>
<td>3.54 ± 0.015</td>
<td>329 ± 1.74</td>
<td>0.104 ± 0.0</td>
<td>0.104 ± 0.0</td>
</tr>
<tr>
<td></td>
<td>YT</td>
<td>126 ± 0.281</td>
<td>3.68 ± 0.032</td>
<td>305 ± 1.94</td>
<td>0.091 ± 0.0</td>
<td>0.093 ± 0.0</td>
</tr>
<tr>
<td>40</td>
<td>In-house</td>
<td>1000 ± 3.21</td>
<td>4.54 ± 0.011</td>
<td>462 ± 6.88</td>
<td>0.111 ± 0.0</td>
<td>0.123 ± 0.0</td>
</tr>
<tr>
<td></td>
<td>YT</td>
<td>928 ± 4.07</td>
<td>4.95 ± 0.014</td>
<td>440 ± 4.45</td>
<td>0.113 ± 0.001</td>
<td>0.116 ± 0.0</td>
</tr>
<tr>
<td>50</td>
<td>In-house</td>
<td>3540 ± 18.1</td>
<td>5.96 ± 0.022</td>
<td>611 ± 1.05</td>
<td>0.125 ± 0.0</td>
<td>0.160 ± 0.0</td>
</tr>
<tr>
<td></td>
<td>YT</td>
<td>3350 ± 11.8</td>
<td>8.3 ± 0.044</td>
<td>581 ± 4.34</td>
<td>0.155 ± 0.0</td>
<td>0.162 ± 0.001</td>
</tr>
</tbody>
</table>

Running times of the algorithms for varying pod sizes. Both in-house and YouTube dataset are listed. Time units are milliseconds.
Conclusions

- Optimal podded ads offer a balance between revenue and consumer engagement.
- Optimizing ads pods can be modelled as a multi-objective knapsack problem.
- Four families of algorithms can be used to solve MOKs - exact, heuristic, greedy and learning-based.
- We have compared greedy, heuristic and exact approaches.
- Greedy approaches are nearly as accurate as exact ones.
- The efficiency of greedy approaches makes them ideal for RTB deployment.
Thank you!