**Introduction and Motivation**

In large-scale social media platforms, the member experience is often controlled by certain parameters in conjunction with machine-learned models. Such parameters could be independent of the machine-learned models. An example is the decision to send or drop a notification depending on a threshold parameter, which is used directly on the output of a machine learning model.

If the score of the notification obtained from a machine-learned model is greater than the threshold, the platform then sends the notification otherwise it is dropped (Near real-time optimization of activity-based notifications, Yan et al, 2018). The machine learning model controls the relevance of the item, while the threshold controls certain business metrics.

In an ideal situation, we would often want such parameters to be fully optimized for the multiple business objectives. One option is to fix a global parameter for all members and then iterate on the choice of this parameter through A/B testing. However, that may very well be a suboptimal solution. We propose a new model-based approach that automatically:

- Identifies member cohorts with heterogeneous causal effects leveraging a wide range of features.
- Selects the best parameter (e.g. threshold) for each cohort through stochastic optimization to personalize member experience.

**Methodology**

**Heterogeneous Cohort Identification**

**Regression Tree (CART)**

- Predict Y (sessions)
- Splitting Objective: MSE(Y)

**Causal Tree**

- Estimate e (sessions)
- Splitting Objective: MSE(e) + Variance regularizer
- Honest estimations

**Stochastic Multi-objective Optimization**

- We start by reframing the optimization problem to a generic form:

  \[
  \text{Minimize } f(x) = \sum_{k=1}^{K} f_k(x, U_k) \\
  \text{subject to } g_k(x) = \sum_{\ell=1}^{L} h_{k\ell}(x, U_{k\ell}) \leq 0 \text{ for } k = 1, \ldots, K.
  \]

- Our algorithm, which we call Multiple Coordinated Stochastic Approximation (MCSA), is an iterative algorithm which runs for N steps. At each step \(t\), we start by estimating the constraint function. Specifically, we simulate \(U_{k\ell}\) for \(\ell = 1, \ldots, L\) and estimate:

  \[
  \hat{g}_{k\ell} = \sum_{\ell=1}^{L} \hat{h}_{k\ell}(x, U_{k\ell}).
  \]

  If all constraints satisfied, we optimize towards the objective. Otherwise we randomly pick a constraint, and optimize towards it, and update \(x\).

**Notation and Definitions**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(K)</td>
<td>Total number of parameter values or choices.</td>
</tr>
<tr>
<td>(C_{ij})</td>
<td>ith cohort for (i = 1, \ldots, K).</td>
</tr>
<tr>
<td>(U_{ij})</td>
<td>Verteclized version of (C_{ij}), which is the causal effect in metric (j) by parameter (i) in cohort (C_{ij}).</td>
</tr>
<tr>
<td>(\mu_j)</td>
<td>Mean of (U_j).</td>
</tr>
<tr>
<td>(\Sigma_j)</td>
<td>Variance of (U_j).</td>
</tr>
<tr>
<td>(x)</td>
<td>The assignment vector.</td>
</tr>
</tbody>
</table>

Based on these notations, we can formulate our optimization problem. Let \(K = 0\) denote the main metric. We wish to optimize the main metric keeping the guardrail metrics at a threshold. Formally, we wish to get the optimal \(x^*\) by solving the above maximization problem: where \(c_{k}\) are known bounds.

**Offline/Online Evaluations**

**Offline Evaluations.** Below figures show the offline estimations.

**Online A/B Testings.** We demonstrated online validation of the approach which has been tested online on the LinkedIn notification system to increase session/visits metric while holding all guardrail metrics neutral.