Dynamic collaborative filtering Thompson Sampling for cross-domain advertisements recommendation

Aug 15, 2022

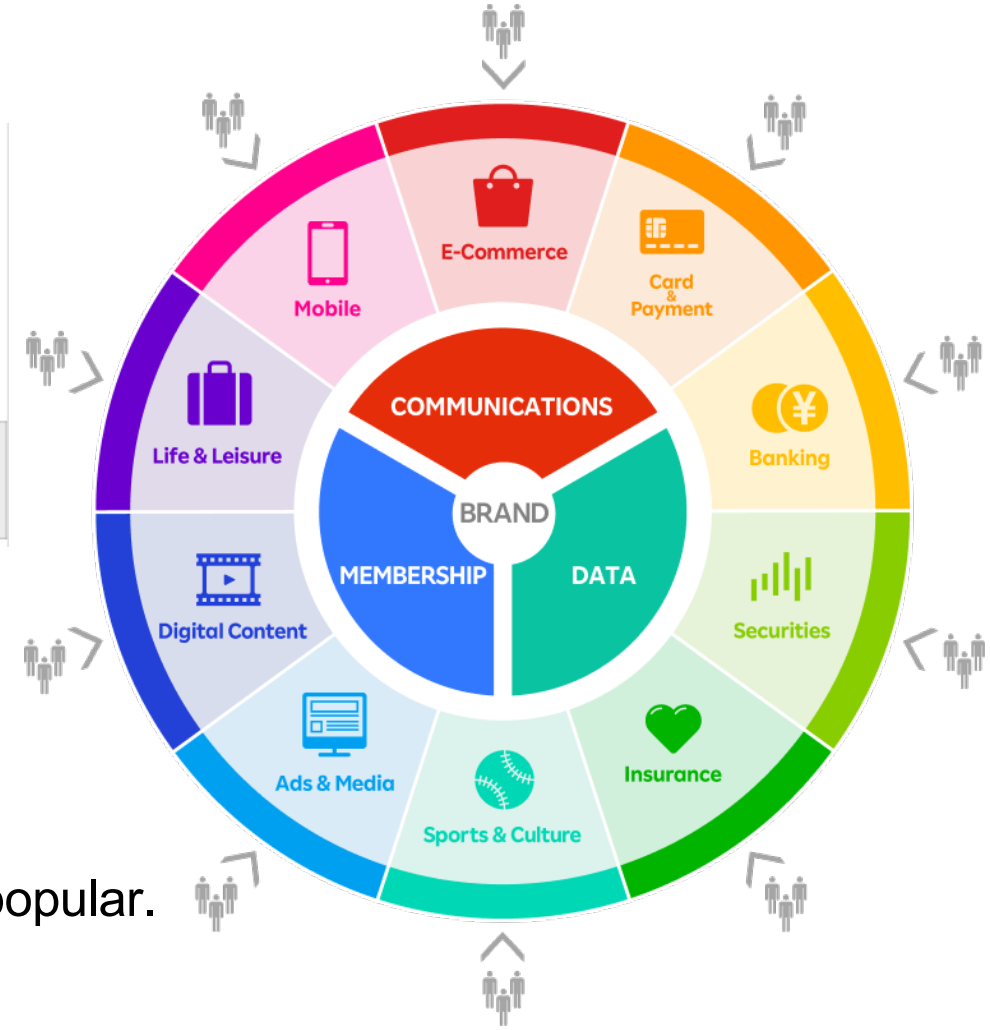
Shion Ishikawa
Rakuten Institute of Technology.
Rakuten, Inc.

Rakuten

Background

Netflix artwork personalization

Rakuten



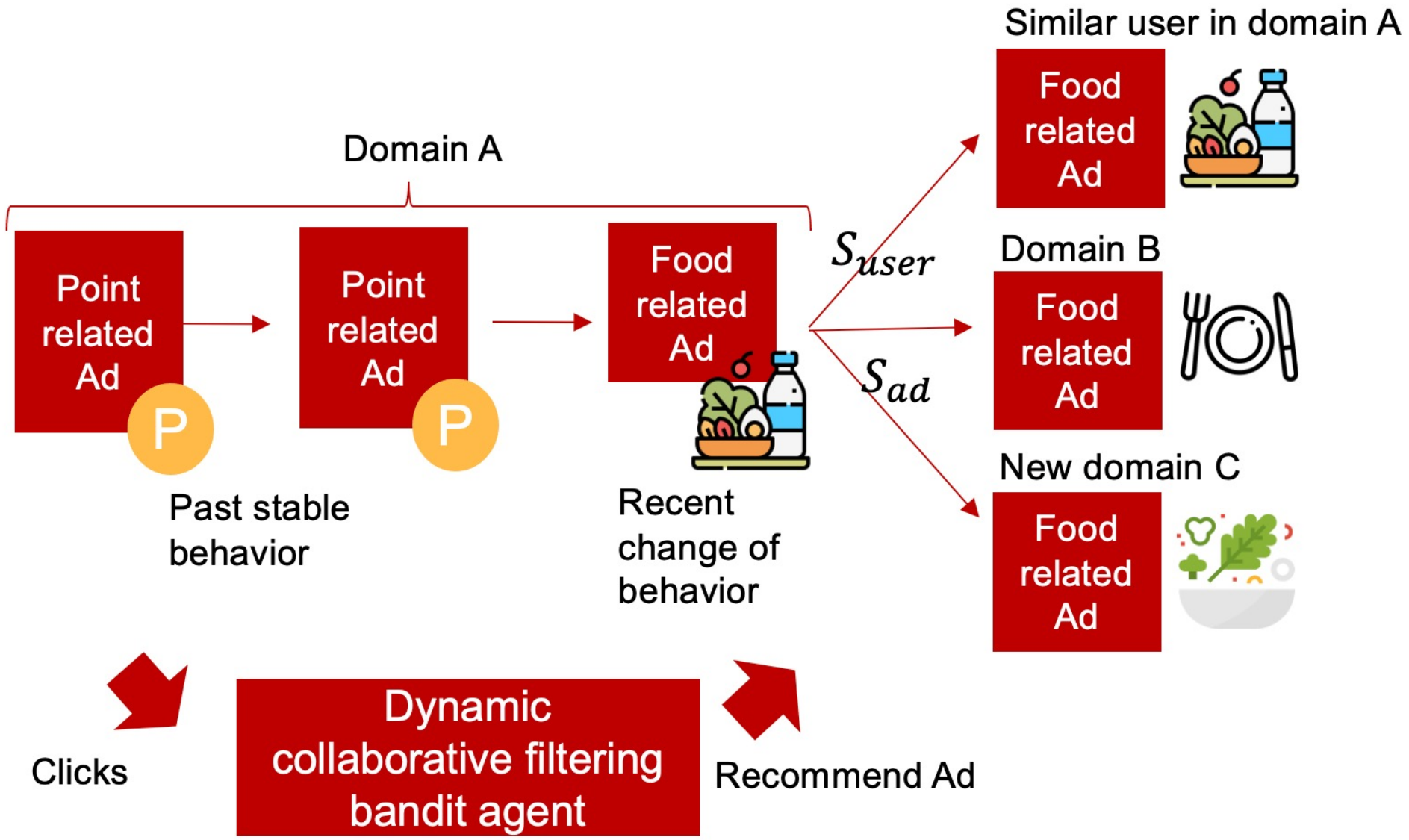
Recommender systems based on **multi-armed bandit** became popular.

Traditionally, these models mainly focused on one domain.

Companies often serve multiple services and a set of users are shared among them.

There is a potential benefit to provide cross-domain recommendation.

Use-case of cross-domain recommendation.



Related works

- Transferable Contextual Bandit for Cross-Domain Recommendation (Bo Liu, et al. AAAI-18)
  - TCB: They introduced the translation matrix among domains to utilize similarities of domains.
  - It's an expansion of Upper Confidence Bound (UCB) policy
- Transferable Contextual Bandits with Prior Observations (Kevin Labille, et al. PAKDD 2021)
  - T-LinUCB: This model is an expansion of LinUCB. This calculates the prior evaluation of arms in the new domain with contextual feature of arms in different domains.

Our approach

- It's an expansion of Thompson's Sampling
- we additionally introduced similarity of users and decaying of rewards
  - User's preference will vary, so it's natural to put a high value on the recent historical transactions.

Problem setting

s : N available sources. A widget where ads are displayed.

$\mathbf{A_s}$: Set of k_s ads (arms) in s

$\mathbf{X} \in \mathbb{R}^{m \times d_u}$: A matrix of m users and d_u features

$\mathbf{Y^s} \in \mathbb{R}^{k_s \times d_a}$: A matrix of k_s ads and d_a features

For each time step t and s, we observe $x_i \in \mathbf{X}$ and $\mathrm{y_a} \in \mathbf{Y}$ as contexts.

User will see ad $a \in \mathbf{A_s}$, then we observe r as an implicit feedback (whether user clicked ad or not).

The objective of our model is to pick up ad a in each user, source, timestep to maximize cumulative rewards. This can be represented as minimizing total regret.

$$\text{minimize } \mathbb{E}[\text{regret}_i(T)] = \sum_{s=0}^{N} \mathbb{E}[\max_{a_t^s \in A_s} \sum_{t=0}^{T} r^*_{i_t^s a_t^s t} - \sum_{t=0}^{T} r_{i_t^s a_t^s t}]$$

where r^* indicates a reward from the best action for user i_t^s.

Background of our method

Bandit algorithm handles cold start case relatively well but standard bandit algorithm still suffers from it because policy assumes non-informative prior.

For example, in Bernoulli Thompson Sampling, the prior distribution is Beta distribution

$$\mathcal{B}(\theta_k; \alpha_k, \beta_k) = \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \theta_k^{\alpha_k - 1} (1 - \theta_k)^{\beta_k - 1}$$

For most cases, $\alpha_k = 1$ and $\beta_k = 1$ were used as non-informative prior.

However, for the most practical cases, these hyperparameters can be estimated by utilizing historical data.

Improve hyper parameters of prior distribution

The parameter for each user i and ad k

$$\alpha_{ik}^{0}(t) = \sum_{l \neq k} S_{ad}(y_k, y_l) s_{il}(t) + \sum_{j \neq i} S_{user}(x_i, x_j) s_{jk}(t)$$

$$\beta_{ik}^{0}(t) = \sum_{l \neq k} S_{ad}(y_k, y_l) f_{il}(t) + \sum_{j \neq i} S_{user}(x_i, x_j) f_{jk}(t) \quad (4)$$

Reward of other ad l by user i

Reward of ad k by other user j

where $s_{il}(t)$ is a discount-aware cumulative reward

$$s_{il}(t) = \sum_{\tau=0}^{t} \gamma^{t-\tau} s_{ij\tau}$$

S_{ad} and S_{user} are the cosine similaritiy among advertisements and users.

Update of hyper-parameters for posterior distribution

$$\alpha_{ik}(t) = \lambda(s)\alpha^0_{ik}(t) + gs_k(t) + s_{ik}(t) + 1$$
$$\beta_{ik}(t) = \lambda(f)\beta^0_{ik}(t) + gf_k(t) + f_{ik}(t) + 1 \qquad (6)$$

Prior knowledge | Global reward | Personal reward

Because we have parameters for each person and ad, the naïve way is to utilize only personal rewards. However, personal rewards are sparse, so we also used global reward with hyper-parameter g.

λ is hyper-parameter which adjusts the importance of prior knowledge

Algorithm

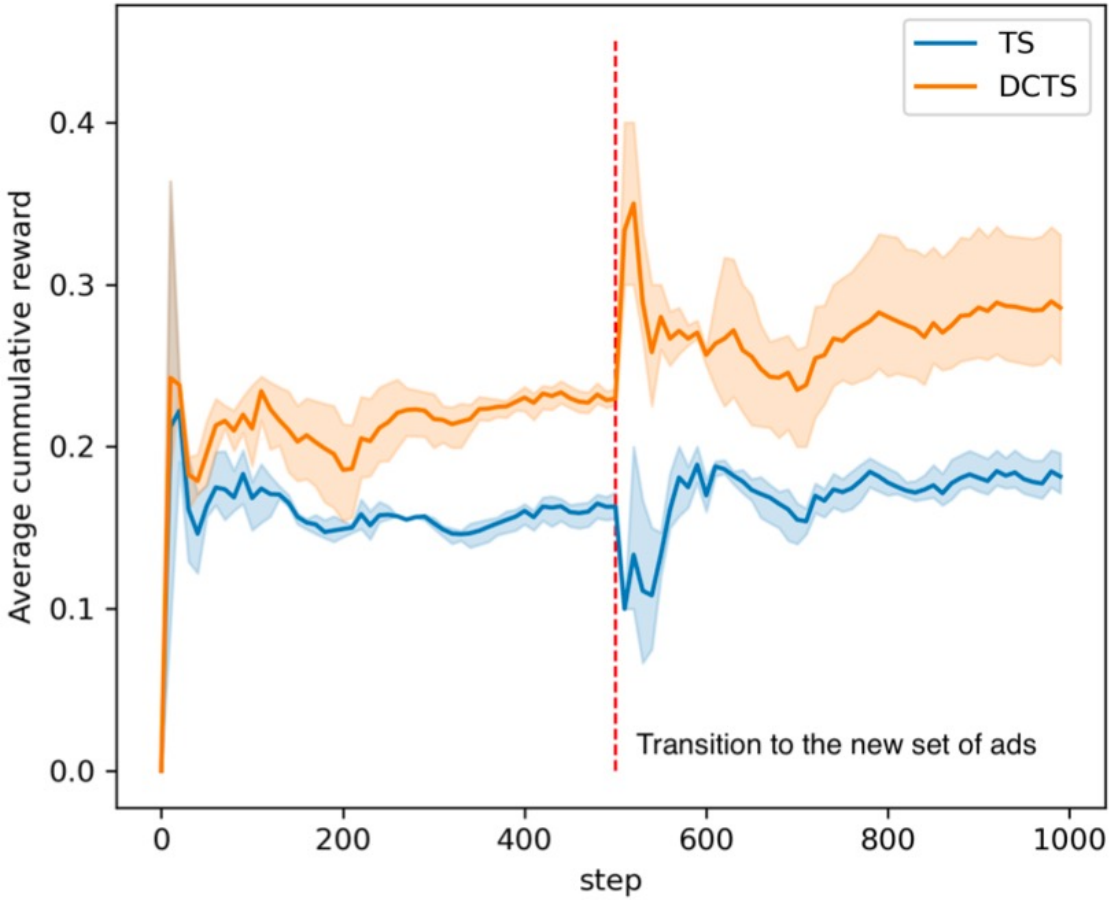
Algorithm 1 Dynamic collaborative filtering Thompson sampling

Input: $\lambda, g, \gamma \in \mathbb{R}_+^0, S_{user}, S_{ad}$, Source observations $\mathbf{O}$

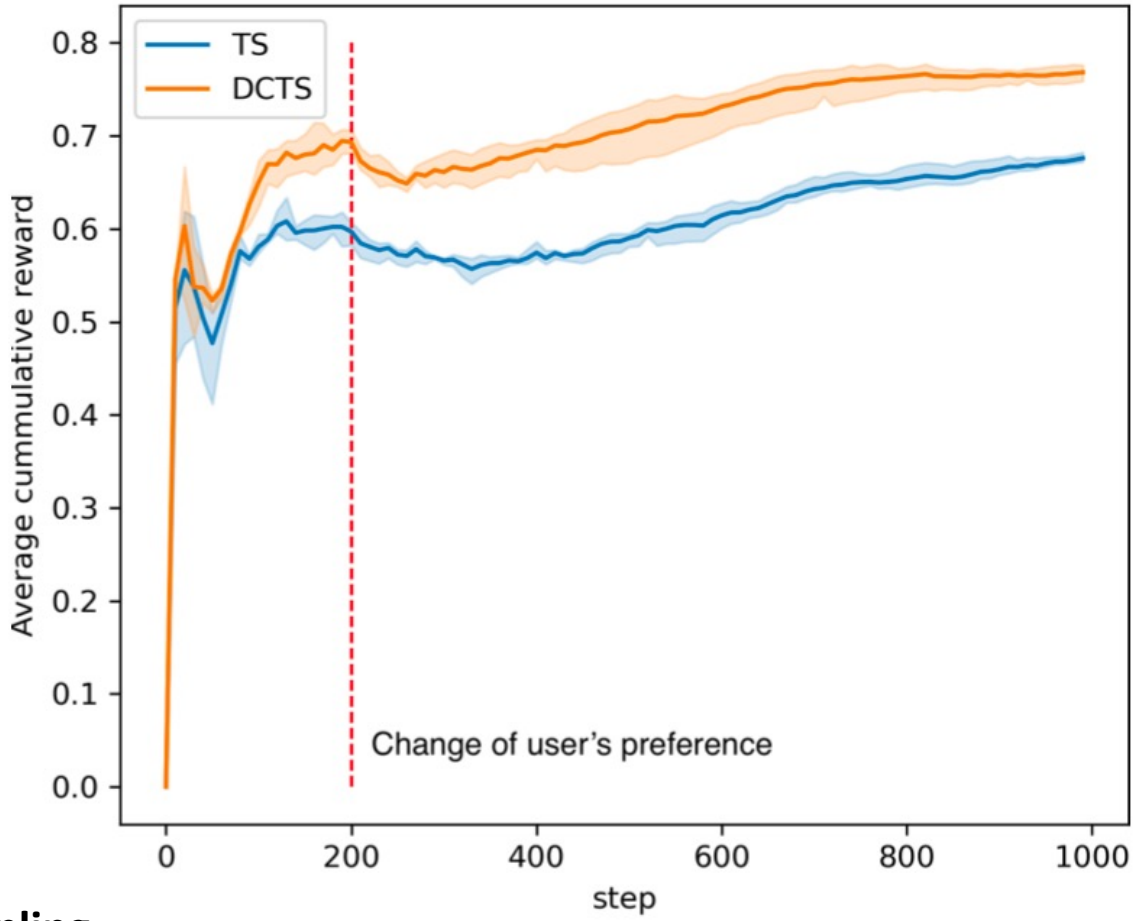
for $t = 0, ..., T$ **do**
  Observe user i_t^s and context $x_{i_t^s}$, action sets $\mathbf{A_s}$ and their contexts $\mathbf{Y^s}$
  for $k \in \mathbf{A_s}$ **do**
    Calculate $\alpha^0_{i_t^s k}(t)$ and $\beta^0_{i_t^s k}(t)$ according to Eq. 4
    Calculate $\alpha_{i_t^s k}(t)$ and $\beta_{i_t^s k}(t)$ according to Eq. 6
    Sample θ_k from the $\mathcal{B}(\theta_k; \alpha_{i_t^s k}(t), \beta_{i_t^s k}(t))$
  end for
  Play action $k = \arg\max_k \theta_k$ and observe reward $r_{i_t^s kt}$
  Add observation $O_t^s \leftarrow (x_{i_t^s}, k, r_{i_t^s kt})$
end for

Simulation with synthetic data

Setting: we showed 5 ads in the first 500 steps and we switched ads to another 5 ads in the latter 500 steps.

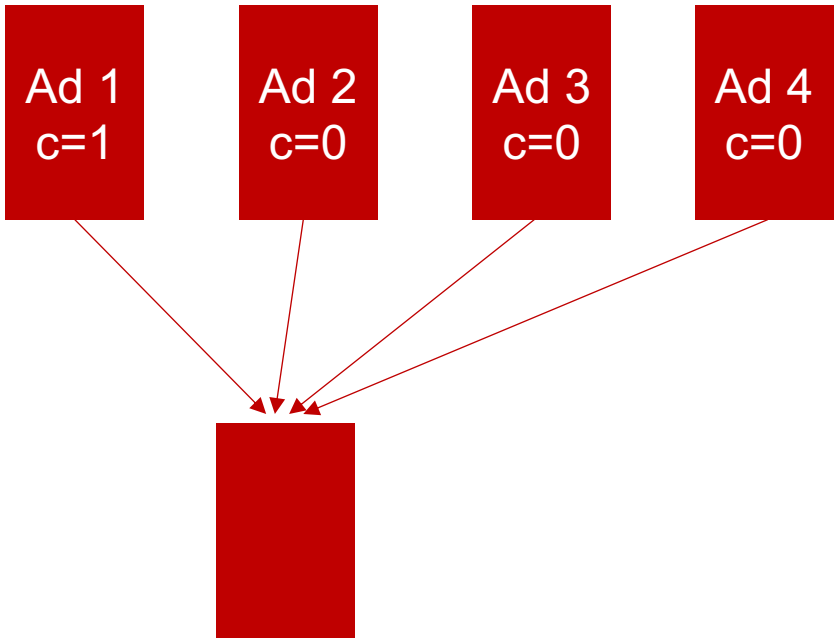


Setting: we prepared 50 ads. At step 200, we switched the user's response function.



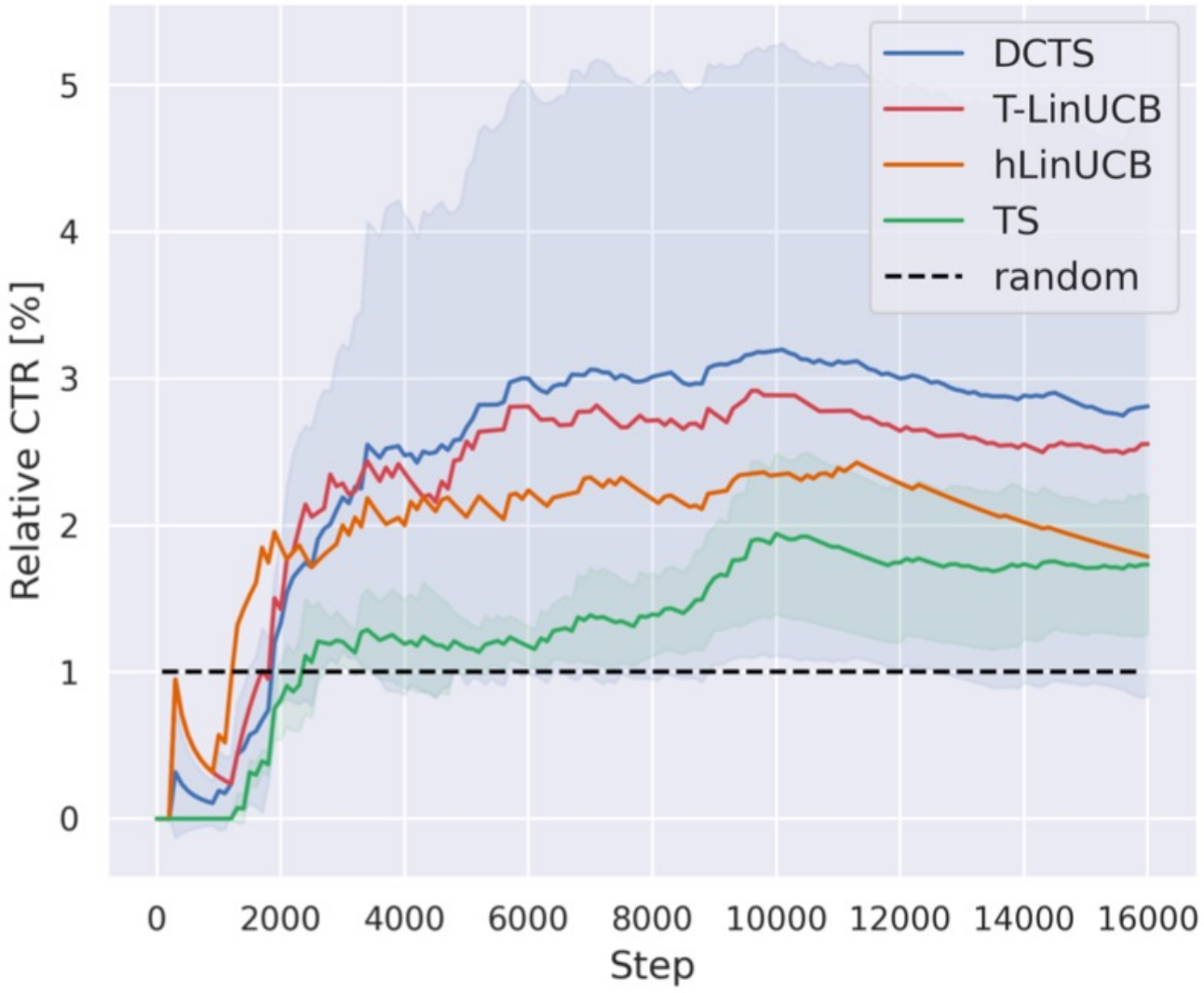
DCTS: **dynamic collaborative filtering Thompson sampling**
TS: Thompson sampling

Method for offline simulation for real-world data



1. Prepared several ads which were displayed together in the carousel
2. We regarded 1. as a standard problem of single slot optimization
3. When we conduct an offline simulation by using this data, we know which ad was clicked among all ads

Offline simulation by using real-world data



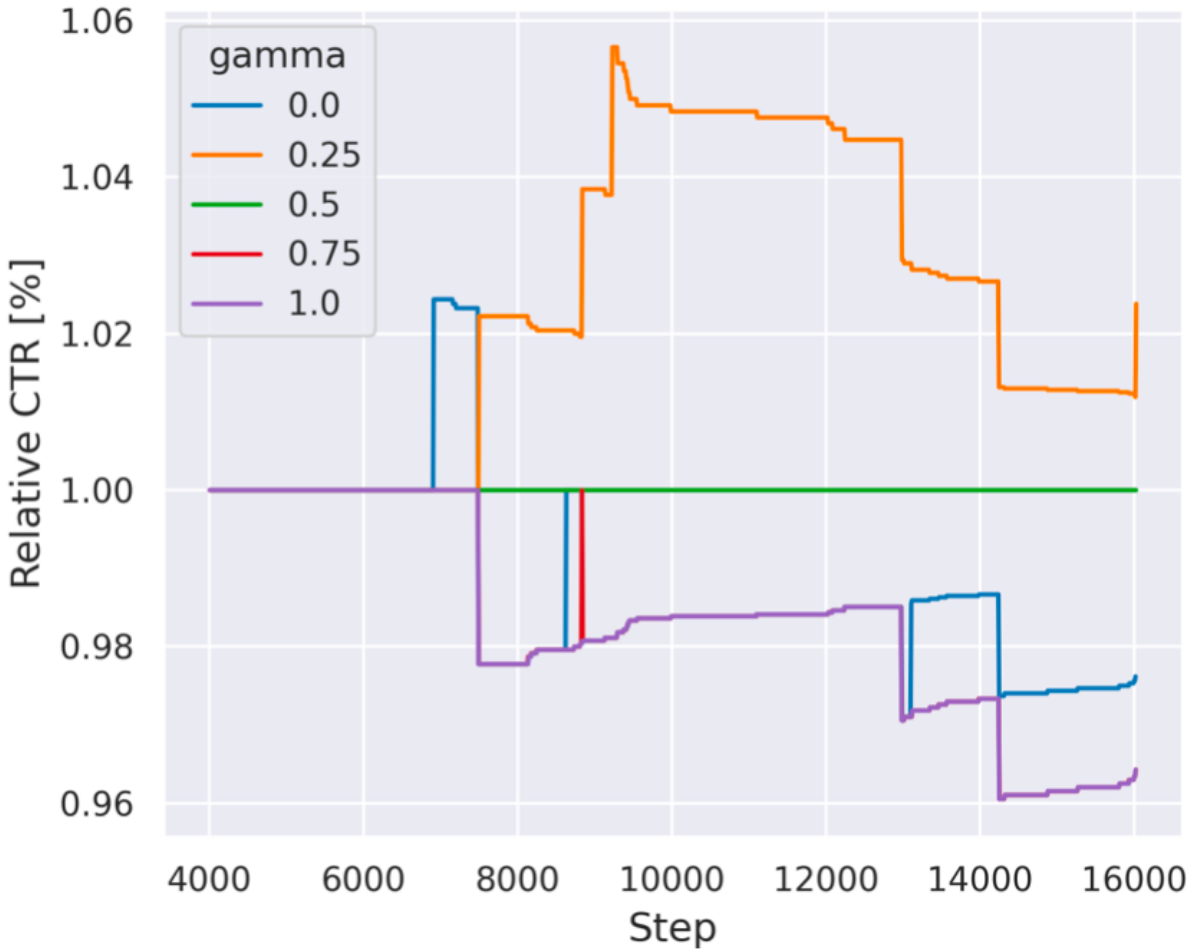
Relative CTR on Rakuten Travel data

We compared dynamic collaborative filtering bandit (DCTS) with Transferable LinUCB (T-LinUCB), hybrid LinUCB (hLinUCB), Thompson Sampling (TS) and random.

We pre-trained DCTS and T-LinUCB by Rakuten Ichiba data first. We executed experiments three times.
Each line indicates the mean of these trials, and the shaded band indicates 95% confidence interval.

As a result, DCTS performed better than T-LinUCB by 9.7 % and better than LinUCB and TS by around 37 %.

Relative CTR on Travel data with various γ parameters



γ is a hyper-parameter which means discount of reward.

$$s_{il}(t) = \sum_{\tau=0}^{t} \gamma^{t-\tau} s_{ij\tau}$$

Figure shows relative CTRs by various parameter gamma, and we normalized values by the value of when γ = 0.5.

We observed the best performance when γ = 0.25. It suggests this value is the most suitable discount for Rakuten Travel dataset.

Conclusions and Takeaway

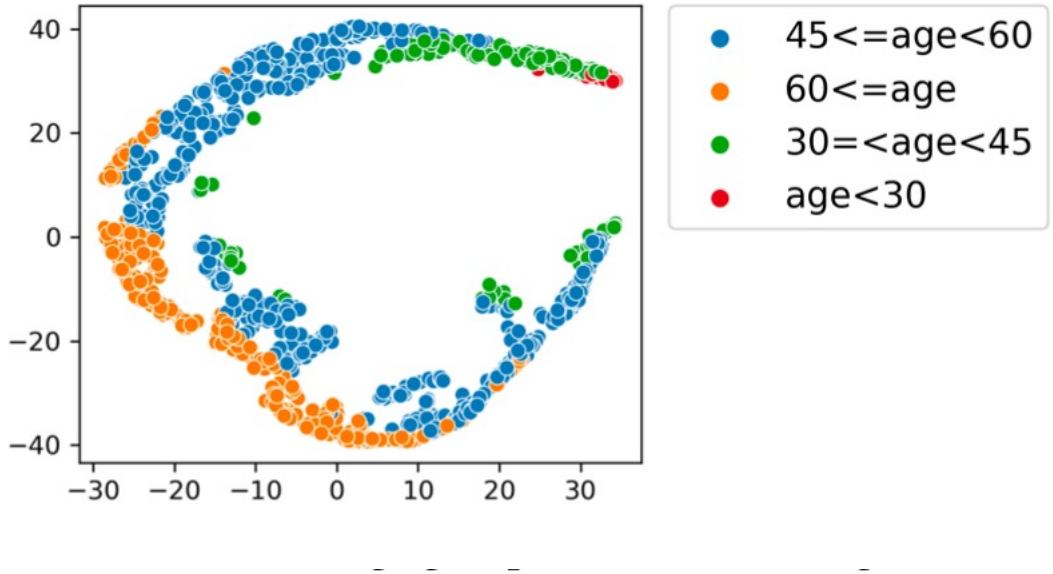
- We proposed Dynamic collaborative filtering Thompson Sampling and improved prior distribution of Thompson sampling by transferring information from other domains.
- We conducted an empirical analysis on a real-world dataset and the result showed that DCTS improved click-through rate by 9.7% than the state-of-the-art models.
- We analyzed hyper-parameters that adjust temporal dynamics and showed the best parameter which maximizes CTR.

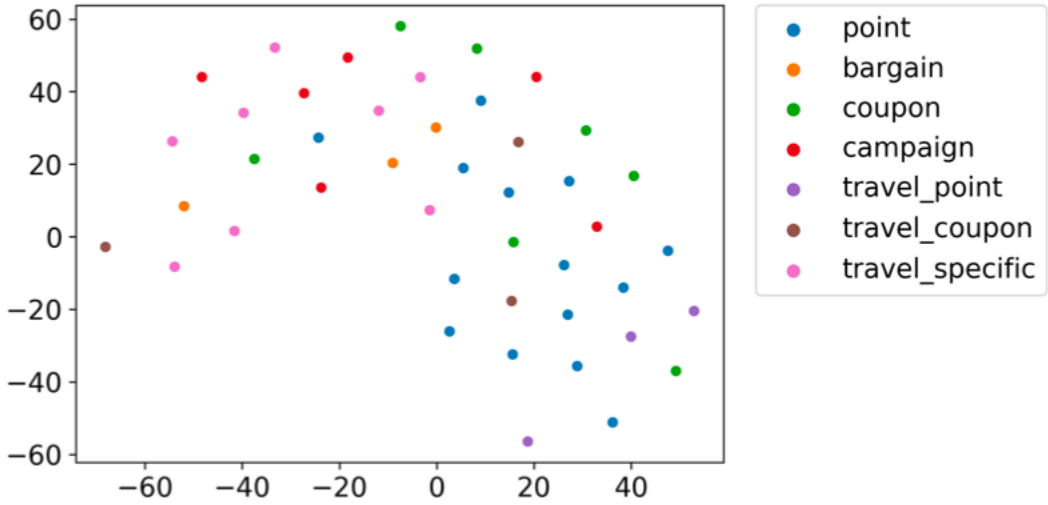
Future work:

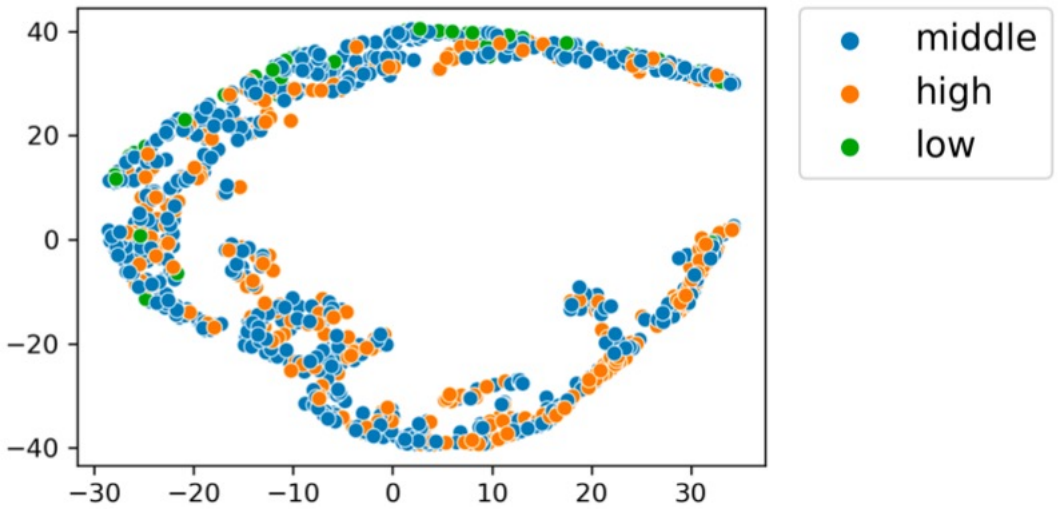
- incorporate historical sequence of user's behavior when transferring knowledge.
- conduct an experiment of off-policy evaluation for the case of transferring rewards


Rakuten

T-SNE project of how similar users and ads were located closely.







Top left: users labeled with age groups.

Top right: users labeled with reward point status.

Left bottom: ads labeled with ad types.

Similarities among users and among ads

We utilized consine similarity

$$\mathcal{S}_{user}(x_i, x_j) = \frac{x_i \cdot x_j}{|x_i||x_j|}$$

$$\mathcal{S}_{ad}(y_i, y_j)$$

The number of users is huge in the real-world dataset, so we leveraged Locally-sensitive Hashing