Optimizing Explorations in Largescale Recommendations System with Information Sharing

Zhihao Huang
Zhihao.Huang@walmart.com
Walmart Global Tech
Sunnyvale, CA, USA

Ankur Bhardwaj
Ankur.Bhardwaj@walmart.com
Walmart Global Tech
Sunnyvale, CA, USA

Abdus Khan
Abdus.Khan@walmart.com
Walmart Global Tech
Sunnyvale, CA, USA

Afroza Ali
Afroza.Ali@walmart.com
Walmart Global Tech
Sunnyvale, CA, USA

Abstract

Large scale recommendation systems rely on robust exploration-exploitation strategies to mitigate the cold start problem caused when new items are added or as customer preferences change. The lack of information on fresh (and new) items needs to be filled in order to surface them to their relevant customer base. However, these systems are often faced with a very large pool of candidates to choose from, making the exploration process quite inefficient. Fortunately, information sharing among similar arms can immensely help in reducing the exploration cost. To address this, we built an online learning system that systematically clusters popular and new products using structures such as taxonomy and product interaction features along with high dimensional semantic embedding, enabling information sharing among eligible popular items and cold/new items. We present in this paper a production ready system (ready for A/B testing) with thorough offline simulation tests that highlights the benefit of sharing information to optimize exploration and improve overall reward metrics.

Keywords

Recommendation System, Multi-arm Bandit, Exploration Exploitation, Reinforcement Learning, Online learning

ACM Reference Format:

Zhihao Huang, Ankur Bhardwaj, Abdus Khan, and Afroza Ali. 2024. Optimizing Explorations in Largescale Recommendations System with Information Sharing. In . CIKM 2024, Boise, Idaho, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Recommendation systems have traditionally been dependent on customer interaction data for training relevant recommendations. This leads to heavily biased promotion of certain high confidence items on the recommendation platforms, while a large number of other items remain minimally exposed. This closed feedback loop [7, 25] results in a small number of items becoming extremely popular, while the vast majority have very few interactions.

With the scale of fresh content uploads on social media platforms or new products getting added into the online marketplaces, recommendation systems have moved to deploy explore-exploit based on multi-arm bandit framework to quickly explore and surface promising new variants across its large and diverse customer base. In the classic formulation of the multi-arm bandit problem, the goal of the agent is to systematically choose among a set of arms or actions to maximize a desired reward. This objective aims to balance between exploration (of new item) and exploitation (of high potential items). In the multi-armed bandit (MAB) problem, the model aims to maximize the cumulative reward by sequentially choosing actions (pulling arms) from a set of options. Each action has an associated reward distribution, and the goal is to identify the action with the highest expected reward. On the other hand, pure exploration solely focuses on identifying the best arm as quickly as possible, rather than maximizing the cumulative reward. This problem is often referred to as the "best-arm identification" problem and has recently gotten attention in research [1–3, 7, 8, 10]. Pure exploration in a MAB problem can be more costly in terms of the number of trials required compared to maximizing the cumulative expected reward.

However, modern day recommendation systems are challenged with tremendously large number of products (several hundreds of millions) where exploring each of them could be intractable. The exploration is both cost inefficient and time consuming and in certain cases the algorithm may never fully leverage its learnings. Further, non-stationarity caused due to seasonality and changes in customer preference prolonge exploration, diminishing its long-term value.[4, 9]

Fortunately, the online learning system can leverage the similarity among the products to share the reward information from popular to new ones, thus helping to optimize the exploration phase across a very vast pool of candidates.

Our Contributions: We consider the problem of optimizing large-scale exploration in recommendation system using multi-arm bandit framework with information sharing. 1)We designed an online learning system that uses product features to form clusters based on similarity metrics. 2) We then define the parameter update process

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM 2024, October 2024, Boise, Idaho, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

within the clusters defining the eligibility criteria on the participants who can share and receive reward information. Through detailed experiments on two different datasets, we show that the share learning algorithm outperforms the traditional Thompson Sampling model. Information sharing among the right arms contribute to the model's faster convergence and robustness over non-stationary.

2 Problem Setup

In the following we introduce the problem setup that utilize the conventional Thompson's Sampling Algorithm (TS) for click through rate (CTR) model on a e-commerce item ranking use case.

Click Through Rate (CTR) Model. We consider n-arm bandit problem for web item ranking, where a_i represent each item as an arm with $i \in \{1, 2, ..., n\}$. For each of the item a_i, the probability of a click given an impression is

$$\mathbb{P}(C_i = 1) = p_i$$

Thompson's Sampling Algorithm (TS) for CTR Model. Following the conventional Thompson's Sampling Algorithm (TS) setup, we use a Bayesian approach to model the click probability of each item. For each arm a_i, it's click probability density distribution follows the Bayesian posterior as a product result of empirical distribution and prior distribution. Under the CTR model, each impression can be considered as a Bernoulli trail with success probability of p_i. Therefore, the likelihood of multiple clicks s and no-click-impression f modelled with a binomial distribution is

$$\mathbb{P}(X|p_i) = \binom{s+f}{s} p_i^s (1-p_i)^f.$$

By utilizing the conjugal properties of beta distribution with binomial distribution, we set the prior of p_i as $\text{Beta}(\alpha, \beta)$, and finally the posterior distribution of p_i is

$$\mathbb{P}(p_i|X) = \text{Beta}(\alpha + s, \beta + f) := \text{Beta}(\alpha, \beta), \tag{1}$$

where we denote $\alpha := \alpha + s$ and $\beta := \beta + f$ for simplicity.

Ranking Model. We consider the ranking of the items determined by the above TS approach. Over the time $t \in \{1, 2, ..., T\}$, the TS algorithm samples the click probabilities $p_i^{(t)}$ as

$$p_i^{(t)} \sim \text{Beta}(\alpha_i^{(t)}, \beta_i^{(t)}) \tag{2}$$

where $\text{Beta}(\alpha_i^{(t)}, \beta_i^{(t)})$ represents the posterior distribution of p_i at time t. The ranking of the items $a_i^{(t)} = \mathcal{A}_{(t)} \in \{1, 2, ..., n\}$ is determined by $p_i^{(t)}$. Therefore, the ranking model based on TS for CTR is an accumulative process of impressions and clicks over time. On most e-commerce platform, time t is a short period of time that a series of impressions $s + f$ and clicks s are observed. The posterior distribution can be updated seamlessly by adding impressions and clicks to the corresponding parameters.

It is obvious that the conventional TS ranking logic is purely determined by the accumulation of impressions and clicks observations. The arm level context is not utilize at all. We denote that for each arm a_i, contextual information $F_i = (SF_i, UF_i)$ is available where SF_i represents structured contextual features and UF_i represents unstructured contextual features. In the next section, we shall discuss how the contextual feature may be used in advancing the TS for CTR model.

3 Algorithm

We introduce a two-stage bandit algorithm on top of the conventional TS for CTR model. The first stage is the creation of arm similarity measurement based on arm context. The second stage is updating parameters with the created similarity measurements.

3.1 Similarity Measurement

The purpose of the similarity measurement is to determine how two arms are alike with each other. A similarity score $S_{i,j} \in (0, 1)$ is calculated between item a_i and a_j. Particularly, we define $S_{i,i} = 1$. However, for some of the items that are completely not related, the similarity score $S_{i,j}$ may still be larger than 0, which may result in some unwanted information sharing in later sections. To prevent this, a more sparse similarity measurement can resolve this issue.

Clustering based on structured features: In case of large number of arms, calculating similarity between all arms is computationally expensive and often includes comparisons between unrelated arms. To facilitate parameter sharing between relevant popular (hot) and less popular (cold) items, we first employ DBSCAN clustering based on structured features [6]. With this approach we ensure that each cluster contains a minimum number of hot items, enabling parameter sharing within the cluster. The structured features includes numerical attributes like item revenue, item price, average rating as well as categorical attributes such as brand, product category, and product type.

Semantic Similarity based on textual features: In this step, we utilize textual features to generate BERT embeddings [5] for each arm. In our use case for e-commerce items, an item title and description based semantic embedding $E_i \in \mathbb{R}^k$ is created.

A similarity matrix is created based on the above steps, where $S_{i,j}$ can be calculated with a cosine similarity as

$$S_{i,j} = I_{(samecluster)}(i, j) * \frac{E_i \cdot E_j}{\|E_i\| \|E_j\|}, \tag{3}$$

where $I_{(samecluster)}(i, j)$ represents the indicator function that the items i and j come from the same cluster.

3.2 Information Sharing and Parameter Update

When sharing the information between items, there are a few criteria that we want to follow, particularly we want to honor the extend of exploration of each item while updating the point estimate of the click probabilities. Specifically,

- **Differentiate hot v.s. cold items**. For hot items, enough exploration has been given, and the exploration result should be honored. Therefore, the information sharing happens only for hot items to cold items, where a threshold is used to differentiate cold and hot items.
- **Preserve the extend of exploration**. Although cold items receive information from hot item, the level of exploration should stays unchanged. We only want to update the point estimate of p_i of the item.
- **Adjust information weight**. Well explored items should play more important role in sharing the information compared to less explored items.

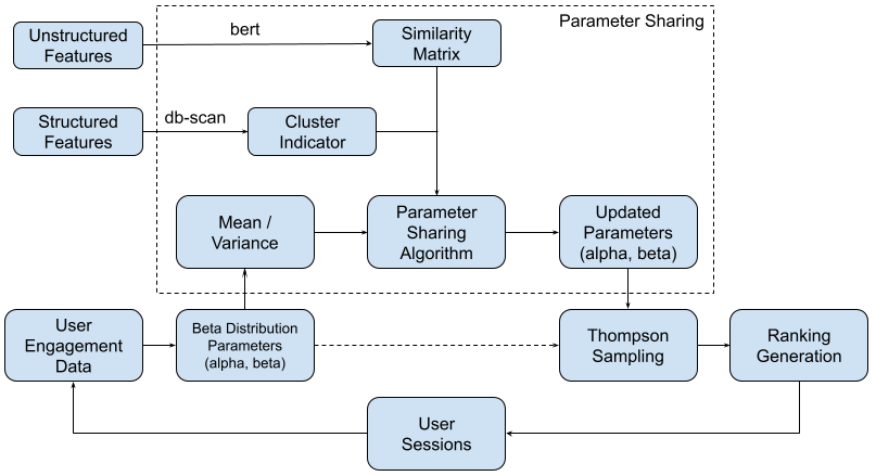

Figure 1: End-to-end design scheme of Thompson's Sampling with Information Sharing that utilizes structured-features-based clustering and semantic-feature-based embedding.

Considering the above logic, we propose the following approach to update the α and β parameters of the eligible items, where the eligibility is determined by a hyperparameter $\zeta \in \mathbb{N}^+$.

For each item a_i at time t, we define the mean and variance of its beta distribution as

$$\mu_i^{(t)} = \frac{\alpha_i^{(t)}}{\alpha_i^{(t)} + \beta_i^{(t)}} \tag{4}$$

$$\sigma_i^{(t)} = \frac{\alpha_i^{(t)} \beta_i^{(t)}}{(\alpha_i^{(t)} + \beta_i^{(t)})(1 + \alpha_i^{(t)} + \beta_i^{(t)})} \tag{4}$$

For cold or somewhat-cold item a_i that satisfy $\alpha_i^{(t)} + \beta_i^{(t)} < \zeta$, a weighted average across the mean μ^t for all eligible items under the same cluster is introduced to update the parameters.

$$\alpha_i^{(t)\prime} = \frac{\sum_{j=1}^{n} S_{i,j} * I_{(samecluster)}(i,j) * \frac{\mu_j^{(t)}}{\sigma_j^{(t)}}}{\sum_{j=1}^{n} S_{i,j} * I_{(samecluster)}(i,j) * \frac{1}{\sigma_j^{(t)}}} \times (\alpha_i^{(t)} + \beta_i^{(t)}) \tag{5}$$

$$\beta_i^{(t)\prime} = \alpha_i^{(t)} + \beta_i^{(t)} - \alpha_i^{(t)\prime} \tag{5}$$

3.3 Overall Algorithm

With the above steps and calculations, we formula the overall algorithm as Algorithm 1 and diagram of the system as Picture 1.

Algorithm 1 Thompson's Sampling with Information Sharing

Require: ζ, DB-SCAN Min Size
  for Each session t where $t \in \{1, 2, ..., n\}$ **do**
    Perform DM-SCAN on SF and obtain $I_{samecluster}$
    for Each arm $a_i^{(t)}$ where $i \in \{1, 2, ..., m\}$ **do**
      Calculate BERT embedding for each arm E_i
      Calculate $\mathbb{P}(p_i|X) = \text{Beta}(\alpha_i^{(t)}, \beta_i^{(t)})$
      for Each arm $a_j^{(t)}$ where $j \in \{1, 2, ..., m\}$ **do**
        Calculate arm similarity $S_{i,j}$ with (3)
      end for
      if $\alpha_i^{(t)} + \beta_i^{(t)} < \zeta$ **then**
        Calculate $\mu_i^{(t)}$ and $\sigma_i^{(t)}$ with (4)
        Update $\alpha_i^{(t)}$ and $\beta_i^{(t)}$ with (5)
      end if
    end for
    Generate ranking based on sampling with (2)
  end for

4 Experiment

Implementation. We conduct simulations to evaluate the performance of the algorithm on various metrics. Given the dynamic nature of online learning, we simulate the learning process when the true rewards distribution of each arm is available but unknown to the algorithm. The algorithm then learns through explore-exploit process. After each time when the algorithm refreshes, impressions are allocated based on the ranking of arms given by the algorithm. Finally, the simulation process generate rewards based on the true rewards distribution. Two simulations are done with synthetic dataset and E-commerce dataset for item recall set generation and re-rank task. The clustering steps are skipped for simplicity.

We also deploy the full algorithm in real-world e-commerce platform for the same task where a much large item pool exists with strong non-stationary performance. In particular, we target the cold items, which are usually introduced on a regularly for promotions purposes, that have no or very little exposure. We collect the performance signals before and after the deployment.

Datasets. The datasets used for simulations consist of multiple arms with known rewards distributions, where a binomial distribution is usually used in the web click model. For comparison purposes, we generate a synthetic dataset that is comparable to the real-world dataset.

- **Synthetic dataset.** The mean of the rewards are generated with a uniform distribution within a reasonable range we

Table 1: Statistics of datasets used for experiments.

Dataset	Arms	Total impression, Model Refresh Frequency and Recall Set Size	Rewards and Similarity
Synthetic Data	377	1 Million, per 500 impressions, 25	Synthetic
E-commerce Data 1	377	1 Million, per 500 impressions, 25	Real-world
E-commerce Data 2	148,824	10 Million, Dynamic, 2500	Real-world

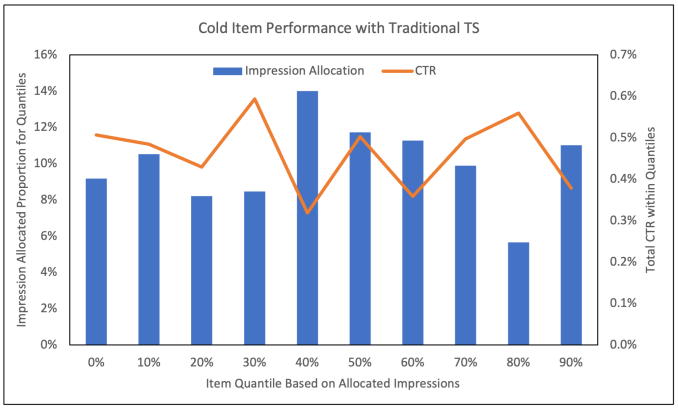

Figure 2: X-axis represents the item quantiles ranked by allocated impressions. Traditional TS requires long exploration phases, resulting randomly allocated impressions.

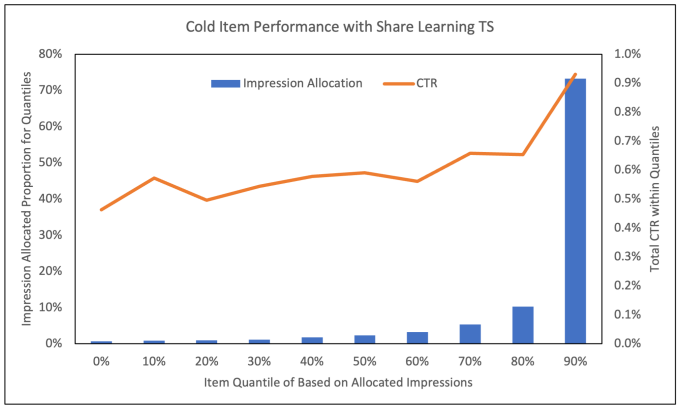

Figure 3: With share learning, the algorithm allocates more than 70% of the total impressions to 10% of the items, where the highest CTRs are observed.

observe in e-commerce. The similarity measurement is generated correlated to the mean rewards.

- **E-commerce Dataset 1.** The arms are actual items in one of the E-commerce Deals Page. We limit the items in Electronic Category with certain amount of impressions only to skip the clustering and ensure the accuracy of the true rewards distribution. The similarity measurement is created based on item semantic cosine similarity.
- **E-commerce Dataset 2.** The arms are actual items in multiple E-commerce Deals Pages. We consider all available items. The similarity measurement is created based on item semantic cosine similarity. The model refresh every few hours with dynamic real world impressions.

The detailed statistics of the datasets are available in Table 1.

Baseline. We compare the performance of the algorithm with traditional TS where no information sharing is conducted.

Evaluation. The simulation is setup with the same total impressions to be allocated and model refresh frequency for comparison purposes. We look at the total generated clicks at 250, 500, 1000 and 2000 model refreshes. The performance of TS with Information Sharing is able to out perform the traditional TS on multiple simulation trials. We observe faster converges to the best performing arms, and better optimization results. Particularly in the initial phase from the cold start, we observe an average of 13% improvements in the rewards, where the maximum of improvements is observed.

In the real-world deployment where a significantly larger item pool exists, the share learning TS outperforms the traditional TS. We observe that the items with the most impressions have the highest CTRs. As a result of the information sharing, the algorithm is able to skip most of the long and costly exploration phase and proceed to exploitation phase much more efficiently than the traditional approach. There is a 88% improvement in the overall CTR as most of the impression are allocated to well-performing items.

Table 2: Click Improvements by Model Refresh Numbers

Dataset	@250	@500	@2000	Overall
Synthetic Data	10.48%	6.77%	1.76%	4.42%
E-commerce Data 1	13.09%	7.43%	2.42%	7.81%
E-commerce Data 2	-	-	-	88.03%

5 Conclusions

This work provides a practical solution for leveraging arm contextual features to share information from well explored arms to under explored arms for the large scale recommendation system. The key is creating an arm-to-arm similarity measurement. By introducing structured features and semantic embedding of arms, the share learning model updates the parameters based on the arm similarity and arm explore-exploit states for under explored arms while preserving the framework of traditional approaches. Through experiments on two datasets, we show that the share learning algorithm outperforms the traditional TS model. Information sharing contributes to the model's faster convergence and robustness over non-stationary. Importantly, information sharing is built on top of the traditional TS framework, making it a scalable solution with great flexibility.

References

[1] Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. 2010. Best Arm Identification in Multi-Armed Bandits. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, Adam Tauman Kalai and Mehryar Mohri (Eds.). Omnipress, 41–53. http://colt2010.haifa.il.ibm.com/papers/COLT2010proceedings.pdf#page=49

[2] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. 2010. Pure Exploration for Multi-Armed Bandit Problems. arXiv:0802.2655 [math.ST] https://arxiv.org/abs/0802.2655

[3] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. 2011. Pure exploration in finitely-armed and continuous-armed bandits. *Theor. Comput. Sci.* 412, 19 (apr 2011), 1832–1852. https://doi.org/10.1016/j.tcs.2010.12.059

[4] Giuseppe Burtini, Jason Loeppky, and Ramon Lawrence. 2015. Improving online marketing experiments with drifting multi-armed bandits. In *International Conference on Enterprise Information Systems*, Vol. 2. SCITEPRESS, 630–636.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL] https://arxiv.org/abs/1810.04805

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (Portland, Oregon) *(KDD'96)*. AAAI Press, 226–231.

[7] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. 2012. Best Arm Identification: A Unified Approach to Fixed Budget and Fixed Confidence. In *Neural Information Processing Systems*. https://api.semanticscholar.org/CorpusID:1168777

[8] Aurélien Garivier and Emilie Kaufmann. 2021. Non-Asymptotic Sequential Tests for Overlapping Hypotheses and application to near optimal arm identification in bandit models. arXiv:1905.03495 [math.ST] https://arxiv.org/abs/1905.03495

[9] Aurélien Garivier and Eric Moulines. 2008. On Upper-Confidence Bound Policies for Non-Stationary Bandit Problems. arXiv:0805.3415 [math.ST] https://arxiv.org/abs/0805.3415

[10] Kevin Jamieson and Robert Nowak. 2014. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *2014 48th Annual Conference on Information Sciences and Systems (CISS)*. 1–6. https://doi.org/10.1109/CISS.2014.6814096