Online Parameter Selection for Web-based Ranking

Joint Work with Kinjal Basu, Souvik Ghosh, Ying Xuan, Yang Yang and Liang Zhang.
LinkedIn Feed

**Mission:** Enable Members to build an active professional community that advances their career.

**The Feed is:**
- The personalized “home page” of LinkedIn
- A heterogenous list of updates
  - Shares from a member’s connections
  - Recommendations including jobs, articles, connections, courses
  - Sponsored content or ads.
Important Metrics

- **Viral Action (VA)** - Sessions where members liked, shared or commented on an item

- **Job Applies (JA)** - Sessions where members applied for a job

- **Engaged Feed Session (EFS)** - Sessions where a member engaged with anything on feed.
Ranking

\[ S(m, u) := P_{VA}(m, u) + x_{EFS} P_{EFS}(m, u) + x_{JA} P_{JA}(m, u). \]

- The weight vector \( x = (x_{EFS}, x_{JA}) \) controls the balance of the metrics EFS, VA and JA.

- Business Strategy:

\[
\begin{align*}
\text{Maximize} & \quad VA(x) \\
\text{subject to} & \quad EFS(x) \geq c_{EFS}, \; JA(x) \geq c_{JA}
\end{align*}
\]
Problem

- The optimal value of x (tuning parameters) changes over time
- Example of changes
  - New content types are added
  - Relevance models are updated

- With every change engineers would manually find the optimal x
  - Run multiple A/B tests
- Not the best use of engineering time
Metrics Modeling

- \( Y^k_{i,j}(x) \in \{0, 1\} \) denote if the i-th member during the j-th session which was served by parameter x, did action k or not. Here k = VA, EFS or Job Clicks.

- We model this data as follows

\[
Y^k_i(x) \sim \text{Binomial}(n_i(x), \sigma(f_k(x)))
\]

where \( n_i(x) \) is the total number of sessions of member i which was served by x and \( f_k \) is a latent function for the particular metric.

- Assume a Gaussian process prior on the latent function f.
We estimate the metrics as:

\begin{align*}
VA(x) &= \sigma(f_{VA}(x)) \\
EFS(x) &= \sigma(f_{EFS}(x)) \\
JA(x) &= \sigma(f_{JA}(x))
\end{align*}

Solve the unconstrained optimization problem:

\[
\max_{x \in \mathcal{X}} VA(x) + \lambda (\sigma_{\xi}(EFS(x) - c_{EFS}) + \sigma_{\xi}(JA(x) - c_{JA}))
\]

**Benefit:** Equivalent to finding the maximum of a continuous functional of a Gaussian process.
Thompson Sampling Algorithm

- Consider a Gaussian Process Prior on each $f_k$, where $k$ is VA, EFS or JA
- Observe the data $Y_i^k(x)$

- Obtain the posterior of each $f_k$ which is another Gaussian Process

- Sample from the posterior distribution and generate samples for the overall objective function.
- We get the next distribution of hyperparameters by maximizing the sampled objectives (over a grid of QMC points).

Continue until the distribution of $x$ converges.
Simulation Results

(a) Trimodal Shekel Function

(b) Decay of log relative square error
The Thompson Sampling approach generates a sampling distribution at every iteration.

\((x_1, \ldots, x_n)\) with probability \((p_1, \ldots, p_n)\) such that \(\sum p_i = 1\)

To serve members with the same distribution, each memberId is mapped to \([0,1]\) using a hashing function \(h\). For example, if

\[ \sum_{i=1}^{k} p_i < h(Deepak) \leq \sum_{i=1}^{k+1} p_i \]

Then the feed is served with parameter \(x_{k+1}\)
Online Architecture
Other Practical Design Considerations

- Consistency in user experience.
  - Randomize at member level instead of session level.

- Offline flow frequency
  - Batch computation where we collect data for an hour and run the offline flow each hour to update the sampling distribution.

- Assume \((f_{VA}, f_{EFS}, f_{JA})\) to be Independent
  - Works well in our setup. Joint modeling might reduce variance.

- Choice of business constraint thresholds.
  - Chosen to allow for a 1% drop.
Online Convergence Result

(a) Iteration = 10

(b) Iteration = 15

(c) Iteration = 20

(d) Iteration = 25

(e) Iteration = 30

(f) Iteration = 35
## Online A/B Test Results

### Table 1: Online A/B results for Online Parameter Selection in LinkedIn Feed Ranking

<table>
<thead>
<tr>
<th>Metric</th>
<th>Lift (%) vs Control $x_{c_1}$</th>
<th>Lift (%) vs Control $x_{c_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viral Actions</td>
<td>+3.3%</td>
<td>+1.2%</td>
</tr>
<tr>
<td>Engaged Feed Sessions</td>
<td>-0.8%</td>
<td>0%</td>
</tr>
<tr>
<td>Job Applies</td>
<td>+12.8%</td>
<td>+6.4%</td>
</tr>
</tbody>
</table>
Key Takeaways

- Removes the human in the loop: Fully automatic process to find the optimal parameters.
- Drastically improves developer productivity.
- Can scale to multiple competing metrics.

Future Direction
- Create a dependent structure on different utilities to better model the variance.
- Automatically identify the primary metric by understanding the models better.
- Allow for relaxing the constraint to get large gain in primary metric.