# Assignment 4: Bachelor Project Report

Theme 2

## Group number: 28

Supervisor: Bob Rombach


Group members:

Maximilian Schütt – 608930

Simran Vaidya – 591216

Shraddha Quandinya – 607776

Valentin Kellermann – 593203

# Contents

# 1 Introduction

## 1.1 Managerial Problem

The digital world has drastically increased the flow of content that users face every day, resulting in consumers having to navigate large amounts of digital noise. Specifically in the online news sector, with content being updated rapidly, the abundance of articles heightens the challenge of not only capturing users' attention but also compelling them to engage with the story.

In the consumer funnel, which describes the user's journey from initial awareness to article selection, this issue is most acute during the consideration stage. At this step, alleviating the decision-making-process for the user becomes paramount. Hence, Yahoo! must use effective recommendation systems to narrow down the consideration set to a selection of articles most likely to engage the user. By doing so, Yahoo! reduces choice overload, thus drastically increasing the likelihood that a customer clicks on an article, which corresponds to a successful conversion in the funnel (Li & Kannan, 2014).

The effectiveness with which a user is guided through the funnel up to that specific point is measured by defining certain business-relevant key performance indicators (KPIs). Generally, in the case of online news, click-through rates (CTRs) for articles are a prominently used metric. The main challenge that arises is that the CTR must be maximised whilst balancing performance with efficient use of organisational resources (i.e., financial and time investments).

## 1.2 Research Objective

On online news platforms, articles are found to have a dynamic and short-lived nature. It is therefore important that a recommendation system can both leverage historic interaction data and simultaneously update recommendations based on new articles added. This is the typical exploration-exploitation tradeoff that is commonly seen with recommendation systems. One system that effectively addresses this dilemma is the Multi-Armed Bandit (MAB).

These bandits dynamically test various content options for users to determine which one is most engaging. Fundamentally, the system evaluates interactions continuously, calculating each option's probability of engaging a user based on prior responses. For instance, a user clicking on an article is considered a "success," indicating high engagement. This success increases the likelihood of the article being shown again.

In light of differences between users and their demands, MABs can make use of "pooling", which considers the extent to which learnings from the preferences for articles (estimated through dynamic testing) should be shared across users. This connects the recommendation system's logic to the widely used concept of segmentation in marketing. The key question that arises here is to what extent can learnings from one "cluster" (i.e. segment) of users be applied to others, to maximise CTR.

One end of the spectrum is the pooled strategy which assumes that all users have the same preferences, indicating that the learnings regarding an article's success from one user can be applied to all users. However, due to the aforementioned potential differences, this may result in highly generalised and suboptimal recommendations, leading to low CTRs. On the other hand, the unpooled strategy is that MABs could consider each user segment as a unique case, hence catering recommendations to segment-specific preferences. However, this prevents cross-cluster information sharing and decelerates the learning process due to unutilized similarities between segments. Playing to the strengths of pooled and unpooled strategies, while mitigating their shortcomings, partial pooling has emerged as an alternative to the two extremes. This approach strikes a balance between the pooled and unpooled logic, utilising both information sharing and individualised learning.

### 1.3 Research Question

Although there is extensive research on MABs with pooled and unpooled strategies, studies combining these approaches dynamically are sparse. Consequently, this exploration is focused on investigating the current gap in pooling strategies to optimise KPIs by striking a balance between isolating and grouping user preferences. To explore this, this paper aims to answer the question: ***"Does applying partial pooling to user clusters maximise the CTR for online news articles on Yahoo!?"***

## 2 Literature Review

### 2.1 Why CTR Maximisation?

For online news platforms such as Yahoo!, advertising revenue is the main revenue stream (Holcomb, 2014). Companies that seek to purchase ad space on such platforms base their decisions on the platforms' customer engagement rates. As explored by Calder et al. (2009), higher engagement signals high user traffic which increases the reach of the advertisements. It is important to note that this paper explores two types of engagement - social interactive (e.g., reacting to other users' comments) and personal (e.g., clicking on an article). Whilst the former is not applicable to the context of Yahoo!, findings of the study show that personal interactions with website content are a strong indicator of advertising engagement.

Nayak et al. (2023) found that in the online news sector, CTR is the main KPI used to track and quantify this engagement. CTR measures the percentage of viewers who click on an online article when it is displayed (Cambridge University Press & Assessment, 2024). This is indicative of users interacting with content on the platform which is reflective of the likelihood that customers will engage with advertisements.

However, CTR does have its drawbacks, which were also explored by Nayak et al. (2023). Firstly, it does not provide context on the click, as there is no information on why an article was clicked (e.g., a reader may have accidentally clicked an article), or if the click resulted in

the user reading the complete article. This leads to the second disadvantage, where the lack of context results in lower assurance of advertisement engagement based on an article's CTR (Nayak et al., 2023). Together, this means that though CTR measures whether or not an article was clicked, the reasoning behind this click is unknown, complicating the process of drawing direct conclusions regarding engagement.

Nonetheless, as stated by Myllylahti (2020), "attention carries monetary value, and [...] can be commodified by exchanging it for advertising dollars or reader revenue". Hence, despite its drawbacks, CTR emerged as the default metric to measure engagement in the context of news articles (Nayak et al., 2023). Therefore, higher CTRs signal higher engagement, and maximising CTR means maximising revenue potential, for both news platforms and advertisers.

### 2.2 Segmentation of Users

Zooming out to the broader user landscape, Beregovskaya & Koroteev (2021) establish that customer preferences greatly vary in the context of online news. This implies that not all articles displayed will be relevant to all users. Therefore, the concept of user segmentation and its effects should be explored.

Additionally, Li & Kannan (2014) found that to maximise CTR, customers must move down the purchase funnel, which starts at consideration and moves down to conversion. Moreover, the field experiment identified that to do so, customers should be exposed to articles that are likely to fall into their consideration set. The field experiment generates insightful data regarding the stages of the consumer funnel (e.g., identifying concrete spillover effects between the stages of the funnel). Although these findings are specific to online marketing businesses, they can still be applied to Yahoo!, given that its primary revenue source is online advertising. Moreover, Yang & Zhai (2022) further support this idea by stating that increasing the number of potentially preferred articles that a user sees results in higher CTRs.

To select these articles that are shown to users, one strategy is to create user segments. As defined by Kotler & Keller (2016), segmentation is a technique utilised in marketing to divide customers on the basis of shared attributes such as preferences or demographic characteristics. This results in relevant articles being displayed to users, hence increasing the likelihood of engagement (John et al., 2023). A benefit of implementing segmentation is that it indirectly improves the user experience because they have to exert less effort to narrow down articles for their consideration set (Kotler & Keller, 2016). This is supported by Hoban & Bucklin (2015), who found that the reduced choice overload encourages users to move down the purchase funnel to the conversion stage (i.e. clicking on an article). Moreover, Liu et al. (2022) identified that the increase in the number of articles that are clicked has a positive effect on the engagement rates communicated to advertisers, thus increasing the likelihood of more ad space being purchased. Therefore, by segmenting, news providers like Yahoo! can improve their CTR and, by extension, revenue.

An alternative is to use a mass marketing strategy where user preferences are ignored or assumed to be homogeneous. As found by Chauhan (2018), the main benefit of this approach is that it allows a platform to reach a wider target audience thus increasing the volume of users that enter the awareness stage of the customer funnel. Given the broad variety of topics available on news platforms, Chauhan (2018) acknowledges that not every user will take a liking to every article, which would be crucial for moving to the consideration stage. One other approach is to cater to each user's preferences individually, however, as explained by Hu et al. (2018), this is highly unfeasible for platforms as large as Yahoo! due to the large amount of interaction data involved. Moreover, the significant computational resources required for real-time personalisation heightens the lack of feasibility. Therefore, it is imperative to strike a balance between addressing individual user preferences and that of the mass market, which can be achieved through segmentation.

### 2.3 Choosing MABs as the Recommendation System

Extensive research has been conducted in the online news sector regarding the effectiveness of various recommendation systems. Recommendation systems are tools that can help businesses simplify the decision-making process for customers by acting as digital curators that suggest personalised content to users (Fayyaz et al., 2020). Their efficacy is measured through organisational KPIs such as CTR, and other model-specific metrics (Ko et al., 2022).

MABs offer a distinct approach to recommendations that effectively address the exploration-exploitation dilemma. They function by treating individual news articles as "arms" to be explored, with user interactions (click-through rate, CTR) acting as the reward signal (Li et al., 2011). The applications of MABs extend beyond news recommendations, playing a vital role in online advertising and marketing. For example, Edupuganti & Sen (n.d.) found that a website displaying ad banners may utilise MABs where each advertisement is considered an arm in the system. Yang & Zhai (2022) found that by tracking user clicks on each advertisement, the system learns which performs better and prioritises it by showing it more frequently. This real-time optimization leads to increased user engagement and potentially higher advertising revenue.

A benefit of MABs is that they perform well in dynamic environments due to their real-time adaptability (Zhang et al., n.d.). For Yahoo!, this is crucial because it balances exploring new articles and exploiting learnings from past user engagement. This results in a recommendation system that can keep up to date with new content whilst ensuring readers are shown relevant content (Slivkins, 2024). This is because MABs continuously update recommendations based on user interactions, ensuring users see content relevant to their evolving interests (Bouneffouf & Rish, 2019). As explained by Schwartz et al. (2017), this is a result of their inherent ability to address the exploration-exploitation dilemma by striking a balance between showing users already-popular articles (exploitation) and exploring new, unseen articles (exploration). This feature is complemented by their ability to effectively combine with contextual information or user segmentation factors like demographics, enabling timely and personalised recommendations that promote user engagement with diverse perspectives within the news domain (Li et

al., 2011).

However, incorporating these contextual factors may negatively impact computational efficiency, which is an important element for the implementation of MABs, which has not been tested in this paper. Hu et al. (2018) identified scalability as another important consideration when implementing MABs, showing that MABs are highly scalable. This makes them ideal for large-scale recommendation systems such as those required by platforms like Yahoo!.

However, MABs are not without limitations. Since they learn through live user interactions, they might not provide optimal recommendations initially (Bouneffouf & Rish, 2019). This is because the system needs to be primed with sufficient user data to understand their preferences. Another consideration is the computational complexity of implementing MAB algorithms. Especially for very large-scale systems such as that of Yahoo!, both the volume of data as well as the constant updating requirement may result in the algorithm requiring longer processing time and hardware support (Yi et al., 2023). However, scaling back on the data inputted can lead to suboptimal recommendations (Shamir & Lin, 2022). Hence, it is imperative to consider the potential trade-offs between accuracy and computational efficiency.

In contrast to MABs, content-based filtering (CBF) takes a content-centric approach. It focuses on analysing the attributes (keywords, categories, topics) of items the user has interacted with in the past (Li & Kim, 2003). However, content-based filtering also faces limitations because its effectiveness heavily relies on the quality and detail of the assigned item attributes (Glauber & Loula, 2019). Poorly defined or inaccurate attributes can lead to irrelevant recommendations. Additionally, because this system is so reliant on keywords, the issue of filter bubbles arises where the balance between offering personalised content and content diversity is not maintained (Jannach et al., 2015).

Conclusively, MABs most effectively address the crucial needs of a recommendation system required by a news platform like Yahoo!. This is due to their ability to dynamically balance exploitation with exploration, a crucial requirement in the newspaper realm, and their ability to incorporate context and segment factors to optimise recommendations in real-time.

## 2.4 Pooling

The combination of MABs with contextual factors such as user segments, introduces 'pooling' as another critical consideration in the recommendation system realm. Pooling, in its simplest form, refers to the sharing and aggregation of data across multiple sources or groups, often with the intention of improving the accuracy of statistical results (Buccapatnam et al., 2015). In the context of MABs, it has mainly been investigated in terms of hierarchical pooling between arms, where information sharing is structured to improve the learning process across different arms of the bandit (Russo & Van Roy, 2014).

A dominant example is the field experiment conducted by Schwartz et al. (2017), which explores different nuances of pooling in the context of online advertising. In their research they distinguish between unpooled, partially pooled and pooled methods in the context of a field experiment, aiming to improve the distribution of impressions across ads and websites. 'Unpooled' refers to a scenario where the learning process is conducted separately for each ad and website, representing arms, without sharing statistical knowledge among them. In 'Pooled' models on the other hand learning is conducted jointly for all ads and websites, assuming they share the same underlying distribution. Furthermore, Schwartz et al. (2017) introduce the concept of partial pooling, which strikes a balance between the unpooled and pooled logic. However, it is important to note that also this paper does not consider the computational efficiencies of these models, which might limit their real-world applicability.

As shown by Schwartz et al. (2017) and Russo & Van Roy (2014), pooling traditionally refers to information sharing across arms, such as articles, advertisements, and websites. Contrastingly, Li et al. (2016)'s work on Collaborative Filtering Bandits, demonstrates a situation in which the concept of pooling – i.e. information sharing across multiple sources or groups – is applied to a different context, namely user segments. By sharing user preferences only within segments of similar users, Li et al. (2016) followed Schwartz et al. (2017)'s logic of an unpooled policy. This approach significantly boosted the bandit's performance, even in environments as dynamic as news recommendations. This is further corroborated by the research of Nguyen and Lauw (2014)'s, who show that clustering a population into specific segments and applying a bandit for each cluster leads to significant boosts in performance.

Hence, the ability to produce tailored recommendations potentially increases the engagement of users (Ban & He, 2021; Nguyen  Lauw, 2014). However, as information sharing across segments is prevented, the learning process could potentially be decelerated, when new information is introduced. This is exemplified by Su & Khoshgoftaar (2009), claiming that systems that employ collaborative filtering in the context of MABs (i.e. that follow the unpooled logic) aggravate the cold start problem.

While the pooled logic introduced by Schwartz et al. (2017), facilitating information sharing across multiple groups, might mitigate the cold start problem, Nguyen & Lauw (2014) claim that such models may be inefficient in the long run, as individual preferences are masked and produce suboptimal recommendations as a result. Overall, the literature describes a trade-off between different pooled and unpooled strategies, highlighting the need for a solution. Schwartz et al. (2017) addressed this problem by introducing a 'partially pooled' model, which considers both arm-specific and joint learnings. While Nguyen & Lauw (2014) demonstrate the advantages of applying the unpooled logic to user segments, little to no research applied the logic of the 'partially pooled' to this context. Thus, the existing body of knowledge exhibits a void for an approach that dynamically balances both extremes in the context of user segments to mitigate their respective issues.

## 2.5 Selection of Bandits

In order to initiate pooling policies, an additional element to consider is the selection of the algorithm. Current research focuses on two types - Upper Confidence Bound (UCB) algorithms, and Thompson Sampling (TS). The investigation of using pooling to improve algorithm performance has been mainly conducted via UCB bandits (Russo & Van Roy, 2014). This algorithm selects the arm with the highest UCB calculated based on the expected reward and the uncertainty associated with that reward based on the number of previous attempts (Sutton & Barto, 1998). As shown by Auer et al. (2002), this uncertainty bonus systematically accounts for variability in returns, encouraging the exploration of uncertain options, and hence effectively balancing exploration and exploitation. More recent work has increasingly focused on posterior sampling, also called Thompson Sampling. These algorithms sample from a distribution that is estimated to reflect the reward distribution for each arm (Russo et al., 2018).

A significant advantage of Thompson Sampling, especially concerning pooled models, is the computational efficiency of the algorithm (Chapelle & Li, 2011). Russo and Van Roy (2014) demonstrate that when the action space becomes large, a condition that is likely to occur when considering the plethora of news articles available online, the UCB's action selection step entails solving a problem of high complexity. In contrast, Thompson Sampling differs in its approach as it simplifies the action selection step by only requiring a solution to a linear problem, which is significantly less complex (Chapelle & Li, 2011). Despite the reduced complexity, both Chapelle and Li (2011), as well as Russo and Van Roy (2014) find that Thompson Sampling performs equally well or even better than UCB, making it a highly attractive method for investigating MAB problems with pooling.

However, both papers use large action spaces in their models, which is beneficial for TS, but disadvantages UCB algorithms. This is because UCB's strength lies in its ability to make highly informed, deterministic decisions from a smaller set of options, where its confidence bounds can be accurately refined (Sutton & Barto, 1998). It is important to consider that especially in environments with smaller action spaces and a relatively constant reward distribution UCB's deterministic nature may allow for a quicker convergence to optimal actions.

## 2.6 Conclusion

This literature review highlights the effectiveness of using CTR as a direct measure of article engagement, which is crucial for maintaining advertisement revenue for online news platforms like Yahoo!. In this context, research has found that the diversity in user preferences results in low generalisability of article preferences. Segmentation has been found to be a strong tool to balance these individual user preferences with that of the mass market. With the recent technological advances, literature has shown that this can be achieved by implementing recommendation systems, specifically Multi-Armed Bandits. This can be attributed to their ability to dynamically balance exploration and exploitation to optimise recommendations.

MABs can be further enhanced when implementing pooling approaches, referring to different degrees of information sharing across articles. With the two extremes of unpooled (no information sharing) and pooled (full information sharing), existing research from Schwartz et al. (2017) has already shown the benefits of partial pooling, which strikes a balance between the two extremes. However, little research exists that applies the logic of partial pooling in the context of user segments.

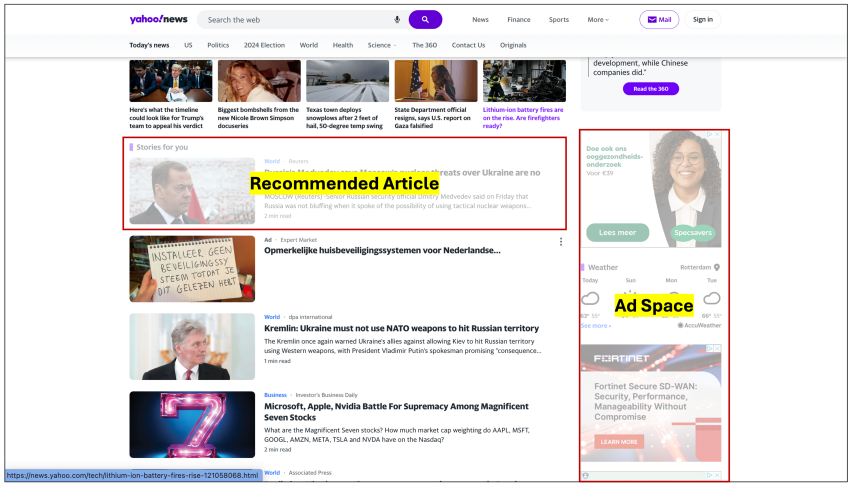
Given these findings, this paper seeks to address this gap by exploring the effectiveness of applying partial pooling on user clusters to maximise CTR for online news articles on Yahoo!. Thus, by answering the question, ***"Does applying partial pooling to user clusters maximise the CTR for online news articles on Yahoo!?"*** this paper will contribute insights into the managerial problem of addressing a diverse set of user preferences, and effectively guiding these users through the customer funnel.

# 3 Problem Formulation

## 3.1 Business Problem Formulation

Having established the importance of CTR maximisation and the right recommendation in the context of online news articles within the literature review, it is paramount to deep-dive into the business problem of Yahoo!. To do so, visualising the Yahoo! website (Figure 1) is helpful to understand the first step of this specific consumer funnel.

Figure 1: Screenshot of 'Stories For You' Section on Yahoo! Homepage (produced by author)




In order to maximise CTR, and with that Yahoo!'s business outcomes, a user (website visitor) must be incentivised to click on an article that is recommended to them on the Yahoo! website. The "Stories for You" section recommends a list of articles that may be appealing to the user. As explained by Li et al. (2011), for each user interaction articles are selected for a personalised recommendation from a list of available options. The first article displayed in this section is deemed most attractive to the respective user, aiming to maximise the likelihood of a

click. Hence, optimising the decision problem of recommending this top article is paramount for Yahoo!'s success. This is also demonstrated by the proximity of the advertisements (ad space) to the recommended articles, which further highlights the strong relationship between article engagement and advertising revenue.

Consequently, by employing different approaches of incorporating context in the form of user segments (i.e., pooling techniques), MABs will be implemented aiming to maximise the CTR for the article that is placed in the top slot in the "Stories for You" section, as highlighted in Figure 1.

### 3.2 Defining the Bandit Problem

Having specified the context from a business perspective, the team formulated the challenge of maximizing the recommended articles' CTR as a bandit problem (find legend of subsequent notations used in Appendix A). As explained, Yahoo has news articles available $a = 1, \ldots, A$, that it can recommend to the users of its site. Consequently, each article was treated as an 'arm' of the bandit, with an unknown rate of success (i.e., CTR). Moreover, having established the value of segmentation due to the highly diverse preferences of online-news consumers, the users of Yahoo were segmented into $k = 1, \ldots, K$ clusters, to which the articles can be shown (elaborated in Section 2.2). Finally, adhering to the conclusion of the Literature Review, two distinct approaches of dealing with these user segments emerged: a 'pooled' approach, assuming similarity of preferences regarding articles across user clusters, and an 'unpooled' approach, assuming inter-cluster heterogeneity.

Accordingly, two bandit problems were formulated:

Pooled Approach (1):

$$\arg\max_{a \in A} \sum_{k=1}^{K} \mathbb{E}[R(a)] \tag{1}$$

Unpooled Approach (2):

$$\sum_{k=1}^{K} \arg\max_{a_k \in A} \mathbb{E}[R(a_k)] \tag{2}$$

- Let $A = \{a_1, \ldots, A\}$ be the set of all available articles
- Let $a \in A$ be the article recommended to all clusters $K$
- Let $a_k \in A$ be the article recommended for cluster $k$
- $\sum R(a)$ is the expected reward (click probability) of the article for all clusters $K$
- $\sum R(a_k)$ is the expected reward (click probability) of the article for cluster $k$

Conclusively, the pooled bandit approach was designed to solve the CTR maximisation problem by identifying the article that maximised the expected reward for all users. On the

other hand, the unpooled bandit approach was designed to solve the CTR maximisation problem by identifying the article for cluster k, for which the expected rewards of that specific cluster was maximised. Hence, the articles recommended were specific to the cluster that each user was assigned to.

# 4 Research Methods

## 4.1 Thompson Sampling

Due to its computational efficiency, the Thompson Sampling algorithm has been evaluated as the most suitable method to utilise. Thus, while the exact implementation differed between the pooling approaches that will be outlined in Section 4.2, the fundamental logic of using Thompson Sampling for news article recommendations is delineated in the following:

TS requires maintaining a probability model for the rewards expected for each arm. Due to the binary nature of the rewards, a Beta distribution was chosen. This distribution is parameterized by two positive shape parameters $\alpha$ (number of successes - i.e., an article was clicked) and $\beta$ (number of failures - i.e., an article was not clicked). As data is collected, the two shape parameters are continuously updated.

When it comes to the implementation, each arm i was associated with its own Beta distribution ($\alpha_i$, $\beta_i$). Upon their introduction, all articles are assumed as equally likely to yield success. Therefore, to initialise the distributions, each arm was set with the same parameters ($\alpha_i = 1$ and $\beta_i = 1$).

To then select an arm (i.e., selecting an article to recommend to the user), the bandit samples a value $\theta_i$ from each article's current Beta distribution. This value reflects the likelihood of that arm resulting in a success. Consequently, the arm with the highest value $\theta_i$ is selected. Following the recommendation, the click (or not click) is used to update the distribution parameters of the selected article. In it's basic form this can be illustrated as follows:

- If the article is clicked: Increment $\alpha_i$ by 1 ($\alpha_i = \alpha_i + 1$)
- If the article is not clicked: Increment $\beta_i$ by 1 ($\beta_i = \beta_i + 1$)

This process is then conducted for every user interaction, with continuous updating of the click probability for each recommended article.

With an increase in the absolute values of the parameters ($\alpha$ & $\beta$), the variance of the Beta distribution of that respective article decreases. Thus, the more an article is tested, the closer the sampled click probabilities $\theta$ will be to the actual (unknown) CTR of that article. In turn, when an article is relatively unexplored (i.e., $\alpha$ and $\beta$ are low), the variance of that article's Beta distribution is higher. This means that a relatively high value of $\theta$ could be sampled, encouraging the exploration of such articles. Consequently, the model appropriately accounts

for the required exploration-exploitation trade off, which has been shown to be particularly relevant for news articles in the Section 2.3.

## 4.2 Models

### 4.2.1 Pooling Models

As demonstrated in Sections 2.4 and 3.2, the pooled and unpooled models differ in their approach in the sharing of learnings between users. In the context of this paper, pooled models share learnings about the 'value' of articles across all users. This can accelerate the learning process, especially in early stages (i.e., when articles are newly introduced) as information that is relevant across segments can be shared. However, as explained, this method assumes homogeneity in preferences and thus masks individual differences amongst users, leading to suboptimal results in the long run (Nguyen & Lauw, 2014). The literature highlighted a trade-off between unpooled and pooled models, calling for an innovative solution. Consequently, inspired by the core idea of Schwartz et al. (2017)'s 'partial pooling', the team investigated the approach of dynamically balancing the benefits of the two extremes. Hence, in the following, the three models (pooled, unpooled, and partially pooled) will be delineated.

### 4.2.2 Pooled Model

Pooled models operate under the assumption that preferences for news articles are uniform across all the user segments. In an organisational context, this is comparable to mass marketing (Section 2.2). Thus, all the learnings from all user interactions were incorporated into a single model. As described in Section 4.1, the model was initialised with the same parameters ($\alpha_i = 1$ and $\beta_i = 1$) for each article i. Thus, irrespective of the user's characteristics, the same beta distribution was used to sample click probabilities and decide on an article to recommend. Consequently, the update process that was implemented was identical to the 'basic' version outlined in Section 4.1.

### 4.2.3 Unpooled Model

In contrast, unpooled models operate under the assumption that preferences vary significantly between user segments, hence learnings are not shared across the segments. This means that different articles are successful in different clusters, representing a more personalised approach. Consequently, based on observable characteristics, 'users' were first associated with a specific cluster and therefore received recommendations that have been performing particularly well for users with similar characteristics.

Intuitively, this suggests that the same article can have a different 'value' to different segments. Hence, to formulate the model, a separate set of parameters was initialised for each cluster k: ($\alpha_{k,i} = 1$ and $\beta_{k,i} = 1$), ensuring that an article's 'value' remained cluster specific. When the bandit then considered an interaction, it was first linked to the user cluster it most closely resembled. In the basic set-up used, this was accomplished by choosing the cluster for which the difference between the observable user characteristics and the centroids of the cluster

was minimal. Based on the beta distributions of that specific cluster k, a random value $\theta_{k,i}$ was then sampled, to evaluate which article should be shown to the user.

Assuming inter-cluster heterogeneity with respect to preferences, the learnings from the results of the user interaction (click or no click) were only relevant for the specific cluster. Consequently, the beta distribution update process for recommended article i, only applied to that specific cluster:

- If the article is clicked: Increment $\alpha_{k,i}$ by 1 ($\alpha_{k,i} = \alpha_{k,i} + 1$)
- If the article is not clicked: Increment $\beta_{k,i}$ by 1 ($\beta_{k,i} = \beta_{k,i} + 1$)

#### 4.2.4 Partial Pooling

Partial pooling was investigated as a new approach to dynamically balance between the pooled and unpooled model, aiming to combine the approaches' benefits while mitigating their shortcomings. The intuition behind the approach is as follows: users are neither completely similar in terms of their preferences, nor is it possible to cluster them in a way that leads to full inter-cluster heterogeneity. Consequently, while personalised recommendations are effective, there are certain learnings that are valuable to share across clusters.

This is especially relevant during the initialization stage (i.e. when articles are newly introduced). As indicated in the Literature Review, preventing inter-cluster information sharing might decelerate the learning process, since each cluster must determine a new article's 'value' through iterative testing on its own. As a result of this fragmented learning, the number of total interactions required until each cluster has sufficiently explored a certain article is drastically higher than that of a generic (pooled) model.

Thus, during this initial phase, while the 'value' of the article is still being tested, solely relying on the cluster-specific distributions to form recommendations might not lead to the optimal outcome. Instead, it can be valuable to rely on cross-cluster tendencies during these stages. As described, general preferences regarding articles can be updated with each interaction, and are therefore likely to stabilise much faster, providing a good estimate shortly after a new article was introduced. This entails not only tracking an article's performance within the respective clusters, but also across the entire population of users. This allows both pooled and unpooled learning to occur simultaneously. Consequently, during the initial period, the generic preferences (i.e., captured by a global beta distribution) should have more weight in influencing the recommendation decision. As the cluster-specific preferences stabilise over time, a gradual transition can be facilitated, shifting towards tailored recommendations, effectively leveraging the strengths of both methods.

Following the intuition behind partial pooling, both cluster-specific ($\alpha_{k,i} = 1$ and $\beta_{k,i} = 1$), as well as global beta distributions ($\alpha_i = 1$ and $\beta_i = 1$) were initialised. At each interaction, these distributions were aggregated into a single distribution based on which the TS algorithm

selected an arm. This aggregation was facilitated by a newly introduced coefficient $\gamma$, regulating the influence of the global distribution relative to the segment-specific one. The value of $\gamma$ was adjusted dynamically via a logistic function, to allow for the intended transition from pooling when information was scarce, to leveraging knowledge about cluster-specific preferences. The adjustment is captured by the following model:

$$\gamma(n_k) = \frac{1}{1 + e^{c(n_k - nt)}} \tag{3}$$

- $n_k$ represents the number of observations (trials) for a specific user segment after initialization
- c is a constant determining the steepness of the logistic curve, dictating how quickly $\gamma$ transitions (was chosen via empirical testing to 0.001, see Figure 5)
- nt is the number of user interactions of a cluster at which $\gamma$ becomes 0.5 and transitions from prioritising the global distribution to focusing more on the cluster-specific distribution (was chosen via empirical testing to 3,000, see Appendix D)

Consequently, for each user interaction, the first step was to form an aggregated distribution of alpha and beta values combined from the global and cluster-specific model. The weight assigned to the respective alpha and beta values was governed by $\gamma$. To ensure that only gamma regulated the ratio of the resulting aggregated alpha and beta values, the alpha and beta values of each distribution (global and cluster-specific) have been scaled in order to have comparable magnitudes. The intuition is that naturally, the absolute values of alpha and beta for the global distribution of a specific article will always be higher than the cluster-specific distribution of that article. Consequently, without scaling, the global distribution would disproportionately influence the aggregated result simply due to its larger numerical values (for proof see Appendix B). The scaling of the alpha and beta values mitigated this, which allowed $\gamma$ to purely determine the weighting between the global and cluster-specific models based on their relevance without the distributions' absolute magnitude interfering with the aggregation.

This can be expressed as follows:

- Compute Scaled Alpha (S$\alpha$) :S$\alpha_{k,i} = \alpha_{k,i} * \frac{(\alpha_i + \beta_i)}{(\alpha_{k,i} + \beta_{k,i})}$
- Compute Scaled Beta (S$\beta$) :S$\beta_{k,i} = \beta_{k,i} * \frac{(\alpha_i + \beta_i)}{(\alpha_{k,i} + \beta_{k,i})}$

Subsequently, gamma can be used as a weighting factor to form the aggregated value of alpha and beta, for the TS algorithm to sample a value. Mathematically, this can be expressed as follows:

- Compute Aggregated Alpha (agg_$\alpha$): $agg\_\alpha_i$ =S$\alpha_{k,i} * (1 - \gamma(\mathbf{n_k})) + \alpha_i * \gamma(\mathbf{n_k})$
- Compute Aggregated Beta (agg_$\beta$): $agg\_\beta_i$ =S$\beta_{k,i} * (1 - \gamma(\mathbf{n_k})) + \beta_i * \gamma(\mathbf{n_k})$

In this case, $\mathbf{n_k}$ denotes the number of observations that have been considered for each cluster, which ensured that the dynamic adjustment of $\gamma$ depended on the degree of certainty of the cluster-specific distributions. This is coherent with the intuition that initially, as segments explore the 'value' of articles on their own, the global beta distribution is more influential (i.e., higher gamma) than the cluster-specific distribution. With increasing interactions, the segment gained more knowledge about the articles, and the cluster-specific distribution was favoured (i.e., lower gamma).

Consequently, for each interaction, this method resulted in an aggregated beta distribution (agg_$\alpha_i$, agg_$\beta_i$) based on which the TS algorithm sampled value $\theta_i$ to determine which article should be shown to the user. Subsequently, the beta distribution update process was conducted for the chosen article in both the global and cluster-specific distribution. This can be summarised as follows:

- Update cluster-specific if the article is clicked: Increment $\alpha_{k,i}$ by 1 ($\alpha_{k,i} = \alpha_{k,i} + 1$)
- Update cluster-specific if the article is not clicked: Increment $\beta_{k,i}$ by 1 ($\beta_{k,i} = \beta_{k,i} + 1$)
- Update global if the article is clicked: Increment $\alpha_i$ by 1 ($\alpha_i = \alpha_i + 1$)
- Update global if the article is not clicked: Increment $\beta_i$ by 1 ($\beta_i = \beta_i + 1$)

# 5 Bandit Implementation and Evaluation

Having delineated the models, it was paramount to determine how these were to be tested. Ideally such evaluations are conducted via bucket tests, running the bandit algorithm on a fraction of the live user traffic to evaluate performance (Li et al., 2012). However, this method is sub-optimal for the case at hand due to several reasons. First, testing on Yahoo!'s site was not possible as it requires high financial investments and substantial efforts to be facilitated. Second, as Li et al. (2012) suggests, reproducible comparisons are difficult to achieve with user metrics varying constantly. Third, a simulation is the most suitable way to evaluate partial pooling – i.e the benefit of dynamically balancing knowledge sharing between user clusters – in maximising CTRs of online news articles. This is, because the model assumes a 'complete cold start', where each article has never been observed by any of the clusters. Since, the number of user interactions per cluster governs the transition from the global to the cluster-specific distribution, the anticipated benefit of the model vanishes after the specified number of instances for a cluster (nt) has been exceeded. Hence, an article introduced after this threshold would not benefit from information sharing across clusters and would simply follow the unpooled logic. Ensuring a 'complete cold start' for all articles simultaneously was best achieved through a simulation and difficult to ensure in a real world setting.

## 5.1 Dataset Description

The dataset for the offline investigation was provided by the thesis supervisor. Attributes in the dataset include users and their characteristics, recommended articles, articles that could

have been recommended, the time of the recommendation, and whether or not the article was clicked on by the user.

## 5.2 Simulation Intuition

To avoid losing observations, a bandit-first approach was chosen. Essentially, this means that when the bandit makes a recommendation based on an instance, an observation must be sampled from the dataset to simulate a likely response. In order to improve the accuracy with which such a response is predicted, the team decided to enhance the simulation by clustering the users into different groups based on their recorded attributes and computed the cluster-specific CTRs based on the available data. The underlying logic is that as the bandit considers a particular instance and formulates a recommendation, the CTR from the cluster the user most closely resembles can be chosen. Consequently, the response is based on past behaviour of similar users, promising a more reliable estimation. Moreover, clustering the users is in line with the pooled and partially pooled approach, which required cluster-specific CTRs in order to effectively cater to individual preferences.

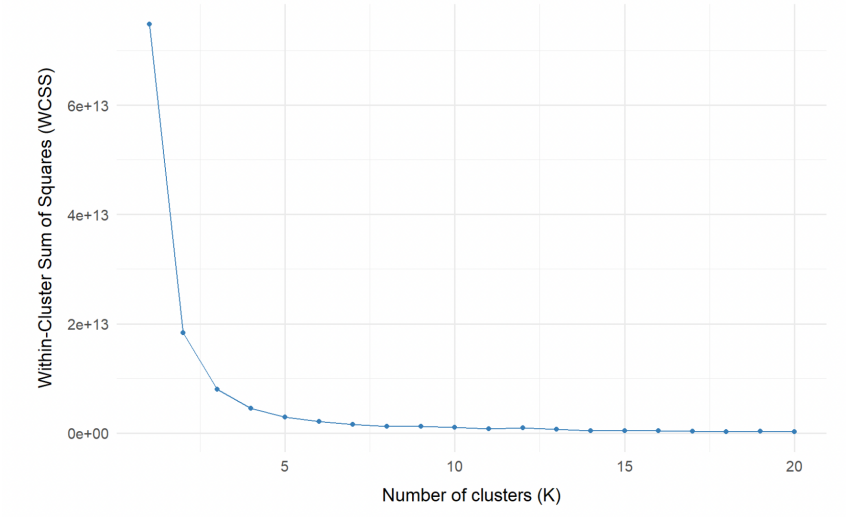
## 5.3 Simulation

### 5.3.1 Cluster Users based on the five user features

Adhering to the intuition of the simulation, the first step is to group users into clusters based on their observable characteristics. To facilitate this, K-Means clustering was chosen by the team due to its computational efficiency and scalability for clustering large datasets (Ioktun et al., 2023). However, this algorithm prerequisites setting the number of segments into which the instances are to be grouped.

Identifying the optimal number of clusters is crucial. Too many clusters would result in unnecessary fragmentation that can dilute meaningful information between data points. On the other hand, too few clusters would lead to over-generalization, preventing unpooled and partially pooled models from effectively catering to the different preferences. To mitigate this threat, the Elbow method was utilised as it is computationally efficient compared to other methods and also provided a graphical representation of the trade-off between the number of clusters and the quality of clustering. The method involves plotting the sum of squared distances of samples to their closest cluster centre against the number of clusters and identifying the “elbow” point where the rate of decrease sharply shifts, indicating the optimal number of clusters.

Figure 2: Elbow Method Graph to Identify the Optimal Number of User Clusters




A visual analysis of Figure 2 concluded that the optimal number of clusters for this dataset would be 5 as it represents the identified 'elbow point' (i.e. the number of clusters at which the rate of decline for the WCSS begins to stagnate). Consequently, at this point, the benefits of adding more clusters diminishes, indicating that further segmentation of the data would not yield substantial improvements in the homogeneity of the clusters, thus optimising the balance between complexity and clustering effectiveness.

Having established 5 as the optimal number of clusters, the K-Means algorithm was applied. Thus, the algorithm initialised 5 random points as cluster centres. Then, it assigned each data point to the nearest centre, recalculated the centres by averaging the assigned points, and repeated these steps until the centres stabilised, effectively grouping the data into 5 distinct clusters based on the observable user features.

### 5.3.2 Calculating CTR for each article within each cluster of users

As described, to accurately simulate user behaviour and enable segment-specific recommendations, it was paramount to compute the average CTR of each article per cluster. To do so, within every cluster, the number of times an article was clicked was divided by the number of times an article was shown. The result was a probability distribution of the 23 articles for each of the 5 clusters.

## 5.4 Cluster Analysis

Having established the clusters and their respective preferences expressed through CTRs, an initial analysis was conducted to provide necessary context to be able to accurately interpret the models' performance.

Table 1: Descriptive Statistics of Identified Clusters

| Statistic | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Cluster 5 |
|---|---|---|---|---|---|
| Mean | 0.0290 | 0.0366 | 0.0087 | 0.0543 | 0.0329 |
| Median | 0.0296 | 0.0350 | 0.0077 | 0.0516 | 0.0336 |
| SD | 0.0105 | 0.0169 | 0.0056 | 0.0217 | 0.0155 |
| N Instances | 226,493 | 94,804 | 348,298 | 168,182 | 162,223 |

As shown in Table 1, the clusters formed are of varying sizes ranging from 94,804 to 348,298 instances. Moreover, the overall engagement of the respective clusters varies, with cluster 4 showing a mean CTR of 5.43%, while Cluster 3 exhibits a mean CTR of only 0.87%. In order to investigate the overall similarity in preferences across clusters, the team computed the pairwise cosine similarity for all possible combinations of clusters (Appendix C). The cosine similarity has been chosen to ensure that the team only accounts for the overall similarity of the preferences between clusters, disregarding the magnitude of the CTRs in the respective clusters. The analysis concluded an average pairwise cosine similarity of 90.2% with the highest correlation existing between cluster 1 and 4 (98.2%). This is consistent with the finding that cluster 1,2 and 4 all have the same "favourite" article.

#### 5.4.1 Simulating an individual instance for an MAB Policy

Having established clusters and their respective CTR for each article was prerequisite to simulate instances for the suggested MAB Policies. The following section provides a step by step breakdown an individual instance was simulated:

- Step 1: A random instance was drawn from the data set.
- Step 2: By finding the cluster that minimised the difference between the user attributes and the clusters' centroids the closest cluster was identified.
- Step 3: Based on that instance, the respective bandit recommended an article .
- Step 4: Based on the click-probability of the recommended article for the instances' cluster, an outcome – click (1) or no click (0) – was simulated.

### 5.4.2 Simulating each MAB Policy

To accurately evaluate each policy, a sample of 500,000 simulated instances ensured sufficient coverage of the whole dataset. Although 100 simulation runs seems to be the norm in the existing body of literature (Li et al., 2011; Schwartz et al., 2017), a total of 10 simulation runs have been chosen, as the results were already sufficiently distinct, indicated by non-overlapping confidence intervals (see Table 2). This further allowed the team to stress test the policies in alternative scenarios, which will be described in the sections hereafter.

## 6 Evaluation Criteria

This section explains the criteria against which the different policies were evaluated within the simulation. Table 2 outlines the statistical and managerial relevance criteria. The simulation tested the three policies (pooled, unpooled, and partially) against a random policy and optimal policy, representing upper and lower bounds for ACTR respectively. The random scenario selected articles on a random basis, and the optimal scenario selected, due to perfect information, the article with the highest conversion likelihood for that instance's cluster.

Table 2: Explanation of Statistical Criteria

| Criteria | Statistical relevance | Managerial relevance |
|---|---|---|
| Aggregated CTR (ACTR)* | Mean aggregated CTR across all simulation runs. Higher ACTR indicates the superiority of a policy. | Higher CTR directly translates into revenue. ACTR will be used as the main evaluation criterion. |
| Standard deviation (STD) | Spread across simulation runs, indicating how much individual performance deviates from the average | Provides additional insight into the risk and variability associated with the ACTR for each policy. |
| Confidence intervals (CI) | The probability that the true ACTR lies within CI (2.5% and 97.5%). | Allows informed decision making when applying the algorithm |
| Relative mean (%) (RM) | Measures the percentage better than the random policy. The higher the positive the RM the better a policy performs against the baseline scenario (Schwartz et al. 2017) | Implementation impact of certain policies over a random allocation of articles, answering the question: "Is it worth it?" |
| Standard error of the mean (SE) | Expected variability from one simulation's mean to the mean of all simulation runs (ACTR). A lower SE represents more certainty. | Paramount to managers, as it indicates the certainty that the ACTR of a certain policy can be expected upon real-world implementation of a certain policy. |

For further insights, the policies' learnings were visualised. This was done by plotting the regret and exploration of the policies across 500,000 simulated instances, representing the averages of the 10 simulation runs.

The team defined regret as the cumulative difference between the CTR of the article chosen by a bandit, and the article chosen by the optimal policy. Secondly, the team defined exploration as the cumulative number of times a policy chose an article that was not optimal according to its knowledge at a specific point in time.

The final evaluation metric is the computational complexity of the policies, which represents the efficiency and computational resources occupied. In line with Computational Complexity Theory (Hartmanis & Stearns, 1965), the policies were evaluated based the times taken by the algorithms, operationalised as total execution time, CPU time, and system overhead. Longer times signaled higher complexity indicating less implementation feasibility for managers.

Besides these evaluation metrics, stress tests were conducted to determine the partially pooled policy's robustness. Firstly, to test the robustness of hyperparameters (used to determine $\gamma$), different values were tested for each parameter, aiming to identify points for which the model breaks or performs particularly well (see Appendix D and Figure 5).

Secondly, to test the performance of the policies in case of homogeneity across user preferences, the CTRs were sampled from the same cluster every time, irrespective of the user attributes present in a particular instance. Thus, all simulated responses were based on the same CTR distribution.

# 7 Research Results

## 7.1 Statistical Criteria

Table 3 reports the summary of the statistical criteria for each policy tested, including the optimal and random policy, as upper and lower bounds respectively. Among the proposed methods, partial pooling achieves the highest ACTR with 4.85%, outperforming both unpooled and pooled models with 4.72% and 4.10% respectively. This result is corroborated due to the non-overlapping confidence intervals amongst the policies' performance distributions. In line with these results, the relative mean follows the same order, with partial pooling achieving 89.22% of the optimal performance. However, with the highest SE (0.0003), the partially pooled policy's ACTR is least robust across simulation runs.

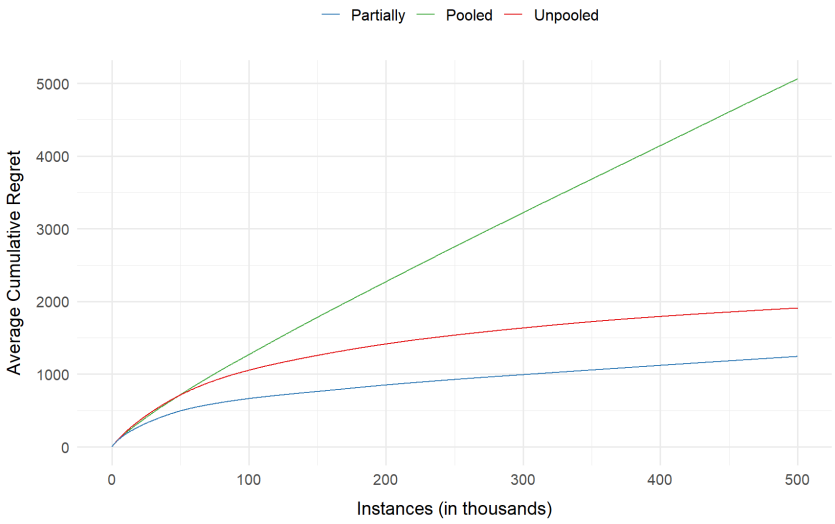
Table 3: Results table showing key performance metrics of Algorithms over 500,000 instances across 10 simulation runs

| **Policy** | **ACTR** | **SE** | **STD** | **CI Lower** | **CI Upper** | **RM** |
|---|---|---|---|---|---|---|
| Random | 0.0276 | 6.1147e-05 | 0.1639 | 0.0278 | 0.0281 | 0.0000 |
| Pooled | 0.0410 | 1.3497e-04 | 0.1983 | 0.0404 | 0.0415 | 56.9808 |
| Unpooled | 0.0472 | 1.3885e-04 | 0.2121 | 0.0463 | 0.0478 | 83.5631 |
| Partially | 0.0485 | 3.0362e-04 | 0.2149 | 0.0479 | 0.0491 | 89.2239 |
| Optimal | 0.0511 | 6.4771e-05 | 0.2201 | 0.0505 | 0.0517 | 100.0000 |

## 7.2 Learning

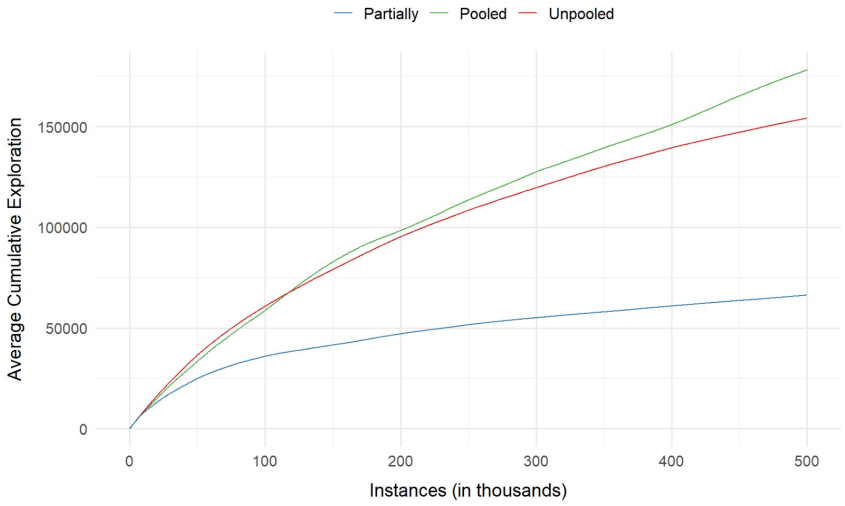
With regards to regret, the key observation of Figure 3 is that the partially pooled policy (blue line) achieved the lowest accumulated regret across the 10 simulation runs when compared with the unpooled (red line) and pooled policy (green line). While both the partially and unpooled policies exhibit a trend towards stabilization, with the rate of regret accumulation diminishing over time, the pooled policy continues to accrue regret at a consistently increasing rate. Furthermore, Appendix E represents a close-up to the first 100,000 instances and illustrates that the pooled policy outperformed the unpooled policy for approximately the first 50,000 instances.

Figure 3: Regret over 500,000 iterations across 10 simulation runs (computed by cumulating the difference of optimal CTR and the CTR of the arm selected)




Considering exploration, the key observation of Figure 4 is that the partially pooled policy minimised exploration, by only allocating 13.27% of the instances to exploration. The pooled policy demonstrates the most extensive exploration, committing 35.64% of its instances to explore new (non-optimal) possibilities, the highest among the three strategies.

Figure 4: Exploration over 500,000 iterations across 10 simulation runs (computed by cumulating the number of instances for which non-optimal arms were selected)




## 7.3 Computational Efficiency

Table 4 summarises the dimensions used to evaluate computational efficiency. While CPU times (User CPU Time) for the pooled and unpooled policy for executing 500,000 observations are relatively similar – 30.9 and 31.9 seconds respectively – the partially pooled policy occupied more than two times as much CPU time (56.96 seconds). The system overhead (System CPU Time) and total wall-clock time (Elapsed Time) display a similar distribution. The partially pooled policy took a total of 88.65 seconds to complete 500,000 instances, while pooled and unpooled only required 35.5 and 35.31 seconds respectively.

Table 4: Computational Efficiency Measures in terms of time in seconds

| **Model** | **User CPU Time** | **System CPU Time** | **Elapsed Time** |
|---|---|---|---|
| Pooled | 30.904 | 0.093 | 35.498 |
| Unpooled | 31.895 | 0.080 | 35.309 |
| Partially Pooled | 56.960 | 0.305 | 88.649 |

## 7.4 Stress Tests

### 7.4.1 Hyperparamters

The hyperparameter stress tests are summarised in Figures 5 and 6. Figure 5 shows the performance in terms of ACTR of the partially pooled policy for different values of c while keeping $nt$ = 3,000 to ensure sufficient certainty about the clusters' preferences (Appendix D). It becomes visible that for very low values of c (i.e. a very smooth transition from the global to the segment-specific distribution governed by $\gamma$) the partially pooled model breaks, as the ACTR fell below the unpooled policies' ACTR (red dotted line). The breaking point approximately lays at c = 1e-04 (0.0001).As indicated by the graph, higher values of c (i.e., faster transitions

between the distributions), resulted in higher ACTRs. The highest ACTR was achieved at c = 0.001.

Figure 5: Hyperparameter stress test - letting the partially pooled model's c vary across nt = 3,000, against the unpooled models ACTR (red dotted line)

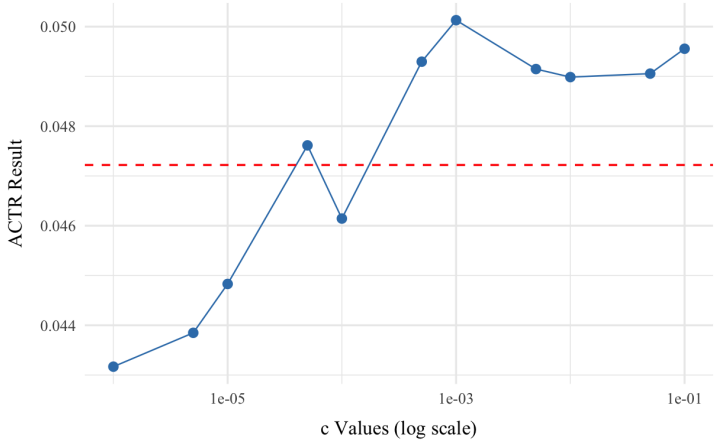
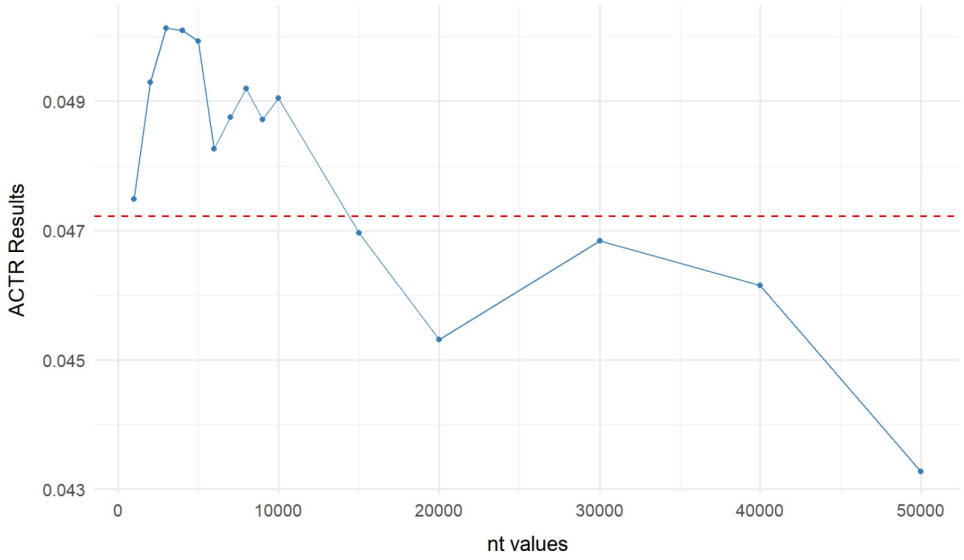



Figure 6 shows the performance in terms of ACTR of the partially pooled policy for different values of nt at c = 0.001 (previously evaluated as a robust choice). It becomes visible that the partially pooled policy has the highest expected ACTR at approximately nt = 3,000. For higher values of nt, the ACTR of the partially pooled policy declined until breaking at approximately 14,000 instances, falling below the unpooled policies' ACTR (red dotted line).

Figure 6: Hyperparameter Stress Test - letting the partially pooled model's nt vary across c = 0.001, against the unpooled models ACTR (red dotted line)



### 7.4.2 Homogeneity across preferences

The second stress test assumed homogeneity across user preferences (full results are summarised in Appendix F). Under this scenario, the pooled policy outperformed the partially pooled policy which still outperformed the unpooled policy, with ACTRs of 4.5%, 4.4%, and 4.2% respectively (Appendix F.1). These results are again validated through the non-overlapping performance distributions.

In this scenario, the pooled policy was further able to minimise regret the strongest - displaying a fast flattening curve - followed by the partially pooled and the unpooled model respectively (Appendix F.2).

# 8 References


[1] Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). *Finite-time Analysis of the Multi-armed Bandit Problem*. Machine Learning, 47(2/3), 235–256. https://doi.org/10.1023/a:1013689704352

[2] Ban, Y., & He, J. (2020). *Generic Outlier Detection in Multi-Armed Bandit*. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/3394486.3403134

[3] Ban, Y., & He, J. (2021). *Local Clustering in Contextual Multi-Armed Bandits*. Proceedings of the Web Conference 2021. https://doi.org/10.1145/3442381.3450058

[4] Ban, Y., He, J., & University of Illinois at Urbana-Champaign. (2021). *Local clustering in contextual Multi-Armed bandits*. In Proceedings of the Web Conference 2021 (p. 13) [Conference proceeding]. https://arxiv.org/pdf/2103.00063.pdf

[5] Beregovskaya, I., & Koroteev, M. (2021). *Review of Clustering-Based Recommender Systems*. Retrieved from https://doi.org/10.48550/arXiv.2109.12839

[6] Bouneffouf, D., & Rish, I. (2019). *A Survey on Practical Applications of Multi-Armed and Contextual Bandits*. Retrieved from https://doi.org/10.48550/arXiv.1904.10040

[7] Buccapatnam, S., Tan, J., & Zhang, L. (2015). *Information sharing in distributed stochastic bandits*. 2015 IEEE Conference on Computer Communications (INFOCOM). https://doi.org/10.1109/infocom.2015.7218651

[8] Calder, B. J., Malthouse, E. C., & Schaedel, U. (2009). *An Experimental Study of the Relationship between Online Engagement and Advertising Effectiveness*. Journal of Interactive Marketing, 23(4), 321–331. https://doi.org/10.1016/j.intmar.2009.07.002

[9] Cambridge University Press & Assessment. (2024). *click-through rate*. In Cambridge Dictionary. Retrieved March 14, 2024 from https://dictionary.cambridge.org/dictionary/english/click-through-rate

[10] Chapelle, O., & Li, L. (2011). *An empirical evaluation of Thompson sampling*. Neural Information Processing Systems, 24, 2249–2257. https://proceedings.neurips.cc/paper/2011/file/e53a0a2978c28872a4505bdb51db06dc-Paper.pdf

[11] Chauhan, J. S. (2018). *Effectiveness of different marketing strategies in reaching target audiences: A quantitative investigation of Ad Agency representatives*. INFORMATION TECHNOLOGY IN INDUSTRY, 6(1). https://doi.org/10.17762/itii.v6i1.836

Calder, B. J., Malthouse, E. C., & Schaedel, U. (2009). An Experimental Study of the Relationship between Online Engagement and Advertising Effectiveness. Journal of Interactive Marketing, 23(4), 321–331. https://doi.org/10.1016/j.intmar.2009.07.002

[12] Edupuganti, V., & Sen, S. (n.d.). *Optimizing Online Advertising Strategies*. Retrieved from https://web.stanford.edu/class/aa228/reports/2019/final76.pdf#:

~:text=URL%3A%20https%3A%2F%2Fweb.stanford.edu%2Fclass%2Faa228%2Freports%2F2019%2Ffinal76.pdf%0AVisible%3A%200%25%20

[13] Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., & Kashef, R. (2020). *Recommendation systems: Algorithms, challenges, metrics, and business opportunities.* Applied Sciences, 10(21), 7748. https://doi.org/10.3390/app10217748

[14] Glauber, R., & Loula, A. (2019). *Collaborative Filtering vs. Content-Based Filtering: differences and similarities.* Retrieved from https://doi.org/10.48550/arXiv.1912.08932

[15] Hartmanis, J., & Stearns, R. E. (1965). *On the computational complexity of algorithms.* Transactions of the American Mathematical Society, 117, 285-306.

[16] Hoban, P. R., & Bucklin, R. E. (2015). *Effects of Internet Display Advertising in the Purchase Funnel: Model-Based Insights from a Randomized Field Experiment.* Journal of Marketing Research, 52(3). https://doi.org/10.1509/jmr.13.0277

[17] Holcomb, J. (2014, March 26). *Revenue sources: A heavy dependence on advertising.* Pew Research Center. Retrieved from https://www.pewresearch.org/journalism/2014/03/26/revenue-sources-a-heavy-dependence-on-advertising/

[18] Hu, J., Liang, J., Kuang, Y., & Honavar, V. (2018). *A user similarity-based top-n recommendation approach for Mobile in-application advertising.* Expert Systems with Applications, 111, 51–60. https://doi.org/10.1016/j.eswa.2018.02.012

[19] Ikotun, A. M., Ezugwu, A. E., AbuAlaigh, L., Abuaijha, B., & Heming, J. (2023). *K-means Clustering Algorithms: A comprehensive review, variants analysis, and advances in the era of Big Data.* Information Sciences, 622, 178–210. https://doi.org/10.1016/j.ins.2022.11.139

[20] Jannach, D., Lerche, L., Kamehkhosh, I., & Jugovac, M. (2015). *What recommenders recommend: An analysis of recommendation biases and possible countermeasures.* User Modeling and User Adapted Interaction, 25(5), 427-491. https://doi.org/10.1007/s11257-015-9165-3

[21] John, J. M., Shobayo, O., & Ogunleye, B. (2023). *An exploration of clustering algorithms for customer segmentation in the UK retail market.* Analytics, 2(4), 809–823. https://doi.org/10.3390/analytics2040042

[22] Ko, H., Lee, S., Park, Y., & Choi, A. (2022). *A survey of recommendation systems: Recommendation models, techniques, and Application Fields.* Electronics, 11(1), 141. https://doi.org/10.3390/electronics11010141

[23] Kotler, P., & Keller, K. L. (n.d.). *Identifying Market Segments and Targets.* In Marketing Management (15th ed., pp. 267–388). essay, Pearson. Retrieved from https://www.edugonist.com/wp-content/uploads/2021/09/Marketing-Management-by-Philip-Kotler-15th-Edition.pdf

[24] Li, H. (Alice), & Kannan, P. K. (2014). *Attributing conversions in a multichannel online marketing environment: An empirical model and a field experiment.* Journal of Marketing Research, 51(1), 40–56. https://doi.org/10.1509/jmr.13.0050

[25] Li, L., Chu, W., Langford, J., & Wang, X. (2011). *Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms.* Conference on Web Search and Data Mining, WSDM 2011, Pp. 297–306. https://doi.org/10.1145/1935826.1935878

[26] Liu, Y., Zhu, C., & Zeng, M. (n.d.). *End-to-end segmentation-based news summarization.* Retrieved from https://www.microsoft.com/en-us/research/uploads/prod/2021/10/AAAI_2021_SegSumm-4.pdf

[27] Myllylahti, M. (2020). *Paying attention to attention: A conceptual framework for studying news reader revenue models related to platforms.* Digital Journalism, 8(5), 567-575.

[28] Nayak, A., Garg, M., & Muni, R. R. D. (2023). *SIGIR '23: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval.* In SIGR'23. Association for Computing Machinery Digital Library. Retrieved from https://doi.org/10.1145/3539618.3591741

[29] Nguyen, T. T., & Lauw, H. W. (2014). *Dynamic Clustering of Contextual Multi-Armed Bandits.* Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. https://doi.org/10.1145/2661829.2662063

[30] Russo, D., & Van Roy, B. (2014). *Learning to optimize via posterior sampling.* Mathematics of Operations Research, 39(4), 1221–1243. https://doi.org/10.1287/moor.2014.0650

[31] Russo, D., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). *A tutorial on Thompson sampling.* https://doi.org/10.1561/9781680834710

[32] Schwartz, E. M., Bradlow, E. T., & Fader, P. S. (2017). *Customer acquisition via display advertising using Multi-Armed Bandit Experiments.* Marketing Science, 36(4), 500–522. https://doi.org/10.1287/mksc.2016.1023

[33] Shamir, G., & Lin, D. (2022). *Real World Large Scale Recommendation Systems Reproducibility and Smooth Activations.* Retrieved from https://doi.org/10.48550/arXiv.2202.06499

[34] Slivkins, A. (2024, April 3). *Introduction to multi-armed bandits.* arXiv.org. Retrieved from https://arxiv.org/abs/1904.07272

[35] Su, X., & Khoshgoftaar, T. M. (2009). *A survey of collaborative filtering techniques.* Advances in Artificial Intelligence, 2009, 1–19. https://doi.org/10.1155/2009/421425

[36] Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning.* Retrieved from http://portal.acm.org/citation.cfm?id=551283

[37] Yahoo. (2024). *Yahoo Advertising — Digital online advertising Platforms.* Yahoo Ad Tech. Retrieved March 17, 2024 from https://www.advertising.yahooinc.com/

[38] Yang, Y., & Zhai, P. (2022a, February 22). *Click-through rate prediction in online advertising: A literature review.* arXiv.org. Retrieved from `https://arxiv.org/abs/2202.10462`

[39] Yang, Y., & Zhai, P. (2022b, February 22). *Click-through rate prediction in online advertising: A literature review.* arXiv.org. Retrieved from `https://arxiv.org/abs/2202.10462`

[40] Yi, X., Wang, S., He, R., Chandrasekaran, H., & Wu, C. (2023). *Online matching: A real-time bandit system for large-scale recommendations.* ar5iv. Retrieved from `https://ar5iv.labs.arxiv.org/html/2307.15893`

[41] Zhang, H., Ding, J., Feng, L., Tan, K. C., & Li, K. (n.d.). *Solving expensive optimization problems in dynamic environments with meta-learning.* Retrieved from `https://ar5iv.labs.arxiv.org/html/2310.12538`

# 9 Appendix

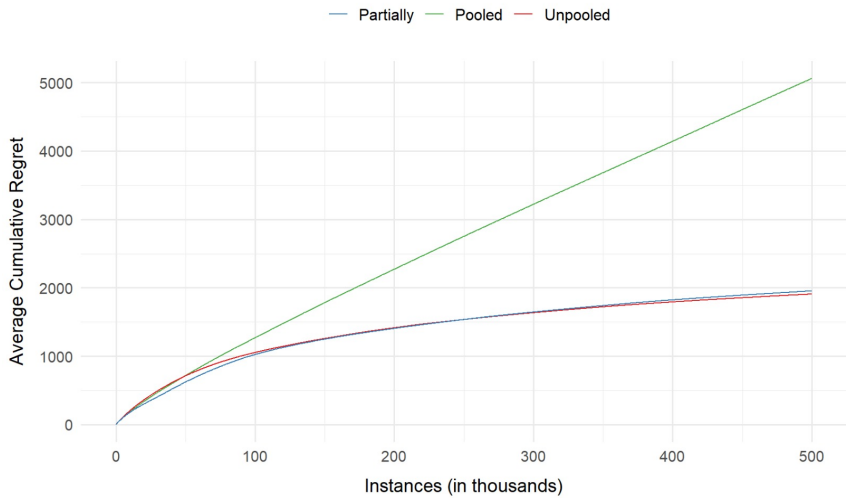
### Appendix A: Legend of Notations Used

| Symbol | Description |
|---|---|
| $A$ | $\{a_1, \ldots, A\}$; set of available articles |
| K | total number of clusters |
| a | article recommended for all clusters |
| $a_k$ | article recommended for cluster k |
| $\alpha$ | number of successes stored in beta distribution |
| $\beta$ | number of failures stored in beta distribution |
| $\alpha_i$ | alpha value for arm i |
| $\beta_i$ | beta value for arm i |
| $\theta_i$ | likelihood of arm i resulting in success |
| $\alpha_{k,i}$ | number of successes for arm i in cluster k |
| $\beta_{k,i}$ | number of failures for arm i in cluster k |
| $\theta_{k,i}$ | likelihood of arm i being shown to the user in cluster k |
| $\gamma$ | weightage to global distributions for partially pooled model |
| $n_k$ | number of observations for a specific segment after initialization |
| c | constant to determine steepness of logistic curve |
| $nt$ | the number of user interactions of a cluster at which $\gamma$ becomes 0.5 and transitions from prioritising the global distribution to cluster specific |
| $S\alpha_{k,i}$ | scaled alpha |
| $S\beta_{k,i}$ | scaled beta |
| agg_$\alpha_i$ | aggregated alpha |
| agg_$\beta_i$ | aggregated beta |

## Appendix B.1: Model Performance if Scaling Mechanism was removed for Partial Pooling

| Policy | ACTR | SE | STD | CI Lower | CI Upper | RM |
|---|---|---|---|---|---|---|
| Random | 0.02764 | 6.115e-05 | 0.1640 | 0.02719 | 0.02810 | 0 |
| Pooled | 0.04099 | 1.350e-04 | 0.1983 | 0.04044 | 0.04154 | 56.981 |
| Unpooled | 0.04722 | 1.389e-04 | 0.2121 | 0.04663 | 0.04781 | 83.563 |
| Partially | 0.04713 | 1.621e-04 | 0.2119 | 0.04654 | 0.04771 | 83.193 |
| Optimal | 0.05107 | 6.477e-05 | 0.2201 | 0.05046 | 0.05168 | 100 |

## Appendix B.2: Cumulative Regret if Scaling Mechanism was removed for Partial Pooling

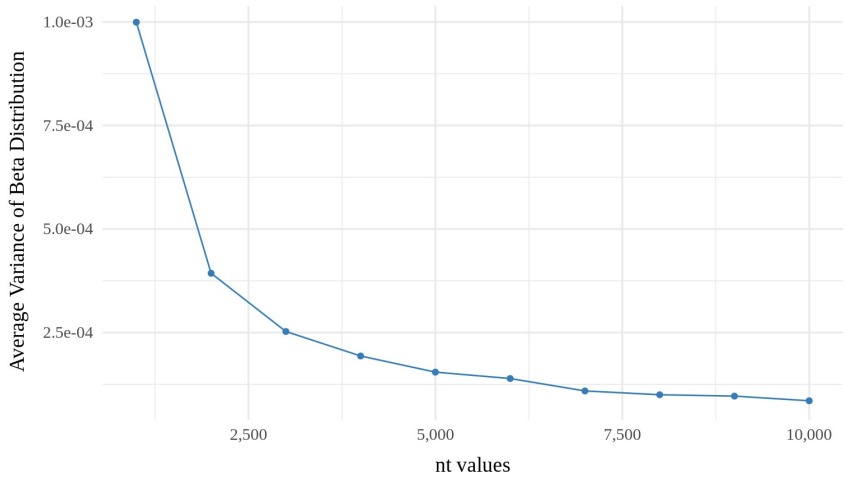



## Appendix C: Analysis of preference similarity between formed clusters using Cosine Similarity Matrix (in terms of CTR per article)

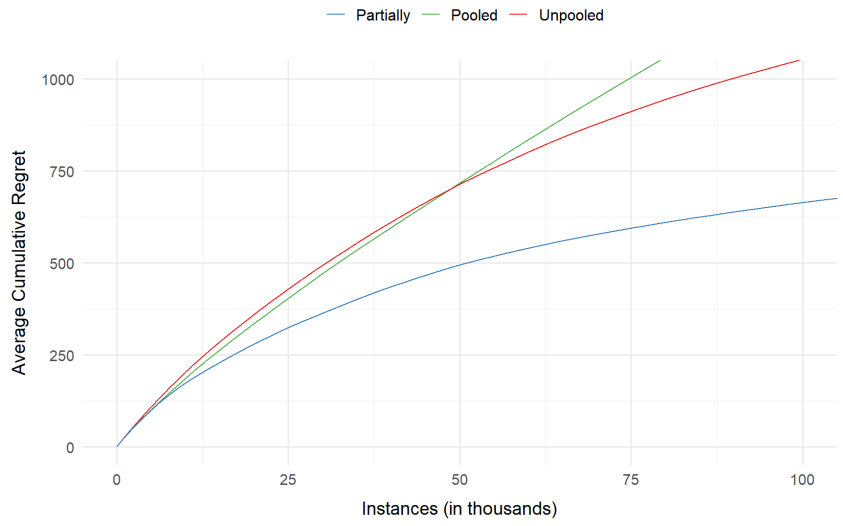
| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **1** | 1.000 | 0.9607 | 0.8709 | 0.9822 | 0.9569 |
| **2** | 0.9607 | 1.0000 | 0.7666 | 0.9702 | 0.8786 |
| **3** | 0.8709 | 0.7666 | 1.0000 | 0.8051 | 0.8795 |
| **4** | 0.9822 | 0.9702 | 0.8051 | 1.0000 | 0.9481 |
| **5** | 0.9569 | 0.8786 | 0.8795 | 0.9481 | 1.0000 |

## Appendix D: Choosing initial estimate of $nt$ value by testing average variance of beta distribution over different $nt$ values across 10 simulation runs

Initially, it was not clear what $nt$ value should be applied to test out applicable values for the scaling factor c. To get a reasonable initial estimate, the team decided to investigate how the beta variance of a segment-specific distribution changes with different number of iterations ($nt$). As the graph shows, the variance of the beta distribution stabilizes at $nt = 3{,}000$, which is why the team chose it as a reasonable estimate to stress-test for varying values of c.




## Appendix E: Zoom-in on regret plot (first 100,000 instances)



**Appendix F.1: Results across 10 simulation runs (500,000 instances each) if all clusters had similar preferences**

| Policy | ACTR | SE | STD | CI Lower | CI Upper | RM |
|---|---|---|---|---|---|---|
| Random | 0.0291 | 0.0001 | 0.1680 | 0.0286 | 0.0295 | 0 |
| Pooled | 0.0451 | 0.0001 | 0.2076 | 0.0446 | 0.0457 | 92.2711 |
| Unpooled | 0.0422 | 0.0001 | 0.2010 | 0.0416 | 0.0427 | 75.2264 |
| Partially | 0.0439 | 0.0002 | 0.2050 | 0.0434 | 0.0445 | 85.4199 |
| Optimal | 0.0465 | 0.0001 | 0.2105 | 0.0459 | 0.0471 | 100 |

**Appendix F.2: Average Cumulative Regret across 10 simulation runs (500,000 instances each) if all clusters had similar preferences**