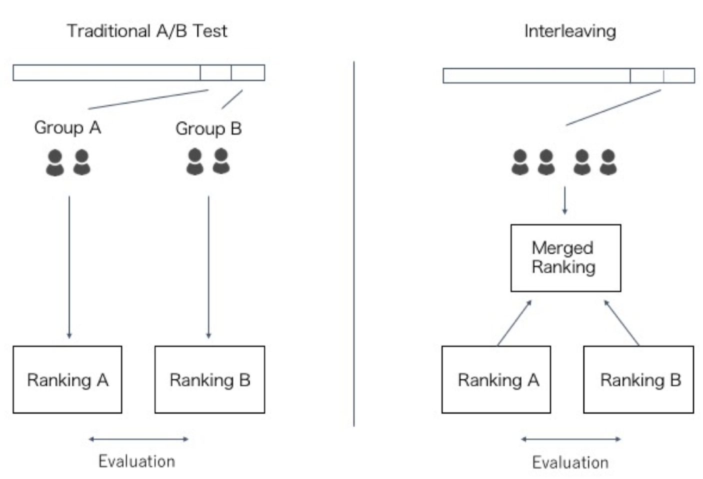
Greedy Optimized Multileaving for Personalization

Kojiro Iizuka[1], Takeshi Yoneda[1], Seki Yoshifumi[1]
[1]Gunosy Inc.
{kojiro.iizuka, takeshi.yoneda, yoshifumi.seki}@gunosy.com

What is Multileaving?

- A method that evaluates multiple rankings efficiently using user click feedback (Schuth, 2014).



Challenges

- Achieving high sensitivity with low computation cost for the rankings, which differ for each user and each time.
- Ensuring stability over the number of rankers and the ranking length.

Contributions

- **New Problems:** Clarifies the challenges of applying the multileaving method to personalized rankings;
- **Algorithm:** Proposes the greedy optimized multileaving (GOM) method with the personalization credit function to solve these problems; and
- **Stability and Sensitivity:** Confirms the stability and sensitivity of GOM throughout offline and online experiments.

Greedy Optimized Multileaving

Optimized Multileaving (Radlinski 2013)

$$\min_{p_k} \quad \alpha \sum_{r=1}^{l} \lambda_r + \sum_{k=1}^{m} p_k \sigma_k^2$$

subject to

$$(\forall r, j, j') \left| \sum_{k=1}^{m} p_k \sum_{i=1}^{r} \delta(O_{k,i}, I_j) - \sum_{k=1}^{m} p_k \sum_{i=1}^{r} \delta(O_{k,i}, I_{j'}) \right| \leq \lambda_r$$

$$\sum_{k=1}^{m} p_k = 1, \quad 0 \leq p_k \leq 1 \quad (k = 1, \ldots, m).$$

Using the previous work's formula has high computational cost to generate multileaved rankings.

δ : Credit function
σ : Insensitivity
λ : Maximum difference of the expected credits
α : Hyperparameter

Re-formulation for personalized settings

Greedy Optimized Multileavning (proposed)

$$\underset{k}{\arg\min} \quad \alpha \sum_{r=1}^{l} \lambda_r + \sigma_k^2$$

subject to

$$(\forall r, j, j') \left| \sum_{i=1}^{r} \delta(O_{k,i}, I_j) - \sum_{i=1}^{r} \delta(O_{k,i}, I_{j'}) \right| \leq \lambda_r .$$

- Personalized ranking is shown to the user only once.
- We can consider ranking output probability as a one-hot vector.

We can solve this problem by greedy strategy.

Inverse credit function

$$\delta(O_{k,i}, I_j) = \frac{1}{rank(O_{k,i}, I_j)},$$

The deeper the click position, the smaller the credit.

personalized credit function

$$\delta(O_{k,i}, I_j) = -|\{j' \mid rank(O_{k,i}, I_{j'}) \leq rank(O_{k,i}, I_j)\}|,$$

The credit can be calculated without position noise.

Experiment

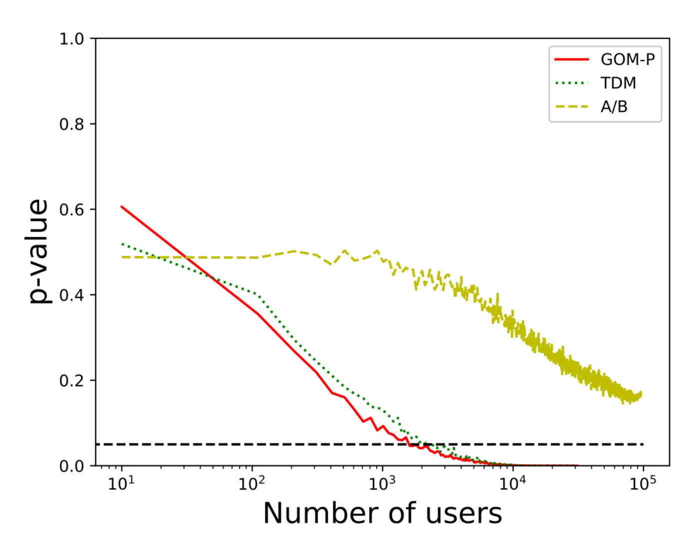
Online Test

Do GOM' s evaluation coincide with A/B Testing? How is the efficiency?

- **TDM:** Team Draft Multileaving (Schuth, 2014)
- **GOM-I:** GOM, using the inverse credit
- **GOM-P:** GOM, using the personalization credit

Table 1: Differences between the sum of credits. The values of GOM-P and TDM were all positive, while some values of GOM-I (written in blue) were negative. The GOM-P and TDM's results were consistent with previous CTR results of A/B testing (algo-A < algo-B < algo-C < algo-D < algo-E), but GOM-I was inconsistent. This means that the inverse credit function which was often used in previous studies is not appropriate for long ranking.

	GOM-I				GOM-P				TDM			
	algo-A	algo-B	algo-C	algo-D	algo-A	algo-B	algo-C	algo-D	algo-A	algo-B	algo-C	algo-D
algo-B	-14,962				21,111				615			
algo-C	-21,642	-6,680			36,561	15,450			1,259	644		
algo-D	-24,179	-9,217	-2,537		44,484	23,373	7,923		1,812	1,197	553	
algo-E	-5,246	9,716	16,396	18,933	52,597	31,486	16,036	8,113	2,117	1,502	858	305

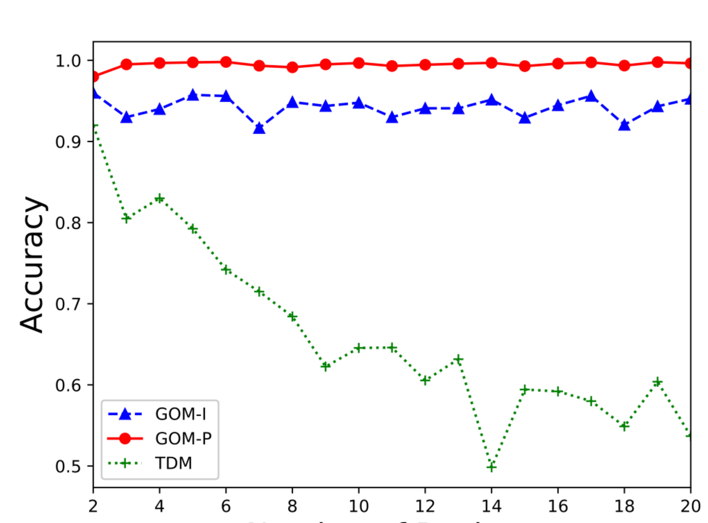


P-value versus the number of users

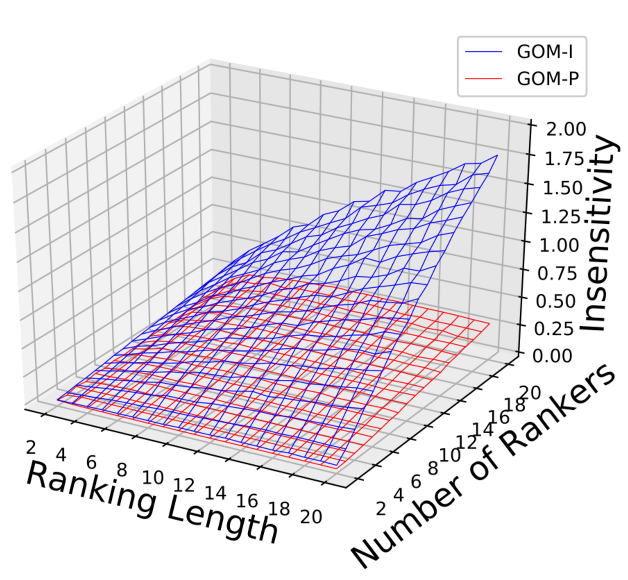
These experiments assumed a practical environment that requires a low computation cost to generate rankings in real-time; therefore, we did not use OM.

Offline Test

How stable and sensitive is GOM?



Accuracy versus the number of rankers



Insensitivity versus the number of ranker and length

Conclusion and Future Work

- We proposed GOM for personalized rankings that require a high degree of freshness, many hyperparameters, and a long ranking length for a rich user experience.
- We confirmed GOM' s stability and sensitivity by online and offline experiments.
- We will try to apply GOM' s feedback schema to speed up online learning to rank.